Edda Leopold

# On Semantic Spaces

## 1 Introduction

This contribution gives an overview about different approaches to semantic spaces. It is not a exhaustive survey, but rather a personal view on different approaches which use metric spaces for the representation of meanings of linguistic units. The aim is to demonstrate the similarities of apparently different approaches and to inspire the generalisation of semantic spaces tailored to the representation of texts to arbitrary semiotic artefacts.

I assume that the primary purpose of a semiotic system is communication. A semiotic system $\tilde{S}$ consists of signs $s$. Signs fulfil a communicative function $f(s)$ within the semiotic system in order to meet the communicative requirements of system's user. There are different similarity relations between functions of signs. In its most general form a semantic space can be defined as follows:

**Definition 1.1** *Let $\tilde{S}$ be a semiotic system, $(S,d)$ a metric space and $r : \tilde{S} \to S$ a mapping from $\tilde{S}$ to $S$. A semantic space $(S,d)$ is a metric space whose elements are representations of signs of a semiotic system, i.e. for each $x \in S$ there is a $s \in \tilde{S}$ such that $r(s) = x$. The inverse metric $(d(x,y))^{-1}$ quantifies some functional similarity of the signs $r^{-1}(x)$ and $r^{-1}(y)$ in $\tilde{S}$.*

Semantic spaces can quantify functional similarities in different respects. If the semiotic system is a natural language, the represented units are usually words or texts — but semantic spaces can also be constructed from other linguistic units like syllables or sentences. The constructions of Semantic spaces leads to a notion of semantic distance, which often cannot easily be made explicit. Some constructions (like the one described in section 6) yield semantically transparent dimensions.

The definition of a semantic space is not confined to linguistic units. Anything that fulfils a function in a semiotic system can be represented in a semantic space. The calculation of a semantic space often involves a reduction of dimensionality and the spaces described in this paper will be ordered with decreasing dimensionality and increasing semantic transparency. In the following section the basic notations will be introduced, that are used in the subsequent sections.

Section 3 roughly outlines the fuzzy linguistic paradigm. Sections 4 and 5 describe shortly the methods of latent semantic indexing and probabilistic latent semantic indexing. In section 6 I show how previously trained classifiers can be used in order to construct semantic spaces.

## 2 Notations

In order to harmonise the presentation of the different approaches I will use the following notations: A text corpus $\mathcal{C}$ consists of a number of $D$ different textual units referred to as *documents* $d_j, j = 1, \ldots, D$. Documents can be complete texts, such as articles in a newspaper, short news as e.g. in the Reuters newswire corpus, or even short text fragments like paragraphs or text blocks of a constant length.

Each document consists of a (possibly huge) number of *terms*. The entire number of different term-types in $\mathcal{C}$ (i.e. the size of the vocabulary of $\mathcal{C}$) is denoted by $W$ and the number of occurrences of a given term $w_i$ in a given document $d_j$ is denoted by $f(w_i, d_j)$. The definition of what is considered as a term may vary, terms can be lemmas, words as they occur in the running text (i.e. strings separated by blanks), tagged words as for instance in Leopold & Kindermann (2002), strings of syllables as in Paaß et al. (2002), or even a mixture of lemmas and phrases as in Neumann & Schmeier (2002). The methods described below are independent from what is considered as a term in a particular application. It is merely assumed that a corpus consists of a set of documents and each of these documents consist of a set of terms[1]. The *term-document matrix* $A$ of $\mathcal{C}$ is a $W \times D$ matrix with $W$ rows and $D$ columns, which is defined as

$$A \quad = \quad (f(w_i, d_j))_{i=1,\ldots,W, j=,\ldots,D}$$

or more explicitly

$$A \quad = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1D} \\ a_{21} & a_{22} & \ldots & a_{2D} \\ \vdots & & \ddots & \vdots \\ a_{W1} & a_{W2} & \ldots & a_{WD} \end{pmatrix}, \qquad \text{where } a_{ij} := f(w_i, d_j) \tag{1}$$

1 Actually the assumption is even weaker: the methods simply focus on the co-occurrences of documents and terms, no matter if one is contained in the other.

The entry in the $i$th row and the $j$th column of the term-document matrix indicates how often term $w_i$ appears in document[2] $d_j$. The rows of $A$ represent terms and its columns represent documents. In the so-called bag-of-words representation, document $d_j$ is represented by the $j$th column of $A$, which is also called the word-frequency vector of document $d_j$ and denoted by $\vec{x}_j$. The sum of the frequencies in the $j$-th row of $A$ is denoted by $f(d_j)$, which is also called the *length* of document $d_j$. The length of corpus $\mathcal{C}$ is denoted by $L$. Clearly

$$f(d_j) = \sum_{i=1}^{W} f(w_i, d_j) \quad \text{and} \quad L = \sum_{j=1}^{D} f(d_j) \tag{2}$$

The $i$th row of $A$ indicates how the term $w_i$ is spread over the documents in the corpus. The rows of $A$ are linked to the notion of *polytexty*, which was defined by Köhler (1986) as the number of contexts in which a given term $w_i$ occurs. Köhler noted that polytexty can be operationalised by the number of *texts* the term occurs in i.e. the number of non-zero entries of the $i$-th row. The $i$th column of $A$ is therefore called *vector of polytexty* of term $w_i$ and the vector of the respective relative frequencies is named *distribution of polytexty*. The sum over the frequencies in the $i$th column, i.e. the total number of occurrences of term $w_i$ in the corpus $\mathcal{C}$, is denoted by

$$f(w_i) = \sum_{j=1}^{D} f(w_i, d_j).$$

The polytexty measured in terms of non-zero entries in a row of the term-document matrix is also called *document-frequency* denoted as $df$. The so-called inverse document frequency, which was defined by Salton & McGill (1983) as $idf = (\log df)^{-1}$, is widely used in the literature on automatic text processing in order to tune term-frequencies according to the thematic relevance of a term. Other term weighting schemes like e.g. the redundancy used by Leopold & Kindermann (2002) consider the entire vector of polytexty rather than solely the number of non-zero elements. An overview about different weighting schemes is given in Manning & Schütze (1999).

Matrix transposition, subsequently indicated by a superscript $\cdot^T$, exchanges columns and rows of a matrix. So the transposed term-document matrix is

2 It should be noticed here that in many cases the term-document matrix does not contain the term-frequencies $f(w, d)$ themselves but a transformation of them like e.g. $\log f(w, d)$ or tfdif.

defined as

$$A^T = (f(w_j, d_i))_{i=1,\ldots,D, j=1,\ldots,W} = \begin{pmatrix} a^t_{11} & a^t_{12} & \cdots & a^t_{1W} \\ a^t_{21} & a^t_{22} & \cdots & a^t_{2W} \\ \vdots & & \ddots & \vdots \\ a^t_{D1} & a^t_{W2} & \cdots & a^t_{DW} \end{pmatrix},$$

$$\text{where } a^t_{ij} := f(w_j, d_i)$$

It is easy to see that the matrix transposition is inverse to itself, i.e. $(A^T)^T = A$. All algorithms presented below are symmetric in documents and terms, i.e. they can be used to estimate semantic similarity of terms as well as of documents depending on whether $A$ or $A^T$ is considered.

There are various measures for judging the similarity of documents. Some measures — the so-called association measures — disregard the term frequencies and just perform set-theoretical operations on the document's term sets. An example for an association measure is the *matching coefficient*, which simply counts the number of terms that two documents have in common (van Rijsbergen 1975).

Other measures take advantage from the vector space model and consider the entire term-frequency vectors of the respective documents. One of the most often used similarity measure, which is also mathematically convenient, is the cosine measure (Manning & Schütze 1999; Salton & McGill 1983) defined as

$$\cos(\vec{x}_i, \vec{x}_j) = \frac{\sum_k^W f(w_k, d_i) f(w_k, d_j)}{\sqrt{\sum_k^W f(w_k, d_i)^2 \sum_k^W f(w_k, d_j)^2}} = \frac{\vec{x}_i \cdot \vec{x}_j}{\|\vec{x}_i\| \|\vec{x}_j\|}, \quad (3)$$

which can also be interpreted as the angle between the vectors $\vec{x}_i$ and $\vec{x}_j$ or, up to centering, as the correlation between the respective discrete probability distributions.

## 3 Fuzzy Linguistics

> [...] the investigation of linguistic problems in general, and that of word-semantics in particular, should start with more or less pre-theoretical working hypotheses, formulated and re-formulated for continuous estimation and/or testing against observable data, then proceed to incorporate its findings tentatively in some preliminary

> theoretical set up which finally may perhaps get formalised to become part of an encompassing abstract theory. Our objective being natural language meaning, this operational approach would have to be what I would like to call *semiotic*. (Rieger 1981)

Fuzzy Linguistics (Rieger & Thiopoulos 1989; Rieger 1981, 1999) aims at a spatial representation of word meanings. I.e. the units represented in the semantic space are *words* as opposed to documents in the other approaches. However from a mathematical point of view there is no formal difference between semantic spaces that are constructed to represent documents and those which are intended to represent terms. One can transform one problem into the other by simply transposing the term-document matrix i.e. by considering $A^T$ instead of $A$.

Rieger has calculated a semantic space of word meanings in two steps of abstraction, which are also implicitly incorporated in the other constructions of semantic spaces described in the sections (4) to (6). The first step of abstraction is the *α-abstraction* or more explicitly *syntagmatic abstraction* which reflects a term's usage regularities in terms of its vector of polytexty. The second abstraction step is the $\delta$-abstraction or *paradigmatic abstraction*, which represents a word's relation to all other words in the corpus.

### 3.1 The Syntagmatic Abstraction

For each term $w_i$ a vector of length $W$ is calculated, which contains the correlations of a term's vector of polytexty with all other terms in the corpus.

$$\alpha_{i,j} = \frac{\sum_{k=1}^{D}(f(w_i,d_k) - E(f(w_i) \mid d_k))(f(w_j,d_k) - E(f(w_j) \mid d_k)}{\sqrt{\sum_{k=1}^{D}(f(w_i,d_k) - E(f(w_i) \mid d_k))^2 \sum_{k=1}^{D}(f(w_j,d_k) - E(f(w_j) \mid d_k))^2}} \tag{4}$$

where $E(f(w_i) \mid d_k) = f(w_i)\frac{f(d_k)}{L}$ is an estimator of the conditioned expectation of the frequency of term $w_i$ in document $d_j$, based on all documents in the corpus. The coefficient $\alpha_{i,j}$ measures the mutual affinity ($\alpha_{i,j} > 0$) or repugnancy ($\alpha_{i,j} < 0$) of pairs of terms in the corpus (Rieger & Thiopoulos 1989).

Substituting $y_{i,j} = f(w_i,d_k) - E(f(w_i) \mid d_k)$ the centralised vector of polytexty of term $w_i$ is defined as $\vec{y}_i = (y_{i,1},\ldots,y_{i,D})^T$. Using this definition equation (4) can be rewritten as

$$\alpha_{i,j} = \frac{\sum_k^D y_{i,k} y_{j,k}}{\sqrt{\sum_k^D y_{i,k}^2 \sum_k^D y_{j,k}^2}} = \frac{\vec{y}_i \cdot \vec{y}_j}{\|\vec{y}_i\| \|\vec{y}_j\|}, \quad (5)$$

which is the definition of the cosine distance as defined in equation (3). The difference between the $\alpha$-abstraction and the cosine distance is merely that in equation (4) the centralised vector of polytexty is considered instead of the word-frequency vector in (3). Using the notion of polytexty one might say more abstractly that $\alpha_{i,j}$ is the correlation coefficient of the polytexty distributions of the types $w_i$ and $w_j$ on the texts in the corpus.

Syntagmatic abstraction realised by equation (4) refers to usage regularities in terms of co-occurrences in the same document. Documents in Rieger's works were in general short texts, like e.g. newspaper texts (Rieger 1981; Rieger & Thiopoulos 1989) or small textual fragments (Rieger 2002). This means that the syntagmatic abstraction solely relies on the distribution of polytexty of the respective terms.

In principle however the approach can be generalised regarding various types of generalised syntagmatic relations. Note that documents were defined as arbitrary disjoint subsets of a corpus. The underlying formal assumption was simply that there is a co-occurrence structure of documents and terms, which is represented in the term-document matrix. Consider for instance a syntactically tagged corpus. In such a corpus documents might be defined e.g. as a set of terms that all carry the same tag. The corresponding "distributions of polytexty" would describe how a term is used in different parts-of-speech and the syntagmatic abstraction $\alpha_{i,j}$ would measure the similarity of $w_i$ and $w_j$ in terms of part-of-speech membership.

### 3.2 The Paradigmatic Abstraction

The $\alpha$-abstraction measures the similarities of the distribution of polytexty over all terms in the corpus. The absolute value of the similarities, however, is not solely a property of the terms themselves, but also of the corpus as a whole. That is if the corpus is confined to a small thematic domain, the documents will be more similar than in the case of a corpus that covers a wide range of themes. In order to attain a paradigmatic abstraction, which abstracts away from the thematic coverage of the corpus, the Euclidean distances to all words in the corpus are summed. This is the $\delta$-abstraction (Rieger 1981; Rieger & Thiopoulos 1989) given by:

$$\delta(y_i, y_j) = \sqrt{\sum_{n=1}^{W} (\alpha_{i,n} - \alpha_{j,n})^2}; \qquad \delta \in [0; 2\sqrt{W}] \tag{6}$$

The $\delta$-abstraction compensates the effect of the corpus' coverage on $\alpha$. The similarity vector of each term is related to the similarity vectors of all other terms in the corpus. In this way the paradigmatic structure in the corpus is evaluated in the sense that every term is paradigmatically related to each other since every term can equally be engaged in a *occurs-in-document* relation.

So the vector $y_i$, is mapped to a vector $(\delta(i,1) \ldots \delta(i,W))$, which contains the Euclidean distance of $x_i$'s $\alpha$ to all other $\alpha$s generated by the corpus and is interpreted as meaning point in a semantic space (Rieger 1988). Rieger concludes that in this way a semantic representation is attained that represents the numerically specified generalised paradigmatic structure that has been derived for each abstract syntagmatic usage regularity against all other in the corpus (Rieger 1999).

Goebl (1991) uses another measurement to anchor similarity measurements of linguistic units (in his case dialectometric data sets) for the completely different purpose of estimating the centrality of dialects in a dialectal network. Let $\alpha_{i,j}$ denote the similarity of dialect $x_i$ and $x_j$, and let $W$ denote the number of dialects in the network. The centrality of $x_i$ is given by:

$$\gamma(x_i) = \sum_{n=1}^{W} \left( \alpha_{i,n} - \frac{1}{W} \sum_{k=1}^{W} \alpha_{i,k} \right)^3 \tag{7}$$

He argues

> The skewness of a similarity distribution has a particular *linguistic* meaning. The more symmetric a similarity distribution is, the greater the centrality of the particular local dialect in the whole network.(Goebl 1991)

Goebl uses (7) in order to calculate the centrality of a local dialect from the matrix $(\alpha_{i,j})_{i,j}$ of similarity measures between pairs of dialects in the network. These centrality measures are employed to draw a choropleth map of the dialectal network. Substituting the delta abstraction in (6) by the skewness in (7) would result in a measure for the centrality of a term in a term-document network: the more typical a term's usage in the corpus the larger the value of $\gamma$. Such a measure could be used as a term-weighting scheme.

Rieger's construction of a semantic space does *not* lead to a reduction of dimensionality. This was not his aim. The meaning of a term is represented by a high-dimensional vector and thus demonstrates the complexity of meaning structures in natural language. Rieger's idea to compute semantic relations from a term-document matrix and represent semantic similarities as distances in a metric space has aspects in common with pragmatically oriented approaches like e.g. latent semantic analysis. The measures of the $\alpha_{i,j}$ can be written in a more condensed way as

$$B^* = A^*(A^*)^T = (\alpha_{i,j})_{i,j=1,...W} \tag{8}$$

$B^*$ is a $W \times W$-matrix which represents the similarity of the words $w_i$ and $w_j$ in terms of their distribution of polytexty. The semantic similarity between words is calculated here in a way similar to the semantic similarity between words in latent semantic indexing, which is described in the next section. The *similarity matrix* $B^* = A^*(A^*)^T$ however is calculated in a slightly different way. The entries of $A^*$ are $y_{i,j} = f(w_i, d_k) - E(f(w_i) \mid d_k)$ rather than the term frequencies $f(w_i, d_j)$ themselves, as can be seen from equation (4).

More advanced techniques within the fuzzy linguistic paradigm (Mehler 2002) extend the concept of the semantic space to the representation of texts. The respective computations, however, are complicated and exceed the scope of this paper.

Fuzzy linguistics aims at a numerical representation of the meaning of terms. Thus the paradigmatic abstraction in equation (6) does not involve a reduction of dimensionality, in contrast to the principal component analysis that is performed in the paradigmatic abstraction step in latent semantic analysis. There is however a close formal relationship.

## 4 Latent Semantic Analysis

> In essence, and in detail, it [latent semantic analysis] assumes that the psychological similarity between any two words is reflected in the way they co-occur in small subsamples of language. (Landauer & Dumais (1997); Words in square brackets added by the author.)

In contrast to fuzzy linguistics latent semantic analysis (LSA) is interested in the semantic nearness of *documents* rather than of words. The method however is symmetric and can be applied to the similarity of words as well.

LSA projects Document frequency vectors into a low dimensional space calculated using the frequencies of word occurrence in each document. The relative distances between these points are interpreted as distances between the topics of the documents and can be used to find related documents, or documents matching some specified query (Berry et al. 1995). The underlying technique of LSA was chosen to fulfil the following criteria:

1. To represent the underlying semantic structure a model with sufficient power is needed. Since the right kind of alternative is unknown the power of the model should be variable.

2. Terms and documents should both be explicitly represented in the model.

3. The method should be computationally tractable for large data sets. Deerwester et al. concluded that the only model which satisfied all these three criteria was the singular value decomposition (SVD), which is a well known technique in linear algebra (Deerwester et al. 1990).

### 4.1 Singular Value Decomposition

Let $A$ be a term-document matrix as defined in section (2) with rank[3] $r$. The singular value decomposition of $A$ is given by

$$A = U\Sigma V, \tag{9}$$

where $\Sigma = diag(\sigma_1, \ldots, \sigma_r)$ is a diagonal matrix with ordered diagonal elements $\sigma_1 > \ldots > \sigma_r$,

$$U = \begin{pmatrix} u_{11} & u_{12} & \ldots & u_{1r} \\ u_{21} & u_{22} & \ldots & u_{2r} \\ \vdots & & \ddots & \vdots \\ u_{W1} & u_{W2} & \ldots & u_{Wr} \end{pmatrix}$$

is a $W \times r$-matrix with orthonormal columns and

$$V = \begin{pmatrix} v_{11} & v_{12} & \ldots & v_{1r} \\ v_{21} & v_{22} & \ldots & v_{2r} \\ \vdots & & \ddots & \vdots \\ v_{r1} & v_{r2} & \ldots & v_{rr} \end{pmatrix}$$

3 In practice one can assume $r = D$, since it is very unlikely that there are two documents in the corpus with linear dependent term-frequency vectors

is a $r \times r$-matrix with orthonormal rows. The diagonal elements $\sigma_1, \ldots, \sigma_r$ of the matrix $\Sigma$ are singular values of $A$. The singular value decomposition can equivalently be written as an eigen-value decomposition of the similarity matrix

$$B = AA^T \tag{10}$$

Note that $U$ and $V$ are orthonormal matrices therefore $UU^T = I$ and $VV^T = I$, where $I$ is the neutral element of matrix-multiplication. According to (9) the singular value decomposition of the transposed term-document matrix $A^T$ is obtained as $A^T = V^T \Sigma U^T$. Hence $AA^T = U\Sigma VV^T \Sigma U^T = U\Sigma^2 U^T$ which is the eigen-value decomposition of $AA^T$ with eigen-values $\sigma_1^2, \ldots, \sigma_r^2$. Term frequency vectors are mapped to the latent space of artificial concepts by multiplication with $U\Sigma$, i.e. $\vec{x} \rightarrow \vec{x}^T U\Sigma$. Each of the $r$ dimensions of the latent space may be thought of as an artificial concept, which represents common meaning components of different words and documents.

### 4.2 Deleting the Smallest Singular Values

A reduction of dimensionality is achieved by deleting the smallest singular values corresponding to the less important concepts in the corpus. In so doing latent semantic analysis reduces the matrix $A$ to a smaller $K$-dimensional ($K < r$) matrix

$$A_K = U_K \Sigma_K V_K, \tag{11}$$

where $U_K$ and $V_K$ are obtained from $U$ and $V$ in equation (9) by deleting respectively columns and/or rows $K+1$ to $r$ and the diagonal matrix is reduced to $\Sigma_K = diag(\sigma_1, \ldots, \sigma_K)$. The mapping of a term-frequency vector to the reduced latent space is now performed by $\vec{x} \rightarrow \vec{x}^T U_K \Sigma_K$. It has been found that $K \approx 100$ is a good value to chose for $K$ (Landauer & Dumais 1997).

LSA leads to vectors with few zero entries and to a reduction of dimensionality ($k$ instead of $W$) which results in a better geometric interpretability. This implies that it is possible to compute meaningful association values between pairs of documents, even if the documents do not have any terms in common.

### 4.3 SVD Minimises Euclidean Distance

Truncating the singular value decomposition as described in equation (11) projects the data onto the best-fitting affine subspace of a specified dimension $K$. It is a well-known theoretical result in linear algebra, that there is no matrix $X$

with $rank(X) < K$ that has a smaller Frobenius distance to the original matrix $A$ i.e. $A_K$ minimises

$$\|A - A_K\|_F = \sum_{i,j}^{K} (a_{i,j} - a_{i,j}^K)^2. \tag{12}$$

Interestingly Rieger's $\delta$-abstraction in equation (6) yields a nice interpretation of this optimality statement. The reduction of dimensionality performed by latent semantic analysis is achieved in such a way that it optimally preserves the inherent meaning (i.e. the sum of the $\delta(x_i, x_j)$). That is the meaning points in the Rieger's $\delta$-space are changed to the minimal possible extent. Another parallel between fuzzy linguistics and LSA is that equation (4) and the corresponding matrix notation of $\alpha_{i,j}$ in equation (8) coincide withe the similarity matrix in equation (10). The only difference is that the entries of $A$ and $A^*$ are defined in a different way. Using Rieger's terminology one may call equation (10) a syntagmatic abstraction, because it reflects the usage regularities in the corpus. The singular value decomposition is then the paradigmatic abstraction, since it abstracts away from the paradigmatic structure of the language's vocabulary which consists of synonymy and polysemy relationships.

One objection to latent semantic indexing is that, along with all other least-square methods, the property of minimising the Frobenius distance makes it suited for normally distributed data. The normal distribution however is unsuitable to model term frequency counts. Other distributions like Poisson or negative binomial are more appropriate for this purpose (Manning & Schütze 1999).

Alternative methods have therefore been developed (Gous 1998), which assume that the term frequency vectors are multinomially distributed and therefore agree with well corroborated models on word frequency distribution developed by Chitashvili and Baayen (Chitashvili & Baayen 1993). Probabilistic Latent Semantic Analysis has advanced further in this direction.

## 5 Probabilistic Latent Semantic Analysis

Whereas latent semantic analysis is based on counts of co-occurrences and uses the singular value decomposition to calculate the mapping of term-frequency vectors to a low-dimensional space, probabilistic latent semantic analysis (see Hofmann & Puzicha (1998); Hofmann (2001)) is based on a probabilistic framework and uses the maximum likelyhood principle. This results in a better lin-

guistic interpretability and makes probabilistic latent semantic analysis (PLSA) compatible with the well-corroborated multinomial model of word frequency distributions.

### 5.1 The Multinomial Model

The assumption that the occurrences of different terms in the corpus are stochastically independent allows to calculate the probability of a given term frequency vector $\vec{x}_j = (f(w_1, d_j), \ldots, f(w_W, d_j))$ according to the multinomial distribution (see Chitashvili & Baayen (1993); Baayen (2001)):

$$p(\vec{x}_j) = \frac{f(d_j)}{\prod_{i=1}^{W} f(w_i, d_j)!} \prod_{i=1}^{W} p(w_i, d_j)^{f(w_i, d_j)}$$

If it is further assumed that the term-frequency vectors of the documents in the corpus are stochastically independent, the probability to observe a given term-document matrix is

$$p(A) = \prod_{j=1}^{D} \frac{f(d_j)}{\prod_{i=1}^{W} f(w_i, d_j)!} \prod_{i=1}^{W} p(w_i, d_j)^{f(w_i, d_j)} \tag{13}$$

### 5.2 The Aspect Model

In order to map high-dimensional term-frequency vectors to a limited number of dimensions PLSA uses a probabilistic framework, called aspect model. The aspect model is a latent variable model which associates an unobserved class variable $z_k, k = 1, \ldots, K$, with each observation an observation being the occurrence of a word in a particular document. The latent variables $z_k$ can be thought of as artificial concepts like the latent dimensions in LSA. Like in LSA the number of artificial concepts $K$ has to be chosen by the experimenter. The following probabilities are introduced: $p(d_j)$ denotes the probability that a word occurrence will be observed in a particular document $d_i$, $p(w_i \mid z_k)$ denotes the conditional probability of a specific term conditioned on the latent variable $z_k$ (i.e. the probability of term $w_i$ given the thematic domain $z_k$), and finally $p(z_k \mid d_j)$ denotes a document-specific distribution over the latent variable space i.e. the distribution of artificial concepts in document $d_j$. A generative model for word/document co-occurrences is defined as follows:

(1) select a document $d_j$ with probability $p(d_j)$,

(2) pick a latent class $z_k$ with probability $p(z_k|d_j)$, and

(3) generate word $w_j$ with probability $p(w_i|z_k)$ (Hofmann 2001).

Since the aspects are latent variables which cannot be observed directly, the conditioned probability $p(w_i \mid d_j)$ has to be calculated as the sum of the possible aspects:

$$p(w_i|d_j) \quad = \quad \sum_{k=1}^{K} p(w_i|z_k)p(z_k|d_j) \tag{14}$$

This implies the assumption, that the conditioned probability of occurrence of aspect $z_k$ in document $d_j$ is independent from the conditioned probability that term $w_i$ is used given that aspect $z_k$ is present (Hofmann 2001).

In order to find the optimal probabilities $p(w_i|z_k)$ and $p(z_k|d_j)$, maximizing the probability of observing a given term-document matrix, the maximum likelihood principle is applied. The multinomial coefficient in equation (13) remains constant when the probabilities $p(w_i, d_j)$ are varied. It can therefore be omitted for the calculation of the likelihood function, which is then given as

$$\mathcal{L} = \sum_{j=1}^{D} \sum_{i=1}^{W} f(w_i, d_j) \log p(w_i, d_j)$$

Using the definition of the conditioned probabilities $p(w_i, d_j) = p(d_j)p(w_i \mid d_j)$ and inserting equation (14) yields

$$\mathcal{L} = \sum_{j=1}^{D} \sum_{i=1}^{W} \left( f(w_i, d_j) \log\Big( p(d_j) \cdot \sum_{k=1}^{K} p(w_i \mid z_k)p(z_k \mid d_j) \Big) \right)$$

Using the additivity of the logarithm and factoring in $f(w_i, d_j)$ gives

$$\mathcal{L} = \sum_{j=1}^{D} \left( \sum_{i=1}^{W} f(w_i, d_j) \log p(d_j) + \sum_{i=1}^{W} f(w_i, d_j) \log \sum_{k=1}^{K} p(w_i \mid z_k)p(z_k \mid d_j) \right)$$

Since $\sum_i f(w_i, d_j) = f(d_j)$ factoring out $f(d_j)$ finally leads to the likelihood function

$$\mathcal{L} = \sum_{j=1}^{D} f(d_j) \left( \log p(d_j) + \sum_{i=1}^{W} \frac{f(w_i, d_j)}{f(d_j)} \log \sum_{k=1}^{K} p(w_i \mid z_k) p(z_k \mid d_j) \right) \tag{15}$$

which has to be maximised with respect to the conditional probabilities involving the latent aspects $z_k$. Maximisation of (15) can be achieved using the EM-algorithm, which is a standard procedure for maximum likelihood estimation in latent variable models (Dempster et al. 1977). The EM-algorithm works in two steps that are iteratively repeated (see e.g. Mitchell (1997) for details).

**Step 1** In the first step (the expectation step) the expected value $E(z_k)$ of the latent variables is calculated, assuming that the current hypothesis $h_1$ holds.

**Step 2** In a second step (the maximisation step) a new maximum likelihood hypothesis $h_2$ is calculated assuming that the latent variables $z_k$ equal their expected values $E(z_k)$ that have been calculated in the expectation step. Then $h_1$ is substituted by $h_2$ and the algorithm is iterated.

In the case of PLSA the the EM-algorithm is employed as follows (see Hofmann (2001) for details): To initialise the algorithm generate $W \cdot K$ random values for the probabilities $p(w_i \mid z_k)$ and $D \cdot K$ random values for the probabilities $p(z_k \mid d_j)$ such that all probabilities are larger than zero and fulfil the conditions $\sum_{i,k} p(w_i \mid z_k) = 1$ and $\sum_{j,k} p(z_k \mid d_j) = 1$ respectively. The expectation step can be obtained from equation (15) by applying Bayes' formula:

$$p(z_k \mid w_i, d_j) = \frac{p(w_i \mid z_k) p(z_k \mid d_j)}{\sum_{k=1}^{K} p(w_i \mid z_k) p(z_k \mid d_j)} \tag{16}$$

In the maximization step the probability $p(z_k \mid w_i, d_j)$ is used to calculate the new conditioned probabilities

$$p(w_i \mid z_k) = \frac{\sum_{j=1}^{N} f(w_i, d_j) p(z_k \mid w_i, d_j)}{\sum_{k=1}^{K} \sum_{j=1}^{D} f(w_i, d_j)(z_k \mid w_i, d_j)} \tag{17}$$

and

$$p(z_k \mid d_j) = \frac{\sum_{i=1}^{W} f(w_i, d_j) p(z_k \mid w_i, d_j)}{f(d_j)}, \tag{18}$$

Then the conditioned probabilities $p(z_k|d_j)$ and $p(w_i|z_k)$ calculated from equation (17) and (18) are inserted into equation (16) to perform the next iteration. The iteration is stopped when a stationary point of the likelihood function is achieved. The probabilities $p(z_k \mid d_j), k = 1, \ldots, K$, uniquely define for each document a $K-1$-dimensional point in continuous latent space.

It is reported that PLSA outperforms LSA in terms of perplexity reduction. Notably PLSA allows to train latent spaces with a continuous increase in performance, in contrast to LSA where the model perplexity increases when a certain number of latent dimensions is exceeded. In PLSA the number of latent dimensions may even exceed the rank of the term-document matrix (Hofmann 2001).

The main difference between LSA and PLSA is the optimisation criterion for the mapping to the latent space, which is defined by $U\Sigma$ and $p(z_k \mid d_j)$ respectively. LSA minimises the least square criterion in equation (12) and thus implicitly assumes an additive Gaussian noise on the term-frequency data. PLSA in contrast assumes multinomially distributed term-frequency vectors and maximises the likelihood of the aspect model. It is therefore in accordance with linguistic word frequency models. One disadvantage of PLSA is, that the EM-algorithm like most iterative algorithms converges only locally. Therefore the solution need not be a global optimum, in contrast to LSA which uses an algebraic solution and ensures global optimality.

## 6 Classifier Induced Semantic Spaces

> [...] problems, in which the task is to classify examples into one of a discrete set of possible categories, are often referred to as *classification problems*.(Mitchell 1997)

The main problem in PLSA approach was to find the latent aspect variables $z_k$ and calculate the corresponding conditioned probabilities $p(w_i|z_k)$ and $p(z_k|d_j)$. It was assumed that the latent variables correspond to some artificial concepts. It was impossible however to specify these concepts explicitly. In the approach described below, the aspect variables can be interpreted semantically. Prerequisite for such a construction of a semantic space is a semantically annotated *training corpus*. Such annotations are usually done manually according to explicitly

defined annotation rules. An example of such a corpus is e.g. the news data of the German Press Agency (dpa) which is annotated according to the categories of the International Press Telecommunications Council (IPTC). These annotations inductively define the concepts $z_k$, or the dimensions, of the semantic space. A *classifier induced semantic space* (CISS) is generated in two steps: In the *training step* classification rules $\vec{x}_j \rightarrow z_k$ are inferred from the training data. In the *classification step* these decision rules are applied to possibly unannotated documents.

This construction of a semantic space is especially useful for practical applications because (1) the space is low-dimensional (up to dozens of dimensions) and thus can easily be visualised, (2) the space's dimension possesses a well defined semantic interpretation, and (3) the space can be tailored to the special requirements of a specific application. The disadvantage of classifier induced semantic spaces (CISS) is that they rely on *supervised* classifiers. Therefore manually annotated training data is required.

Classification algorithms often use an internal representation of degree of membership. They internally calculate how much a given input vector $\vec{x}$ belongs to a given class $z_k$. This internal representation of degree of membership can be exploited to generate a semantic space.

A Support Vector Machine (SVM) is a supervised classification algorithm that recently has been applied successfully to text classification tasks. SVMs have proven to be an efficient and accurate text classification technique (Dumais et al. 1998; Drucker et al. 1999; Joachims 1998; Leopold & Kindermann 2002). Therefore Support Vector Machines appears to be the best choice for the construction of a semantic space for textual documents.

### 6.1 Using an SVM to Quantify the Degree of Membership

Like other supervised machine learning algorithms, an SVM works in two steps. In the first step — the *training* step — it learns a decision boundary in input space from preclassified training data. In the second step — the *classification* step — it classifies input vectors according to the previously learned decision boundary. A *single* support vector machine can only separate *two* classes — a positive class ($y = +1$) and a negative class ($y = -1$). This means that for each of the $K$ classes $z_k$ a new SVM has to be trained separating $z_k$ from all other classes.

**In the training step** the following problem is solved: Given is a set of training examples $S_\ell = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_\ell, y_\ell)\}$ of size $\ell \leq W$ from a fixed but unknown distribution $p(\vec{x}, y)$ describing the learning task. The term-frequency

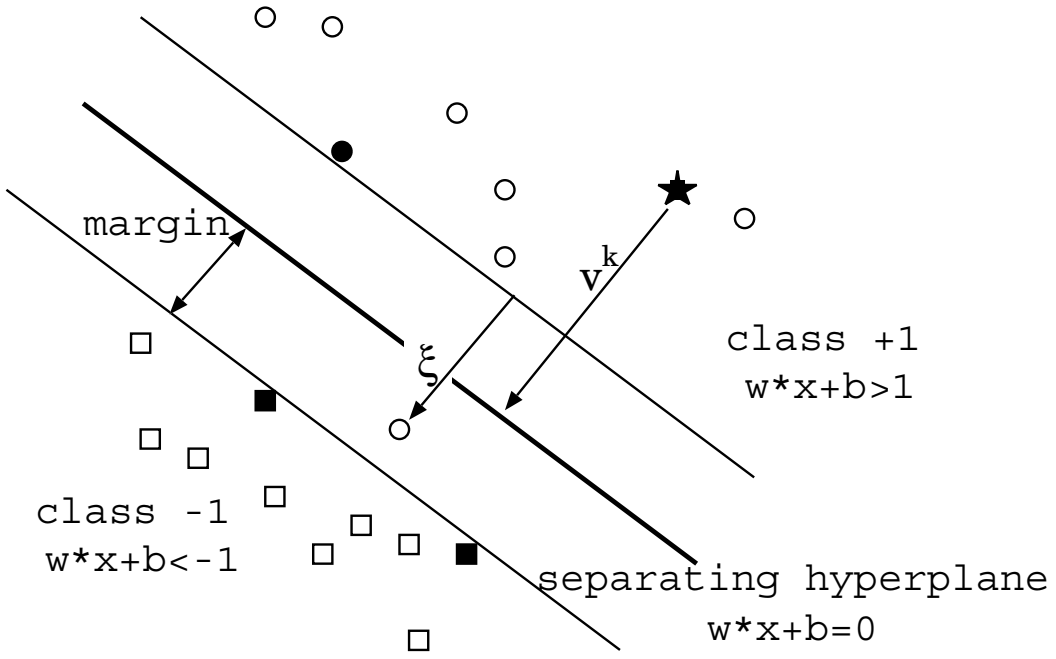


**Figure 1: Generating a CISS with a support vector machine.** The SVM algorithm seeks to maximise the margin around a hyperplane that separate a positive class (marked by circles) from a negative class (marked by squares). Once an SVM is trained, $v^k = \vec{w}^k\vec{x} + b$ is calculated in the classification step. The quantity $v^k$ measures the rectangular distance between the point marked by a star and the hyperplane. It can be used to generate a CISS.

vectors $\vec{x}_i$ represent documents and $y_i \in \{-1, +1\}$ indicates whether a document has been annotated as belonging to the positive class or not. The SVM aims to find a decision rule $h_{\mathcal{L}} : \vec{x} \rightarrow \{-1, +1\}$ based on $S_\ell$ that classifies documents as accurately as possible.

The hypothesis space is given by the functions $f(\vec{x}) = \text{sgn}(\vec{w}\vec{x} + b)$, where $\vec{w}$ and $b$ are parameters that are learned in the training step and which determine the class separating hyperplane. Computing this hyperplane is equivalent to solving the following optimisation problem (Vapnik 1998; Joachims 2002):

$$\begin{aligned} \text{minimise:} \quad & V(\vec{w}, b, \vec{\xi}) = \frac{1}{2}\vec{w}\vec{w} + C\sum_{i=1}^{\ell}\xi_i \\ \text{subject to:} \quad & \forall_{i=1}^{\ell} : y_i(\vec{w}\vec{x} + b) \geq 1 - \xi_i \\ & \forall_{i=1}^{\ell} : \xi_i \geq 0 \end{aligned}$$

The constraints require that all training examples are classified correctly allowing for some outliers, symbolised by the slack variables $\xi_i$. If a training example lies on the wrong side of the hyperplane, the corresponding $\xi_i$ is greater or equal to 0. The factor $C$ is a parameter that allows one to trade off training error against model complexity. Instead of solving the above optimization problem directly, it is easier to solve the following dual optimisation problem (Vapnik 1998; Joachims 2002).

$$\begin{aligned} \text{minimise:} \quad & W(\vec{\alpha}) = -\sum_{i=1}^{\ell} \alpha_i + \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} y_i y_j \alpha_i \alpha_j \vec{x}_i \vec{x}_j \\ \text{subject to:} \quad & \sum_{i=1 \; \underline{0} \leq \alpha_i \leq C}^{\ell} y_i \alpha_i = 0 \end{aligned} \tag{19}$$

All training examples with $\alpha_i > 0$ at the solution are called support vectors. The support vectors are situated right at the margin (see the solid squares and the circle in figure (1)) and define the hyperplane. The definition of a hyperplane by the support vectors is especially advantageous in high dimensional feature spaces because a comparatively small number of parameters — the $\alpha$s in the sum of equation (19) — is required.

**In the classification step** an unlabeled term-frequency vector is estimated to belong to the class

$$\hat{y} = \text{sgn}(\vec{w}\vec{x} + b) \tag{20}$$

Heuristically the estimated class membership $\hat{y}$ corresponds to whether $\vec{x}$ belongs on the lower or upper side of the decision hyperplane. Thus estimating the class membership by equation (20) consists of a loss of information since only the algebraic sign of right-hand term is evaluated. However the value of $v = \vec{w}\vec{x} + b$ is a real number and can be used in order to create a real valued semantic space, rather than just to estimate if $\vec{x}$ belongs to a given class or not.

### 6.2 Using Several Classes to Construct a Semantic Space

Suppose there are several, say $K$, classes of documents. Each document is represented by an input vector $\vec{x}_j$. For each document the variable $y_j^k \in \{-1, +1\}$ indicates whether $\vec{x}_j$ belongs to the $k$-th class ($k = 1, \ldots, K$) or not. For each class $k = 1, \ldots, K$ an SVM can be learned which yields the parameters $\vec{w}^k$ and

$b^k$. After the SVMs have been learned, the classification step (equation (20)) can be applied to a (possibly unlabeled) document represented by $\vec{x}$ resulting in a $K$-dimensional vector $\vec{v}$, whose $k$th component is given by $v^k = \vec{w}^k \cdot \vec{x} + b^k$ The component $v^k$ quantifies how much a document belongs to class $k$. Thus the document represented by the term frequency vector $\vec{x}_j$ is mapped to the $K$-dimensional vector in the classifier induced semantic space. Each dimension in this space can be interpreted as the membership degree of the document to each of the $K$ classes.

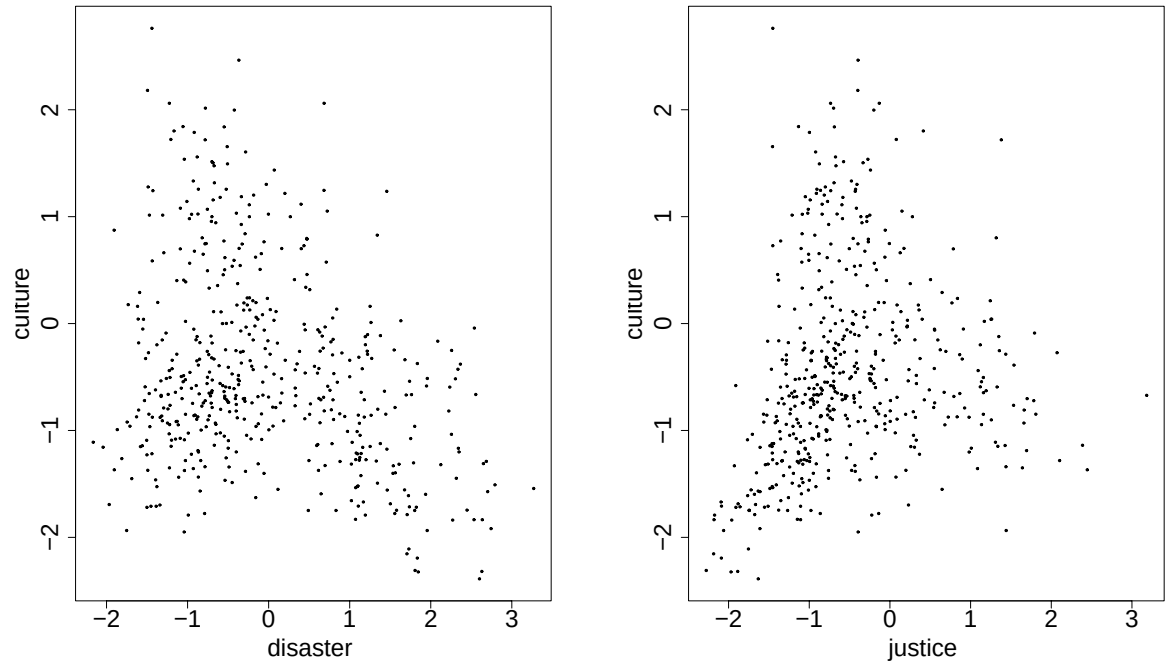


**Figure 2: A classifier induced semantic space.** 17 classifiers have been trained according to the highest level of the IPTC classification scheme. The projection to two dimensions "culture" and "disaster" is displayed on the right, and the projection to "culture" and "justice" on the left. The calculation is based on 68778 documents from the "Basisdienst" of the German Press Agency (dpa) July-October 2000.

The relation between PLSA and CISS is given by the latent variable $z_k$. In the context of CISS the latent variable $z_k$ is interpreted as the thematic domain, in accordance with semantic annotations in the corpus. Statistical learning theory assumes, that each class $k$ is learnable because there is an underlying conditional

distribution $p(\vec{x}_j \mid z_k)$, which reflects the special characteristics of the class $z_k$. The classification rules that are learned from the training data minimise the expected error. In PLSA the aspect variables are not previously defined. The conditioned probabilities $p(w_i \mid z_k)$ and $p(z_k \mid \vec{x}_j)$ are chosen in such a way that they maximise the likelihood of the multinomial model.

### 6.3 Graphical Representation of a CISS

Self-organising Maps (SOM) were invented in the early 80s (Kohonen 1980). They use a specific neural network architecture to perform a recursive regression leading to a reduction of the dimension of the data. For practical applications SOMs can be considered as a distance preserving mapping from a more than three-dimensional space to two-dimensions. A description of the SOM algorithm and a thorough discussion of the topic is given by Kohonen (1995).

Figure 3 shows an example of a SOM visualising the semantic relations of news messages. SVMs for the four classes 'culture', 'economy', 'politics', and 'sports' were trained by news messages from the 'Basisdienst' of the German Press Agency (dpa) April 2000. Classification and generation of the SOM was performed for the news messages of the first 10 days of April. 50 messages were selected at random and displayed as white crosses. The categories are indicated by different grey tone. Then the SOM algorithm is applied (with $100 \times 100$ nodes using Euclidean metric) in order to map the four-dimensional document representations to two dimensions admitting a minimum distortion of the distances. The grey tone indicates the topic category. Shadings within the categories indicate the confidence of the estimated class membership (dark = low confidence, bright = high confidence).

It can be seen that the change from sports (15) to economy (04) is filled by documents which cannot be assigned confidently to either classes. The area between politics (11) and economy (04), however, contains documents, which definitely belong to both classes. Note that classifier induced semantic spaces go beyond a mere extrapolation of the annotations found in the training corpus. It gives an insight into how typical a certain document is for each of the classes. Furthermore Classifier induced semantic spaces allow one to reveal previously unseen relationships between classes. The bright islands in area 11 on Figure 3 show, for example, that there are messages classified as economy which surely belong to politics.

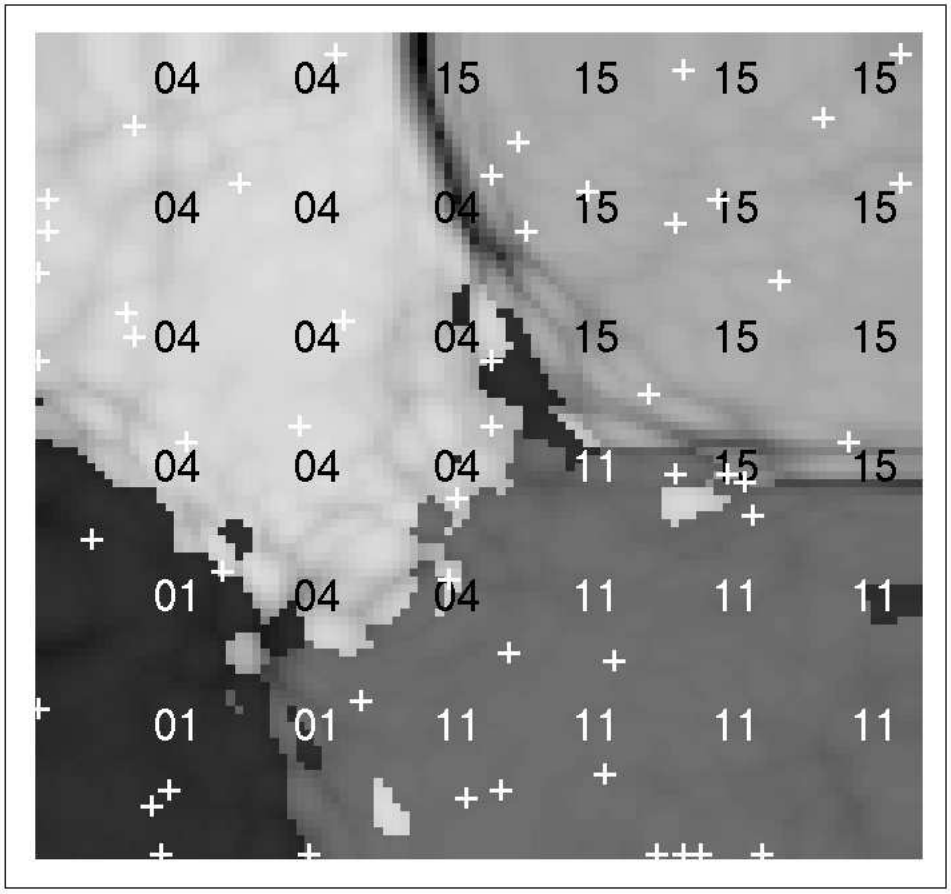


**Figure 3: Self-organising map of a classifier induced semantic space.** 4 classifiers have been trained according to the highest level of the IPTC classification scheme. The shadings and numbers indicate the "true" topic annotations of the news messages. 01: culture, 04: economy, 11: politics, 15: sports. (The figure was taken from Leopold et al. (2004)).

## 7 Conclusion

Fuzzy Linguistics, LSA, PLSA, and CISS map documents to the semantic space in a different manner. Fuzzy Lintuistics computes a vector for each word which consists of the cosine distances to every other word in the corpus. Then it calculates the Euclidean Dinstances between the vectors which gives the meaning point. Documents are represented by summing up the meaning points of the document's words.

In the case of LSA the representation of the document in the semantic space is achieved by matrix multiplication: $d_j \rightarrow \vec{x}_j^T U_K \Sigma_K$. The dimensions of the semantic space correspond to the $K$ largest eigen-values of the similarity matrix $AA^T$. The projection employed by LSA always leads to a global optimum in terms of the Euclidean distance between $A$ and $A_k$.

PLSA maps a document to the vector of the conditional probabilities, which indicate how probable aspect $z_k$ is, when document $d_j$ is selected: $d_j \rightarrow (p(z_1 \mid d_j), \ldots, p(z_K \mid d_j))$. The probabilities are derived from the aspect model using the maximum likelihood principle and the assumption of multinomially distributed word frequency distributions. The the likelihood function is maximised using the EM-algorithm, which is an iterative algorithm that leads only to a local optimum.

CISS requires a training corpus of documents annotated according to their membership of classes $z_k$. The classes have to be explicitly defined by the human annotation rules. For each class $z_k$ a classifier is trained, i.e. parameters $\vec{w}^k$ and $b^k$ are calculated from the training data. For each document $d_j$ the quantities $v^k = \vec{w}^k \cdot \vec{x} + b^k$ are calculated, which indicate how much $d_j$ belongs the previously learned classes $z_k$. The mapping of document $d_j$ to the semantic space is defined as $d_j \rightarrow (v_1, \ldots v_K)$. The dimensions can be interpreted according to the annotation rules.

## 8 Acknowledgements

This study is part of the project InDiGo which is funded by the German ministry for research and technology (BMFT) grant number 01 AK 915 A.

## References

Baayen, H. (2001). *Word Frequency Distributions*. Dordrecht: Kluwer.

Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review*, *37*(4), 573–595.

Chitashvili, R. J. & Baayen, R. H. (1993). Word frequency distributions. In G. Altmann & L. Hřebíček (Eds.), *Quantitative Text Analysis* (pp. 54–135). Trier: wvt.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshmann, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B*, *39*, 1–38.

Drucker, H., Wu, D., & Vapnik, V. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, *10*, 1048–1054.

Dumais, S., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In *Proceedings of the ACM-CIKM*, (pp. 148–155).

Goebl, H. (1991). Dialectrometry: A short overview of the principles and practice of quantitative classification of linguistic atlas data. In Köhler, R. & Rieger, B. B. (Eds.), *Contributions to quantitative linguistics, Proceedings of the first international conference on quantitative linguistics*, (pp. 277–315)., Dordrecht. Kluwer.

Gous, A. (1998). *Exponential and Spherical Subfamily Models*. PhD thesis, Stanford University.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, *42*, 177–196.

Hofmann, T. & Puzicha, J. (1998). Statistical models for co-occurrence data. A.I. Memo No. 1625., Massachusetts Institute of Technology.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In *Proceedings of the Tenth European Conference on Machine Learning (ECML 1998)*, (pp. 137–142)., Berlin. Springer.

Joachims, T. (2002). *Learning to classify text using support vector machines*. Boston: Kluwer.

Köhler, R. (1986). *Zur linguistischen Synergetik: Struktur und Dynamik der Lexik*. Bochum: Brockmeyer.

Kohonen, T. (1980). *Content-addressable Memories*. Berlin: Springer.

Kohonen, T. (1995). *Self-Organizing Maps*. Berlin: Springer.

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.

Leopold, E. & Kindermann, J. (2002). Text categorization with support vector machines. How to represent texts in input space? *Machine Learning*, *46*, 423–444.

Leopold, E., May, M., & Paaß, G. (2004). Data mining and text mining for science and technology research. In H. F. Moed, W. Glänzel, & U. Schmoch (Eds.), *Handbook of Quantitative Science and Technology Research* (pp. 187–214). Dordrecht: Kluwer.

Manning, C. D. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: MIT Press.

Mehler, A. (2002). Hierarchical orderings of textual units. In *Proceedings of the 19th International Conference on Computational Linguistics, COLING'02, Taipei*, (pp. 646–652)., San Francisco. Morgan Kaufmann.

Mitchell, T. M. (1997). *Machine Learning*. New York: McGraw-Hill.

Neumann, G. & Schmeier, S. (2002). Shallow natural language technology and text mining. *Künstliche Intelligenz, 2(2)*, 23–26.

Paaß, G., Leopold, E., Larson, M., Kindermann, J., & Eickeler, S. (2002). SVM classification using sequences of phonemes and syllables. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD), Helsinki*, (pp. 373–384)., Berlin. Springer.

Rieger, B. B. (1981). Feasible fuzzy semantics. On some problems of how to handle word meaning empirically. In H. Eikmeyer & H. Rieser (Eds.), *Words, Worlds, and Contexts. New Approaches in Word Semantics (Research in Text Theory 6)* (pp. 193–209). Berlin: de Gruyter.

Rieger, B. B. (1988). Definition of terms, word meaning, and knowledge structure. On some problems of semantics from a computational view of linguistics. In Czap, H. & Galinski, C. (Eds.), *Terminology and Knowledge Engineering. Proceedings International Congress on Terminology and Knowledge Engineering (Volume 2)*, (pp. 25–41)., Frankfurt a. M. Indeks.

Rieger, B. B. (1999). Computing fuzzy semantic granules from natural language texts. A computational semiotics approach to understanding word meanings. In Hamza, M. H. (Ed.), *Artificial Intelligence and Soft Computing, Proceedings of the IASTED International Conference, Anaheim/Calgary/Zürich*, (pp. 475–479). IASTED/Acta Press.

Rieger, B. B. (2002). Perception based processing of NL texts. Discourse understanding as visualized meaning constitution in scip systems. In Lotfi, A., John, B., & Garibaldi, J. (Eds.), *Recent Advances in Soft Computing (RASC-2002 Proceedings), Nottingham (Nottingham Trent UP)*, (pp. 506–511).

Rieger, B. B. & Thiopoulos, C. (1989). Situations, topoi, and dispositions: on the phenomenological modeling of meaning. In Retti, J. & Leidlmair, K. (Eds.), *5th Austrian Artificial Intelligence Conference, ÖGAI '89, Innsbruck, KI-Informatik-Fachberichte 208*, (pp. 365–375)., Berlin. Springer.

Salton, G. & McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. New York: McGraw Hill.

van Rijsbergen, C. J. (1975). *Information Retrieval*. London, Boston: Butterworths.

Vapnik, V. N. (1998). *Statistical Learning Theory*. New York: Wiley & Sons.