Chapter 4
Representing and Analysing Meaning with LSA

Semantics,—the study of meaning communicated through language (Saeed 2009)—, is usually defined to investigate the relation of signs to the objects they represent.

Such representation of objects and their relations, interactions, properties, as well as states can be created in various ways. For example, they emerge naturally in form of neural activity in the human brain. Expressing thoughts again in language and other formalisms materialises them intellectually. And they can be created automatically using data processing techniques and computation. One branch of such automated techniques for generating semantic representations uses the co-occurrence of the words in language to derive information on the semantic structure. This branch is often entitled 'heuristics-based approaches'. More recently, they are also called 'distributional semantics' (Sahlgren 2008). Latent Semantic Analysis (LSA) is one of the methods in this branch. LSA was introduced to facilitate the investigation of meaning in texts, originally in the context of indexing and information retrieval.

While Social Network Analysis, as presented in the previous chapter, is a powerful instrument to represent and analyse the purposive context of learning activity, Latent Semantic analysis is blind to such social and relational aspects. LSA lacks the elaborate instruments and measures provided by network analyses to further investigate the characteristics of structure found. Moreover, no clear guidance is provided on determining *before calculation* an optimal number of singular values to retain in the truncation of the dimensional system resolved.

Still, it is a time-tested algorithm for representing and analysing meaning from text, with its closeness in mathematical foundation being a natural candidate for further integration (see Chap. 5). These foundations as well as the analysis workflow with the *lsa* package developed and standard use cases are following in the Sects. 4.1 and 4.2.

Two demos are used within this chapter to foster understanding and allow derivation of the main restrictions applying to LSA. The foundational example presented in Sect. 4.3 picks up the usage scenario of the foundational SNA demo

F. Wild, *Learning Analytics in R with SNA, LSA, and MPIA*,
DOI 10.1007/978-3-319-28791-1_4

presented in Sect. 3.2. It will be revisited in the Chapter on application examples for MPIA (Sect. 9.2).

Following the summary of the state of the art in application of LSA to technology-enhanced learning in Sect. 4.4, a second, real-life application example in essay scoring will be added in Sect. 4.5. A summary outlining also the key limitations of LSA concludes this chapter in Sect. 4.6.

The academic discourse around latent semantic analysis started even more recently than that around social network analysis, as can be seen from the line plot depicted at the beginning of this the previous chapter in Fig. 3.1. The expression was coined and the method was developed by a group of researchers at the Bell Laboratories in the late 1980s (Deerwester et al. 1990) as an attempt to overcome synonymy and polysemy problems of—at that time—state-of-the-art information retrieval and navigation systems, following their interest in statistical semantic representation techniques expressed already in earlier articles (e.g. Furnas et al. 1983). The original patent for latent semantic analysis was granted in the US (Deerwester et al., patented 1989, filed 1988). The project leader of the group was Landauer (Landauer et al. 2008, p. ix).

The initial interest in LSA's application areas was on indexing, primarily in information retrieval (see, e.g., Dumais 1992). Though more complicated indexing applications follow soon: the Bellcore Advisor, for example, is an indexing system that uses technical memos to map human expertise by domains (Dumais et al. 1988, p. 285). From there focus broadens over the next years to encompass additional application areas such as information filtering (Foltz 1990), as a measure for textual comprehension (Foltz et al. 1998, p. 304), and technology-enhanced learning. Landauer et al. (1997), for example, investigated how well LSA can be used to evaluate essays. In two experiments with 94 and 273 participants, results of the LSA-based measures were found to have near human performance or to even outperform the human raters in their correlation to a "40 point short answer test" (p. 413 and p. 416). Foltz (1996, p. 200) finds that "grading done by LSA is about as reliable as that of the graders" in an experiment with four human graders and 24 essays.

In a special issue edited by Foltz (1998, p. 128, 129) about "quantitative approaches for semantic knowledge representation", a rich body of technology-enhanced learning applications is introduced: Landauer et al. (1998a) in particular describe several different methods for scoring essays with LSA (p. 279) and for assigning appropriate instructional material to learners (p. 280).

In the same issue, Rehder et al. (1998) investigate how the following set of influencing factors impact on scoring: focusing solely on the technical vocabulary (ignoring non-technical words) did not improve scoring, an essay length of 200 words was (under the conditions of the study) related to result in the best prediction of scores. Furthermore, the article finds both the cosine distance between the vectors representing model solution and essay, as well as the vector length of the essay itself to be important components in predicting scores.

Wolfe et al. (1998)—in the same special issue—investigate how text complexity influences learning, testing this with undergraduates as well as medical students and

assessing high reliability and validity (p. 331). Different assessment tests were conducted: an assessment questionnaire was developed and scored, and the essays were evaluated with a score by professional graders from educational testing service. The cosine between a model solution and the essay was used as the LSA measure. The measure correlated with $r = 0.63$ to essay score and $r = 0.68$ to questionnaire score. This was quite similar to $r = 0.74$, with which the human grader score correlated to questionnaire score (and the $r = 0.77$ of the interrater correlation between the human evaluators), therefore leading Wolfe et al. to conclude that likely all three "were measuring largely the same thing" (p. 331).

4.1 Mathematical Foundations

The mathematical and statistical foundations are made accessible in a series of publications, most notably the seminal article Deerwester et al. (1990) and Berry et al. (1995).

The basic working principle of latent semantic analysis is to map texts to their bag-of-words vector space representation (of document vectors along term 'axes' and term vectors along document 'axes') and then rotate (and scale) the axes of the resulting vector space according to the insights gained from a two-mode factor analysis—using singular value decomposition. The rotation and scaling is done in such way that a new dimensional system factorizes those term and document vectors together along a new system of coordinate axes that appear frequently together. Typically, the resulting coordinate system is an approximation of the original vector space, deliberately neglecting the term and document loadings onto lower ranking factors.

The assumption behind this truncation of lower ranking factors is that it compensates for synonymy and other forms of word variation with same intended meaning, as this is suppressed through the approximation. At the same time it is assumed that the factorisation as such deals with polysemy and homonymy, splitting differing usage contexts across different factors or combinations thereof. This way, the resulting higher-order LSA vector space is assumed to better reflect the (latent) semantic structure previously obscured by the variability in word use.

In more detail, the following steps need to be conducted to produce a latent semantic analysis. First, a document-term matrix M is constructed from a given collection of n documents containing m terms (see Fig. 4.1). This matrix, also called 'text matrix', has m rows (representing the $t_1, t_2, \ldots, t_m$ terms) and n columns (representing the $d_1, d_2, \ldots, d_n$ documents of the corpus), thus denoting in its cells the frequency with which each term appears in each document. This text matrix M is typically sparse, i.e. few of the cells contain values greater than 0. The text matrix M holds the basic vector space of the corpus.

This text matrix M of size $m \times n$ is then resolved with *singular value decomposition* into three constituent matrices T, S, and D, such that their product is M (see Fig. 4.2). The constituent T thereby holds the left-singular vectors (term vector

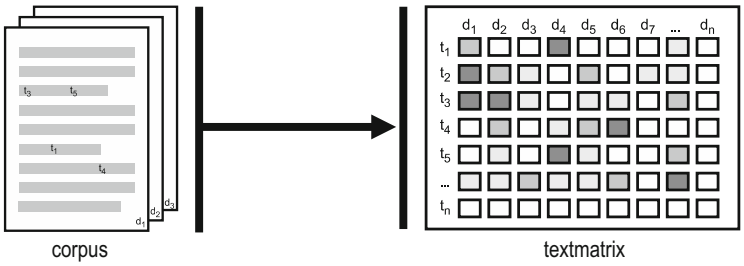

Fig. 4.1 Mapping of document collection to document-term matrix

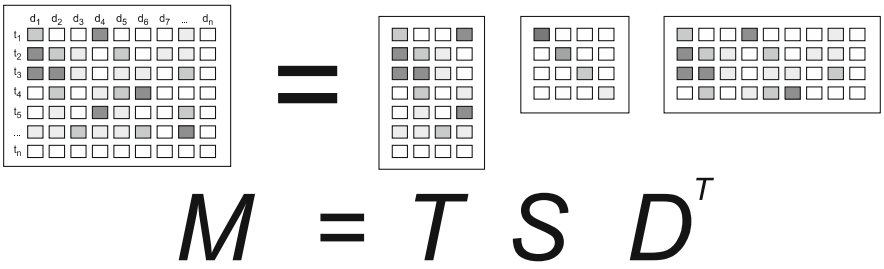

Fig. 4.2 Singular value decomposition (SVD, own graphic, modified from Wild et al. 2005b)

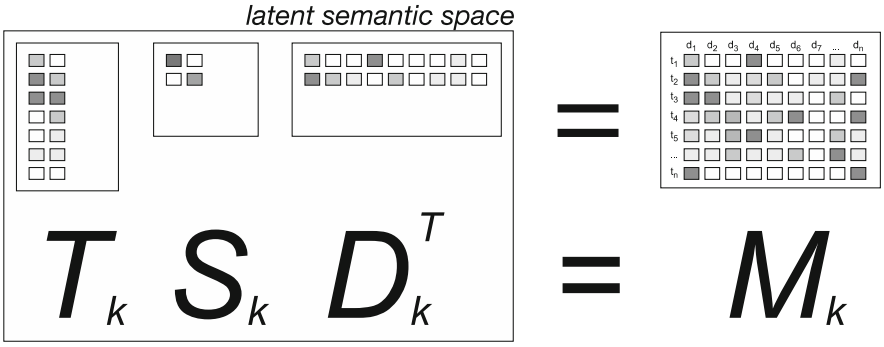

Fig. 4.3 Factor reduction (own graphic, modified from Wild et al. 2005b)

'loadings' onto the singular values in S) and the constituent D holds the right-singular vectors (the document vector 'loadings'). S is the orthonormal, diagonal matrix, listing the square roots of the eigenvalues of MM^T and M^TM (in descending order).

The diagonal matrix S is subsequently truncated to k diagonal values, effectively truncating T to T_k and D to D_k as well (see Fig. 4.3). This set of truncated matrices T_k, S_k, and D_k establishes the latent semantic space—the least-squares best-fit approximation of M with k singular values.

Multiplying T_k, S_k, and D_k^T produces the text matrix M_k, which is of the same format as M: rows representing the same terms and columns representing the same documents. The cells, however, now contain corrected frequencies that better reflect the (latent) semantic structure.

Since M_k is no longer sparse (other than M), in many cases it can be more memory efficient, to just hold the truncated space matrices to produce only those text matrix vectors in M_k that are actually needed.

The computationally costly part of the latent semantic analysis process is the step applying in the singular value decomposition. With rising corpus size through a larger number of documents and the typically resulting larger number of terms connected to it, the computation has a complexity of up to $O(mn \min\{m,n\})$, depending on the actual algorithm and basic linear algebra implementations used (see Menon and Elkan 2011, for a comparison of SVD algorithm complexity).

To avoid the bulk of calculations for the singular value decomposition, it is possible to project new documents into an existing latent semantic space—a process called 'folding in' (Berry et al. 1995, p. 577). This is particularly useful, where it is necessary to keep additional documents from changing the previously calculated factor distribution—for example, when evaluating low-scored student essays. Given that the reference corpus from which the latent semantic analysis is calculated were representative, such projection produces identical results.

To conduct a *fold-in*, the following equations need to be resolved [see Berry et al. 1995, p. 577, particularly Eq. (7)]. First, a document vector v needs to be constructed for the additional documents, listing their term occurrence frequencies along the controlled (and ordered!) vocabulary provided by T_k (and shared by M and M_k). This document vector v is effectively an additional column to the input text matrix M. The projections target document vector m' is then calculated by applying Eqs. (4.1) and (4.2), effectively mapping v to a new right-singular vector d' in D_k [Eq. (4.1)]—and then to a document-term vector m' in M_k [Eq. (4.2)].

$$d' = v^{\mathrm{T}} T_k {S_k}^{-1} \tag{4.1}$$

$$m' = T_k S_k {d'}^T \tag{4.2}$$

T_k and S_k thereby refer to the truncated space matrices from the existing latent semantic space.

Using fold in or not, the latent semantic space and its vectors allows for several ways to conduct *proximity measurement* of how close certain documents, terms, or documents and terms are.

Evaluations can be performed utilising the truncated partial matrices of the latent semantic space or—less memory efficient—in the re-multiplied text matrix M_k that reflects the underlying latent semantic structure. Same as in the 'pure' vector space model, various proximity measurement algorithms can be used: the cosine, for example, utilises the angle between vectors to provide a measure for their relatedness [see Fig. 4.4 and Eq. (4.3)].

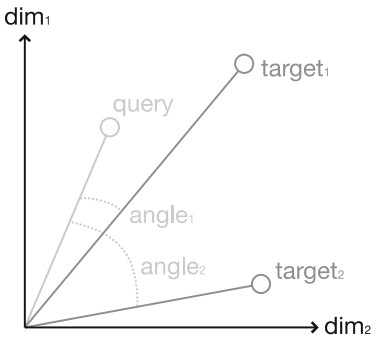

Fig. 4.4 Cosine proximity measurement of a query vector to two target vectors

When performed over the reconstituted text matrix M_k, the dimensions depicted in the figure relate to the terms (for term-to-term comparisons) and documents (for document-to-document comparisons). When performed in the latent semantic space, the dimensions refer to the factors of the singular value decomposition.

Other popular measures (see Leydesdorff 2005; Klavans and Boyack 2006; Tao and Zhai 2007) include, for example, Pearson's r, Euclidian distances, and the Jaccard coefficient. One of the advantages of the cosine measure is the reduced sensitivity for zeros (Leydesdorff 2005), a difference coming to effect particularly for large, sparse textmatrices.

$$\cos \propto = \frac{\sum_{i=1}^{m} a_i b_i}{\sqrt{\sum_{i=1}^{m} a_i^2} \sqrt{\sum_{i=1}^{m} b_i^2}} \tag{4.3}$$

The interpretation of value ranges provided by any of the measures depends largely on the underlying data. Given that the latent semantic space is valid in providing a representation of the meaning structures under investigation, high proximity values in the vector space between terms, documents, or both indicate associative closeness.

Only very high values indicate identity, whereas lower positive proximity values can be seen as to indicate whether certain features are associated, i.e., whether they are likely to appear in the same contexts.

For example, although the words 'wolf' and 'dog' are *semantically* very close, in a generic newspaper corpus, however, they cannot be expected to be *associatively* very close (not least to the widespread metaphoric use of 'wolf' and the rare use of wolfs as pets): it is much more likely that 'dog' will be found in closer proximity to any other popular pet (such as 'cat'), though not identical to them.

Landauer and Dumais (1997) demonstrated, that latent semantic spaces can be trained to perform a synonym test on a level required for the admission to U.S. universities in the Test Of English as a Foreign Language (TOEFL). Their findings, however, should not be overrated: it remains largely a question of selecting and sampling a representative corpus (and space) for the domain of interest.

4.2 Analysis Workflow with the R Package 'lsa'

In support of this book, the author has implemented the *lsa* package for R as Open Source (Wild 2014). This subsection provides an overview on the functionality covered by the package. More detailed documentation of the individual package routines can be found in Annex C.

The workflow of an analyst applying latent semantic analysis is depicted in Fig. 4.5. An 'analyst' thereby refers to any user of LSA, who applies the technique to investigate meaning structures of texts, for example being learner, tutor, teacher, faculty administrator, system designer, researcher, or the like.

Typically, analysis involves a set of filtering and other pre-processing operations (weighting, sanitising) in order to select and prepare the document collection to map to a text matrix, the calculation of the LSA space (singular value decomposition and factor truncation), and then—subsequently—the application of similarity measurement operations in interchange with further filtering of document and term vectors for the desired scope of analysis. This latter analysis may or may not involve folding in of additional documents.

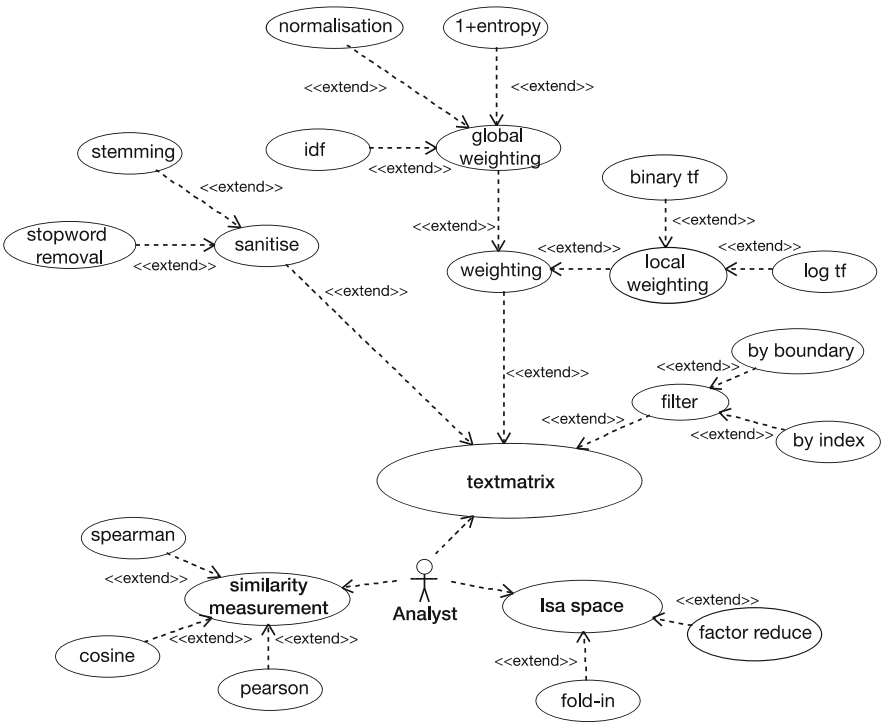

Fig. 4.5 Analysis use cases

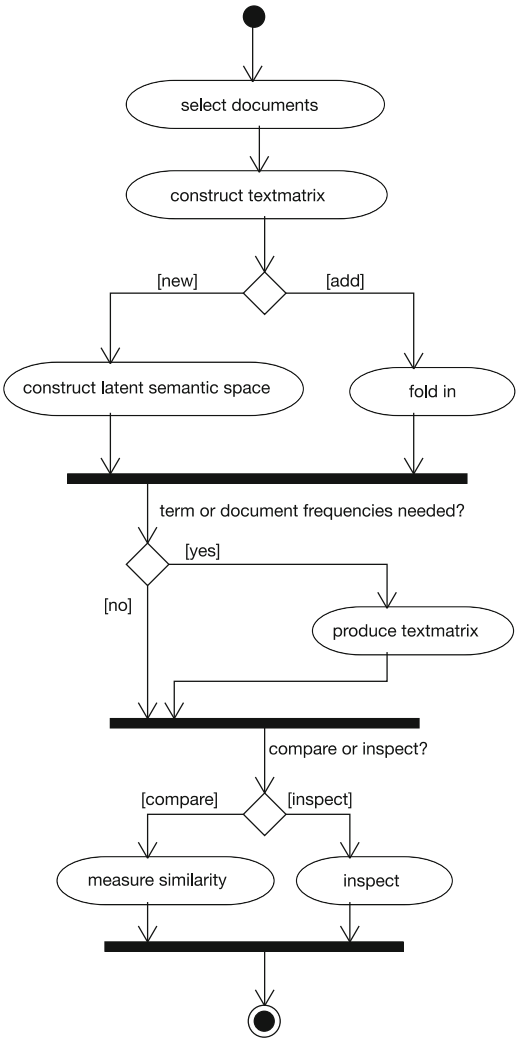

Fig. 4.6 Activity diagram of a simple latent semantic analysis

Both the text matrix operations as well as the factor reduction of the lsa space creation offer a wide choice in configuration options. Their interdependency will be further investigated in the Chaps. 5 and 7 (Fig. 4.6).

Although the number of use cases depicted in Fig. 4.5 looks complex, their application in the actual analysis is not. Typically, an analyst starts off with selecting the document collection to be analysed, then constructs the 'raw' text matrix to then construct the latent semantic space. Depending on the scope of analysis the factor reduced text matrix M_k can be produced (or not). The analysis of the resulting vector space then is conducted with a mixture of or either of similarity measurements and frequency inspections. The resulting frequencies

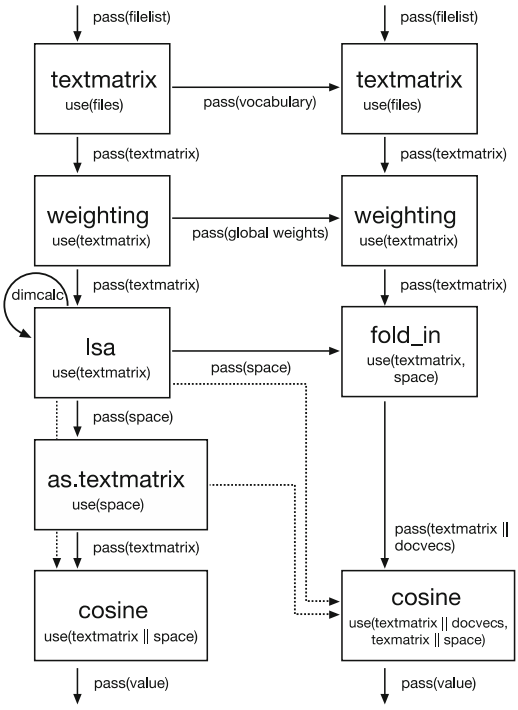

Fig. 4.7 Workflow data handling

thereby reflect the 'term activations' of each document vector in M_k and/or—in case of adding documents with fold-ins—m'.

The handling of the data required in the analysis steps involving fold-ins is thereby not trivial. Fig. 4.7 uses workflow data patterns (following the recommendation of Russell et al. 2004) to show what data are passed on in between the different functional activities.

The left hand side depicts the standard workflow, whereas on the right hand side, the fold in workflow is described. The standard workflow starts with parsing a list of files into a text matrix, weighting it, to then construct a latent semantic space. The fold-in process is dependent on this: the text matrix construction requires the controlled, ordered vocabulary of the original text matrix (otherwise projection is prevented). Moreover and noteworthy, if a global weighting schema was applied, then the resulting global weights have to be handed over to be used for the global weighting of the new text matrix to be projected into the existing latent semantic space. Bot these data hand-overs are depicted in the figure through a 'pass' statement.

When turning to the similarity measurement and text matrix production, it is evident, that either the space or the reconstructed text matrix are needed to allow for comparison and inspection.

More details on data handling are provided in Annex C, the 'lsa' package documentation and source code.

4.3 Foundational Example

Turning back to the foundational example introduced in the previous chapter (Sect. 3.2), the human resource manager of the large multinational decides now to look deeper into the contents of the learning activity the ten employees under scrutiny have been involved in. Therefore, the human resource manager has asked the nine employees to write a short memo about the fourteen trainings, summarising the learning experience. Luckily, Christina, who is still off sick, produced such memo already for her last career development meeting with the human resource manager.

To keep this example simple, only the document titles will be used. The example is slightly more complex[1] than the standard one used repeatedly in the introductory papers of the Bellcore group (e.g. in Deerwester et al. 1990, p. 396).

This is to illustrate better, what indexing and classification with latent semantic analysis looks like in the context of the example introduced above in the social network analysis chapter. Still it is small enough to follow.

The personnel provide fourteen memos for the fourteen different learning opportunities already introduced in the previous chapter. To keep it simple, only the document titles are used. Moreover, all fourteen documents visible fall into three subject areas—i.e. computing ('c'), mathematics ('m'), and pedagogy ('p')—and they are accordingly labelled and numbered. This is to illustrate the working principle of latent semantic analysis, as in a real life case, such classification would be known only *ex post*, following the analysis.

These document titles are pre-processed, such that only those terms are selected that appear in more than one document title (the underlined terms in the Table 4.1). The documents are filed in and converted to a document-term matrix using the `textmatrix` function of the *lsa* package, just as shown in Listing 1.

Listing 1 Reading text files into a document-term matrix.

```
dtm = textmatrix("lsa-example/", minWordLength = 1)
```

The result *dtm* is a sparsely populated text matrix such as the one depicted in Table 4.2. The order of terms and documents can vary slightly when running this example on different machines as it is basically constituted following the order of appearance in the documents (which again is driven by the file system ordering provided by the operating system used). Reshuffling rows and columns in this matrix is of course possible using the native R matrix manipulation routines. Since this has no impact whatsoever on the singular value decomposition, it will not be demonstrated here.

Listing 2 Singular-value decomposition.

```
space = lsa(dtm, dims = dimcalc_raw())
```

[1] 14 instead of 9 documents.

Table 4.1 Memos about the learning experiences

Titles
c1: A web interface for social media applications
c2: Review of access time restrictions on web system usage
c3: Content management system usage of the HTML 5 interface
c4: Error spotting in HTML: social system versus software system
c5: Barriers to access and time spent in social mobile apps
m1: The generation of random unordered trees
m2: A survey of divisive clustering along the intersection of partial trees
m3: Width and height of trees in using agglomerative clustering with Agnes
m4: Agglomerative clustering algorithms: a review
p1: The intersection of learning and organisational knowledge sharing
p2: A transactional perspective on teaching and learning
p3: Innovations in online learning: moving beyond no significant difference
p4: Tacit knowledge management in organisational learning
p5: Knowledge building: theory, pedagogy, and technology

Table 4.2 Document-term matrix

	c1	c2	c3	c4	c5	m1	m2	m3	m4	p1	p2	p3	p4	p5
interface	1	0	1	0	0	0	0	0	0	0	0	0	0	0
social	1	0	0	1	1	0	0	0	0	0	0	0	0	0
web	1	1	0	0	0	0	0	0	0	0	0	0	0	0
access	0	1	0	0	1	0	0	0	0	0	0	0	0	0
review	0	1	0	0	0	0	0	0	1	0	0	0	0	0
system	0	1	1	2	0	0	0	0	0	0	0	0	0	0
time	0	1	0	0	1	0	0	0	0	0	0	0	0	0
usage	0	1	1	0	0	0	0	0	0	0	0	0	0	0
html	0	0	1	1	0	0	0	0	0	0	0	0	0	0
management	0	0	1	0	0	0	0	0	0	0	0	0	1	0
trees	0	0	0	0	0	1	1	1	0	0	0	0	0	0
clustering	0	0	0	0	0	0	1	1	1	0	0	0	0	0
intersection	0	0	0	0	0	0	1	0	0	1	0	0	0	0
agglomerative	0	0	0	0	0	0	0	1	1	0	0	0	0	0
knowledge	0	0	0	0	0	0	0	0	0	1	0	0	1	1
learning	0	0	0	0	0	0	0	0	0	1	1	1	1	0
organisational	0	0	0	0	0	0	0	0	0	1	0	0	1	0

In the next step, this text matrix *dtm* is resolved using the singular-value decomposition, effectively resulting in the three partial matrices listed in Tables 4.3, 4.4, and 4.5. Typically, the resulting three 'space' matrices are immediately truncated to the desired number of factors. Together SVD and truncation from the core of the LSA process, the *lsa* package encapsulates them therefore in the `lsa` function.

Table 4.3 The term loadings T on the factors

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}
interface	−0.21	0.00	−0.06	0.14	−0.01	−0.69	0.05	−0.07	0.08	0.16	−0.22	0.14	−0.19	0.12
social	−0.31	0.07	−0.03	−0.06	0.69	−0.16	0.32	0.12	−0.09	0.09	0.08	−0.10	−0.16	0.21
web	−0.23	0.05	0.05	−0.31	0.03	−0.38	0.02	−0.25	0.38	−0.39	0.19	−0.05	0.36	−0.33
access	−0.23	0.06	0.08	−0.47	0.06	0.09	−0.16	0.14	−0.28	0.14	−0.08	0.06	0.06	0.14
review	−0.19	0.01	0.25	−0.28	−0.35	0.13	0.25	−0.07	0.26	0.01	0.15	−0.26	−0.68	0.00
system	−0.65	0.08	−0.08	0.31	0.01	0.44	−0.02	−0.09	0.15	−0.20	0.08	−0.01	0.18	0.10
time	−0.23	0.06	0.08	−0.47	0.06	0.09	−0.16	0.14	−0.28	0.14	−0.08	0.06	0.06	−0.35
usage	−0.31	0.02	0.01	−0.08	−0.43	−0.15	−0.29	−0.06	−0.02	0.07	−0.22	0.20	0.08	0.44
html	−0.32	0.02	−0.09	0.38	−0.01	0.06	0.01	0.04	−0.08	0.18	−0.17	0.09	−0.19	−0.40
management	−0.18	−0.23	−0.12	0.17	−0.31	−0.28	−0.04	0.27	−0.38	0.09	0.39	−0.36	0.07	−0.26
trees	−0.01	−0.12	0.5	0.19	0.21	−0.10	−0.47	−0.05	−0.27	−0.50	0.04	0.04	−0.30	0.00
clustering	−0.04	−0.12	0.62	0.14	−0.01	−0.01	0.16	0.03	0.03	0.26	0.01	−0.38	0.40	0.26
intersection	−0.02	−0.29	0.17	0.03	0.22	0.05	−0.41	−0.11	0.44	0.53	0.01	0.07	−0.04	−0.26
agglomerative	−0.03	−0.06	0.43	0.07	−0.14	0.01	0.49	0.11	−0.10	−0.03	−0.13	0.56	0.10	−0.26
knowledge	−0.05	−0.51	−0.11	−0.08	0.02	0.03	0.03	0.53	0.27	−0.30	−0.49	−0.18	0.01	0.00
learning	−0.05	−0.59	−0.13	−0.11	0.03	0.09	0.20	−0.66	−0.32	−0.01	−0.17	−0.07	0.00	0.00
organisational	−0.04	−0.45	−0.09	−0.07	0.01	0.02	0.02	0.19	0.09	0.01	0.59	0.46	−0.03	0.26

Table 4.4 The document 'loadings' *D* on the factors

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}
c1	−0.22	0.04	−0.01	−0.10	0.39	−0.75	0.27	−0.16	0.30	−0.15	0.06	−0.03	0.02	0.00
c2	−0.55	0.10	0.15	−0.59	−0.35	0.14	−0.24	−0.15	0.17	−0.23	0.06	0.00	0.13	0.00
c3	−0.50	−0.04	−0.13	0.42	−0.42	−0.38	−0.19	0.07	−0.20	0.31	−0.22	0.10	−0.10	0.00
c4	−0.58	0.09	−0.11	0.42	0.39	0.48	0.20	−0.02	0.11	−0.13	0.10	−0.05	0.02	0.00
c5	−0.23	0.06	0.05	−0.45	0.45	0.01	0.01	0.33	−0.51	0.37	−0.12	0.03	−0.11	0.00
m1	0.00	−0.04	0.19	0.09	0.12	−0.06	−0.32	−0.04	−0.22	−0.51	0.06	0.09	−0.72	0.00
m2	−0.02	−0.19	0.50	0.16	0.23	−0.04	−0.49	−0.10	0.16	0.29	0.09	−0.50	0.13	0.00
m3	−0.02	−0.10	0.60	0.18	0.03	−0.06	0.12	0.08	−0.27	−0.28	−0.13	0.43	0.46	0.00
m4	−0.07	−0.06	0.50	−0.04	−0.28	0.08	0.60	0.06	0.15	0.24	0.04	−0.14	−0.42	0.00
p1	−0.05	−0.65	−0.06	−0.11	0.16	0.12	−0.12	−0.03	0.39	0.22	−0.08	0.53	−0.14	0.00
p2	−0.02	−0.21	−0.05	−0.05	0.02	0.05	0.13	−0.53	−0.25	−0.01	−0.26	−0.14	0.01	0.71
p3	−0.02	−0.21	−0.05	−0.05	0.02	0.05	0.13	−0.53	−0.25	−0.01	−0.26	−0.14	0.01	−0.71
p4	−0.09	−0.62	−0.18	−0.04	−0.13	−0.08	0.14	0.27	−0.27	−0.21	0.48	−0.29	0.13	0.00
p5	−0.01	−0.18	−0.04	−0.04	0.01	0.02	0.02	0.43	0.21	−0.31	−0.72	−0.34	0.01	0.00

Table 4.5 The singular values *S*

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}
1	3.36	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	2.85	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	2.58	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	2.21	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1.79	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1.63	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1.48	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1.25	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1.24	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0.98	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0.67	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0.53	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0.42	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Listing 2 shows how this function is used to retain all available singular values, passing through the `dimcalc_raw` function to the dimensionality selection interface. The result *space* is a list with three slots containing the matrices decomposing *dtm* into the matrices *T*, *S*, and *D* as shown above in Fig. 4.2. The matrix *T* (accessible via *space$tk*) thereby holds the term 'loadings' onto the factors.

The partial matrix *D* (access via *space$dk*) contains the document 'loadings' onto the factors, with the columns holding the right singular vectors of the matrix decomposition. The according matrix from the example is presented in Table 4.4. The list *S* then contains the singular values of the decomposition, sorted descending. The values in *S* constitute a diagonal matrix depicted in Table 4.5.

As already indicated, this set of matrices (aka the 'latent semantic space') can be used to reconstruct the original document-term matrix, using the type casting operator `as.textmatrix` provided in the package, which effectively re-multiplies the three partial matrices again (as indicated in the second line in Listing 3).

Listing 3 Reconstruction of the document-term matrix from the space.

```
X = as.textmatrix(space)
X = space$tk %*% diag(space$sk) %*% t(space$dk)
```

To confirm, whether the re-multiplication in fact reconstructed the original document-term matrix, all elements in the reconstructed matrix *X* can be compared with all elements in *dtm* (values are rounded to three digits to avoid impact of minimal rounding errors resulting from the decomposition).

Listing 4 Confirming whether reconstruction succeeded.

```
X = round(X, 3)
all((dtm == X) == TRUE)
## [1] TRUE
```

The 'trick' of LSA is to factor-reduce this space, i.e. to eliminate those lower-ranking factors that obscure the semantic structure, while retaining those high-ranking factors that constitute the differences. In our example, a useful number of factors to retain is three.

Listing 5 Truncating the space.

```
space_red = lsa(dtm, dims = 3)
```

This reduced space now reflects better the semantic structure than the original document-term matrix, as can be seen from the following table which depicts the re-multiplied matrix. To be a bit more precise: this truncated space reflects better the 'latent semantic' structure contained in the documents—and of course construed by the boundaries of the semantics surfacing in this example (Table 4.6).

Table 4.6 Reconstructed document-term matrix of the factor-reduced space

	c1	c2	c3	c4	c5	m1	m2	m3	m4	p1	p2	p3	p4	p5
interface	0.2	0.4	0.4	0.4	0.2	0.0	−0.1	−0.1	0.0	0.0	0.0	0.0	0.1	0.0
social	0.2	0.6	0.5	0.6	0.2	0.0	−0.1	0.0	0.0	−0.1	0.0	0.0	0.0	0.0
web	0.2	0.5	0.4	0.4	0.2	0.0	0.1	0.1	0.1	−0.1	0.0	0.0	0.0	0.0
access	0.2	0.5	0.4	0.4	0.2	0.0	0.1	0.1	0.1	−0.1	0.0	0.0	−0.1	0.0
review	0.1	0.4	0.2	0.3	0.2	0.1	0.3	0.4	0.4	0.0	0.0	0.0	−0.1	0.0
system	0.5	1.2	1.1	1.3	0.5	0.0	−0.1	−0.1	0.0	0.0	0.0	0.0	0.1	0.0
time	0.2	0.5	0.4	0.4	0.2	0.0	0.1	0.1	0.1	−0.1	0.0	0.0	−0.1	0.0
usage	0.2	0.6	0.5	0.6	0.2	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.1	0.0
html	0.2	0.6	0.6	0.6	0.2	0.0	−0.1	−0.1	0.0	0.0	0.0	0.0	0.1	0.0
management	0.1	0.2	0.4	0.3	0.1	0.0	0.0	−0.1	−0.1	0.5	0.2	0.2	0.5	0.1
trees	0.0	0.2	−0.1	−0.1	0.1	0.3	0.7	0.8	0.7	0.1	0.0	0.0	0.0	0.0
clustering	0.0	0.3	−0.1	−0.1	0.1	0.3	0.9	1.0	0.8	0.1	0.0	0.0	−0.1	0.0
intersection	0.0	0.0	0.0	−0.1	0.0	0.1	0.4	0.4	0.3	0.5	0.2	0.2	0.4	0.1
agglomerative	0.0	0.2	−0.1	−0.1	0.1	0.2	0.6	0.7	0.6	0.0	0.0	0.0	−0.1	0.0
knowledge	0.0	−0.1	0.2	0.0	−0.1	0.0	0.1	0.0	0.0	1.0	0.3	0.3	1.0	0.3
learning	0.0	−0.1	0.2	0.0	−0.1	0.0	0.1	0.0	−0.1	1.1	0.4	0.4	1.1	0.3
organisational	0.0	−0.1	0.2	0.0	−0.1	0.0	0.1	0.0	0.0	0.8	0.3	0.3	0.8	0.2

Table 4.7 Proximity matrix for the original vector space (rounded)

	c1	c2	c3	c4	c5	m1	m2	m3	m4	p1	p2	p3	p4	p5
c1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
c2	0	1	0	0	0	0	0	0	0	0	0	0	0	0
c3	0	0	1	1	0	0	0	0	0	0	0	0	0	0
c4	0	0	1	1	0	0	0	0	0	0	0	0	0	0
c5	0	0	0	0	1	0	0	0	0	0	0	0	0	0
m1	0	0	0	0	0	1	1	1	0	0	0	0	0	0
m2	0	0	0	0	0	1	1	1	0	0	0	0	0	0
m3	0	0	0	0	0	1	1	1	1	0	0	0	0	0
m4	0	0	0	0	0	0	0	1	1	0	0	0	0	0
p1	0	0	0	0	0	0	0	0	0	1	0	0	1	0
p2	0	0	0	0	0	0	0	0	0	0	1	1	0	0
p3	0	0	0	0	0	0	0	0	0	0	1	1	0	0
p4	0	0	0	0	0	0	0	0	0	1	0	0	1	0
p5	0	0	0	0	0	0	0	0	0	0	0	0	0	1

The document ‘c3’ contains now, for example, also a value of 0.4 for the term ‘web’, which previously was not there: the document is about the HTML interface of a content management system, but does not use the term ‘web’ in its title.

The effect of this truncation becomes even more evident, when looking at proximity relations between documents. The proximity of the documents is calculated as follows: in the first line for the original, non-truncated vector space (as already established in *dtm*); in the second line for the truncated space (Tables 4.7 and 4.8).

Listing 6 Calculating cosine proximities.

```
proximity = cosine(dtm)
proximitySpaceRed = cosine(as.textmatrix(space_red))
```

When looking at the proximity table of the documents in the original, unreduced vector space and compare them with the proximity of documents in the factor-reduced space, the difference becomes clearly visible: the ‘computing’ documents (starting with ‘c’) can be much better be differentiated for the latter space from the ‘math’ (starting with ‘m’) and ‘pedagogy’ documents (starting with ‘p’). Moreover, the computing, math, and pedagogy documents respectively have become more similar within their own groups—and more dissimilar from each other.

Since this example uses only three factors, we can use the factor loadings of both documents and terms to draw a 3D perspective plot.

Table 4.8 Proximity table for the factor-reduced space (rounded)

	c1	c2	c3	c4	c5	m1	m2	m3	m4	p1	p2	p3	p4	p5
c1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
c2	1	1	1	1	1	0	0	0	0	0	0	0	0	0
c3	1	1	1	1	1	0	0	0	0	0	0	0	0	0
c4	1	1	1	1	1	0	0	0	0	0	0	0	0	0
c5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
m1	0	0	0	0	0	1	1	1	1	0	0	0	0	0
m2	0	0	0	0	0	1	1	1	1	0	0	0	0	0
m3	0	0	0	0	0	1	1	1	1	0	0	0	0	0
m4	0	0	0	0	0	1	1	1	1	0	0	0	0	0
p1	0	0	0	0	0	0	0	0	0	1	1	1	1	1
p2	0	0	0	0	0	0	0	0	0	1	1	1	1	1
p3	0	0	0	0	0	0	0	0	0	1	1	1	1	1
p4	0	0	0	0	0	0	0	0	0	1	1	1	1	1
p5	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Listing 7 Generating the perspective plot of terms and documents.

```
p = persp(
   x = -1:1,y = -1:1,
   z = matrix(
      c(
       -1,0,1,
       -1,0,1,
       1,0,-1
      ),3,3),
   col = "transparent", border = "transparent",
   xlim = range(c(space$dk[,1], space$tk[,1])),
   ylim = range(c(space$dk[,2], space$tk[,2])),
   zlim = range(c(space$dk[,3], space$tk[,3])),
   theta = 35, phi = 20,
   xlab = "dim 1", ylab = "dim 2", zlab = "dim 3",
   expand = 0.5, scale = F,
   axes = TRUE, nticks = 10, ticktype = "simple"
)

points(trans3d(space$dk[,1], space$dk[,2],
   space$dk[,3], pmat = p), bg = "red", col = "red",
   pch = 22, cex = 1)

points(trans3d(space$tk[,1], space$tk[,2],
   space$tk[,3], pmat = p), bg = "blue", col = "blue",
   pch = 21, cex = 1)
```

```
text(trans3d(space$tk[,1], space$tk[,2],
   space$tk[,3], pmat = p), rownames(space$tk),
   col = "blue", cex = 0.8)

text(trans3d(space$dk[,1], space$dk[,2],
   space$dk[,3], pmat = p), rownames(space$dk),
   col = "red", cex = 0.8)
```

The code of Listing 7 thereby first creates an empty perspective plot using `persp` (to ensure that the limits of all axes are set up so that the projected points are all visible). Then the two commands `points` and `text` are used to plot the term and document positions (and according labels) into the perspective plot.

The resulting visualisation (Fig. 4.8) shows, how the factors separate the three clusters of documents and terms: to the top right, the math documents and math-related terms cluster together; in the bottom right corner, the pedagogy-related documents and terms; and in the top left corner, the computing ones. The axes thereby represent the new base of the Eigensystem resulting from the singular value decomposition.

One term is depicted in between the two clusters of pedagogy and computing, i.e. 'management'. This term is found in both a computing as well as a pedagogy document—bridging between the two clusters.

Calculation of spaces can be a resource intense endeavour, preventing recalculation in real-time: a space with a few million documents and terms can easily take one or more hours on a fast machine with big memory (not speaking of how long it can take on a slow machine with scarce memory).

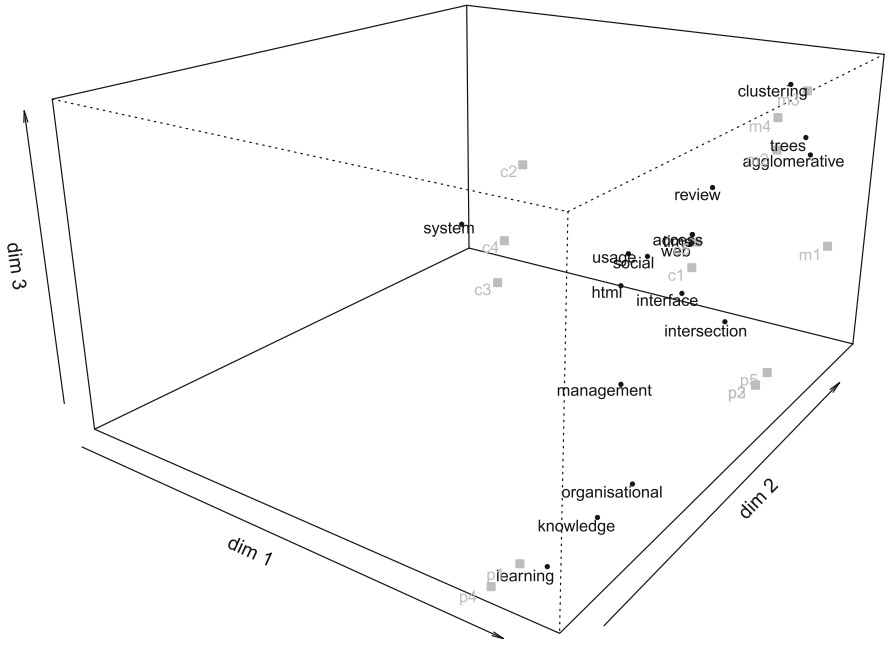

Fig. 4.8 Perspective plot of terms and documents (three factors)

There is, however, an efficient updating method that can be used to project new data into an existing space. Given that the space captures its semantic, this projection is almost lossless. Moreover, it prevents influencing the semantics of a given space, thus ensuring stability its semantics and validity.

To update, new data can be filed in (reusing the vocabulary of the existing document-term matrix, to ensure that the new data can be projected into the existing latent semantic space).

Listing 8 Reading additional data with a controlled vocabulary.

```
data = "Review of the html user interface of the system"
pdoc = query(data, rownames(dtm))
```

The title of the new document 'c6' to add is "Review of the html user interface of the system", which results in the following column vector (Table 4.9).

This new vector can be projected into the latent-semantic space using the `fold_in` procedure.

Listing 9 Folding in.

```
newY = fold_in(pdoc, space_red)
```

Once this is done, comparisons of the new document with the existing ones become possible. Therefore, the new column vector that was just folded in is bound to the reconstructed document-term matrix of the factor-reduced space—to then calculate all cosine proximities (see Listing 10).

Listing 10 Calculating proximity to the existing documents.

```
allY = cbind(newY, as.textmatrix(space_red))
allCos = cosine(allY)
allCos["c6",]
```

Table 4.9 'Raw' document vector of an additional document

	c6
interface	1
social	0
web	0
access	0
review	1
system	1
time	0
usage	0
html	1
management	0
trees	0
clustering	0

Table 4.10 Proximity of 'c6' to the existing documents (rounded to one digit)

	c6	c1	c2	c3	c4	c5	m1	m2	m3	m4	p1	p2	p3	p4	p5
c6	1	1	1	1	1	1	0	0	0	0.2	0	0	0	0.1	0

As visible from the last table, the new document is evaluated to be very close to the computing documents (and far from the math and pedagogy ones)—just as expected (Table 4.10).

4.4 State of the Art in the Application of LSA for TEL

While the application focus on essay scoring is taken up more widely over the next decade in the late 1990s and early 2000s with prototypes and original research contributed from research groups around the globe, a period of theoretical reflection follows in the research group in Colorado, ultimately leading up to the formulation of LSA as a representation theory.

In parallel, the mathematical/statistical foundation introduced above in Sect. 4.1 are further elaborated and generic implementations of the underlying linear algebra routines and its application in LSA/LSI are made available. First derivate algorithms such as probabilistic LSA and topic models arrive. Moreover, the parametric configuration of LSA such as applied in weighting schemes or dimensionality selection are starting to get investigated.

In this section, the state of the art of LSA in technology-enhanced learning and as a representation theory will be described. The mathematical and statistical foundations, as well as selected extensions were already described above in Sect. 4.1. The state of the art on parametric configuration of LSA and the derived MPIA is documented in Chap. 7: 'Calibrating for specific domains'.

To summarise the first area of growth in the body of literature, i.e. essay scoring, the following can be stated.

From the original research group at Bell Communications Research holding the patent, Dumais, Landauer, and Streeter/Lochbaum are the most active members, spinning off their own research teams.

Dumais focuses more, though not exclusively on the information retrieval aspects (as evident in Dumais et al. 1988, p. 282; Dumais 1991, 1992, p. 230). Dumais sometimes uses the notion 'LSI' for the application of LSA in indexing (Dumais 1995, p. 219). Dumais further investigates other information access problems such as categorisation and question answering using latent semantic analysis (see Dumais 2003, for a summary).

Landauer teams up with Foltz[2] and Laham in pursuing educational applications, reflected in multiple co-authorships as well as in the joint spin off company Knowledge Assessment Technologies (today: Pearson Knowledge Technologies):

[2] Landauer and Foltz first met at Bellcore: http://kt.pearsonassessments.com/whyChooseUs.php

as already introduced above, Landauer et al. (1997) provide experimental results on automated essay scoring; Landauer et al. 1998b, p. 48). Laham is based at the time—same as Landauer—at the University of Colorado at Boulder, whereas Foltz is based at New Mexico State University. Other prominent members of the LSA research group at the University of Colorado are Kintsch, Rehder, and Schreiner (see e.g. Landauer et al. 1997; Foltz et al. 1998; Wolfe et al. 1998; Rehder et al. 1998).

Streeter and Lochbaum both start publishing on further experiments and prototypes together with their army counter part Psotka in connection with their contract-research to develop tutoring systems for the army (Lochbaum et al. 2002; Lochbaum and Streeter 2002; Psotka et al. 2004; Streeter et al. 2002).

Of the remaining co-authors of the original patent, Furnas focused more on information visualisation research in his subsequent career (see his publication list[3]). Deerwester continued to Hongkong University of Science and Technology. He holds another patent on a related technology (indexing collections of items: Deerwester 1998) and filed another application (on optimal queries: Deerwester 2000), but stopped publishing about LSA-related topics and, ultimately, moved into a different career (see his Wikipedia entry[4]). Harshman focused before as after more on data analysis aspects (see publication list[5] and obituary Sidiropoulos and Bro 2009).

Regarding additional, new research groups, the following can be concluded. At the very end of the 1990ies, the Tutoring Research Group at the University of Memphis is formed around Graesser, visible often with co-authorship of P. -Wiemer-Hastings (aka Hastings in more recent years) and K. Wiemer-Hastings. They develop AutoTutor (and other prototypes and products): Wiemer-Hastings et al. (1998) and Graesser et al. (1999) introduce into the system architecture and way of functioning of AutoTutor, thereby putting more emphasis on tutorial dialogue moves as in the earlier works from the Bellcore-affiliated research groups. This is driven by the research interest of the group, as evident e.g. in Graesser et al. (1995), an extensive study of actual (human) tutorial dialogues and protocols.

With the beginning of the twenty-first century, the community using LSA in technology-enhanced learning quickly broadens: around 1999/2000, a French research group starts publishing at the Université Pierre-Mendès-France in Grenoble around Dessus and Lemaire follows, building the research prototype 'Assistant for Preparing Exams' (Apex, Dessus and Lemaire 1999; Dessus et al. 2000, p. 66ff; Lemaire and Dessus 2001, p. 310ff; Dessus and Lemaire 2002). Later, additional prototypes such as Apex-2 (Trausan-Matu et al. 2008, p. 46ff) and Pensum (Trausan-Matu et al. 2009, 2010, p. 9, 21ff) will follow.

[3] http://furnas.people.si.umich.edu/BioVita/publist.htm

[4] http://en.wikipedia.org/wiki/Scott_Deerwester

[5] http://psychology.uwo.ca/faculty/harshman/

In the UK, a research group at the Open University (Haley et al. 2003, 2005; Haley 2008) starts conducting LSA-based e-assessment research, building EMMA, in the context of the EC-funded eLeGi project (see Haley et al. 2007, p. 12).

In the Netherlands, a group at the Open University of the Netherlands around van Bruggen (2002) starts conducting essay scoring and question answering research. With Van Bruggen et al. (2004), they coin the term 'positioning' for assessing prior learning with the help of automated scoring techniques. Kalz et al. (2006) looks at how positioning can be done using LSA combined with a meta-data approach. van der Vegt et al. (2009) describe how to conduct placement experiments with the R implementation of the author of this book and with a PHP implementation made available.

With support from the Memphis group, van Lehn starts conducting essay-scoring research with LSA at the University of Pittsburgh, thereby creating Why2-Atlas (van Lehn et al. 2002) and CarmelTC (Rose et al. 2003).

At the University of Massachusetts, Larkey (1998) conducts essay-scoring experiments.

In Germany, Lenhard et al. (2007a, b, 2012), trained by the Kintschs and Landauer, builds up a research group at the University of Wuerzburg—focusing on a writing-trainer in German supported with automated feedback. The system is called conText.

In Austria, the author of this book Wild starts building up research around LSA and essay scoring (Wild et al. 2005a, b; Wild and Stahl 2007), thereby releasing the lsa package for R (Wild 2014), building an Essay Scoring Application (ESA) module for .LRN/OpenACS (Wild et al. 2007b; Koblischke 2007). Later this will lead to the joint development of LEApos and Conspect (LTfLL d4.3), when Wild moves on to the Open University. Conspect focuses on monitoring conceptual development from student writings, whereas LEApos sets focus on short answer scoring, but with a stronger focus on knowledge-rich technology with LSA playing a limited role.

In Australia, Williams and Dreher at the Curtin University of Technology in Perth start development of MarkIT (Williams 2001; Palmer et al. 2002; Williams and Dreher 2004, 2005; Dreher 2006).

In Germany, at the Goethe University of Frankfurt, Holten et al. (2010) start using LSA to assess professional expertise and application domain knowledge.

In Spain, a cross-university group forms staffed by members from mainly the Autonomous University of Madrid and from the National Distance Education University (UNED). From 2006 onwards, mainly Olmos, Jorges-Botano, and Leon publish about their findings (Leon et al. 2006; Olmos et al. 2009; Jorge-Botana et al. 2010a, b; Olmos et al. 2011). The system developed is named Gallito.

Several special events support community building over time. Besides the 1998 special issue introduced above, a second dedicated set of articles is published in 2000. It features seven articles (spread out across two issues). Landauer and Psotka (2000) introduce into the seven articles, thereby also providing an short introduction into LSA. Foltz et al. (2000) document different essay scoring methods implemented in a real system and its evaluation with undergraduates. Graesser

et al. (2000) describe how AutoTutor uses LSA for the dialogue interaction with the learners. Wiemer-Hastings and Graesser (2000) show how LSA is used in Select-a-Kibitzer to provide feedback on student writings. Kintsch et al. (2000) describe the systems and evaluation results of summary writing tools called 'Summary Street' and 'State the Essence'. Laham et al. (2000) describe a system for matching job and competence profiles with the help of LSA. Freeman et al. (2000) describe how to model expert domain knowledge from written short texts.

The Handbook on Latent Semantic Analysis (Landauer et al. 2008) brings together contributions from the original and extended circle.

In 2007, the 1st European Workshop on Latent Semantic Analysis takes place (Wild et al. 2007a), from which a European research group around Wild, Kalz, Koper, van Rosmalen, and van Bruggen forms, teaming up with Dessus for the EC-funded LTfLL project (funded from 2008 to 2011).

Several review articles and debates support spreading of research over time. Whittington and Hunt (1999) compare LSA with shallow surface feature detection mechanisms offered by PEG (Page 1966) and hybrid systems that take syntactical or even discourse-oriented features into account. They acknowledge LSA "impressive results" (p. 211), but at the same time criticise its need for large amounts of data and computational resources.

Hearst (2000) moderates a debate for the column 'Trends and Controversies' of IEEE Intelligent Systems, with position statements from Kukich (2000) (the director of the natural language processing group at Educational Testing Service), Landauer et al. (2000), and (MITRE, Hirschmann et al. 2000), rounded up with a comment by Calfee (Dean of the School of Education at UC Riverside). The message of the debate is clear: it is surprising how well the automated techniques work, but there is still "room for improvement" (Calfee 2000, p. 38). Yang et al. (2001) review validation frameworks for essay scoring research. Miller (2003) provides a review of the use of LSA for essay scoring and compares the technique against earlier, computer-based techniques. Valenti et al. (2003) review findings for ten different essay-scoring systems. They particularly criticize the lack of test and training collections, a problem common to all techniques surveyed. Dumais (2005) provides an extensive review over a wide range of LSA applications, including educational ones. Landauer and Foltz (2007) review the applications available at the University of Colorado and at Pearson Knowledge Technologies.

In parallel to the application-oriented research, basic research leads to establishing LSA as a representation theory.

Following the seminal articles listed above, research is conducted to reflect on the capabilities of LSA to be a semantic representation theory, starting the first attempt in Landauer and Dumais (1997): using Plato's poverty of stimulus problem,[6] they investigate the validity of LSA as a theory of "acquired similarity and knowledge representation" (p. 211). They come to the conclusion that LSA ability

[6] The poverty of stimulus problem is paraphrased as: "How do people know as much as they do with as little information as they get?" (Landauer and Dumais 1997, p. 211)

to use local co-occurrence stimuli for the induction of previously learnt global knowledge provides evidence of its applicability.

Kintsch (1998) discusses how LSA can be used as a surrogate for the construction of a propositional network of "predicate-argument structures with time and location slots" (p. 411), thereby grounding LSA in this representation theory.

Landauer (1998) proposes that word meaning is acquired from experience and therefore it is valid to represent them through their positions in a semantic vector space. When the space is calculated from a text body large enough, similar performance to humans can be found. He reports results from an experiment where an LSA space was trained from a psychology text book, then used to evaluate a multiple-choice test used in real assessments: LSA passed.

In Landauer (1999), in a reply to Perfetti's critique (1998), defends LSA as a theory of meaning and mind. He particularly defends the criticised grounding in co-occurrences (though not mere 1st order co-occurrences) as an fundamental principle in establishing meaning from passages. He also clearly states that LSA is not a complete theory of discourse processing, as this would include turning perceived discourse into meaning and turning ideas into discourse. In other words, one might say, it would have to provide means to model purpose.

Kintsch (2000, 2001) and Kintsch and Bowles (2002) discuss the interpretation of the mathematical and statistical foundations of LSA with respect to "metaphor interpretation, causal inferences, similarity judgments, and homonym disambiguation" (Kintsch 2001, p. 1).

Landauer (2002) further elaborates on LSA as a theory of learning and cognition, again basing its fundamental principle in the co-occurrence of words within passage contexts and in the analysis of these. They characterise the shortcomings of LSA as a theory of language and verbal semantics (p. 28) to lie in the lack of a model of production and the dynamic processes of comprehension, discourse, and conversation conventions (p. 28). The chapter also discusses the relation of LSA to visual perception and physical experience.

Quesada et al. (2001, 2002a, b, 2005) and Quesada (2003) elaborate on complex problem solving with LSA, implemented—quite similar to case-based reasoning—as contextual similarity.

4.5 Extended Application Example: Automated Essay Scoring

Automated essay scoring is one of the popular application areas of latent semantic analysis, see the overview presented below in Table 4.11. Over time, a wide variety of scoring methods were proposed and evaluated. While many roads lead to Rome, already naïve scoring approaches work astoundingly well—such as comparing a student-written essay with a model solution (a.k.a. gold standard). This subsection

Table 4.11 The state of the art of TEL applications of LSA: Summary

Application	System(s)	Type[a]	Group
Expertise mapping, people recommender	Bellcore advisor	P	Dumais
Essay scoring	Intelligent Essay Assessor, Summary Street, WriteToLearn, Open-Cloze, Meaningful Sentences, Team Communications, Knowledge Post	PEC	Landauer
Learning object search, summaries	SuperManual	P	Landauer
Identifying learning standards	Standard seeker	P	Landauer
Matching learning experience and training programmes	Career map	P	Landauer
Tagging learning objects	Metadata tagger	P	Landauer
Summary writing	Apex, Apex-II, Pensum	PE	Dessus
Essay scoring	EMMA	PE	Haley
Essay scoring	conText	PE	Lenhard
Locating tutors	ASA-ATK	PEC	Van Rosmalen
Dialogue tutoring	AUTOTUTOR, Select-a-Kibitzer, State the Essence	PE	Graesser
Assessing conversations	PolyCAFe	PE	Trausan-Matu
Essay scoring	ESA, R	PEC	Wild
Monitoring conceptual development	CONSPECT	PE	Wild
Essay scoring	MarkIT	P	Dreher
Essay scoring	Gallito	PE	Jorge-Botano, Olmos, Leon

P TEL Prototype, *E* TEL Evaluation study, *C* Configuration

provides an exemplification of such naïve scoring method.[7] The subsequent Chap. 9 on MPIA's application examples will revisit this example to unveil the here unspoken assumptions underlying this approach.

Emphasis thereby will lie on the algorithmic details and therefore the following limitations apply: From a didactical, instructional perspective, delivering a 'naked' single score to learners is more than just a bit questionable. Assessment for learning requires much more than that and even assessment of learning should—if not for

[7] The author made this example available online at: http://crunch.kmi.open.ac.uk/people/~fwild/services/lsa-essay-scoring.Rmw

accuracy reasons, then for acceptance reasons—better rely on the advice and guidance of skilled human evaluators.

Still, to improve quality in assessment, similar scoring systems are in use around the globe (Steedle and Elliot 2012, p. 3), mostly where assessment situations can be standardised to scale, so the investment into training an LSA-based system with high precision pays off. In such scenario, the machine scores typically serve as an extra rater in addition to the human ones (and mediating reviews are called where human and machine differ).

Figure 4.9 gives an overview on the process of analysis. First, a training corpus is created, holding generic, domain specific documents, or both. In case of this example, only domain specific documents were used. They had been gathered from the first hits of a major search engine, thereby splitting larger documents in about paragraph sized units (yielding 71 documents). Additionally, copies of the three model solutions were added to the training corpus. In step 1, a text matrix is constructed from this document collection, to then—step 2—calculate the latent semantic space from it. Step 3 allows to inspect the LSA-revised resulting text matrix.

The student essays (and—for convenience—the three model solutions) are subsequently filed in (step A) and then folded into this space (step B), not least to avoid bad essays from distorting the associative closeness proximity relations in the space. The resulting appendable document vectors are then used to calculate the proximity of each student essay with the three model solutions (step C). The average Spearman Rho rank correlation is thereby used as the score for each essay.

The 74 documents used for training the latent semantic space are converted to their document-term matrix representation with the help of the *lsa* package's `textmatrix` routine.

Listing 11 Filing in the training corpus.

```
corpus_training = textmatrix("corpus/corpus.6.base",
    stemming = FALSE, minWordLength = 3, minDocFreq = 1)
```

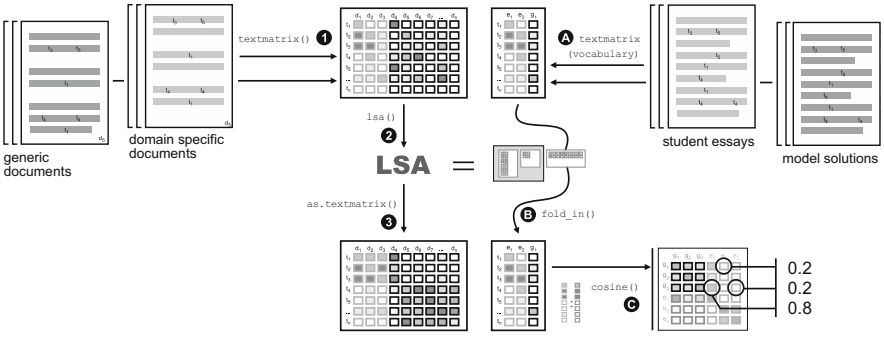

Fig. 4.9 Naïve essay scoring process (revised and extended from Wild and Stahl 2007, p. 388)

Table 4.12 The text matrix of the training corpus

Term	data6_18.txt	data6_19.txt	data6_20.txt	data6_21.txt
zerlegt	0	1	0	1
zweite	0	1	1	0
art	3	1	1	1
auftretens	0	0	0	0
ausgerichtet	0	0	0	0

The resulting text matrix has 1056 terms in the rows and 74 documents in the columns. A subset of these is shown in Table 4.12: four documents and their term frequencies for the subset of five terms.

Following the construction of the text matrix, weighting measures are applied and the space is calculated (see Listing 12). The chosen weighting measure is 1 + entropy (see package documentation in the annex for more detail).

Listing 12 Applying a weighting measure and calculating the latent semantic space.

```
weighted_training = corpus_training *
    gw_entropy(corpus_training)
space = lsa(weighted_training,
    dims = dimcalc_share(share = 0.5))
```

The next step is to map the essays into this existing latent semantic space, in order to enable their analysis in the structure of the space. This prevents 'bad' essays from distorting the structure of the space. Therefore, a text matrix representation of the essays is constructed using the controlled and ordered vocabulary of the training corpus is done with the line of code shown in Listing 13.

Listing 13 Text matrix construction with a controlled, ordered vocabulary.

```
corpus_essays = textmatrix("essays/",
    stemming = FALSE,
    minWordLength = 3,
    vocabulary = rownames(corpus_training)
)
```

Since the training corpus was weighted using a global weighting scheme, this essay text matrix has to be weighted with the same (!) global weights, otherwise the mapping would suffer from an unwanted distortion. This is done with the following line of code.

Listing 14 Weighting of essay corpus with existing global weights.

```
weighted_essays = corpus_essays *
    gw_entropy(corpus_training)
```

Subsequently, the weighted essay matrix can be mapped into the existing latent semantic space with the following command.

Listing 15 Fold in of the essay text matrix into the space.

```
lsaEssays = fold_in(weighted_essays, space)
```

As a naïve scoring method, the average Spearman rank correlation of each student essay to the three model solutions (the three 'gold standards') is used:

Listing 16 Assigning a score to each essay.

```
essay2essay = cor(lsaEssays, method = "spearman")

goldstandard = c("data6_golden_01.txt",
    "data6_golden_02.txt", "data6_golden_03.txt")

machinescores = colSums(essay2essay[goldstandard,])/3
```

To evaluate, how well the machine-assigned scores perform in comparison to the human raters, first the human-assigned scores for each of the essays in the collection are loaded.

Listing 17 Filing in the scores of the human raters.

```
corpus_scores = read.table("corpus/corpus.6.scores",
    row.names = "V1")
```

Then the correlation of machine-assigned to human-assigned scores is calculated. The human scores range from 0 to 4 points in half-point steps and the machine scores with a real number between 0 and 1. Consequently, their correlation is best measured with Spearman's rank correlation coefficient Rho, as shown in Listing 18.

Listing 18 Measuring the correlation between human and machine scores.

```
cor.test(
     humanscores[names(machinescores),],
     machinescores,
     exact = FALSE,
     method = "spearman",
     alternative = "two.sided"
)
```

```
##
## Spearman's rank correlation rho
##
## data: humanscores[names(machinescores),] and
   machinescores
## S = 914.6, p-value = 0.0001049
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
## 0.6873
```

Listing 18 also lists the output of the correlation test: a measured Spearman's Rho of 0.69 for this case of scoring essays in the latent semantic space.

The interesting question is, would a comparable result have been reached without the base change and singular value decomposition: how well is the 'pure' vector space model performing? The following Listing 19 provides an answer to this question. In fact, the pure vector space performs with a measured Spearman's Rho of 0.45 visibly lower.

Listing 19 Correlation of human and machine scores in the 'pure' vector space.

```
essay2essay = cor(corpus_essays, method = "spearman")

machinescores = colSums(essay2essay[goldstandard,])/3

cor.test(
   humanscores[names(machinescores),],
   machinescores,
   exact = FALSE,
   method = "spearman",
   alternative = "two.sided"
)

##
## Spearman's rank correlation rho
##
## data: humanscores[names(machinescores),]
   and machinescores
## S = 1616, p-value = 0.02188
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
## 0.4475
```

4.6 Limitations of Latent Semantic Analysis

In this chapter, an introduction to LSA was delivered, covering the foundations and application techniques. Comprehensive examples illustrate the description. The open source implementation of LSA provided by the author was described in further detail. Two application demos illustrate the use of the *lsa* package for R.

Further information on how to use the 'lsa' package with sparse matrices (using the *tm* package, see Feinerer et al. 2008) and partial SVDs for Linux and Mac OsX (interfacing with the svdlibc of Rhode 2014) is available from the author. The same applies for the binding of the lsa R routines to a REST-ful (Fielding 2000) web service using, for example, the Apache web server (Wild et al. 2008, p. 19ff).

The core restriction of the means for content analysis provided by LSA are its blindness to purpose and social relations and the instruments for interaction analysis that SNA is so popular for.

Moreover, there is no clear rule available, which number of factors to retain and to which to truncate, a shortcoming to which a lot of the criticism of the method can be attributed, often leading to unsuccessful attempts of utilising LSA.

Both these shortcomings will be resolved in the subsequent chapter, which introduces meaningful, purposive interaction analysis as implementation model of the theoretical foundations presented before in Chap. 2.

References

Berry, M., Dumais, S., O'Brien, G.: Using linear algebra for intelligent information retrieval. SIAM Rev. **37**(4), 573–595 (1995)

Calfee, R.: To grade or not to grade. In: Hearst, M. (ed.) The Debate on Automated Essay Grading. IEEE Intelligent Systems, Sep/Oct 2000, pp. 35–37 (2000)

Deerwester, S.: Method and system for revealing information structures in collections of data items, US patent number 5,778,362, dated July 7, 1998, filed June 21, 1996 (1998)

Deerwester, S.: Apparatus and method for generating optimal search queries, Application (deemed withdrawn), Number EP0978058, European Patent Office (2000)

Deerwester, S., Dumais, S., Furnas, G., Harshman, R., Landauer, T., Lochbaum, K., Streeter, L.: Computer information retrieval using latent semantic structure, United States Patent, No. 4,839,853, Appl. No.: 07/244,349, filed: September 15, 1988, Date of Patent: June 13 (1989)

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 391–407 (1990)

Dessus, P., Lemaire, B.: Apex, un système d'aide à la préparation d'examens (Apex, a system that helps to prepare exams). Sciences et Techniques Éducatives **6–2**, 409–415 (1999)

Dessus, P., Lemaire, B.: Using production to assess learning: an ILE that fosters self-regulated learning. In: Cerri, S.A., Gouarderes, G., Paraguacu, F. (eds.) ITS 2002, LNCS 2363, pp. 772–781. Springer, Berlin (2002)

Dessus, P., Lemaire, B., Vernier, A.: Free-text assessment in a Virtual Campus. In: Zreik, K. (ed.) Proceedings of the Third International Conference on Human System Learning (CAPS'3), Europia, Paris, pp. 61–76 (2000)

Dreher, H.: Interactive on-line formative evaluation of student assignments. Issues Inform. Sci. Inform. Technol. (IISIT) **3**(2006), 189–197 (2006)
Dumais, S.: Improving the retrieval of information from external sources. Behav. Res. Methods Instrum. Comput. **23**(2), 229–236 (1991)
Dumais, S.: Enhancing performance in Latent Semantic Indexing (LSI) retrieval. Bellcore Technical memo (sometimes this is dated 1989, not 1992). Online at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.8278 (1992)
Dumais, S.: Latent Semantic Indexing (LSI): TREC-3 Report. In: Harman, M. (ed.) The Third Text REtrieval Conference (TREC3), NIST Special Publication 500–226, pp. 219–230 (1995)
Dumais, S.: Data-driven approaches to information access. Cogn. Sci. **27**(3), 491–524 (2003)
Dumais, S.: Latent semantic analysis. Annu. Rev. Inform. Sci. Technol. **38**(1), 188–230 (2005)
Dumais, S., Furnas, G., Landauer, T., Deerwester, S., Harshman, R.: Using latent semantic analysis to improve access to textual information, In: CHI '88 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 281–285, ACM, New York, NY (1988)
Feinerer, I., Hornik, K., Meyer, D.: Text mining infrastructure in R. J. Stat. Softw. **25**(5), 1–54 (2008)
Fielding, R.T.: Architectural styles and the design of network-based software architectures. Doctoral dissertation, University of California, Irvine (2002)
Foltz, P.: Latent semantic analysis for text-based research. Behav. Res. Methods Instrum. Comput. **28**(2), 197–202 (1996)
Foltz, P.: Quantitative approaches to semantic knowledge representation. Discource Process. **25** (2–3), 127–130 (1998)
Foltz, P.: Using latent semantic indexing for information filtering. In: COCS'90: Proceedings of the ACM SIGOIS and IEEE CS TC-OA conference on Office information systems, pp. 40–47, ACM, New York, NY (1990)
Foltz, P., Kintsch, W., Landauer, T.: The measurement of textual coherence with latent semantic analysis. Discource Process. **25**(2-3), 285–307 (1998)
Foltz, P., Gilliam, S., Kendall, S.: Supporting content-based feedback in on-line writing evaluation with LSA. Interact. Learn. Environ. **8**(2), 111–127 (2000)
Freeman, J., Thompson, B., Cohen, M.: Modeling and diagnosing domain knowledge using latent semantic indexing. Interact. Learn. Environ. **8**(3), 187–209 (2000)
Furnas, G., Landauer, T., Dumais, S., Gomez, L.: Statistical semantics: analysis of the potential performance of key-word information systems. Bell Syst. Tech. J. **62**(6), 1753–1806 (1983)
Graesser, A., Person, N., Magliano, J.: Collaborative dialogue patterns in naturalistic one-to-one tutoring. Appl. Cogn. Psychol. **9**, 295–522 (1995)
Graesser, A., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R., Tutoring Research Group: AutoTutor: a simulation of a human tutor. J. Cogn. Syst. Res. **1**, 35–51 (1999)
Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Tutoring Research Group, Person, N.: Using latent semantic analysis to evaluate the contributions of students in AutoTutor. Interact. Learn. Environ. **8**(2), 129–147 (2000)
Haley, D.: Applying latent semantic analysis to computer assisted assessment in the computer science domain: a framework, a tool, and an evaluation. Dissertation, The Open University, Milton Keynes (2008)
Haley, D., Thomas, P., Nuseibeh, B., Taylor, J., Lefrere, P.: E-Assessment using Latent Semantic Analysis. In: Proceedings of the 3rd International LeGE-WG Workshop: Towards a European Learning Grid Infrastructure, Berlin, Germany (2003)
Haley, D., Thomas, P., De Roeck, A., Petre, M.: A research taxonomy for latent semantic analysis-based educational applications. Technical Report 2005/09, The Open University, Milton Keynes (2005)
Haley, D., Thomas, P., Petre, M., De Roeck, A.: EMMA—a computer assisted assessment system based on latent semantic analysis. In: ELeGI Final Evaluation, Technical Report 2008/14, The Open University, Milton Keynes (2007)

Hearst, M.: The Debate on Automated Essay Grading. IEEE Intelligent Systems, Sep/Oct 2000, pp. 22–37 (2000)
Hirschmann, L., Breck, E., Light, M., Burger, J., Ferro, L.: Automated grading of short-answer tests. In: Hearst, M. (ed.) The Debate on Automated Essay Grading. IEEE Intelligent Systems, Sep/Oct 2000, pp. 31–35 (2000)
Holten, R.; Rosenkranz, C.; Kolbe, H.: Measuring application domain knowledge: results from a preliminary experiment. In: Proceedings of ICIS 2010, Association for Information Systems (2010)
Jorge-Botana, G., Leon, J., Olmos, R., Escudero, I.: Latent semantic analysis parameters for essay evaluation using small-scale corpora. J. Quant. Linguist. **17**(1), 1–29 (2010a)
Jorge-Botana, G., Leon, J., Olmos, R., Hassan-Montero, Y.: Visualizing polysemy using LSA and the predication algorithm. J. Am. Soc. Inf. Sci. Technol. **61**(8), 1706–1724 (2010b)
Kalz, M., Van Bruggen, J., Rusman, E., Giesbers, B., Koper, R.: Positioning of learners in learning networks with content analysis, metadata and ontologies. In: Koper, R., Stefanov, K. (eds.) Proceedings of International Workshop "Learning Networks for Lifelong Competence Development" pp. 77–81, Mar 30–31, 2006, TENCompetence Conference, Sofia, Bulgaria (2006)
Kintsch, W.: The representation of knowledge in minds and machines. Int. J. Psychol. **33**(6), 411–420 (1998)
Kintsch, W.: Metaphor comprehension: a computational theory. Psychon. Bull. Rev. **7**(2), 257–266 (2000)
Kintsch, W.: Predication. Cogn. Sci. **25**(2001), 173–202 (2001)
Kintsch, W., Bowles, A.: Metaphor comprehension: what makes a metaphor difficult to understand? Metaphor. Symb. **17**(2002), 249–262 (2002)
Kintsch, E., Steinhart, D., Stahl, G., LSA Research Group, Matthews, C., Lamb, R.: Developing summarization skills through the use of LSA-based feedback. Interact. Learn. Environ. **8**(2), 87–109 (2000)
Klavans, R., Boyack, K.: Identifying a better measure of relatedness for mapping science. J. Am. Soc. Inf. Sci. **57**(2), 251–263 (2006)
Koblischke, R.: Essay Scoring Application @ DotLRN—Implementierung eines Prototyps, Diploma Thesis, Vienna University of Economics and Business (2007)
Kukich, K.: Beyond automated essay scoring. In: Hearst M (ed.) The Debate On Automated Essay Grading. IEEE Intelligent Systems, Sep/Oct 2000, pp. 22–27 (2000)
Laham, D., Bennett, W., Landauer Jr., T.: An LSA-based software tool for matching jobs, people, and instruction. Interact. Learn. Environ. **8**(3), 171–185 (2000)
Landauer, T.: Learning and representing verbal meaning: the latent semantic analysis theory. Curr. Dir. Psychol. Sci. **7**(5), 161–164 (1998)
Landauer, T.: Latent semantic analysis: a theory of the psychology of language and mind. Discource Process. **27**(3), 303–310 (1999)
Landauer, T.: On the computational basis of learning and cognition: arguments from LSA. Psychol. Learn. Motiv. **41**(2002), 43–84 (2002)
Landauer, T., Dumais, S.: A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychol. Rev. **1**(2), 211–240 (1997)
Landauer, T., Psotka, J.: Simulating text understanding for educational applications with latent semantic analysis: introduction to LSA. Interact. Learn. Environ. **8**(2), 73–86 (2000)
Landauer, T., Laham, D., Rehder, B., Schreiner, M.: How well can passage meaning be derived without using word order? A comparison of latent semantic analysis and humans. In: Proceedings of the 19th annual meeting of the Cognitive Science Society, pp. 412–417, Erlbaum, Mahwah, NJ (1997)
Landauer, T., Foltz, P., Laham, D.: An introduction to latent semantic analysis. In: Discource Process. 25(2–3), 259–284 (1998a)
Landauer, T., Laham, D., Foltz, P.: Learning human-like knowledge by singular value decomposition. In: Jordan, M.I., Kearns, M.J., Solla, S.A. (eds.) Advances in Neural Information

Processing Systems 10. Proceedings of the 1997 Conference, pp. 45–51, The MIT Press (1998b)
Landauer, T., Laham, D., Foltz, P.: The intelligent essay assessor. In: Hearst M. (ed.) The Debate on Automated Essay Grading. IEEE Intelligent Systems, Sep/Oct 2000, pp. 27–31 (2000)
Landauer, T., McNamara, D., Dennis, S., Kintsch, W.: Handbook of latent semantic analysis. Lawrence Erlbaum Associates, Mahwah (2008)
Landauer, T., McNamara, D., Dennis, S., Kintsch, W.: Preface. In: LAndauer, T., McNamara, D., Denns, S., Kintsch, W. (eds.) Handbook of Latent Semantic Analysis. Lawrence Erlbaum Associates, Mahwah (2008)
Larkey, L.: Automated essay grading using text categorization techniques. Proc SIGIR **98**, 90–95 (1998)
Lemaire, B., Dessus, P.: A system to assess the semantic content of student essays. J. Educ. Comput. Res. **24**(3), 305–320 (2001). SAGE Publications
Lenhard, W., Baier, H., Hoffmann, J., Schneider, W.: Automatische Bewertung offener Antworten mittels Latenter Semantischer Analyse. Diagnostica **53**(3), 155–165 (2007a)
Lenhard, W., Baier, H., Hoffmann, J., Schneider, W., Lenhard, A.: Training of Summarisation skills via the use of content-based feedback. In: Wild, F., Kalz, M., van Bruggen, J., Koper, R. (eds.) Mini-Proceedings of the 1st European Workshop on Latent Semantic Analysis in Technology-Enhanced Learning, pp. 26–27, Open University of the Netherlands, Heerlen (2007b)
Lenhard, W., Baier, H., Endlich, D., Lenhard, A., Schneider, W., Hoffmann, J.: Computerunterstützte Leseverständnisförderung: Die Effekte automatisch generierter Rückmeldungen. Zeitschrift für Pädagogische Psychologie **26**(2), 135–148 (2012)
Leon, J., Olmos, R., Escudero, I., Canas, J., Salmeron, L.: Assessing short summaries with human judgments procedure and latent semantic analysis in narrative and expository texts. J. Behav. Res. Methods **38**(4), 616–627 (2006)
Leydesdorff, L.: Similarity measures, author cocitation analysis, and information theory. J. Am. Soc. Inf. Sci. **56**(7), 69–772 (2005)
Lochbaum, K., Psotka, J., Streeter, L.: Harnessing the power of peers. In: Interservice/Industry, Simulation and Education Conference (I/ITSEC), Orlando, FL (2002)
Lochbaum, K., Streeter, L.: Carnegie Hall: an intelligent tutor for command-reasoning practice based on latent semantic analysis. United States Army Research Institute for Behavioral and Social Sciences, ARI Research Note 2002-18 (2002)
Menon, A.K., Elkan, C.: Fast algorithms for approximating the singular value decomposition. ACM Trans. Knowl. Discov. Data **5**(2), 136 (2011)
Miller, T.: Essay assessment with latent semantic analysis. Technical Report (2003)
Olmos, R., Leon, J., Jorge-Botana, G., Escudero, I.: New algorithms assessing short summaries in expository texts using latent semantic analysis. J. Behav. Res. Methods **41**(3), 944–950 (2009)
Olmos, R., Leon, J., Escudero, I., Jorge-Botana, G.: Using latent semantic analysis to grade brief summaries: some proposals, In: Int. J. Cont. Eng. Educ. Life-Long Learn. **21**(2/3), 192–209 (2011)
Page, E.: The imminence of grading essays by computer. Phi Delta Kappan **47**(5), 238–243 (1966)
Palmer, J., Williams, R., Dreher H.: Automated essay grading system applied to a first year university subject—how can we do it better. In: Proceedings of the Informing Science and IT Education (InSITE) Conference, pp. 1221–1229, Cork, Ireland (2002)
Psotka, J., Robinson, K., Streeter, L., Landauer, T., Lochbaum, K.: Augmenting electronic environments for leadership. In: Advanced Technologies for Military Training, RTO meeting proceedings, MP-HFM-101, pp. 307–322 (2004)
Quesada, J.: Introduction to latent semantic analysis and latent problem solving analysis: chapter 2. In: Latent Problem Solving Analysis (LPSA): a computational theory of representation in complex, dynamic problem solving tasks, pp. 22–35, Dissertation, Granada, Spain (2003)

Quesada, J., Kintsch, W., Gomez, E.: A computational theory of complex problem solving using the vector space model (part I): latent semantic analysis, through the path of thousands of ants. Technical Report (2001)
Quesada, J., Kintsch, W., Gomez, E.: A computational theory of complex problem solving using latent semantic analysis. In: Gray, W.D., Schunn, C.D. (eds.) 24th Annual Conference of the Cognitive Science Society, pp. 750–755, Lawrence Erlbaum Associates, Mahwah, NJ (2002a)
Quesada, J., Kintsch, W., Gomez, E.: A computational theory of complex problem solving using the vector space model (part II): latent semantic analysis applied to empirical results from adaptation experiments. Online at: http://lsa.colorado.edu/papers/EMPIRICALfinal.PDF (2002b)
Quesada, J., Kintsch, W., Gomez, E.: Complex problem-solving: a field in search of a definition? Theor. Issues Ergon. Sci. **6**(1), 5–33 (2005)
Rehder, B., Schreiner, M., Wolfe, M., Laham, D., Landauer, T.K., Kintsch, W.: Using latent semantic analysis to assess knowledge: Some technical considerations. Discourse Process. **25** (2–3), 337–354 (1998)
Rhode, D.: SVDLIBC: A C library for computing singular value decompositions, version 1.4. Online at: http://tedlab.mit.edu/~dr/SVDLIBC/ (2014). Last access 31 Jan 2014
Rose, C., Roque, A., Bhembe, D., VanLehn, K.: A hybrid text classification approach for analysis of student essays. In: Proceedings of the HLT-NAACL'03 workshop on Building educational applications using natural language processing, Vol. 2, pp. 68–75, ACM (2003)
Russell, N., ter Hofstede, A., Edmond, D., van der Aalst, W.: Workflow data patterns. QUT Technical report, FIT-TR-2004-01, Queensland University of Technology, Brisbane (2004)
Saeed, J.: Semantics. Wiley-Blackwell, Chichester (2009)
Sahlgren, M.: The distributional hypothesis. Rivista di Linguistica **20**(1), 33–53 (2008)
Sidiropoulos, N., Bro, R.: In memory of Richard Harshman. J. Chemom. **23**(7–8), 315 (2009)
Steedle, J., Elliot, S.: The efficacy of automated essay scoring for evaluating student responses to complex critical thinking performance tasks. Whitepaper, Council for Aid to Education (CAE), New York (2012)
Streeter, L., Psotka, J., Laham, D., MacCuish, D.: The credible grading machine: automated essay scoring in the DOD. In: Interservice/Industry, Simulation and Education Conference (I/ITSEC), Orlando, FL (2002)
Tao, T., Zhai, C.: An exploration of proximity measures in information retrieval. In: SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands (2007)
Trausan-Matu, S., Dessus, P., Lemaire, B., Mandin, S., Villiot-Leclercq, E., Rebedea, T., Chiru, C., Mihaila, D., Gartner, A., Zampa, V.: Writing support and feedback design, Deliverable d5.1 of the LTfLL project, LTfLL consortium (2008)
Trausan-Matu, S., Dessus, P., Rebedea, T., Mandin, S., Villiot-Leclercq, E., Dascalu, M., Gartner, A., Chiru, C., Banica, D., Mihaila, D., Lemaire, B., Zampa, V., Graziani, E.: Learning support and feedback, Deliverable d5.2 of the LTfLL project, LTfLL consortium (2009)
Trausan-Matu, S., Dessus, P., Rebedea, T., Loiseau, M., Dascalu, M., Mihaila, D., Braidman, I., Armitt, G., Smithies, A., Regan, M., Lemaire, B., Stahl, J., Villiot-Leclercq, E., Zampa, V., Chiru, C., Pasov, I., Dulceanu, A.: Support and feedback services (version 1.5), Deliverable d5.3 of the LTfLL project, LTfLL consortium (2010)
Valenti, S., Neri, F., Cucchiarelli, A.: An overview of current research on automated essay grading. J. Inf. Technol. Educ. **2**(2003), 319–330 (2003)
Van Bruggen, J.: Computerondersteund beoordelen van essays. Technical Report, OTEC 2002/1, Open Universiteit Nederland, Heerlen (2002)
Van Bruggen, J., Sloep, P., van Rosmalen, P., Brouns, F., Vogten, H., Koper, R., Tattersall, C.: Latent semantic analysis as a tool for learner positioning in learning networks for lifelong learning. Br. J. Educ. Technol. **35**(6), 729–738 (2004)
van der Vegt, W., Kalz, M., Giesbers, B., Wild, F., van Bruggen, J.: Tools and techniques for placement experiments. In: Koper, R. (ed.) Learning Network Services for Professional Development, pp. 209–223. Springer, London (2009)

van Lehn, K., Jordan, P., Rosé, C., Bhembe, D., Böttner, M., Gaydos, A., Makatchev, M., Pappuswamy, U., Ringenberg, M., Roque, A., Siler, S., Srivastava, R.: The architecture of why2-atlas: a coach for qualitative physics essay writing. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds.) ITS 2002, LNCS 2363, pp. 158–167, Springer, Berlin (2002)

Whittington, D., Hunt, H.: Approaches to the computerized assessment of free text responses. In: Proceedings of the Third Annual Computer Assisted Assessment Conference (CAA'99), pp. 207–219, Loughborough University (1999)

Wiemer-Hastings, P., Graesser, A.: Select-a-Kibitzer: a computer tool that gives meaningful feedback on student compositions. Interact. Learn. Environ. **8**(2), 149–169 (2000)

Wiemer-Hastings, P., Graesser, A., Harter, D., Tutoring Research Group: The foundations and architecture of AutoTutor. In: Proceedings of the 4th International Conference on Intelligent Tutoring Systems, pp. 334–343, San Antonio, TX, Springer, Berlin (1998)

Wild, F.: lsa: Latent Semantic Analysis: R package version 0.73 (2014). http://CRAN.R-project.org/package=lsa

Wild, F., Stahl, C.: Investigating unstructured texts with latent semantic analysis. In: Lenz, H.J., Decker, R. (eds.) Advances in Data Analysis, pp. 383–390. Springer, Berlin (2007)

Wild, F., Stahl, C., Stermsek, G., Neumann, G.: Parameters driving effectiveness of automated essay scoring with LSA. In: Proceedings of the 9th International Computer Assisted Assessment Conference (CAA), pp. 485–494, Loughborough (2005a)

Wild, F., Stahl, C., Stermsek, G., Penya, Y., Neumann, G.: Factors influencing effectiveness in automated essay scoring with LSA. In: Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED), Amsterdam, The Netherlands (2005b)

Wild, F., Kalz, M., van Bruggen, J., Koper, R.: Latent semantic analysis in technology-enhanced learning. In: Mini-Proceedings of the 1st European Workshop, Mar 29–30, 2007, Heerlen, NL (2007a)

Wild, F., Koblischke, R., Neumann, G.: A Research prototype for an automated essay scoring application in .LRN. In: OpenACS and .LRN Spring Conference, Vienna (2007b)

Wild, F., Dietl, R., Hoisl, B., Richter, B., Essl, M., Doppler, G.: Services approach & overview general tools and resources, deliverable d2.1, LTfLL consortium (2008)

Williams, R.: Automated essay grading: an evaluation of four conceptual models. In: Kulski, M., Herrmann, A. (eds.) New Horizons in University Teaching and Learning: Responding to Change. Curtin University of Technology, Perth (2001)

Williams, R., Dreher, H.: Automatically grading essays with Markit©. Issues Inform. Sci. Inform. Technol. (IISIT) **1**(2004), 693–700 (2004)

Williams, R., Dreher, H.: Formative assessment visual feedback in computer graded essays. Issues Inform. Sci. Inform. Technol. (IISIT) **2**(2005), 23–32 (2005)

Wolfe, M., Schreiner, M., Rehder, B., Laham, D., Foltz, P., Kintsch, W., Landauer, T.: Learning from text: matching readers and texts by latent semantic analysis. Discourse Process. **25**(2–3), 309–336 (1998)

Yang, Y., Buckendahl, C., Juszkiewicz, P.: A review of strategies for validating computer automated scoring. In: Proceedings of the Annual Meeting of Midwestern Educational Research Association, Chicago, IL (2001)