Jerome R. Bellegarda

A data-driven framework for modeling global relationships implicit in large volumes of data

Latent Semantic Mapping

Originally formulated in the context of information retrieval, latent semantic analysis (LSA) arose as an attempt to improve upon the common procedure of matching words in queries with words in documents [1]–[7]. The problem is that users typically retrieve documents on the basis of conceptual content, and individual words provide unreliable evidence about the conceptual topic or meaning of a document. There are usually many ways to express a given concept, so the literal terms in a user's query may not match those of a relevant document. This tends to decrease the "recall" performance of retrieval systems. In addition, most words have multiple meanings, so the literal terms in a user's query may match terms in documents that are not of interest to the user. This results in less precise performance.

The LSA paradigm operates under the assumption that there is some underlying latent semantic structure in the data that is partially obscured by the randomness of word choice with respect to retrieval. Algebraic and/or statistical techniques are used to estimate this latent structure and get rid of the obscuring "noise." The outcome is a parsimonious, continuous parameter description of terms and documents based on the underlying structure. This low-dimensionality description then replaces the original parameterization in indexing and retrieval [2], [3].

In practice, the "latent structure" is conveyed by correlation patterns, derived from the way individual words appear in documents; this entails an elementary "bag-of-words" model of the language. Moreover, in line with the data-driven nature of the approach, "semantic" merely implies that terms in a document may be taken as referents to the document itself or to its topic. These simplifications notwithstanding, there is abundant evidence that the resulting description is beneficial. It indeed takes advantage of implicit higher-order structure in the association of terms with documents, and thereby improves the detection of relevant

1053-5888/05/$20.00©2005IEEE

documents on the basis of terms found in queries [4], [5]. It is well documented, for instance, that LSA improves separability among different topics, as in the simple information-retrieval example given in "Improved Topic Separability Under LSA."

The success of LSA in information retrieval led to the application of the same paradigm in other areas of natural language processing, including word clustering, document/topic clustering, large-vocabulary speech recognition language modeling, automated call routing, and semantic inference for spoken interface control [8]–[16]. (To illustrate word clustering, for instance, Table 1 lists two typical LSA word clusters obtained using a training collection of about 20,000 news articles from the *Wall Street Journal* database [8].) These solutions all leverage LSA's ability to expose global relationships in the language in order to extract useful metadata concerning topic context and meaning; they thereby make progress toward more intelligent human-machine communication.

More specifically, three unique factors seem to make LSA particularly attractive: 1) the mapping of discrete entities (in this case, words and documents) onto a continuous parameter space, where efficient machine learning algorithms can be applied; 2) the dimensionality reduction inherent in the process, which makes complex natural language problems tractable; and 3) the intrinsically global outlook of the approach, which tends to complement the local optimization performed by more conventional techniques.

These are, of course, fairly generic properties, which are desirable in a variety of different contexts. Categorical problems are notoriously hard to model statistically, especially when they comprise an extremely large number of categories [17]. Dimensionality reduction helps uncover important structural aspects of a problem while filtering out "noise" [4]. And a global outlook can often reveal discriminative characteristics that are essentially impossible to recognize locally [10].

These observations have, in turn, sparked interest in several other potential uses of the basic paradigm [18]–[22]. In such applications, the task at hand may not be as directly language-related, but the approach has nonetheless proved very appealing. This motivates a change of terminology to the more generic "latent semantic mapping" (LSM). The new wording conveys increased reliance on the general properties listed earlier, as opposed to a narrower interpretation in terms of specific topic context and meaning. The purpose of this article is to review such properties and their applications under the broad perspective of unsupervised information extraction.

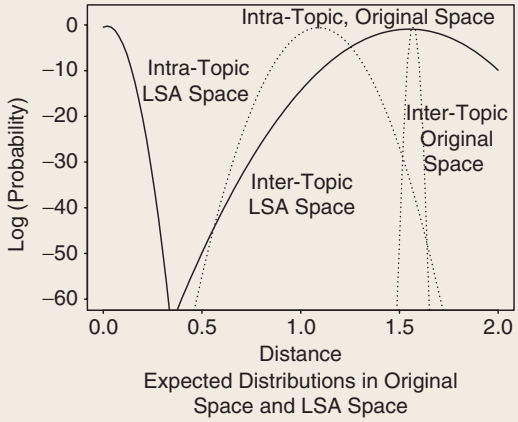
LSM

Let $\mathcal{M}$ be an inventory of M individual units, such as words in some underlying vocabulary, and $\mathcal{N}$ be a collection of N "meaningful" compositions of said units, such as documents in some relevant text corpus. While the order of M and N vary greatly with the application considered, it usually lies between 1,000 and 100,000; the collection $\mathcal{N}$ might comprise up to about 100 million units in total. The LSM paradigm defines a mapping between the discrete sets $\mathcal{M}$, $\mathcal{N}$ and a continuous vector space

IMPROVED TOPIC SEPARABILITY UNDER LSA (AFTER [6]).

Consider an (artificial) information retrieval task with 20 distinct topics and a vocabulary of 2,000 words. A probabilistic corpus model generates 1,000 documents, each 50 to 100 words long. The probability distribution for each topic is such that 0.95 of its probability density is equally distributed among topic words, and the remaining 0.05 is equally distributed among all 2,000 words. How does the average distance between documents evolve upon exploiting the LSA paradigm?

To answer this question, a suitable version (see the section "Closeness Measures") of the distance between all pairs of documents is calculated both before ("original space") and after ("LSA space") applying LSA. This leads to the expected distance distributions below, where a pair of documents is considered "Intra-Topic" if the two documents were generated from the same topic and "Inter-Topic" otherwise.



Expected Distributions in Original Space and LSA Space

Using LSA does not appreciably affect the average distance between inter-topic pairs, but dramatically reduces the average distance between intra-topic pairs. In addition, the standard deviation of the intra-topic distance distribution also becomes substantially smaller. As a result, separability between intra- and inter-topic pairs is much better under LSA, despite a sharp increase in the standard deviation of the inter-topic distance distribution.

[TABLE 1] TWO TYPICAL LSA WORD CLUSTERS (SEE [8]).

CLUSTER 1: Andy, antique, antiques, art, artist, artist's, artists, artworks, auctioneers, Christie's, collector, drawings, gallery, Gogh, fetched, hysteria, masterpiece, museums, painter, painting, paintings, Picasso, Pollock, reproduction, Sotheby's, van, Vincent, Warhol

CLUSTER 2: appeal, appeals, attorney, attorney's, counts, court, court's, courts, condemned, convictions, criminal, decision, defend, defendant, dismisses, dismissed, hearing, here, indicted, indictment, indictments, judge, judicial, judiciary, jury, juries, lawsuit, leniency, overturned, plaintiffs, prosecute, prosecution, prosecutions, prosecutors, ruled, ruling, sentenced, sentencing, suing, suit, suits, witness

$\mathcal{L}$, whereby each unit r_i in $\mathcal{M}$ is represented by a vector $\bar{u}_i$ in $\mathcal{L}$ and each composition c_j in $\mathcal{N}$ is represented by a vector $\bar{v}_j$ in $\mathcal{L}$.

FEATURE EXTRACTION

The starting point is the construction of a matrix (W) of co-occurrences between units and compositions. In line with the semantic nature of the original formulation [5], this construction normally disregards colocational information in neighboring units; the context for each unit essentially becomes the entire composition in which it appears. Thus, the matrix W is accumulated from the available training data by simply keeping track of which unit is found in which composition.

This accumulation involves some suitable function of the number of times each unit appears in each composition. Although many such functions are possible [2], it is often desirable to normalize for composition length and unit entropy [15], in which case the (i, j) cell of W is obtained as

$$w_{i,j} = (1 - \varepsilon_i) \frac{\kappa_{i,j}}{\lambda_j}, \tag{1}$$

where $\kappa_{i,j}$ is the number of times r_i occurs in c_j, λ_j is the total number of units present in c_j, and ε_i is the normalized entropy of r_i in the collection $\mathcal{N}$. The global weighting implied by $1 - \varepsilon_i$ reflects the fact that two units appearing with the same count in c_j do not necessarily convey the same amount of information about the composition; this is subordinated to the distribution of the units in the collection $\mathcal{N}$.

If we denote by $\tau_i = \sum_j \kappa_{i,j}$ the total number of times r_i occurs in $\mathcal{N}$, the expression for ε_i is easily seen to be

$$\varepsilon_i = -\frac{1}{\log N} \sum_{j=1}^{N} \frac{\kappa_{i,j}}{\tau_i} \log \frac{\kappa_{i,j}}{\tau_i}. \tag{2}$$

By definition, $0 \le \varepsilon_i \le 1$, with equality if and only if $\kappa_{i,j} = \tau_i$ and $\kappa_{i,j} = \tau_i / N$, respectively. A value of ε_i close to one indicates a unit distributed across many compositions throughout the collection, while a value of ε_i close to zero means that the unit is present only in a few specific compositions. The global weight $1 - \varepsilon_i$ is therefore a measure of the indexing power of the unit r_i.

SINGULAR VALUE DECOMPOSITION

The $(M \times N)$ unit-composition matrix W resulting from the above feature extraction defines two vector representations for the units and compositions. Each unit r_i can be uniquely associated with a row vector of dimension N, and each composition c_j can be uniquely associated with a column vector of dimension M. Unfortunately, these vector representations are impractical for three related reasons: 1) the dimensions M and N can be extremely large; 2) the vectors r_i and c_j are typically very sparse; and 3) the two spaces are distinct from one other.

To address these issues, it is useful to employ singular value decomposition (SVD), a technique closely related to eigenvector decomposition and factor analysis [23]. We proceed to perform the (order-R) SVD of W as follows:

$$W \approx \hat{W} = U S V^T, \tag{3}$$

where U is the $(M \times R)$ left singular matrix with row vectors u_i $(1 \le i \le M)$, S is the $(R \times R)$ diagonal matrix of singular values $s_1 \ge s_2 \ge \cdots \ge s_R > 0$, V is the $(N \times R)$ right singular matrix with row vectors v_j $(1 \le j \le N)$, $R < \min(M, N)$ is the order of the decomposition, and ^{T} denotes matrix transposition. As is well known, both left and right singular matrices U and V are column orthonormal, i.e., $U^T U = V^T V = I_R$ (the identity matrix of order R). Thus, the column vectors of U and V each define an orthonormal basis for the vector space of dimension R spanned by the vectors u_i and v_j, referred to as the LSM space $\mathcal{L}$. Furthermore, it can be shown [23] that the matrix $\hat{W}$ is the best rank-R approximation to the original matrix W for any unitarily invariant norm. This entails, for any matrix A of rank R,

$$\min_{\{A:\, \mathrm{rank}(A)=R\}} \|W - A\| = \|W - \hat{W}\| = s_{R+1}, \tag{4}$$

where $\|\cdot\|$ refers to the L_2 norm and s_{R+1} is the smallest singular value retained in the order-$(R+1)$ SVD of W. Obviously, $s_{R+1} = 0$ if R is equal to the rank of W.

INTERPRETATION

The decomposition in (3) can be illustrated as in "Singular Value Decomposition." It amounts to representing each unit and each composition as a linear combination of (hidden) abstract concepts, incarnated by the singular vectors. This in turn defines a mapping $(\mathcal{M}, \mathcal{N}) \longrightarrow \mathcal{L}$ between the high-dimensional discrete entities $\mathcal{M}$, $\mathcal{N}$ and the lower-dimensional continuous vector space $\mathcal{L}$. Since the SVD (by definition) provides a parsimonious description of the linear space spanned by W, the singular vectors are specified to *minimally* span the units in the inventory $\mathcal{M}$ and the compositions in the collection $\mathcal{N}$. Thus, the dual one-to-one mapping between units/compositions and unit/composition vectors resulting from (3) corresponds to an efficient representation of the training data. The dimension R is bounded from above by the (unknown) rank of the matrix W, and from below by the amount of distortion tolerable in the decomposition. The most common values of R range from 100 to 1,000.

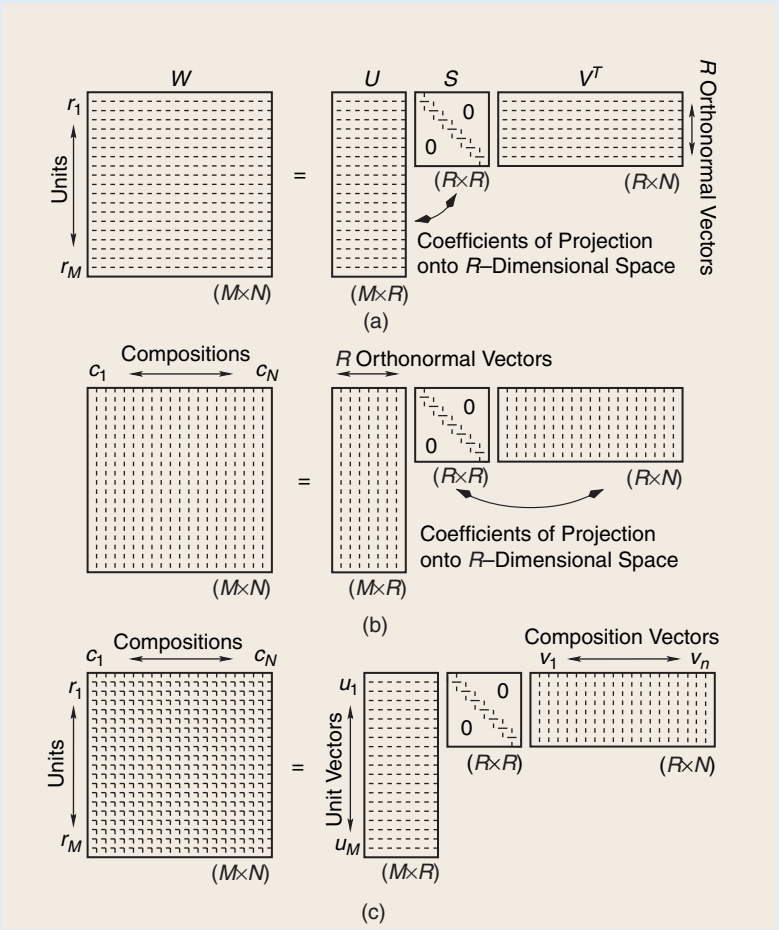
The basic idea behind this mapping is that $\hat{W}$ captures the major structural associations in W and ignores higher-order effects. The "closeness" of vectors in $\mathcal{L}$ is therefore determined by the overall pattern of the composition language used in $\mathcal{N}$, as opposed to specific constructs. Hence, two units whose representations are "close" (in some suitable metric) tend to appear in the same kind of compositions, whether or not they actually occur within identical unit contexts in those compositions. Conversely, two compositions whose representations are

SINGULAR VALUE DECOMPOSITION

(a) Row vectors (units) are projected onto the orthonormal basis formed by the column vectors of the right singular matrix V, i.e., the row vectors of V^T. This defines a new representation for the units, in terms of their coordinates in this projection, namely the rows of US. Thus, the row vector $u_i S$ characterizes the position of unit r_i in $\mathcal{L}$, for $1 \le i \le M$.

(b) Similarly, column vectors (compositions) are projected onto the orthonormal basis formed by the column vectors of the left singular matrix U. The coordinates of the compositions in this space are therefore given by the columns of SV^T. Thus, the column vector Sv_j^T, i.e., the row vector $v_j S$, characterizes the position of composition c_j in $\mathcal{L}$, for $1 \le j \le N$.

(c) Hence, each unit $r_i \in \mathcal{M}$ is uniquely mapped onto the *unit vector* $\bar{u}_i = u_i S$, and each composition $c_j \in \mathcal{N}$ is uniquely mapped onto the *composition vector* $\bar{v}_j = v_j S$.



"close" tend to convey the same meaning, whether or not they contain the same unit constructs. As a result, units and compositions whose representations are "close" in the LSM space $\mathcal{L}$ tend to be meaningfully related, whether or not such units actually occur in such compositions.

LSM FEATURE SPACE

The mapping $(\mathcal{M}, \mathcal{N}) \longrightarrow \mathcal{L}$ automatically resulting from (3) opens up the opportunity to apply familiar machine learning techniques in the continuous vector space $\mathcal{L}$. But to do that, it is necessary to first define within $\mathcal{L}$ some suitable metrics consistent with the SVD formalism.

CLOSENESS MEASURES

Since the matrix W embodies (by construction) all structural associations between units and compositions, it follows that for a given training collection, WW^T characterizes all cooccurrences between units and W^TW characterizes all cooccurrences between compositions. Thus, the extent to which units r_i and r_j have a similar pattern of occurrence across the entire collection of compositions can be inferred from the (i, j) cell of WW^T; the extent to which compositions c_i and c_j contain a similar pattern of units from the entire inventory can be inferred from the (i, j) cell of W^TW; and the extent to which unit r_i globally relates to composition c_j across the entire training collection can be inferred from the (i, j) cell of W itself. This leads to the following comparisons involving units and compositions.

UNIT-UNIT COMPARISONS

Expanding WW^T using the SVD expression (3), we obtain

$$WW^T = US^2U^T. \tag{5}$$

(Please note that henceforth we ignore the distinction between W and $\hat{W}$. From (4), this is without loss of generality as long as R approximates the rank of W.) Since S is diagonal, this means that the (i, j) cell of WW^T can be obtained by taking the dot product between the ith and jth rows of the matrix US, namely u_iS and u_jS. A natural metric to consider for the "closeness" between units is therefore the cosine of the angle between $\bar{u}_i$ and $\bar{u}_j$:

$$K(r_i, r_j) = \cos(u_iS, u_jS) = \frac{u_i S^2 u_j^T}{\|u_iS\| \, \|u_jS\|}, \tag{6}$$

for any $1 \le i, j \le M$. A value of $K(r_i, r_j) = 1$ means the two units always occur in the same kind of compositions, while a value of $K(r_i, r_j) < 1$ means the two units are used in increasingly different contexts. While (6) does not define a bona fide distance measure in the space $\mathcal{L}$, it easily leads to one. For example, over the interval $[\pi, 2\pi]$, the measure

$$\mathcal{D}(r_i, r_j) = \cos^{-1} K(r_i, r_j) \tag{7}$$

satisfies the properties of a distance on $\mathcal{L}$ [15].

COMPOSITION-COMPOSITION COMPARISONS

Proceeding as above, expanding $W^T W$ using (3) yields

$$W^T W = V S^2 V^T, \tag{8}$$

which means that the (i, j) cell of $W^T W$ can be obtained by taking the dot product between $v_i S$ and $v_j S$. As a result, a natural metric for the "closeness" between compositions is

$$K(c_i, c_j) = \cos(v_i S, v_j S) = \frac{v_i S^2 v_j^T}{\|v_i S\| \, \|v_j S\|}, \tag{9}$$

for any $1 \le i, j \le N$. Here, a value of $K(c_i, c_j) = 1$ means the two compositions contain the same kind of units, while a value of $K(c_i, c_j) < 1$ means they increasingly do not. The metric (9) has the same functional form as (6), and therefore the distance (7) is equally valid for both unit and composition comparisons. (In fact, the measure (7) is precisely the one used in "Improved Topic Separability Under LSA." Thus, the distances on the x-axis are $\mathcal{D}(c_i, c_j)$ expressed in radians.)

UNIT-COMPOSITION COMPARISONS

Since $W = USV^T$, the (i, j) cell of W can be obtained by taking the dot product between $u_i S^{1/2}$ and $v_j S^{1/2}$. Thus, a natural metric for the "closeness" between unit r_i and composition c_j is

$$K(r_i, c_j) = \cos(u_i S^{1/2}, v_j S^{1/2}) = \frac{u_i S v_j^T}{\|u_i S^{1/2}\| \, \|v_j S^{1/2}\|}, \tag{10}$$

for any $1 \le i \le M$ and $1 \le j \le N$. A value of $K(r_i, c_j) = 1$ means that there is a strong relationship between r_i and c_j, while a value of $K(r_i, c_j) < 1$ means that there is increasingly less evidence that they are meaningfully linked across the entire training collection. Interestingly, (10) is functionally equivalent to (6) and (9), but involves scaling by $S^{1/2}$ instead of S. As before, the transformation (7) can be used to turn (10) into an actual distance measure.

LSM FRAMEWORK EXTENSION

Clearly, the training collection $\mathcal{N}$ will not comprise all conceivable events that can be produced in the composition language. To find adequate representations in the space $\mathcal{L}$ for unobserved units and compositions, it is necessary to devise an appropriate extension to the LSM framework.

Suppose we observe a new composition $\tilde{c}_p$, with $p > N$, where the tilde symbol reflects the fact that the composition was not part of the original $\mathcal{N}$. First, we construct a feature vector containing, for each unit in the underlying inventory $\mathcal{M}$, the weighted counts (1) with $j = p$. This feature vector $\tilde{c}_p$, a column vector of dimension M, can be thought of as an additional column of the matrix W. Thus, provided the matrices U and S do not change, (3) implies

$$\tilde{c}_p = U S \tilde{v}_p^T, \tag{11}$$

where the R-dimensional vector $\tilde{v}_p^T$ act as an additional column of the matrix V^T. This in turn leads to the definition

$$\tilde{\tilde{v}}_p = \tilde{v}_p S = \tilde{c}_p^T U. \tag{12}$$

The vector $\tilde{\tilde{v}}_p$, indeed seen to be functionally similar to a composition vector, corresponds to the representation of the new composition in the space $\mathcal{L}$.

To convey the fact that it was not part of the SVD extraction, the new composition $\tilde{c}_p$ is referred to as a pseudo-composition, and the new representation $\tilde{\tilde{v}}_p$ is referred to as a pseudo-composition vector. Figure 1 illustrates the resulting extension to the LSM framework, along with the following caveat. Recall that the singular vectors in the SVD expansion (3) are specified to minimally span $\mathcal{M}$ and $\mathcal{N}$. As a result, if the new composition contains language patterns that are inconsistent with those extracted from W, (3) will no longer apply. Similarly, if the addition of $\tilde{c}_p$ causes the major structural associations in W to shift in some substantial manner, the singular vectors will become inadequate. Then U and S will no longer be valid, in which case it would be necessary to recompute (3) to find a proper representation for $\tilde{c}_p$. If, on the other hand, the new composition generally conforms to the rest of the collection $\mathcal{N}$, then $\tilde{\tilde{v}}_p$ in (12) will be a reasonable representation for $\tilde{c}_p$. (Note that any newly observed unit can similarly be treated as an additional row of the matrix W, giving rise to an analogous pseudo-unit vector.)

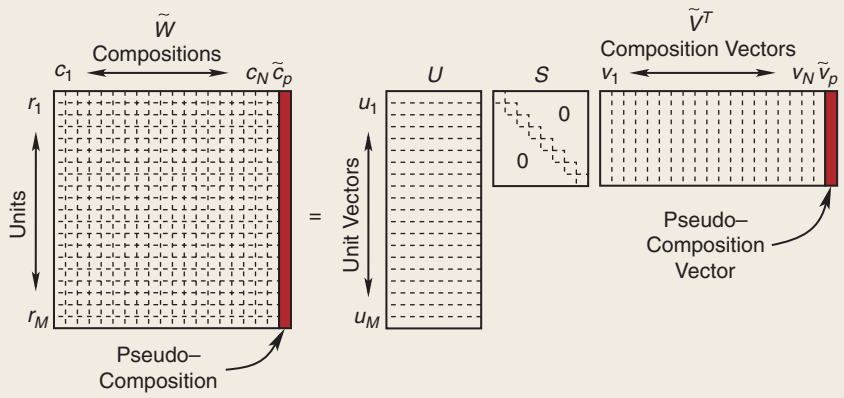
SALIENT CHARACTERISTICS

Overall, the LSM feature space has a number of interesting and fairly generic characteristics. To summarize, the mapping entails: 1) a single vector embedding for both units and compositions in the same continuous vector space $\mathcal{L}$; 2) a relatively low dimensionality, which makes operations such as clustering meaningful and practical; and 3) an underlying topological

structure reflecting globally meaningful relationships, with well-motivated, natural similarity metrics to measure the distance between units, between compositions, or between units and compositions in $\mathcal{L}$.

NATURAL LANGUAGE PROCESSING

In natural language applications like information retrieval, $\mathcal{M}$ is a vocabulary of words and $\mathcal{N}$ is a text corpus of documents; thus, units are words and compositions are documents. It is in this context that the term "semantic" in LSM comes closest to its ordinary meaning. Recall, however, that it still has a rather narrow interpretation in terms of words taken as referents to the general topic of the document. This caveat notwithstanding, the LSM properties can be exploited in several areas of spoken language processing.



Caveat: Suppose training is carried out for a banking application involving the word "bank" taken in a financial context. Now suppose $\tilde{c}_p$ germane to a fishing application, where "bank" refers to the edge of a river or a lake. Clearly, the closeness of "bank" to, e.g., "money" and "account," would become irrelevant. Conversely, adding $\tilde{c}_p$ to W would likely cause such structural associations to shift substantially, and perhaps even disappear altogether. In that case the re-computation of (3) would become necesary.

[FIG1] **LSM framework extension.**

SEMANTIC CLASSIFICATION

(Data-driven) semantic classification refers to the task of determining, for a given (newly observed) document, with which of several predefined topics the document is most closely aligned. Each topic is typically specified during training via a cluster of documents from the collection $\mathcal{N}$. Semantic classification is useful for many tasks, such as automated call routing [11], [24], desktop interface control [16], and unsupervised tutoring [25]. It is also an integral, if often implicit, part of spoken language understanding [26] and audio content indexing [27].

Suppose that a document cluster can be uniquely associated with a particular outcome in the task. Then the centroid of each cluster can be viewed as the semantic representation of this outcome in the LSM space. Phrased another way, each centroid becomes a semantic anchor for the corresponding outcome. A particularity of these semantic anchors is that they are automatically derived from the evidence presented during training, without regard for the particular syntax used to express the semantic link between various word sequences and the corresponding outcome. This opens up the possibility of mapping a newly observed word sequence onto an outcome by computing the distance (7) between that document and each semantic anchor, and then choosing the minimum. In this scenario, domain knowledge is automatically encapsulated in the LSM space in a data-driven fashion.

In the context of desktop interface control, this approach is referred to as semantic inference [16]. It relaxes some of the interaction constraints typically attached to the domain and thus allows the end user more flexibility in expressing the desired command/query. This, in turn, tends to reduce the associated cognitive load, thereby enhancing user satisfaction. A simple example of semantic inference is illustrated in "Semantic Inference in Latent Semantic Space $\mathcal{L}$ (R=2)."

MULTISPAN LANGUAGE MODELING

The LSM framework is also useful in statistical language modeling, where it can advantageously be combined with the standard n-gram approach [15]. The purpose of a language model is to compute the probability of the current word, say r_q, based on the admissible history (context) available, H_{q-1} [17]. In a standard n-gram, the history is the string $H_{q-1}^{(n)} = r_{q-1}r_{q-2}\ldots r_{q-n+1}$. In LSM language modeling, the history is the current document up to word r_{q-1}, $H_{q-1}^{(l)} = \tilde{c}_{q-1}$, where the (pseudo-)document $\tilde{c}_{q-1}$ is continually updated as q increases. Assuming the document being processed is semantically homogeneous, eventually we can expect the associated pseudo-document vector, which follows from (12), to settle down in the vicinity of the document cluster corresponding to the closest semantic content.

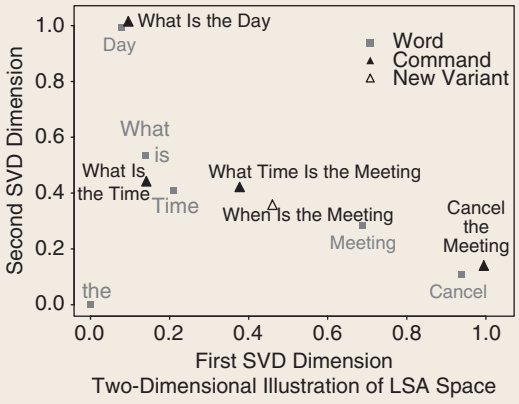
Under relatively mild conditions, it is possible to derive an integrated n-gram + LSM formulation for the overall language model probability. Starting with the definition

$$\Pr(r_q|H_{q-1}) = \Pr\left(r_q|H_{q-1}^{(n)}, H_{q-1}^{(l)}\right), \tag{13}$$

we make the assumption that the probability of the document history given the current word is not affected by the immediate context preceding it. This reflects the fact that, for a given word, different syntactic constructs (immediate context) can be used to carry the same meaning (document history). This is obviously reasonable for content words. The degree to which it matters for function words is less clear, but we conjecture that if the document history is long enough, the semantic anchoring is sufficiently strong for the assumption to hold.

SEMANTIC INFERENCE IN LATENT SEMANTIC SPACE $\mathcal{L}$ ($R = 2$).

Consider a task with $N = 4$ outcomes, each represented by a unique command: 1) "what is the time," 2) "what is the day," 3) "what time is the meeting," and 4) "cancel the meeting." There are only $M = 7$ words in the vocabulary, "what" and "is" always co-occur, and "the" appears in all four commands. Additionally, only 1) and 4) contain a unique word, and 1) is a proper subset of 3). Constructing the (7×4) word-command matrix and performing the SVD, we obtain the following latent semantic space $\mathcal{L}$.



Two-Dimensional Illustration of LSA Space

The two words that each uniquely identify a command—"day" for 2) and "cancel" for 4)—each have a high coordinate on a different axis. Conversely, the word "the," which conveys no information about the identity of a command, is located at the origin. The semantic anchors for 2) and 4) fall "close" to the words which predict them best,"day" and "cancel," respectively. Similarly, the semantic anchors for 1) and 3) fall in the vicinity of their meaningful components: "what is" and "time" for 1) and "time" and "meeting" for 3), with the word "time," which occurs in both, indeed appearing "close" to both.

Consider now a variant of command 3) unobserved during training, such as "when is the meeting," which maps onto the hollow triangle in $\mathcal{L}$. As expected, this point is closest to the anchor of command 3); the variant is deemed semantically most related to 3), and the correct outcome is selected accordingly. In essence, the system has performed elementary, "bottom-up" natural language understanding by inferring that "when" is a synonym for "what time."

With this assumption, (13) can be written as (see [15]):

$$\Pr(r_q|H_{q-1}) = \frac{\Pr(r_q|r_{q-1}r_{q-2}\ldots r_{q-n+1})\ \Pr(\tilde{c}_{q-1}|r_q)}{\sum_{r_i \in \mathcal{M}} \Pr(r_i|r_{q-1}r_{q-2}\ldots r_{q-n+1})\ \Pr(\tilde{c}_{q-1}|r_i)}. \tag{14}$$

If $\Pr(\tilde{c}_{q-1}|r_q)$ is viewed as a prior probability of the current document history, then (14) simply translates the classical Bayesian estimation of the n-gram (local) probability using a prior distribution obtained from (global) LSM. The end result, in effect, is a modified n-gram language model incorporating large-span semantic information; this approach is referred to as multispan language modeling.

The expression $\Pr(\tilde{c}_{q-1}|r_q)$ can be easily derived from the "closeness" between the associated word vector and pseudo-document vector in $\mathcal{L}$. Yet to express the outcome as a probability, it is necessary to go from the distance measure (7) to an actual probability measure. We opt to use the empirical multivariate distribution constructed by allocating the total probability mass in proportion to the distances observed during training. In essence, this reduces the complexity to a simple histogram normalization, at the expense of introducing a potential "quantization-like" error. Of course, such error can be minimized through a variety of histogram-smoothing techniques. Also note that the dynamic range of the distribution typically needs to be controlled by a parameter that is optimized empirically, e.g., by an exponent on the distance term, as carried out in [12]. The reader is referred to [15] for a more thorough discussion.

Intuitively, the expression $\Pr(\tilde{c}_{q-1}|r_q)$ reflects the "relevance" of word r_q to the admissible history, as observed through $\tilde{c}_{q-1}$. As such, it will be highest for words whose meaning aligns most closely with the semantic fabric of $\tilde{c}_{q-1}$ (i.e., relevant "content" words) and lowest for words that do not convey any particular information about this fabric (e.g., "function" words like "the"). This behavior is exactly the opposite of that observed with the conventional n-gram formalism, which tends to assign higher probabilities to (frequent) function words than to (rarer) content words. Hence, the attractive synergy between the two paradigms: use of the integrated trigram + LSM language model version of (14) resulted in a reduction in word error rate of up to 16% relative to a standard trigram when evaluated on a subset of the *Wall Street Journal* database [15].

ADDITIONAL APPLICATIONS

The fairly generic nature of the LSM properties makes the framework applicable to a variety of tasks beyond those just discussed. This section explores three recent forays into increasingly general applications.

JUNK E-MAIL FILTERING

The goal of junk e-mail filtering is to sort out and discard unsolicited commercial e-mail, commonly known as spam [19], [28], [29]. More specifically, LSM-based filtering aims to curtail the number of spam messages seen by a user by assessing whether or not the latent subject matter is consistent with the user's interests [19]. It can be viewed as a degenerate case of semantic classification where there are only two semantic categories: legitimate and junk. The specificity of e-mail, however, sets it apart from applications like call routing and semantic inference.

In junk e-mail filtering, $\mathcal{M}$ is an inventory of words, symbols, and combination thereof, and $\mathcal{N}$ is a (binary) collection of legitimate and junk e-mail messages. Two semantic anchors are established in the space $\mathcal{L}$, one for each category. During classification, each incoming e-mail message is mapped onto $\mathcal{L}$ using (12), and the resulting message vector is classified against the two anchors. If it is deemed "closer" [in the sense of (9)] to the anchor representing legitimate e-mail, it is permitted through. Otherwise, it is tagged as junk, and optionally discarded, if confidence is high enough. The procedure is completely automatic and requires no input from the user. In case the user chooses to correct misclassifications, however, the framework can also be adapted to learn new junk topics and the semantic anchors can be updated accordingly for more robust performance.

LSM-based filtering was investigated in detail in [19] and was found to be competitive with state-of-the-art spam filtering methods based on machine learning [28], [29]. The parsimonious nature of LSM, in particular, seems to contribute to a more robust generalization from the seen to the unseen. The resulting system has been part of the e-mail client bundled with MacOS X since August 2002.

PRONUNCIATION MODELING

Pronunciation modeling is the process of assigning phonemic/phonetic transcriptions to graphemic word forms [18], [30], [31]. Also called grapheme-to-phoneme conversion (GPC), it is of critical importance to all spoken language applications. For most languages, especially English, GPC is a challenging task, because local correspondences between graphemes and phonemes are difficult to encapsulate within a manageable number of general principles. Hence, the interest of a global outlook such as that provided by LSM.

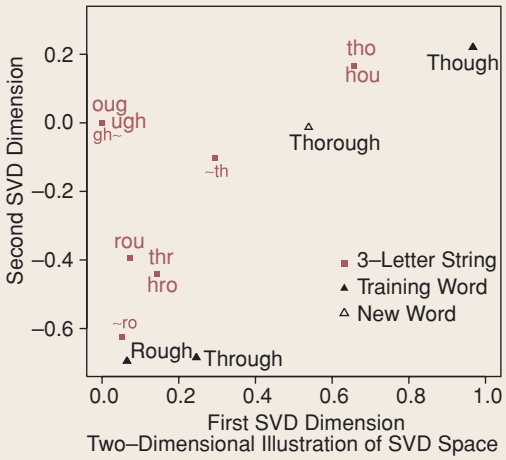
In pronunciation by latent analogy [18], $\mathcal{M}$ is an inventory of letter n-tuples and $\mathcal{N}$ is a collection of words from a given vocabulary. The LSM framework is used to determine which letter n-tuples are globally most characteristic of words and map all in-vocabulary words onto the space of all characteristic letter n-tuples. The outcome is a set of orthographic anchors (one for each in-vocabulary word), determined automatically from the underlying vocabulary. Each out-of-vocabulary (OOV) word for which a pronunciation is sought is then compared to each orthographic anchor, and the corresponding "closeness" is evaluated in the resulting LSM space $\mathcal{L}$. If this closeness is high enough, the associated in-vocabulary word is added to the so-called orthographic neighborhood of the OOV word. A simple example is illustrated in "Orthographic Neighbors in Orthographic Space $\mathcal{L}$ (R=2)."

Orthographic neighbors thus obtained can be viewed as the equivalent of similar words in classical pronunciation by analogy [30], except that LSM now defines the concept of similarity in a global rather than a local sense. This means, among other things, that specific matched substrings are no longer available as a byproduct of the lexical match. Once an orthographic neighborhood is available for a given OOV word, however, it is straightforward to gather the corresponding set of pronunciations from the existing dictionary. By construction, phonetic expansions in this pronunciation neighborhood have the property to contain at least one substring that is "locally close" to the pronunciation sought. Phoneme transcription thus follows via locally optimal sequence alignment and maximum likelihood position scoring, in which the influence of the entire neighborhood is implicitly and automatically taken into account [21].

The overall procedure is entirely data-driven and requires no human supervision. Compared to classical pronunciation by analogy [30], [31], it essentially decouples the two underlying tasks of neighborhood generation and pronunciation assembly.

ORTHOGRAPHIC NEIGHBORS IN ORTHOGRAPHIC SPACE $\mathcal{L}$ ($R = 2$).

Consider the collection of the N = 3 words; i) "*rough*," ii) "*though*," and iii) "*through*" for which local graphemic evidence is particularly uniformative. Counting those formed with the word boundary marker ~, there are M = 10 strings of n = 3 letters in the unit inventory. Some, for example "*tho*" and "*hou*," always co-occur; "*ough*," and "*gh~*" appear in all three words. This leads to the orthographic space $\mathcal{L}$ below.



Two–Dimensional Illustration of SVD Space

Note the parallels with the semantic inference example of "Semantic Inference in Latent Semantic Space," despite the very different choices of $\mathcal{M}$ and $\mathcal{N}$.

While "*ough/ugh/gh~*" is located at the origin, as expected, letter strings unique to a word score relatively high on the main axes. The orthographic anchors for the three words fall "close" to the letter strings which predict them best. And, not surprisingly, letter strings which are present in two different words indeed appear in the vicinity of both.

Consider now the new word "*thorough*" unobserved in training but consistent with the global graphemic structure. It maps onto the hollow triangle in $\mathcal{L}$. Since this point is closer to (ii) than (iii), the new word is deemed most related to "*through*," despite a closer Levenshtein relationship with "*through*." It thereby inherits the correct pronounciation for the substring "*ough*."

Now, neighborhood generation involves gathering globally pertinent information on the grapheme side, while pronunciation assembly involves exploiting locally consistent constraints on the phoneme side. This method was observed to be particularly effective on a difficult test collection of proper names with a large diversity of origin [18], [21].

TTS UNIT SELECTION

In modern concatenative TTS synthesis, unit selection identifies which speech segment from a large prerecorded database is the most appropriate at any given point in the speech waveform [20], [32], [33]. As these segments are extracted from disjoint phonetic contexts, discontinuities in spectral shape as well as phase mismatches tend to occur at segment boundaries. Because such artifacts have a deleterious effect on perception, it is important to ensure that they are minimized. This, in turn, requires a high-fidelity discontinuity metric for characterizing the acoustic (dis-)similarity between two segments [33].

We have recently investigated [20], [22] the feasibility of exploiting the LSM paradigm for that purpose. In that work, LSM is used as a mechanism for feature extraction. Because the resulting transform framework is better suited to preserve globally relevant properties in the region of concatenation, this approach proves beneficial when comparing concatenation candidates [20].

The implementation proceeds as follows. For a given boundary point, we gather all frames in the vicinity of this point for all instances from the database that straddle the boundary. This leads to a matrix W where each row corresponds to a particular (centered) pitch period near the given boundary. LSM then performs a modal (eigen-)analysis via a pitch synchronous SVD in each boundary region of interest. In this application, $\mathcal{M}$ is therefore an inventory of centered pitch periods from the boundary region and $\mathcal{N}$ is a collection of time slices, each comprising the relevant time samples from the pitch period inventory.

The resulting decomposition acts as a data-driven alternative to traditional Fourier analysis. To see this, denote the sinusoidal transform kernel by the symmetric complex matrix Φ such that $\Phi_{k\ell} = (1/\sqrt{N}) \exp(-j2\pi k\ell/N)$. In classical Fourier analysis, this kernel is applied to the ith row r_i of the matrix W to yield

$$X_i = r_i \Phi , \tag{15}$$

where X_i is the Fourier transform vector associated with r_i. In other words, the Fourier-derived features (X_i) are obtained via projection onto a set of signal-independent complex sinusoids. The inverse transform yields $r_i = X_i \Phi^H$, where ^{H} denotes Hermitian transposition and Φ is (column-)orthornormal just like U and V.

In contrast [from (3)], each row r_i of W can be expressed as $r_i = u_i S V^T = \bar{u}_i V^T$, which leads to the expression

$$\bar{u}_i = r_i V . \tag{16}$$

Clearly, the inner product of r_i with the kth right singular vector can be interpreted as a measure of the strength of the signal at the mode represented by this right singular vector. Thus, $\bar{u}_i$ can be viewed as the transform vector associated with r_i, given the (data-driven) transform kernel represented by V. Phrased another way, the new features ($\bar{u}_i$) are derived in terms of a separately optimized set of basic components for each boundary region of interest.

We readily acknowledge that the SVD kernel (V) is most likely inferior to the Fourier kernel as a general-purpose signal analysis tool, if only because it does not explicitly expose the concept of frequency. However, its use has several benefits of interest in the present application: 1) the singular vectors are, by construction, inherently tailored to the boundary region considered, in contrast with the traditional set of signal-independent complex sinusoids; 2) the kernel offers a global view of what is happening in this boundary region, as encapsulated in the vector space $\mathcal{L}$; 3) this representation is parsimonious, to the extent that an empirically consistent value is selected for the dimension R of the space; and 4) since this is a real-valued transform, both amplitude and phase information are retained and in fact contribute simultaneously to the outcome.

These properties lead to an efficient, optimized (in the L_2 sense), boundary-centric representation of the problem, as well as several discontinuity metrics in $\mathcal{L}$, expressed in terms of cumulative differences in closeness before and after concatenation. Experimental evaluation of these metrics has underscored a better correlation with perceived discontinuity, as compared with a widely used baseline measure [22]. This has confirmed the viability and effectiveness of the LSM framework for TTS unit selection.

INHERENT TRADEOFFS

As evidenced by the above, the LSM paradigm has wider applicability than initially contemplated in its original word-document incarnation. Generalizing the formulation to generic units and compositions enables the underlying mapping to potentially capture hidden patterns in all types of large collections. How effective LSM will be in other areas of signal processing, however, hinges on the various tradeoffs associated with the approach.

DESCRIPTIVE POWER

The descriptive power of LSM is subordinated to the particular constraints intrinsic to LSM modeling. One such limitation is the L_2 norm arising from (4). Since there is no reason to believe it is the best choice in all situations, the generic optimality of the LSM framework can be debated.

Consider the case of linguistic phenomena. Depending on many subtly intertwined factors like frequency and recency, linguistic cooccurrences may not always have the same interpretation. When comparing word counts, for example, observing 100 versus 99 occurrences is markedly different from observing one versus zero. In information retrieval, this has motivated the investigation of an alternative objective function based on the Kullback-Leibler divergence. This approach

has the advantage of providing an elegant probabilistic interpretation of (3), at the expense of requiring a conditional independence assumption on the words and the documents, given a hidden variable representing the topic of discourse [7]. In practice, unfortunately, this independence assumption is almost always violated. (Incidentally, while more sophisticated models have since been proposed (see, e.g., [34]), they tend to rely on somewhat arcane text generation scenarios that are similarly difficult to justify.)

Another fundamental restriction stems from the elementary "bag-of-units" modeling, which is by nature unable to capitalize on the local constraints present in the composition language. This is fine in applications like information retrieval, where they are discarded anyway. But in some situations, forgoing local constraints is simply not acceptable; in spoken interface control, for example, different commands would often map onto the same point in LSM space if word order were completely disregarded [16]. This forces somewhat awkward extensions such as word agglomeration, a solution not necessarily feasible in all environments [16].

Such optimality and modeling limitations can be viewed as instances of the familiar tradeoff between descriptive power and mathematical tractability.

DOMAIN SENSITIVITY

Like all data-driven techniques, LSM is inherently dependent on the quality of the training data, expressed here in terms of the particular unit-composition cooccurrences observed in the collection $\mathcal{N}$. Basically, pertinent patterns not present in $\mathcal{N}$ cannot be captured. This entails a relatively high sensitivity to the general domain sampled during training, and accordingly restricts cross-domain applicability.

In language modeling, for instance, the framework exhibits a fairly narrow semantic specificity, in the sense that the space $\mathcal{L}$ does not appear to reflect any of the pragmatic characteristics of the task considered. Hence, there exists a relatively high sensitivity to the style of composition [15]. Perhaps the LSM framework would benefit from explicitly including a "composition style" component. In [6], for example, it has been suggested to define an $(M \times M)$ stochastic matrix (a matrix with nonnegative entries and row sums equal to one) to account for the way style modifies the frequency of units. This solution, however, makes the (not always valid) assumption that this influence is independent of the underlying subject matter.

Further contributing to domain sensitivity is polysemy, the fact that some units have multiple meanings in the composition language. This phenomenon can be readily observed in the clustering instances presented in Table 1 and helps explain why some obvious words seem to be missing from the clusters. Take, for example, the singular noun "drawing" from cluster 1 and the present tense verb "rule" from cluster 2. As a result of polysemy, "drawing" and "rule" were more likely to appear in the training corpus with an alternative meaning (as in "drawing a conclusion" and "breaking a rule," respectively), thus leading to different cluster assignments.

UPDATING THE LSM SPACE

Also of interest is what happens as new data becomes available. Clearly, it would be desirable to update the LSM space to reflect any new information uncovered. The scale of the data in a specific application, however, obviously impacts the amount of computations involved. (For a detailed account of the computational complexity of LSM, see [15].) At the present time, recomputing a very large SVD on the fly is usually not practical. In such cases, there is an implicit assumption that actual usage will only involve new compositions that closely conform to the training collection. It thus becomes all the more critical to have a sufficient amount of training data up-front in order to achieve an acceptable level of performance.

This limitation has been extensively analyzed in the context of semantic inference [16], where it was found that, in particular, complete outcome coverage was required to avoid systematic errors. In practice, this requires recomputing the LSM space every time a new outcome is added. On this topic, we note that some adaptive techniques have been proposed, including a linear transform solution based on Cholesky factorizations [35]. Unfortunately, they appear to be limited in their applicability and in particular cannot compete with recomputation when the amount of new data is large.

CONCLUSION

This article has described LSM, a data-driven framework for modeling globally meaningful relationships implicit in large volumes of data. LSM generalizes a paradigm originally developed to capture hidden word patterns in a text document corpus. Over the past decade, this paradigm has proven effective in an increasing variety of fields, gradually spreading from query-based information retrieval to word clustering, document/topic clustering, large-vocabulary speech recognition language modeling, automated call routing, semantic inference for spoken interface control, and several other speech processing applications.

This success can be largely attributed to three unique characteristics: 1) the mapping of discrete entities (such as words and documents) onto a continuous parameter space, where efficient machine learning algorithms can be applied; 2) the dimensionality reduction inherent in the process, which makes complex speech and language problems tractable; and 3) the intrinsically global outlook of the approach, which tends to complement the local optimization performed by more conventional techniques.

Such properties are potentially attractive in more general instances of unsupervised information extraction. How effective LSM is to a particular application hinges on three caveats: 1) the L_2 optimality criterion and "bag-of-units" modeling characteristic of the framework, which may not be appropriate in certain situations; 2) the narrow semantic specificity of the cooccurrence approach, which is vulnerable to such phenomena as composition style and polysemy; and 3) the computational complexity of updating the LSM space as new data becomes available, despite ongoing efforts to derive efficient algorithms for this purpose.

Fortunately, these limitations can be somewhat mitigated through careful attention to the expected domain of use and the size of the training collection. In that sense, the advantages of the framework substantially outweigh its drawbacks. As a result, LSM is likely to have wide applicability in any situation involving discrete entities, complex parameterizations with high dimensionality, and conventional algorithms focused on local phenomena. It is hoped that the discussion of the various benefits and trade-offs of LSM presented in this article will stimulate application to such tasks.

AUTHOR

Jerome R. Bellegarda received the Diplôme d'Ingénieur degree from the Ecole Nationale Supérieure d'Electricité et de Mécanique, Nancy, France, in 1984, and the M.S. and Ph.D. degrees in electrical engineering from the University of Rochester, New York, in 1984 and 1987, respectively. From 1988–1994, he was with the IBM T.J. Watson Research Center, Yorktown Heights, New York. In 1994 he joined Apple Computer, Cupertino, California, where he is currently Apple Distinguished Scientist in Speech and Language Technologies. His general interests span the areas of voice-driven man-machine communications, multiple input/output modalities, and multimedia knowledge management. He has written over 120 publications and holds 25 patents. He has been a member of many international scientific committees, review panels, and editorial boards. He was associate editor for *IEEE Transactions on Speech and Audio Processing*, served on the IEEE Signal Processing Society Speech Technical Committee, and is currently a member of the *Speech Communication* editorial board. He is a Fellow of the IEEE.

REFERENCES

[1] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Am. Soc. Inform. Science*, vol. 41, no. 6, pp. 391–407, 1990.

[2] S.T. Dumais, "Latent semantic indexing (LSI) and TREC–2," in *Proc. Second Text Retrieval Conf. (TREC–2)*, 1994, pp. 105–116.

[3] M.W. Berry, S.T. Dumais, and G.W. O'Brien, "Using linear algebra for intelligent information retrieval," *SIAM Rev.*, vol. 37, no. 4, pp. 573–595, 1995.

[4] T.K. Landauer and S.T. Dumais, "Solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge," *Psychol. Rev.*, vol. 104, no. 2, pp. 211–240, 1997.

[5] T.K. Landauer, D. Laham, B. Rehder, and M.E. Schreiner, "How well can passage meaning be derived without using word order: A comparison of latent semantic analysis and humans," in *Proc. Cognit. Science Soc.*, 1997, pp. 412–417.

[6] C.H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, "Latent semantic indexing: A probabilistic analysis," in *Proc. 17th ACM Symp. Principles Database Systems.*, Seattle, WA, June 1998, pp. 159–168.

[7] T. Hofmann, "Probabilistic topic maps: Navigating through large text collections," *Lecture Notes Comp. Science* (no. 1642). Heidelberg, Germany: Springer–Verlag, pp. 161–172.

[8] J.R. Bellegarda, J.W. Butzberger, Y.L. Chow, N.B. Coccaro, and D. Naik, "A novel word clustering algorithm based on latent semantic analysis," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Atlanta, GA, May 1996, pp. I172–I175.

[9] Y. Gotoh and S. Renals, "Document space models using latent semantic analysis," in *Proc. 5th European Conf. Speech Communication Technology*, Rhodes, Greece, Sept. 1997, vol. 3, pp. 1443–1448.

[10] J.R. Bellegarda, "A multi-span language modeling framework for large vocabulary speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 6, no. 5, pp. 456–467, Sept. 1998.

[11] B. Carpenter and J. Chu–Carroll, "Natural language call routing: A robust, self–organized approach," in *Proc. Int. Conf. Spoken Language Proc.*, Sydney, Australia, Dec. 1998, pp. 2059–2062.

[12] N. Coccaro and D. Jurafsky, "Towards better integration of semantic predictors in statistical language modeling," in *Proc. Int. Conf. Spoken Language Processing*, Sydney, Australia, Dec. 1998, pp. 2403–2406.

[13] D. Gildea and T. Hofmann, "Topic-based language modeling using EM," in *Proc. 6th Euro. Conf. Speech Communication Technology*, Budapest, Hungary, Sept. 1999, vol. 5, pp. 2167–2170.

[14] J.R. Bellegarda, "Large vocabulary speech recognition with multi-span statistical language models," *IEEE Trans. Speech Audio Processing*, vol. 8, no. 1, pp. 76–84, Jan. 2000.

[15] J.R. Bellegarda, "Exploiting latent semantic information in statistical language modeling," *Proc. IEEE (Special Issue on Speech Recognition Understanding)*, vol. 88, no. 8, pp. 1279–1296, Aug. 2000.

[16] J.R. Bellegarda and K.E.A. Silverman, "Natural language spoken interface control using data–driven semantic inference," *IEEE Trans. Speech Audio Processing*, vol. SAP–11, no. 3, pp. 267–277, May 2003.

[17] R. Rosenfeld, "Two decades of statistical language modeling: Where do we go from here," *Proc. IEEE (Special Issue on Speech Recognition Understanding)*, vol. 88, no. 8, pp. 1270–1278, Aug. 2000.

[18] J.R. Bellegarda, "A latent analogy framework for grapheme-to-phoneme conversion," in *Proc. 8th European Conf. Speech Communication Technology*, Geneva, Switzerland, Sept. 2003, pp. 2029–2032.

[19] J.R. Bellegarda, D. Naik, and K.E.A. Silverman, "Automatic junk e–mail filtering based on latent content," in *Proc. 2003 IEEE Automatic Speech Recognition Understanding Workshop*, St. Thomas, U.S. Virgin Islands, Dec. 2003, pp. 465–470.

[20] J.R. Bellegarda, "A novel discontinuity metric for unit selection text–to–speech synthesis," in *Proc. 5th ISCA Speech Synthesis Workshop*, Pittsburgh, PA, June 2004, pp. 133–138.

[21] J.R. Bellegarda, "Unsupervised, language-independent grapheme–to–phoneme conversion by latent analogy," *Speech Commun.*, vol. 46, no. 2, pp. 140–152, June 2005.

[22] J.R. Bellegarda, "A global, boundary–centric framework for unit selection text–to–speech synthesis," *IEEE Trans. Speech Audio Processing*, to be published.

[23] J.K. Cullum and R.A. Willoughby, "Real Rectangular Matrices," *Lanczos Algorithms for Large Symmetric Eigenvalue Computations—Vol. 1 Theory*. Boston: Birkhauser, 1985, ch. 5.

[24] A.L. Gorin, G. Riccardi, and J.H. Wright, "How may I help you?," *Speech Commun.*, vol. 23, no. 1–2, pp. 113–127, 1997.

[25] A.C. Graesser, K. Wiemer–Hastings, P. Wiemer–Hastings, and R. Kreuz, "AutoTutor: A simulation of a human tutor," *J. Cognit. Syst. Research*, vol. 1, no. 1, pp. 35–51, 1999.

[26] Y.-Y. Wang, L. Deng, and A. Acero, "Spoken language understanding," *IEEE Signal Processing Mag.*, vol. 22, no. 5, pp. 16–31, 2005.

[27] L.-S. Lee and B. Chen, "Spoken document understanding and organization," *IEEE Signal Processing Mag.*, vol. 22, no. 5, pp. 42–60, 2005.

[28] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C.D. Spyropoulos, and P. Stamatopoulos, "Learning to filter spam e-mail: A comparison of a naive Bayesian and a memory-based approach," in *Proc. Workshop Machine Learning Textual Information Access*, Lyon, France, 2000, pp. 1–13.

[29] K.-M. Schneider, "A comparison of event models for naive Bayes anti-spam e-mail filtering," in *Proc. 11th Conf. European Chap. ACL*, Budapest, Hungary, 2003, pp. 307–314.

[30] F. Yvon, "Pradigmatic cascades: A linguistically sound model of pronunciation by analogy," in *Proc. 35th Ann. Meeting Association Computational Linguistics*, 1997, pp. 428–435.

[31] R.I. Damper, C.Z. Stanbridge, and Y. Marchand, "A pronunciation-by-analogy module for the festival text-to-speech synthesiser," in *Proc. 4th Int. Workshop Speech Synthesis*, Pitlochry, Scotland, Aug. 2001, pp. 97–102.

[32] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using large speech database," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Atlanta, GA, 1996, pp. 373–376.

[33] E. Klabbers and R. Veldhuis, "Reducing audible spectral discontinuities," *IEEE Trans. Speech Audio Processing (Special Issue on Speech Synthesis*), vol. 9, no. 1, pp. 39–51, Jan. 2001.

[34] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent dirichlet allocation," in *Advances Neural Info. Proc. Systems (NIPS) 14*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds., Cambridge, MA: The MIT Press, pp. 601–608.

[35] J.R. Bellegarda, "Fast update of latent semantic spaces using a linear transform framework," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Orlando, FL, May 2002, pp. 769–772. SP