Form Latent Semantic Indexing to Language Models and Back

Thomas Hofmann
Department of Computer Science, Brown University, Providence, RI 02912
& RecomMind Inc., Berkeley, CA 94710
th@cs.brown.edu

1 Introduction

One of the key challenges in information retrieval is the problem of *automated indexing*. How can computers be used to automatically extract relevant index terms from documents? How should documents be represented to facilitate information access? Primarily, a good document representation should capture the topical and semantical relationships between documents. Thereby, it should support the computation of similarities between documents and queries or other documents.

From the early years of information retrieval, it has been realized that automated indexing should get to the semantic level of the meaning of words. An important example is the idea of *notional families* in the work of H.P. Luhn [5]. Ideally, notional families group together words of similar and related meaning and use these *concepts* to encode documents.

In this paper, we will put our own work [4] in a new context and show how a combination of ideas from latent semantic indexing [2] with the language modeling approach to information retrieval [6] leads to a statistical retrieval model that is very close in spirit to notional families.

2 Latent Semantic Indexing

Latent Semantic Indexing (LSI) [2] is a well-known information retrieval technique that attempts to partially implement semantic or concept-based retrieval. The essential ingredient is a dimension reduction technique: Starting from a standard vector space representation of documents, LSI maps documents – and by duality terms – to a low-dimensional, semantic space. The specific form of this mapping is learned from a given document collection by applying a Singular Value Decomposition (SVD) to the term-document matrix. The idea and hope is that axis in this latent space will correspond to meaningful concepts and that directions in the original vector space representation that correspond to synonyms and semantical related words will be mapped to a common direction in the semantic space.

Formally, denote by $\mathbf{N}$ the term-document matrix, for example using a *tfidf*–representation $n_{ij} = \mathrm{tf}(w_j, d_i) \cdot \mathrm{idf}(w_j)$, where d_i $(1 \leq i \leq n)$ refers to the i-th document and w_j $(1 \leq j \leq m)$ refers to the k-th term. Then LSI computes an approximation $\hat{\mathbf{N}}$ according to

$$\hat{\mathbf{N}} = \mathbf{U}\hat{\mathbf{\Sigma}}\mathbf{V}^t \approx \mathbf{U}\hat{\mathbf{\Sigma}}\mathbf{V}^t = \mathbf{N}\,. \tag{1}$$

Here, $\mathbf{U}$ and $\mathbf{V}$ denote matrices such that $\mathbf{U}^t\mathbf{U} = \mathbf{V}^t\mathbf{V} = \mathbf{I}$, $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0)$ is a diagonal matrix which contains the singular values, $\sigma_i \geq \sigma_j$ for $i < j$, $r = \mathrm{rank}(\mathbf{N})$, and $\hat{\mathbf{\Sigma}} = (\sigma_1, \ldots, \sigma_s, 0, \ldots, 0)$, $s < r$. The approximation $\hat{\mathbf{N}}$ is known to be optimal among rank s matrices with respect to the Frobenius and L_2 matrix norms.

In LSI, document similarities $s(d_i, d_j)$ are computed according to

$$s(d_i, d_j) = s_{ij}\ ,\ \mathbf{S} = (s_{ij})_{i,j}\ ,\ \mathbf{S} = \hat{\mathbf{N}}\hat{\mathbf{N}}^t = \mathbf{U}\hat{\mathbf{\Sigma}}^2\mathbf{U}^t\ . \tag{2}$$

Effectively, a document d_i is mapped to the s–dimensional representation $\vec{u}_i\hat{\mathbf{\Sigma}}$ and inner products are computed in this low-dimensional space.

LSI has been successfully used in a number of applications, although it has not consistently outperformed standard retrieval systems. As illustrated by many examples, LSI is often able to discover non-trivial relationships between words (and documents) such as synonyms. Yet, notional families are not explicitly represented in LSI and the non-probabilistic nature of the method raises some issues with respect to a principled foundation of the approach. Moreover, due to its linear nature, LSI is inherently unable to represent and model polysemy.

3 Language Models in Information Retrieval

In the language modeling approach to information retrieval, each document is modeled as an *information source*. Typically sources are assumed to be memoryless, in which case a document d_i can be represented by symbol emission probabilities $p(w_j|d_i)$, where w_j denotes a term in a vocabulary, for example, a word or phrase. Effectively, this associates a m-dimensional probability vector $\vec{p}_i = (p(w_1|d_i), p(w_2|d_i), \ldots, p(w_m|d_i))$ with each document. Due to the normalization constraint $\|\vec{p}_i\|_1 = 1$ each $\vec{p}_i$ can be thought of as a point on the $m-1$ dimensional probability simplex.

For a given query $q = q_1, \ldots, q_{l(q)}$ consisting of $l(q)$ query terms, one can thus easily compute the probability of generating q from a document d_i,

$$p(q|d_i) = \prod_{k=1}^{l(q)} p(q_k|d_i)\,. \tag{3}$$

Conceptually, (3) can be thought of as the probability of *generating* the query q from document d_i, i.e., it models how likely it is that a user for whom document d_i is relevant would ask for it using query q.

The above probabilistic model due to Ponte and Croft [6] has been further refined by Berger and Lafferty [1] to capture the effect that queries are typically distilled versions of documents. Query generation is hence thought of as a translation process which maps terms occurring in a document to corresponding terms of the query. One of the models proposed in [1] (called *Model 1*) simply introduces an additional Markov kernel $t(q_k|w_j)$ to model the translation. A simplified version of their model can be written as

$$\bar{p}(q|d_i) = \psi(l(q)) \prod_{k=1}^{l} \sum_{j=1}^{m} t(q_k|w_j)p(w_j|d_i)\,, \tag{4}$$

where $\psi(l(q))$ is a function that models the probability to generate a query of length $l(q)$.

To use language models for ranking documents in *ad hoc* retrieval, one can apply Bayes' rule to compute the most likely documents given the query,

$$p(d_i|q) = \frac{\bar{p}(q|d_i)p(d_i)}{\sum_{i'} \bar{p}(q|d_{i'})p(d_{i'})}\,, \tag{5}$$

where $p(d_i)$ allows to include a prior relevance probability for each document.

The language modeling approach to information retrieval is well-founded in statistics and information theory, yet the crucial question is how to estimate the required probabilities, i.e., the language model $p(w_j|d_i)$

and the Markov kernel $t(q_k|w_j)$, given the intrinsic sparseness problem. The focus for estimating document-specific language models has been on smoothed versions of the maximum likelihood estimator. While this seems to be a valid first steps, it does not address or exploit the fact that a certain *context* of words might increase the probability to find semantically related terms. As demonstrated in [1], the translation model is able to capture semantic relations between words based on term co-occurrences. We consider it to be a major weakness of this approach though that semantic relations between words are learned from (synthetically created) query/document pairs and are not directly based on co-occurrences within the document collection. Conceptually, the semantics should be captured by the language model and not by the translation model which merely deals with the distillation process of generating queries.

4 The Best of Both Worlds

How are the two retrieval models presented so far related? We have proposed a technique called Probabilistic Latent Semantic Indexing (PLSI) [4] which combines dimension reduction with language model estimation. The key idea is that a probabilistic dimension reduction technique can be utilized to overcome the sparseness problem and to simultaneously estimate document-specific language models by exploiting domain/collection-specific statistical regularities.

Formally, PLSI is based on the following parameterized statistical model

$$P(w_j|d_i) = \sum_{r=1}^{R} P(w_j|z_r)P(z_r|d_i)\,. \tag{6}$$

Here z_r refers to R possible states of a latent variable, each modeling a concept or notional family. A concept z_r is characterized by a distribution over terms $P(w_j|z_r)$, such that $\sum_{j=1}^{m} P(w_j|z_r) = 1$. Terms that are likely to occur in the context of a notional family z_r will have high probabilities $P(w_j|z_r)$, while unrelated terms will have a probability close to zero. Documents participate in concepts according to the probability $P(z_r|d_i)$. In order to estimate the multinomial probabilities in (6), one can use the Expectation Maximization algorithm along with a temperature control technique to avoid overfitting. Details can be found in [3]. This technique directly optimizes the average perplexity of the document-specific language models.

PLSI has a geometric interpretation that relates it to LSI. The probabilities $P(w_j|z_r)$ can be thought of as spanning a low-dimensional concept space, namely the $R-1$ dimensional convex hull of the points $\vec{z}_r = (P(w_1|z_r), \ldots, P(w_m|z_r))$. In this view, the probability vectors $\vec{d}_i = (P(z_1|d_i), \ldots, P(z_R|d_i))$ can be thought of as coordinates that define a unique point in the convex hull of $\{\vec{z}_1, \ldots, \vec{z}_R\}$. They correspond to a low-dimensional representation for documents in the concept space.

Let us first point out some key differences between LSI and PLSI. Besides the technical issues of a likelihood-based approach vs. a least squares method, what are the main conceptual differences? First of all, notice that PLSI learns a low-dimensional subspace in the space of all memoryless information sources, which allows to use it seamlessly in the context of the language modeling paradigm. Secondly, there is no notion of orthogonality in the Euclidean sense involved, the vectors $\vec{z}_r$ can not be simply rotated without changing the model. As a consequence, we have found that they often capture true concepts or notional families, in the precise meaning of defining a probability distribution over the set of terms. Thirdly, the probability vectors $\vec{d}_i$ are typically sparse in the sense that most entries are zero or close to zero. This is highly desirable, since it reflects the assumption that each individual document will only deal with a small subset of all possible concepts. Fourthly, PLSI is able to deal with the polysemy of words. For a potentially ambiguous word, one may compute the posterior probabilities $P(z_r|d_i, w_j) \propto P(w_j|z_r)P(z_r|d_i)$, i.e., the probability that a particular term occurrence w_j in document d_i is associated with concept z_r. For the same word, different contexts d_i correspond to different probabilities $P(z_r|d_i)$ and hence yield different posterior probabilities.

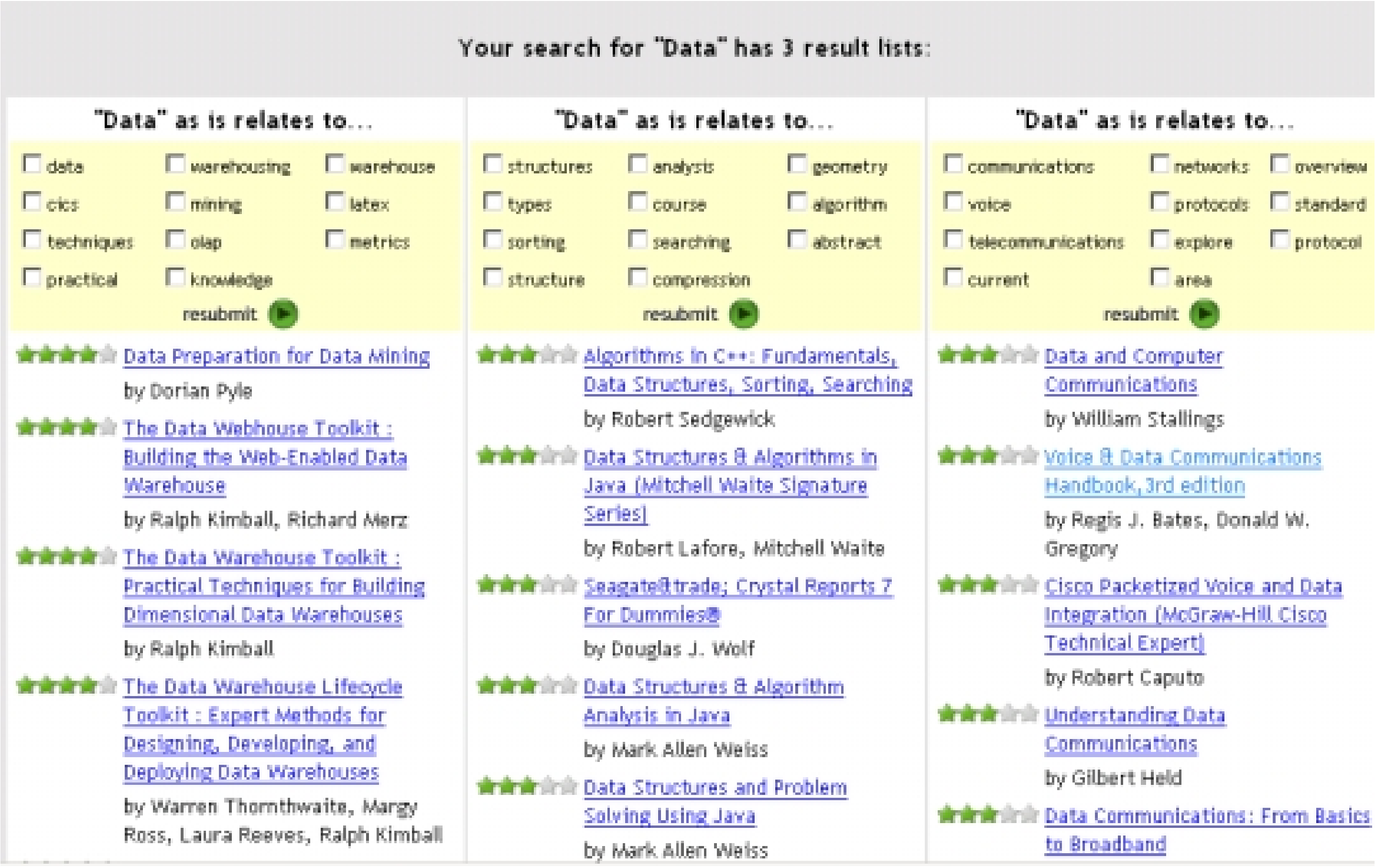

Figure 1: Retrieval results for a query "data" in a book database.

In summary, PLSI re-introduces the idea of automatically extracting concepts by dimension reduction, yet in a way that makes it compatible with and complementary to the language model information retrieval paradigm. It addresses the important issue of how to overcome data sparseness, a problem that plagues most probabilistic retrieval models.

Finally, we would like to show examples to supplement the quantitative evaluation in [4], of how the extracted notional families can be used to support information retrieval and increase the usability of retrieval systems. Figure 1 shows the result of a query "data" on a database of computer books. The query is inherently ambiguous, activating multiple concepts dealing with (i) data mining, warehousing, etc., (ii) data structures and algorithms, (iii) data exchange and communication. Because of the "concept awareness" of the PLSI-based system, this ambiguity can be made explicit, thereby helping users to make sense of the returned results by distributing them automatically over multiple result lists and helping users to refine their query, if necessary.

To give a better idea of the notional families that one might extract with PLSI, Figure 2 shows exemplary concepts extracted from a corpus of *Science Magazine* papers.

universe	0.0439	drug	0.0672	cells	0.0675	sequence	0.0818	years	0.156
galaxies	0.0375	patients	0.0493	stem	0.0478	sequences	0.0493	million	0.0556
clusters	0.0279	drugs	0.0444	human	0.0421	genome	0.033	ago	0.045
matter	0.0233	clinical	0.0346	cell	0.0309	dna	0.0257	time	0.0317
galaxy	0.0232	treatment	0.028	gene	0.025	sequencing	0.0172	age	0.0243
cluster	0.0214	trials	0.0277	tissue	0.0185	map	0.0123	year	0.024
cosmic	0.0137	therapy	0.0213	cloning	0.0169	genes	0.0122	record	0.0238
dark	0.0131	trial	0.0164	transfer	0.0155	chromosome	0.0119	early	0.0233
light	0.0109	disease	0.0157	blood	0.0113	regions	0.0119	billion	0.0177
density	0.01	medical	0.00997	embryos	0.0111	human	0.0111	history	0.0148

bacteria	0.0983	male	0.0558	theory	0.0811	immune	0.0909	stars	0.0524
bacterial	0.0561	females	0.0541	physics	0.0782	response	0.0375	star	0.0458
resistance	0.0431	female	0.0529	physicists	0.0146	system	0.0358	astrophys	0.0237
coli	0.0381	males	0.0477	einstein	0.0142	responses	0.0322	mass	0.021
strains	0.025	sex	0.0339	university	0.013	antigen	0.0263	disk	0.0173
microbiol	0.0214	reproductive	0.0172	gravity	0.013	antigens	0.0184	black	0.0161
microbial	0.0196	offspring	0.0168	black	0.0127	immunity	0.0176	gas	0.0149
strain	0.0165	sexual	0.0166	theories	0.01	immunology	0.0145	stellar	0.0127
salmonella	0.0163	reproduction	0.0143	aps	0.00987	antibody	0.014	astron	0.0125
resistant	0.0145	eggs	0.0138	matter	0.00954	autoimmune	0.0128	hole	0.00824

Figure 2: Notional families extracted from *Science Magazine* papers, numbers indicate probabilities $P(w_j|z_r)$.

References

[1] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 222–229, 1999.

[2] S. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):381–407, 1990.

[3] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI'99)*, 1999.

[4] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 50–57, 1999.

[5] H.P. Luhn. The automatic derivation of information retrieval encodement from machine readable text. *Information Retrieval and Machine Translation*, 3(2):1021–1028, 1961.

[6] J. Ponte and W.B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 275–281, 1998.