Form Latent Semantic Indexing to Language Models and Back

Thomas Hofmann
Department of Computer Science, Brown University, Providence, RI 02912
& RecMind 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 $N$ the term-document matrix, for example using a tf-idf-representation $n_{ij} = tf(w_j, d_i) \cdot idf(w_j)$, 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 $\tilde{N}$ according to

$$\tilde{N} = U \Sigma V' \approx U \Sigma V' = N. \quad (1)$$

Here, $U$ and $V$ denote matrices such that $U'U = V'V = I$, $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0)$ is a diagonal matrix which contains the singular values, $\sigma_i \geq \sigma_j$ for $i < j$, $r = \text{rank}(N)$, and $\Sigma = (\sigma_1, \ldots, \sigma_s, 0, \ldots, 0)$, $s < r$. The approximation $N$ 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
\[
s(d_i, d_j) = s_{ij}, \quad S = (s_{ij})_{i,j}, \quad S = \mathbf{N} \mathbf{N}^T = \mathbf{U}_s \mathbf{\Sigma}_s \mathbf{U}_s^T.
\] (2)

Effectively, a document \( d_i \) is mapped to the \( s \)-dimensional representation \( \mathbf{u}_s \mathbf{\Sigma}_s \) 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 \( \mathbf{p}_i = (p(w_1 | d_i), p(w_2 | d_i), \ldots, p(w_m | d_i)) \) with each document. Due to the normalization constraint \( ||\mathbf{p}_i||_1 = 1 \) each \( \mathbf{p}_i \) can be thought of as a point on the \( m-1 \) dimensional probability simplex.

For a given query \( q = q_1, \ldots, q_{|q|} \) consisting of \( l(q) \) query terms, one can thus easily compute the probability of generating \( q \) from a document \( d_i \),
\[
p(q | d_i) = \prod_{k=1}^{l(q)} p(q_k | d_i).
\] (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
\[
\tilde{p}(q | d_i) = \psi(l(q)) \prod_{k=1}^{l(q)} \sum_{j=1}^{m} t(q_k | w_j) p(w_j | d_i),
\] (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,
\[
p(d_i | q) = \frac{\tilde{p}(q | d_i) p(d_i)}{\sum_{\tilde{p}(q | d_i) p(d_i)}},
\] (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 step, 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

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

(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 $\sum_{r=1}^{R} P(w_j | z_r) = 1$. 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 $\mathbf{Z} = (P(w_1 | z_r), \ldots, P(w_m | z_r))$. In this view, the probability vectors $d_i = (P(z_1 | d_i), \ldots, P(z_R | d_i))$ can be thought of as coordinates that define a unique point in the convex hull of $\mathbf{Z}$. 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. Second, there is no notion of orthogonality in the Euclidean sense involved, the vectors $\mathbf{Z}$ 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 $d_i$ 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 polysemous 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.
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.
Figure 2: Notional families extracted from Science Magazine papers, numbers indicate probabilities $P(w_j | z_r)$.

References


