Probabilistic Latent Semantic Indexing

Thomas Hofmann
International Computer Science Institute, Berkeley, CA &
EECS Department, CS Division, UC Berkeley
hofmann@cs.berkeley.edu

Abstract

Probabilistic Latent Semantic Indexing is a novel approach to automated document indexing which is based on a statistical latent class model for factor analysis of count data. Fitted from a training corpus of text documents by a generalization of the Expectation Maximization algorithm, the utilized model is able to deal with domain–specific synonymy as well as with polysemous words. In contrast to standard Latent Semantic Indexing (LSI) by Singular Value Decomposition, the probabilistic variant has a solid statistical foundation and defines a proper generative data model. Retrieval experiments on a number of test collections indicate substantial performance gains over direct term matching methods as well as over LSI. In particular, the combination of models with different dimensionalities has proven to be advantageous.

1 Introduction

With the advent of digital databases and communication networks, huge repositories of textual data have become available to a large public. Today, it is one of the great challenges in the *information sciences* to develop intelligent interfaces for human–machine interaction which support computer users in their quest for relevant information. Although the use of elaborate ergonomic elements like computer graphics and visualization has proven to be extremely fruitful to facilitate and enhance information access, progress on the more fundamental question of *machine intelligence* is ultimately necessary to ensure substantial progress on this issue. In order for computers to interact more *naturally* with humans, one has to deal with the potential ambivalence, impreciseness, or even vagueness of user requests, and has to recognize the difference between what a user might say or do and what she or he actually meant or intended.

One typical scenario of human–machine interaction in information retrieval is by *natural language queries*: the user formulates a request, e.g., by providing a number of keywords or some free-form text, and expects the system to return the relevant data in some amenable representation, e.g., in form of a ranked list of relevant documents. Many retrieval methods are based on simple word matching strategies to determine the rank of relevance of a document with respect to a query. Yet, it is well known that literal term matching has severe drawbacks, mainly due to the ambivalence of words and their unavoidable lack of precision as well as due to personal style and individual differences in word usage.

Latent Semantic Analysis (LSA) [1] is an approach to automatic indexing and information retrieval that attempts to overcome these problems by mapping documents as well as terms to a representation in the so–called *latent semantic space*. LSA usually takes the (high dimensional) vector space representation of documents based on term frequencies [14] as a starting point and applies a dimension reducing linear projection. The specific form of this mapping is determined by a given document collection and is based on a *Singular Value Decomposition* (SVD) of the corresponding term/document matrix. The general claim is that similarities between documents or between documents and queries can be more reliably estimated in the reduced latent space representation than in the original representation. The rationale is that documents which share frequently co-occurring terms will have a similar representation in the latent space, even if they have no terms in common. LSA thus performs some sort of noise reduction and has the potential benefit to detect synonyms as well as words that refer to the same topic. In many applications this has proven to result in more robust word processing.

Although LSA has been applied with remarkable success in different domains including automatic indexing (Latent Semantic Indexing, LSI) [1, 3], it has a number of deficits, mainly due to its unsatisfactory statistical foundation. The primary goal of this paper is to present a novel approach to LSA and factor analysis – called *Probabilistic Latent Semantic Analysis* (PLSA) – that has a solid statistical foundation, since it is based on the likelihood principle and defines a proper generative model of the data. This implies in particular that standard techniques from statistics can be applied for questions like model fitting, model combination, and complexity control. In addition, the factor representation obtained by PLSA allows to deal with polysemous words and to explicitly distinguish between different meanings and different types of word usage.

SIGIR '99 8/99 Berkley, CA USA

2 The Aspect Model

The core of PLSA is a statistical model which has been called *aspect model* [7, 15]. The latter is a latent variable model for general co-occurrence data which associates an unobserved class variable $z \in \mathcal{Z} = \{z_1, \ldots, z_K\}$ with each observation, i.e., with each occurrence of a word $w \in \mathcal{W} = \{w_1, \ldots, w_M\}$ in a document $d \in \mathcal{D} = \{d_1, \ldots, d_N\}$. In terms of a generative model it can be defined in the following way:

- select a document d with probability $P(d)$,
- pick a latent class z with probability $P(z|d)$,
- generate a word w with probability $P(w|z)$.

As a result one obtains an observed pair (d, w), while the latent class variable z is discarded.

Translating this process into a joint probability model results in the expression

$$P(d,w) = P(d)P(w|d)\,, \text{ where} \tag{1}$$

$$P(w|d) = \sum_{z \in \mathcal{Z}} P(w|z)P(z|d)\ . \tag{2}$$

Essentially, to derive (2) one has to sum over the possible choices of z which could have generated the observation. The aspect model is a statistical *mixture model* [9] which is based on two independence assumptions: First, observation pairs (d, w) are assumed to be generated independently; this essentially corresponds to the 'bag–of–words' approach. Secondly, the *conditional independence* assumption is made that conditioned on the latent class z, words w are generated independently of the specific document identity d. Given that the number of states is smaller than the number of documents ($K \ll N$), z acts as a bottleneck variable in predicting w conditioned on d.

Notice that in contrast to *document clustering* models document–specific word distributions $P(w|d)$ are obtained by a convex combination of the *aspects* or *factors* $P(w|z)$. Documents are not assigned to clusters, they are characterized by a specific mixture of factors with weights $P(z|d)$. These mixing weights offer more modeling power and are conceptually very different from posterior probabilities in clustering models and (unsupervised) naive Bayes models (cf. [7]).

Following the likelihood principle, one determines $P(d)$, $P(z|d)$, and $P(w|z)$ by maximization of the log–likelihood function

$$\mathcal{L} = \sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{W}} n(d,w) \log P(d,w)\,, \tag{3}$$

where $n(d, w)$ denotes the term frequency, i.e., the number of times w occurred in d. It is worth noticing that an equivalent symmetric version of the model can be obtained by inverting the conditional probability $P(z|d)$ with the help of Bayes' rule, which results in

$$P(d,w) = \sum_{z \in \mathcal{Z}} P(z)P(w|z)P(d|z)\,. \tag{4}$$

This is just a re-parameterized version of the generative model described by (1), (2).

3 Model Fitting with Tempered EM

The standard procedure for maximum likelihood estimation in latent variable models is the Expectation Maximization (EM) algorithm [2]. EM alternates two steps: (i) an expectation (E) step where posterior probabilities are computed for the latent variables z, based on the current estimates of the parameters, (ii) an maximization (M) step, where parameters are updated for given posterior probabilities computed in the previous E–step.

For the aspect model in the symmetric parameterization Bayes' rule yields the E–step

$$P(z|d,w) = \frac{P(z)P(d|z)P(w|z)}{\sum_{z'} P(z')P(d|z')P(w|z')}\,, \tag{5}$$

which is the probability that a word w in a particular document or context d is explained by the factor corresponding to z. By standard calculations one arrives at the following M–step re-estimation equations

$$P(w|z) = \frac{\sum_d n(d,w)P(z|d,w)}{\sum_{d,w'} n(d,w')P(z|d,w')}, \tag{6}$$

$$P(d|z) = \frac{\sum_w n(d,w)P(z|d,w)}{\sum_{d',w} n(d',w)P(z|d',w)}, \tag{7}$$

$$P(z) = \frac{1}{R}\sum_{d,w} n(d,w)P(z|d,w),\ R \equiv \sum_{d,w} n(d,w)\,. \tag{8}$$

Alternating (5) with (6)–(8) defines a convergent procedure that approaches a local maximum of the log–likelihood in (3).

So far we have focused on maximum likelihood estimation or, equivalently, word perplexity reduction. One has, however, to distinguish between the predictive performance of the model on training data and the expected performance on unseen test data. In particular, it is to naive to assume that a model will generalize well on new data just based on the fact that it might achieve low perplexity on training data. To derive conditions under which generalization on unseen data can be guaranteed is actually *the* fundamental problem of statistical learning theory. Here, we propose a generalization of maximum likelihood for mixture models – called *tempered EM* (TEM) – which is based on entropic regularization and is closely related to a method known as *deterministic annealing* [13].

Since a principled derivation of TEM is beyond the scope of this paper (the interested reader is referred to [12, 7]), we will present the necessary modification of standard EM in an *ad hoc* manner. Essentially, one introduces a control parameter β (inverse computational temperature) and modifies the E-step in (5) according to

$$P_\beta(z|d,w) = \frac{P(z)\left[P(d|z)P(w|z)\right]^\beta}{\sum_{z'} P(z')\left[P(d|z')P(w|z')\right]^\beta}. \tag{9}$$

Notice that $\beta = 1$ results in the standard E–step, while for $\beta < 1$ the likelihood part in Bayes' formula is discounted (additively on the log–scale).

It can be shown, that TEM minimizes an objective function known as the free energy [11] and hence defines a convergent algorithm. While temperature–based generalizations of EM and related algorithms for optimization are often used

as a homotopy or continuation method to avoid unfavorable local extrema, the main advantage of TEM in our context is to avoid overfitting. Somewhat contrary to the spirit of annealing as a continuation method we propose to utilize (9) to temper EM by "heating". In order to determine the optimal value of β we propose to make use of some held–out portion of the data. This idea can be implemented by the following scheme:

i. Set $\beta \leftarrow 1$ and perform EM until the performance on held–out data deteriorates (*early stopping*).

ii. Decrease β, e.g., by setting $\beta \leftarrow \eta\beta$ with some rate parameter $\eta < 1$.

iii. As long as the performance on held-out data improves continue TEM iterations at this value of β.

iv. Stop on β, i.e., stop when decreasing β does not yield further improvements, otherwise goto step (ii).

v. Perform some final iterations using both, training and held-out data.

In our experiments, the typical number of iterations TEM performed starting from randomized initial conditions was $40-60$, where each iteration requires one pass through the data, i.e., of the order of $R \cdot K$ arithmetical operations.

4 Probabilistic Latent Semantic Analysis

4.1 Latent Semantic Analysis

As mentioned in the introduction, the key idea of LSA [1] is to map documents (and by symmetry terms) to a vector space of reduced dimensionality, the *latent semantic space*. This mapping is computed by decomposing the term/document matrix $\mathbf{N}$ with SVD, $\mathbf{N} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^t$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices $\mathbf{U}^t\mathbf{U} = \mathbf{V}^t\mathbf{V} = \mathbf{I}$ and the diagonal matrix $\mathbf{\Sigma}$ contains the singular values of $\mathbf{N}$. The LSA approximation of $\mathbf{N}$ is computed by thresholding all but the largest K singular values in $\mathbf{\Sigma}$ to zero ($= \hat{\mathbf{\Sigma}}$), which is rank K optimal in the sense of the L_2-matrix norm as is well-known from linear algebra, i.e., one obtains the approximation $\hat{\mathbf{N}} = \mathbf{U}\hat{\mathbf{\Sigma}}\mathbf{V}^t \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^t = \mathbf{N}$. Note that the L_2–norm approximation does not prohibit entries of $\hat{\mathbf{N}}$ to be negative.

4.2 Geometry of the Aspect Model

Now consider the class-conditional multinomial distributions $P(\cdot|z)$ over the vocabulary in the aspect model which can be represented as points on the $M-1$ dimensional simplex of all possible multinomials. Via its convex hull, this set of K points defines a $K-1$ dimensional sub-simplex. The modeling assumption expressed by (2) is that all conditional distributions $P(\cdot|d)$ are approximated by a multinomial representable as a convex combination of the class-conditionals $P(\cdot|z)$. In this geometrical view, the mixing weights $P(z|d)$ correspond exactly to the coordinates of a document in that sub-simplex. A simple sketch of the geometry is shown in Figure 1. This demonstrates that despite of the discreteness of the latent variables introduced in the aspect model, a *continuous latent space* is obtained within the space of all multinomial distributions. Since the dimensionality of the sub-simplex is $K-1$ as opposed to $M-1$ for the complete

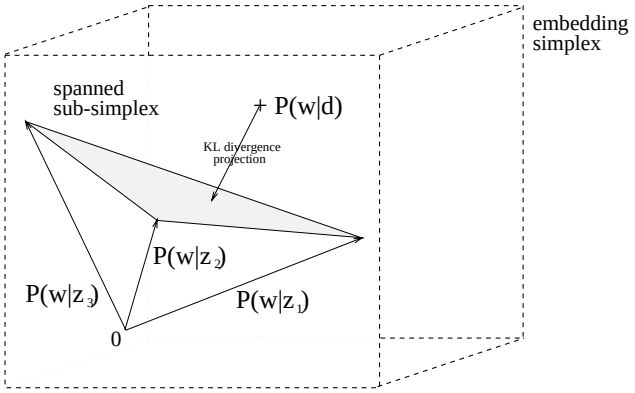

Figure 1: Sketch of the probability sub-simplex spanned by the aspect model.

probability simplex, this can also be thought of in terms of dimensionality reduction and the sub-simplex can be identified with a *probabilistic latent semantic space*.

4.3 Mixture Decomposition vs. Singular Value Decomposition

To stress this point and to clarify the relation to LSA, let us rewrite the aspect model as parameterized by (4) in matrix notation. Hence define matrices by $\hat{\mathbf{U}} = (P(d_i|z_k))_{i,k}$, $\hat{\mathbf{V}} = (P(w_j|z_k))_{j,k}$, and $\hat{\mathbf{\Sigma}} = \mathrm{diag}(P(z_k))_k$. The joint probability model $\mathbf{P}$ can then be written as a matrix product $\mathbf{P} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^t$. By comparing this decomposition with the SVD decomposition in LSA, one can point out the following re-interpretation of concepts of linear algebra:

i. The weighted sum over outer products between rows of $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ reflects conditional independence in PLSA.

ii. The left/right eigenvectors in SVD are seen to correspond to the factors $P(w|z)$ and the component distributions $P(d|z)$ of the aspect model.

iii. The mixing proportions $P(z)$ in PLSA substitute the singular values of the SVD in LSA.

Despite this similarity, there is also a fundamental difference between PLSA and LSA, which is the objective function utilized to determine the optimal decomposition/approximation. In LSA, this is the L_2–norm or Frobenius norm, which corresponds to an implicit additive Gaussian noise assumption on counts. In contrast, PLSA relies on the likelihood function of multinomial sampling and aims at an explicit maximization of the predictive power of the model. On the modeling side this offers important advantages, for example, the mixture approximation $\mathbf{P}$ of the co-occurrence table is a well-defined probability distribution and factors have a clear probabilistic meaning in terms of mixture component distributions.

4.4 Kullback–Leibler Projection vs. Orthogonal Projection

Returning to the geometrical view of the aspect model as sketched in Figure 1, it is interesting to reveal the projection principle which is implicitly used in the aspect model.

"plane"	"space shuttle"	"family"	"Hollywood"
plane	space	home	film
airport	shuttle	family	movie
crash	mission	like	music
flight	astronauts	love	new
safety	launch	kids	best
aircraft	station	mother	hollywood
air	crew	life	love
passenger	nasa	happy	actor
board	satellite	friends	entertainment
airline	earth	cnn	star

Table 1: Four factors from a 128 factor decomposition of the TDT-1 corpus. Factor are represented by their 10 most probable words, i.e., the words are ordered according to $P(w|z)$.

"Bosnia"	"Iraq"	"Rwanda"	"Kobe"
un	iraq	refugees	building
bosnian	iraqi	aid	city
serbs	sanctions	rwanda	people
bosnia	kuwait	relief	rescue
serb	un	people	buildings
sarajevo	council	camps	workers
nato	gulf	zaire	kobe
peacekeepers	saddam	camp	victims
nations	baghdad	food	area
peace	hussein	rwandan	earthquake

Table 2: Four additional factors from the 128 factor decomposition of the TDT-1 corpus (cf. Table 1).

Rewriting the log–likelihood in (3) one arrives at

$$\mathcal{L} = \sum_{d \in \mathcal{D}} n(d) \left[\sum_{w \in \mathcal{W}} \frac{n(d,w)}{n(d)} \log P(w|d) + \log P(d) \right] . \quad (10)$$

The first term in brackets corresponds to the negative Kullback–Leibler (KL) divergence (or cross–entropy) between the empirical distribution of words in a document $\hat{P}(w|d) \equiv n(d,w)/n(d)$ and the model distribution $P(w|d)$. For fixed factors $P(w|z)$ maximizing the log–likelihood w.r.t the mixing proportions $P(z|d)$ thus amounts to projecting $\hat{P}(w|d)$ on the subspace spanned by the factors based on the KL–divergence. This is very different from any type of squared deviation which would result in an orthogonal projection (cf. [10] for more details on the geometry of statistical models).

4.5 Factor Representation: An Example

In order to visualize the factor solution found by PLSA we present an elucidating example. We have performed experiments with the TDT-1 collection, which contains 15,862 documents of broadcast news stories [8].[1] Stop words have been eliminated by a standard stop word list, no stemming or further preprocessing has been performed. Table 1 shows a reduced representation of 4 factors from a 128 factor solution.

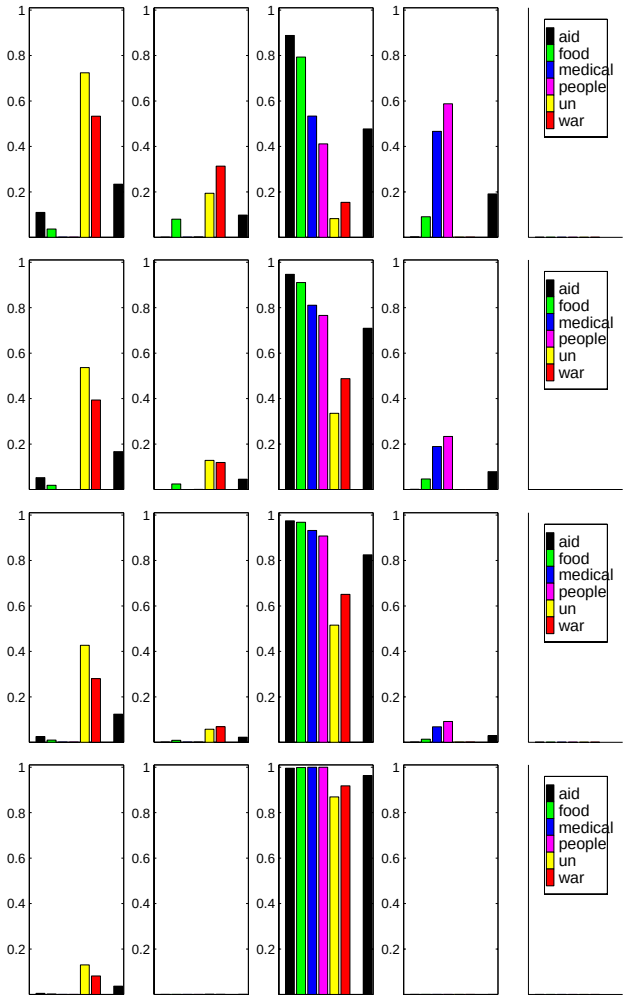

Figure 2: Folding in a query conisting of the terms "aid", "food", "medical", "people", "UN", and "war": evolution of posterior probabilities and the mixing proportions $P(z|q)$ (rightmost column in each bar plot) for the four factors depicted in Table 2 after 1 (first row), 2 (second row), 3 (third row), and 20 (fourth row) iterations.

The first two factors have been selected as the ones with the highest probablity to generate the word "flight", the last two factors have the highest probability to generate the word "love". It is interesting to see that the first two factors indeed capture two different types of usage for the term "flight": flights with planes and flights with space ships/shuttles. Similarly the last two factors capture two distinguishable contexts in which the word "love" occurs in

[1] Since the TDT-1 collection contains documents on topics and events most readers will be familiar with, this collection has been preferred over the test collections utilized in Section 6.

	MED		CRAN		CACM		CISI	
	precision	improvement	precision	improvement	precision	improvement	precision	improvement
cos+tf	44.3	-	29.9	-	17.9	-	12.7	-
LSI	51.7	+16.7	*28.7	-4.0	*16.0	-11.6	12.7	±0.0
PLSI-U	63.1	+42.4	32.8	+9.7	19.2	+7.2	14.0	+10.2
PLSI-Q	63.9	+44.2	35.1	+17.4	22.9	+27.9	18.8	+48.0
PLSI-U*	**67.5**	**+52.4**	33.3	+11.4	19.5	+8.9	14.7	+15.7
PLSI-Q*	66.3	+49.7	**37.5**	**+25.4**	**26.8**	**+49.7**	**20.1**	**+58.3**
cos+tfidf	49.0	-	35.2	-	21.9	-	20.2	-
LSI	64.6	+31.8	38.7	+9.9	23.8	+8.7	21.9	+8.4
PLSI-U	69.5	+41.8	38.9	+10.5	25.3	+15.5	23.3	+15.3
PLSI-Q	63.2	+29.0	38.6	+9.7	26.6	+21.5	23.1	+14.4
PLSI-U*	**72.1**	**+47.1**	**40.4**	**+14.8**	27.6	+26.0	**24.6**	**+21.8**
PLSI-Q*	66.3	+35.3	40.1	+13.9	**28.3**	**+29.2**	24.4	+20.8

Table 3: Average precision results and relative improvement w.r.t. the baseline method (cos+tf and cos+tfidf, respectively) for the 4 standard test collections. Compared are LSI, PLSI, and the two PLSI variants (PLSI-U, PLSI-Q) as well as results obtained by combining PLSI models (PLSI-U* and PLSI-Q*, respectively). An asterix for LSI indicates that no performance gain could be achieved over the baseline, the result at 256 dimensions with a 1 : 2 combination with the baseline score is reported in this case.

the TDT-1 collection: real love in the context of family life as opposed to staged love in the sense of "Hollywood".

4.6 Folding-In Queries

Folding-in refers to the problem of computing a representation for a document or query that was not contained in the original training collection. In the LSA approach, this is simply done by a linear mapping that effectively represents a document or query by the center of its constituent terms (with an appropriate term weighting) [1]. In PLSA, mixing proportions can be computed by EM iteration, where the factors are fixed such that only the mixing proportions $P(z|q)$ are adapted in each M–step.

Table 2 shows some more factors for the TDT-1 collection which clearly reflect the vocabulary dealing with certain events: the war in Bosnia and Iraq, the crisis in Rwanda, and the earthquake in Kobe. Based on this four factors, we have computed a representation for a test query consisting of the terms "aid", "food", "medical", "people", "UN", and "war". Figure 2 visualizes the evolution of the posterior probabilities and the mixing proportions in the course of the EM procedure. The query has been designed such that only the "Rwanda" factor is matching all query terms (e.g., the UN was not involved in the Kobe earthquake, there was no medical aid provided for the Iraq during the Gulf war, etc.). As can be seen this factor has indeed the highest weight after the first iteration, but notice that the other factors still account for more than half of the probability. However this changes after some EM iterations, since the aspect model introduces feedback between the terms. For example, although a term like "UN" would by itself be best explained by the "Bosnia" factor, the context of the other query terms drastically increases the probability that this particular occurrence of "UN" is related to the events in Rwanda. The same mechanism is able to detect "true" polysems [6].

5 Probabilistic Latent Semantic Indexing

5.1 Vector-Space Models and LSI

One of the most popular families of information retrieval techniques is based on the *Vector–Space Model* (VSM) for documents [14]. A VSM variant is characterized by three ingredients: (i) a transformation function (also called local term weight), (ii) a term weighting scheme (also called global term weight), and (iii) a similarity measure. In our experiments we have utilized (i) a representation based on the (untransformed) term frequencies (tf) $n(d,w)$ which has been combined with (ii) the popular *inverse document frequency* (idf) term weights, and the (iii) standard cosine matching function. The same representation applies to queries q such that the matching function for the baseline methods can be written as

$$s(d,q) = \frac{\sum_w \hat{n}(d,w)\hat{n}(q,w)}{\sqrt{\sum_w \hat{n}(d,w)^2}\sqrt{\sum_w \hat{n}(q,w)^2}}, \qquad (11)$$

where $\hat{n}(d,w) = \mathrm{idf}(w) \cdot n(d,w)$ are the weighted word frequencies.

In latent semantic indexing, the original vector space representation of documents is replaced by a representation in the low–dimensional latent space and the similarity is computed based on that representation. Queries or documents which were not part of the original collection can be *folded in* by a simple matrix multiplication (cf. [1] for details). In

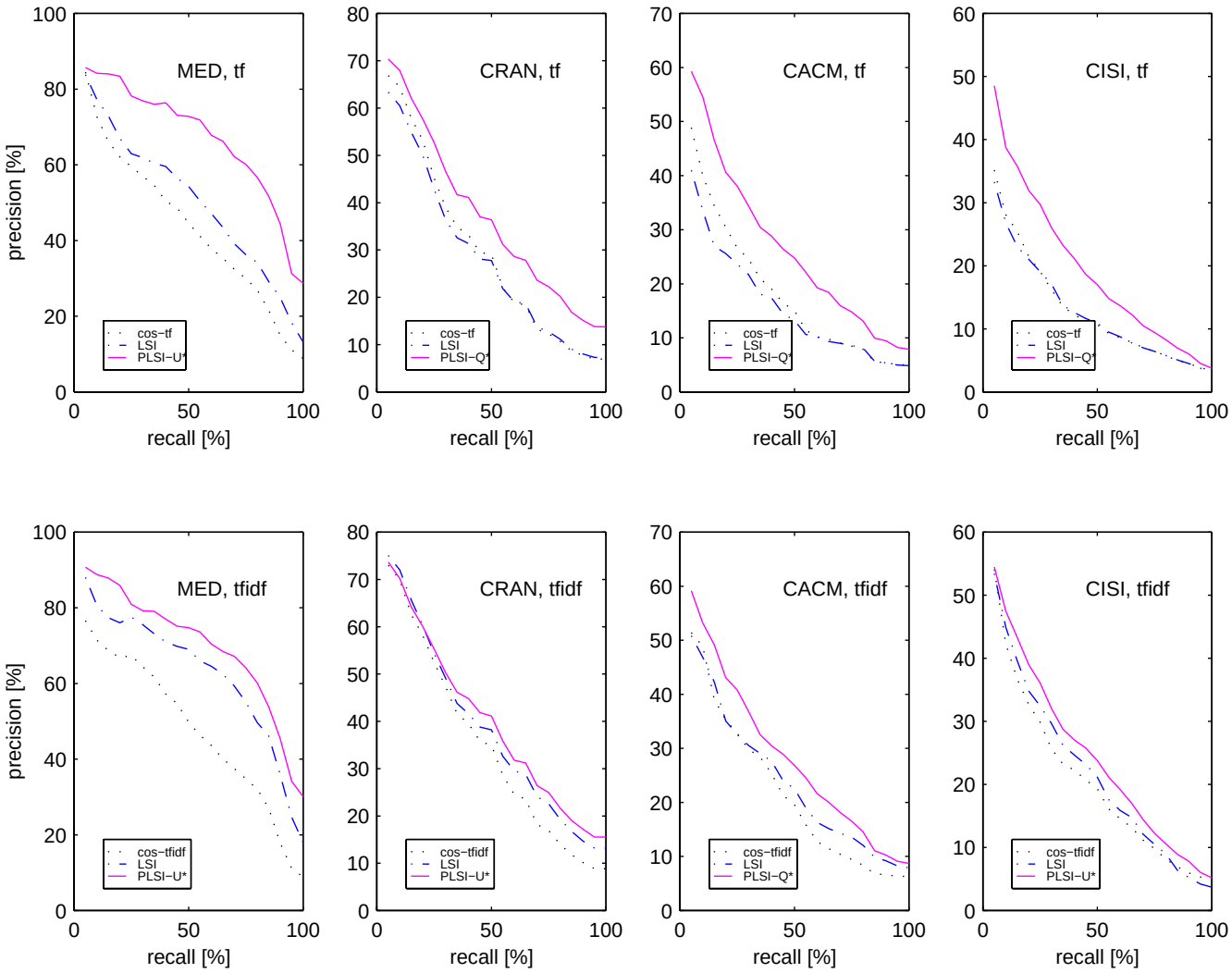

Figure 3: Precision–recall curves for the 4 test collections with idf term weighting (lower row) and without (upper row). Depicted are curves for direct term matching, LSI, and the best performing PLSI* variant.

our experiments, we have actually considered linear combinations of the original similarity score (11) (weight λ) and the one derived from the latent space representation (weight $1-\lambda$), as suggested in [3] (cf. [16] for a more detailed empirical investigation of linear combination schemes for information retrieval systems).

5.2 Variants of Probabilistic Latent Semantic Indexing

Two different schemes to exploit PLSA for indexing have been investigated: (i) as a context–dependent *unigram model* to smoothen the empirical word distributions in documents (PLSI-U), (ii) as a latent space model which provides a low–dimensional document/query representation (PLSI-Q):

PLSI-U For each document d in the collection, PLSA provides a multinomial distribution $P(w|d)$ over the vocabulary as given by (2). This distribution will in general be a smooth version of the empirical distribution $\hat{P}(w|d) = n(d,w)/n(d)$. We propose to utilize $P(w|d)$ (thought of as a document vector) in order to compute a matching score between a document an a query. Notice that $P(w|d)$ is a representation in the original (word) space obtained by back–projection from the probabilistic latent space. The vector $P(\cdot|d)$ can (optionally) be weighted with the inverse document frequencies and compared with the (weighted) query by the cosine.[2] We have considered two ways of combining PLSA-U with the standard VSM: (i) by linearly combining the cosine similarities as discussed above for LSI, and (ii) by additively combining the multinomials like in interpolation methods for language modeling, i.e., by using the representation $\tilde{P}(w|d) = \lambda\hat{P}(w|d) + (1-\lambda)P(w|d)$. Both methods have empirically shown almost identical performance and we will only report results of variant (i), because this scheme has also been used in the case of LSI.

[2] Folding–in queries, though possible, has empirically shown no advantages in the PLSI-U scheme.

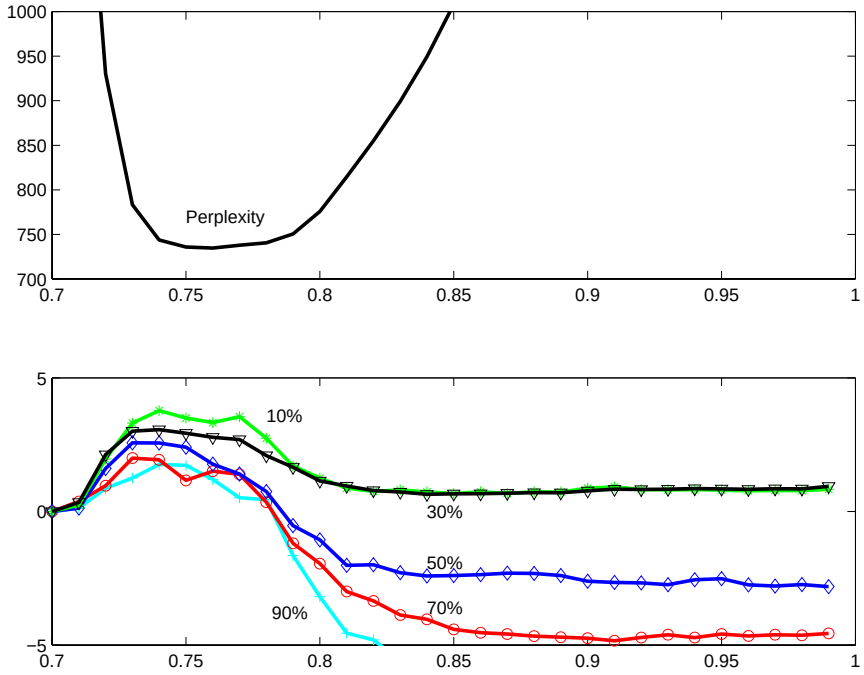

Figure 4: Model performance $K = 256$ on the Cranfield collection in terms of perplexity (upper plot) and precision (lower plot, absolute gain vs. baseline for different recall levels) at different values of β. The model has been annealed ($0.7 \rightarrow \beta \rightarrow 1.0$) and trained up to convergence; no early stopping was performed.

PLSI-Q In this scheme we use the low–dimensional representation $P(z|d)$ and $P(z|q)$ to evaluate similarities. Therefore, queries have to be folded in, which is done by fixing the $P(w|z)$ parameters and calculating weights $P(z|q)$ by TEM. How to optimally take into account (global) term weights in PLSI-Q is an only partially resolved problem. We have used the *ad hoc* approach to reweight the different model components by the quantities $\sum_w P(w|z) \cdot \mathrm{idf}(w)$, but this may not make optimal use of the term weight priors.

One advantage of using statistical models vs. SVD techniques is that it allows us to systematically combine different models. While this should optimally be done according to a Bayesian model combination scheme, we have utilized a much simpler approach in our experiments which has nevertheless shown excellent performance and robustness. In the PLSI-U we have combined the probability estimates $P(w|d)$ for models with different number of components K additively with uniform weights. In the PLSI-Q scheme, we have simply combined the cosine scores of all models with a uniform weight. The resulting methods are referred to as PLSI-U* and PLSI-Q*, respectively. Empirically we have found the performance to be very robust w.r.t. different (non-uniform) weights and also w.r.t. the λ–weight used in combination with the original cosine score. This is due to the noise reducing benefits of (model) averaging. Notice that LSA representations for different K form a nested sequence, which is not true for the statistical models which are expected to capture a larger variety of reasonable decompositions.

6 Experimental Results

The performance of PLSI has been systematically compared with the standard term matching method based on the raw term frequencies (tf) and their combination with the inverse document frequencies (tfidf), as well as with LSI. We have utilized the following four medium–sized standard document collection: (i) MED (1033 document abstracts from the National Library of Medicine), (ii) CRAN (1400 document abstracts on aeronautics from the Cranfield Institute of Technology), (iii) CACM (3204 abstracts from the CACM journal), and (iv) CISI (1460 abstracts in library science from the Institute for Scientific Information). The condensed results in terms of average precision recall (at the 9 recall levels $10\% - 90\%$) are summarized in Table 3. A selection of average precision recall curves can be found in Figure 3.

Here are some details of the experimental setup: PLSA models at $K = 32, 48, 64, 80, 128$ have been trained by TEM for each data set with 10% held–out data. For PLSI-U/PLSI-Q we report the best result obtained by any of these models, for LSI we report the best result obtained for the optimal dimension (exploring 32–512 dimensions at a step size of 8). The combination weight λ with the cosine baseline score has been coarsely optimized by hand, MED, CRAN: $\lambda = 1/2$, CACM, CISI: $\lambda = 2/3$; in general slightly smaller weights have been utilized for the combined models.

The experiments consistently validate the advantages of PLSI over LSI. Substantial performance gains have been achieved for all 4 data sets and both term weighting schemes. In particular, PLSI-Q/PLSI-Q* work particularly well on the raw term frequencies, where LSI on the other hand may even fail completely (in accordance with the results

reported in [1]). We explain this by the fact that large frequencies dominate the squared error deviation used in SVD and a dampening (e.g., by idf weighting) is necessary to get a reasonable decomposition of the term/document matrix. Since PLSI-Q can not take much advantage from the term weighting scheme, PLSI-U/PLSI-U* performs slightly better in this case. We suspect that even better results could be achieved by an improved integration of term weights in PLSI-Q. The benefits of model combination are also very substantial. In all cases the (uniformly) combined model performed better than the best single model. As a sight-effect, model averaging also deliberates from selecting the "optimal" model dimensionality.

In terms of computational complexity, despite of the iterative nature of EM, the computing time for TEM model fitting at $K = 128$ was roughly comparable to SVD in a standard implementation. For larger data sets one may also consider speeding up TEM by on-line learning [11]. Notice that the PLSI-Q scheme has the advantage that documents can be represented in a low–dimensional vector space (as in LSI), while PLSI-U requires the calculation of the high–dimensional multinomials $P(w|d)$ which offers advantages in terms of the space requirements for the indexing information that has to be stored.

Finally, we have also performed an experiment to stress the importance of tempered EM over standard EM–based model fitting. Figure 4 plots the performance of a 128 factor model trained on CRAN in terms of perplexity and in terms of precision as a function of β. It can be seen that it is crucial to control the generalization performance of the model, since the precision is inversely correlated with the perplexity. In particular, notice that the model obtained by maximum likelihood estimation (at $\beta = 1$) actually deteriorates the retrieval performance.

7 Conclusion and Outlook

We have presented a novel method for automated indexing based on a statistical ~~latent class~~ model. This approach has important theoretical advantages over standard LSI, since it is based on the likelihood principle, defines a generative data model, and directly minimizes word perplexity. It can also take advantage of statistical standard methods for model fitting, overfitting control, and model combination. The empirical evaluation has clearly confirmed the benefits of Probabilistic Latent Semantic Indexing which achieves significant gains in precision over both, standard term matching and LSI. Further investigation is needed to take full advantage of the prior information provided by term weighting schemes. Recent work has also shown that the benefits of PLSA extend beyond document indexing and that a similar approach can be utilized, e.g., for language modeling [4] and collaborative filtering [5].

Acknowledgment

This work has been supported by a DAAD postdoctoral fellowship.

References

[1] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* (1990).

[2] Dempster, A., Laird, N., and Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. B 39* (1977), 1–38.

[3] Dumais, S. T. Latent semantic indexing (lsi): Trec-3 report. In *Proceedings of the Text REtrieval Conference (TREC-3)* (1995), D. Harman, Ed., pp. 219–30.

[4] Gildea, D., and Hofmann, T. Topic-based language models using em. In *Proceedings of the 6th European Conference on Speech Communication and Technology(EUROSPEECH)* (1999).

[5] Hofmann, T. Latent class models for collaborative filtering. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)* (1999).

[6] Hofmann, T. Probabilistic latent semantic analysis. In *Proceedings of the 15th Conference on Uncertainty in AI* (1999).

[7] Hofmann, T., Puzicha, J., and Jordan, M. I. Unsupervised learning from dyadic data. In *Advances in Neural Information Processing Systems* (1999), vol. 11.

[8] Linguistic Data Consortium. TDT pilot study corpus. Catalog no. LDC98T25, 1998.

[9] McLachlan, G., and Basford, K. E. *Mixture Models*. Marcel Dekker, INC, New York Basel, 1988.

[10] Murray, M. K., and Rice, J. W. *Differential geometry and statistics*. No. 48 in Monographs on statistics and applied probability. Chapman & Hal, 1993.

[11] Neal, R., and Hinton, G. A view of the EM algorithm that justifies incremental and other variants. In *Learning in Graphical Models*, M. Jordan, Ed. Kluwer Academic Publishers, 1998, pp. 355–368.

[12] Pereira, F., Tishby, N., and Lee, L. Distributional clustering of english words. In *Proceedings of the ACL* (1993), pp. 183–190.

[13] Rose, K., Gurewitz, E., and Fox, G. A deterministic annealing approach to clustering. *Pattern Recognition Letters 11*, 11 (1990), 589–594.

[14] Salton, G., and McGill, M. J. *Introduction to Modern Information Retrieval*. McGraw–Hill, 1983.

[15] Saul, L., and Pereira, F. Aggregate and mixed–order Markov models for statistical language processing. In *Proceedings of the 2nd International Conference on Empirical Methods in Natural Language Processing* (1997).

[16] Vogt, C. C., and Cottrell, G. W. Predicting the performance of linearly combined IR systems. In *Proceedings of the 21st ACM-SIGIR International Conference on Research and Development in Information Retrieval, Melbourne, Australia* (1998).