A Multispan Language Modeling Framework for Large Vocabulary Speech Recognition

Jerome R. Bellegarda, *Senior Member, IEEE*

***Abstract*— A new framework is proposed to construct multispan language models for large vocabulary speech recognition, by exploiting both local and global constraints present in the language. While statistical n-gram modeling can readily take local constraints into account, global constraints have been more difficult to handle within a data-driven formalism. In this work, they are captured via a paradigm first formulated in the context of information retrieval, called *latent semantic analysis* (LSA). This paradigm seeks to automatically uncover the salient semantic relationships between words and documents in a given corpus. Such discovery relies on a parsimonious vector representation of each word and each document in a suitable, common vector space. Since in this space familiar clustering techniques can be applied, it becomes possible to derive several families of large-span language models, with various smoothing properties. Because of their semantic nature, the new language models are well suited to complement conventional, more syntactically oriented n-grams, and the combination of the two paradigms naturally yields the benefit of a multispan context. An integrative formulation is proposed for this purpose, in which the latent semantic information is used to adjust the standard n-gram probability. The performance of the resulting multispan language models, as measured by perplexity, compares favorably with the corresponding n-gram performance.**

***Index Terms*— Latent semantic analysis, n-gram adaptation, perplexity reduction, statistical language modeling.**

I. Introduction

STOCHASTIC language modeling plays a central role in large vocabulary speech recognition, where it is usually implemented using the n-gram paradigm. In a typical application, the purpose of an n-gram language model may be to constrain the acoustic analysis, guide the search through various (partial) text hypotheses, and/or contribute to the determination of the final transcription [1]. Success in these endeavors depends on the ability of the language model to suitably discriminate between different strings of n words. This ability is in turn critically influenced by the two familiar issues of coverage and estimation.

The coverage issue reflects the fact that current systems cannot recognize any "unknown" word. The vocabulary must therefore be chosen so that the expected text (e.g., to be dictated) has as few unknown words as possible [2]. In this paper, we primarily address the other issue, which centers around the choice of n. Due to practical constraints on the size of available text databases, an inherent trade-off arises between weak predictive power (low n) and unreliable estimation (higher n). This is because many events (i.e., occurrences of n-word strings) are seen infrequently, yielding questionable probabilities; hence the need for fairly sophisticated parameter estimation and smoothing, cf. [3]. One common solution is to group words into classes and accumulate statistics at the class level rather than the word level. This makes the frequency counts more reliable and thereby improves the robustness of the estimation (e.g., see [4]). Broadly speaking, the underlying strategy is to better estimate the conditional probability of a word given some context by taking advantage of observations of other words that behave "like" this word in this particular context.

A number of variants have been developed on this theme, using grammatical constraints such as part-of-speech, or morphological units such as lemma, or both [5]. More recently, algorithms have evolved to automatically determine word classes without explicit syntactic or semantic knowledge: cf., e.g., [6] and [7]. In [6], for example, all words are gathered into a single class at the beginning of the procedure, and are successively split to maximize the average mutual information of adjacent classes. In [7], a similar divisive clustering is proposed, based on binomial posteriori distributions on word co-occurrences. A number of other papers have described related approaches, with different variations in the optimization criterion or distance metric used for clustering [8]–[10].

Such techniques make it possible to estimate the necessary probabilities from relatively sparse text data bases. Still, it remains extremely challenging to go beyond, say $n \leq 4$, with currently available data bases and processing power [9]. This imposes an artificially local horizon to the language model and thereby limits its predictive power. Consider, for instance, predicting the word "fell" from the word "stocks" in the two equivalent phrases:

stocks fell sharply as a result of the announcement (1)

and

stocks, as a result of the announcement, sharply fell. (2)

In (1), the prediction can be done with the help of a bigram language model ($n = 2$), which is rather straightforward [11]. In (2), however, the value $n = 9$ would be necessary, a rather unrealistic proposition at the present time.

At the other end of the spectrum, it is possible to consider the entire sentence, as opposed to just the n preceding words. This requires a paradigm shift toward parsing and rule-based

Manuscript received July 29, 1996; revised November 17, 1997. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Douglas D. O'Shaughnessy.

The author is with the Spoken Language Group, Apple Computer, Inc., Cupertino, CA 95014 USA (e-mail: jerome@apple.com).

Publisher Item Identifier S 1063-6676(98)05937-9.

grammars, such as are routinely and successfully employed in small vocabulary recognition applications. This solution, unfortunately, is not (yet) practical for large vocabulary recognition [2], which is precisely the reason why the n-gram framework was so widely adopted in the first place. What seems to be needed is an intermediate approach, where the effective context is expanded from three or four words to a larger span, say an entire sentence or even a whole document, without resorting to a formal parsing mechanism. This in turn would allow for the extraction of suitable long distance information.

One approach recently proposed in that direction is based on the concept of word triggers [12]. In the above example, suppose that the training data reveals a significant correlation between "stocks" and "fell," so that the pair ("stocks, fell") forms a trigger pair. Then the presence of "stocks" in the document could automatically trigger "fell," causing its probability estimate to change. Because this behavior would occur indifferently in (1) and in (2), the two phrases would lead to the same result. Unfortunately, trigger pair selection is a complex issue: different pairs display markedly different behavior, which limits the potential of low frequency word triggers [13]. Still, self-triggers have been shown to be particularly powerful and robust [12], which underscores the desirability of exploiting correlations between the current word and features of the document history.

This paper proposes a different approach along the same lines, based on a paradigm originally formulated in the context of information retrieval, called *latent semantic analysis* (LSA) [14]–[18]. In some respect, this approach can in fact be viewed as an extension of the word trigger concept, where a more systematic framework is used to handle the trigger pair selection. The paper is organized as follows. In the next section we discuss our general strategy to expand the effective context without resorting to a formal parsing mechanism. In Section III, we present the vector representation derived from LSA. Section IV develops the general modeling framework, and reports on a preliminary, qualitative evaluation. In Section V, we use this framework to derive several families of large-span semantic language models and discuss their relative prediction power. Section VI addresses the integration of the new framework with conventional n-gram language models. Finally, in Section VII, a series of experimental results illustrates some of the benefits associated with the integrated language models, using both $n = 2$ and $n = 3$ as examples.

II. General Strategy

In a nutshell, we would like to expand the effective context while avoiding syntactic analysis. This goal constrains the approach sought to be semantically derived, which entails a departure from the largely structural n-gram paradigm. Even class n-grams, which often exhibit "semantic-like" classes, inherently rely on the position information in the sentence (cf., e.g., [6]). To further develop the example in (1) and (2), consider the slightly modified phrase:

bonds, as a result of the announcement, sharply increased. (3)

It is likely that "stocks" and "bonds" would belong to the same class of the class n-gram. Furthermore, it is intuitively appealing to postulate that the prediction of "increased" from "bonds" in this phrase is related to the prediction of "fell" from "stocks" in (2). Yet, as mentioned before, the only way to express this relationship in the n-gram paradigm would be to derive a class 9-gram, a challenging proposition. On the other hand, accounting for this kind of relationship might be substantially easier in a semantically derived approach to language modeling.

The operating principle is to relate to one another those words which are found to be semantically linked from the evidence presented in the training text database, without regard to the particular syntax used to express that semantic link. In the above case, for instance, let us assume that the training database is a collection of financial news articles. Then it will comprise many articles with the words "stocks," "bonds," "fell," "decreased," "rose," "increased," etc. As a result, these words will either co-occur frequently (although not necessarily within the same syntactic relationship), or appear in articles within similar semantic contexts, or both. The crux of the problem is to harness this evidence to derive the probability of seeing the word "fell" (respectively, "increased") given an occurrence of the word "stocks" (respectively, "bonds"), even when the two words do not appear near each other in the text.

Clearly, the trigger approach mentioned earlier does provide a solution to this problem for those trigger pairs that have been selected by the algorithm [13]. However, trigger pair selection entails a number of practical constraints. First, only word pairs that co-occur in a sufficient number of documents are considered. This means that even though "stocks" may often co-occur with "decreased," and "decreased" may often co-occur with "fell," the pair ("stocks, fell") will not be included unless it has itself been frequently seen in the training data. In addition, a mutual information criterion is typically used to further confine the list of candidate pairs to a manageable size. This may result in too much "filtering" of the data. What seems to be needed is a somewhat more flexible framework to exploit the long distance information present in the history.

This is where the latent semantic paradigm comes into play. In latent semantic indexing [14]–[18], co-occurrence analysis takes place across much larger spans than with a traditional n-gram approach (i.e., spans of two words as in [4] or three words as in [7]), and on a much larger scale than with the trigger approach (i.e., about 1.4 million trigger pairs as in [13]). The span of choice is a *document* that can be defined as a semantically homogeneous set of sentences embodying a given storyline. Thus, each article mentioned above would be considered a document. As for scale, every combination of words from the vocabulary is viewed as a potential trigger combination. This amounts to addressing the problem of trigger pair selection as part of the analysis, as opposed to a postprocessing step. These extensions (in span and scale) lead to the systematic integration of long-term dependencies into the analysis.

To take advantage of the concept of *document*, we of course have to assume that the available training data is tagged at the document level, i.e., there is a way to identify

article boundaries. This is the case, for example, with the ARPA *North American Business News* (*NAB News*) corpus [19]. This assumption enables the construction of a matrix of co-occurrences between words and documents. This matrix is accumulated from the available training data by simply keeping track of which word is found in what document. Said another way, the context for each word becomes the document in which it appears. Note that, in marked contrast with n-gram modeling, word order is ignored, which is of course in line with the semantic nature of the approach [20]. This means that the LSA paradigm not only does not exploit syntactic information, but effectively throws it away. Thus, it should not be expected to replace conventional n-grams, but rather to complement them.

After the word-document matrix of co-occurrences is constructed, the LSA approach proceeds by computing the singular value decomposition (SVD) of the word-document matrix. The left singular vectors in this SVD represent the words in the given vocabulary, and the right singular vectors represent the documents in the given corpus. The role of the SVD, intrinsically, is therefore to establish a one-to-one mapping between words/documents and some vectors in a space of appropriate dimension. Specifically, this space is spanned by the singular vectors resulting from the SVD.

An important property of this space is that two words whose representations are "close" (in some suitable metric) tend to appear in the same kind of documents, whether or not they actually occur within identical word contexts in those documents. Conversely, two documents whose representations are "close" tend to convey the same semantic meaning, whether or not they contain the same word constructs. Thus, we can expect that the respective representations of words and documents that are semantically linked would also be "close" in that space. This property is what makes the framework useful for language modeling purposes.

III. Latent Semantic Analysis

Let $\mathcal{V}$, $|\mathcal{V}| = M$, be some vocabulary of interest and $\mathcal{T}$ a training text corpus, i.e., a collection of N articles (documents) from a variety of sources. (Typically, M and N are on the order of 10 000 and 100 000, respectively; $\mathcal{T}$ might comprise 100 million words or so.) The task at hand is to define a mapping between the sets $\mathcal{V}$, $\mathcal{T}$, and a vector space $\mathcal{S}$, whereby each word in $\mathcal{V}$ and each document in $\mathcal{T}$ is represented by a vector in $\mathcal{S}$.

A. Feature Representation

We first construct a word-document matrix W associated with $\mathcal{V}$ and $\mathcal{T}$. This is done by computing, for each word $w_i \in \mathcal{V}$, the weighted count W_{ij} of w_i in each of the documents $d_j \in \mathcal{T}$. Following results from information retrieval (cf., e.g., [21]), this weighted count is expressed as

$$W_{ij} = G_i L_{ij} \tag{4}$$

where G_i is a global weight, indicating the overall importance of w_i as an indexing term for the collection $\mathcal{T}$, and L_{ij} is a local value, which may reflect a possible normalization within d_j.

The global weighting G_i translates the fact that two words appearing with the same count in d_j do not necessarily convey the same amount of information about the document; this is subordinated to the distribution of the words in the collection $\mathcal{T}$. Let us denote by c_{ij} the number of times w_i occurs in document d_j, and by t_i the total number of times w_i occurs in the entire collection $\mathcal{T}$. Then the relative frequency of w_i in d_j is obtained as

$$f_{ij} = \frac{c_{ij}}{t_i} \tag{5}$$

and the associated normalized entropy of w_i is seen to be

$$E_i = -\frac{1}{\log(N)} \sum_{j=1}^{N} f_{ij} \log f_{ij}. \tag{6}$$

By definition, $0 \leq E_i \leq 1$, with equality if and only if $f_{ij} = 1$ and $f_{ij} = 1/N$, respectively. A value of E_i close to one underscores a word distributed across many documents throughout the corpus, and therefore of little indexing value. Conversely, a value of E_i close to zero indicates a word present only in a few specific documents, i.e., of suitable indexing value. Hence

$$G_i = 1 - E_i \tag{7}$$

is a reasonable global weight for the word w_i.

The local value L_{ij} is a transformed version of c_{ij} which may reflect any adjustment to the raw count c_{ij}. For example, it is common to use $L_{ij} = \log(1+c_{ij})$, where the log dampens the effects of large differences in counts [21]. It is also possible to normalize for document length. If we denote by n_j the number of words in document d_j, then

$$L_{ij} = \log_2 \left(1 + \frac{c_{ij}}{n_j}\right) \tag{8}$$

is such that $0 \leq L_{ij} \leq 1$. This functional avoids implicitly favoring long documents in text corpora containing documents of greatly variable length.

B. Singular Value Decomposition

The $(M \times N)$ word-document matrix W with entries W_{ij} given by (4) fully describes, for the training corpus $\mathcal{T}$, which words appeared in what contexts. Clearly, this matrix defines two vector representations for the words and the documents. Each word w_i can be uniquely associated with a row vector of dimension N, and each document d_j can be uniquely associated with a column vector of dimension M. For the sake of simplicity, we will also refer to these row and column vectors as w_i and d_j, respectively. Unfortunately, these vector representations are impractical for three related reasons. First, the dimensions M and N can be extremely large; second, the vectors w_i and d_j are typically very sparse; and third, 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 [22]. We proceed to

perform the SVD of W as follows:

$$W \approx \hat{W} = USV^T \tag{9}$$

where U is the $(M \times R)$ matrix of left singular vectors u_i $(1 \leq i \leq M)$, S is the $(R \times R)$ diagonal matrix of singular values, V is the $(N \times R)$ matrix of right singular vectors v_j $(1 \leq j \leq N)$, $R \ll M(\ll N)$ is the order of the decomposition, and ^{T} denotes matrix transposition. By definition, the matrix S is positive definite, and the matrices U and V are unitary; hence $U^TU = V^TV = I_R$, the identity matrix of dimension R. As is well known, the matrix $\hat{W}$ is the best rank-R approximation to the word-document matrix W, for any unitarily invariant norm [22].

This decomposition has a dual benefit. First, it eliminates the sparseness issue, by isolating the meaningful components of W. Second, it defines a single vector space with a relatively small dimension R, namely the space spanned by both left and right singular vectors. The ith left singular vector u_i can be viewed as the representation of w_i in this vector space. Similarly, the jth right singular vector v_j can be viewed as the representation of d_j in the *same* space. Thus, this space of dimension R is the space $\mathcal{S}$ which we sought. The dimension R is bounded from above by the rank of the matrix W, and from below by the amount of distortion tolerable in the decomposition. Values of R in the range $R = 200$ to $R = 300$ are typically used for information retrieval [23]. In the present context, we have found $100 \leq R \leq 200$ to work reasonably well.

The basic idea behind (9) is that $\hat{W}$ captures the major structural associations in W and ignores higher order effects. As a result, the "closeness" of vectors in $\mathcal{S}$ is determined by the overall pattern of the language used in $\mathcal{T}$, as opposed to specific constructs. In particular, this means that two words which do not co-occur in $\mathcal{T}$ will still be "close" if that is otherwise consistent with the major patterns of the language (e.g., if they tend to co-occur with a common set of words). This has the important benefit of alleviating the effects of polyzemy. For example, a financial news corpus will be more likely to contain the word "bank" in patterns comprising, e.g., "loan" and "interest" than "river" and "lake," thus forcing the vector representing "bank" to get closer to the appropriate region in the space $\mathcal{S}$.

C. Pseudodocument Representation

While the matrices U and V in (9) are usually obtained simultaneously using a numerical SVD solver, note that (9) provides a way to derive the left or right singular vectors separately if the others are known. For example, the ith row of the matrix W can be written as

$$w_i = u_iSV^T \tag{10}$$

where u_i is as above, and, without loss of generality, we have dropped the approximation symbol. Thus, taking into account the fact that V is unitary, simple algebraic manipulations show that

$$u_i = w_iVS^{-1} \tag{11}$$

assuming, of course, that R is chosen so that S is invertible. Similarly, the jth column of the matrix W is given by

$$d_j = USv_j^T \tag{12}$$

which, since U is unitary, implies

$$v_j = d_j^TUS^{-1}. \tag{13}$$

The latter relation is particularly useful to extend the vector space representation just constructed to new documents, which have not been seen in the training corpus.

Let us assume that the new document $\tilde{d}_p$ (with $p > N$) was not used to derive the space $\mathcal{S}$ through the SVD process outlined above. Can we still find a representation for this document in the space $\mathcal{S}$? The answer is yes. It is easy to construct a feature vector containing, for each word in the underlying vocabulary, the weighted counts (4) with $j = p$. With the convention specified earlier, this feature vector can be simply denoted by $\tilde{d}_p$, a column vector of dimension M. Then the representation of the new document in the space $\mathcal{S}$ is the associated vector $\tilde{v}_p$ given by

$$\tilde{v}_p = \tilde{d}_p^TUS^{-1} \tag{14}$$

through straightforward application of (13).

To convey the fact that it was not part of the SVD extraction, the new document $\tilde{d}_p$ is referred to as a *pseudodocument*. Clearly, if this document contains language patterns which are inconsistent with those extracted from W, the representation $\tilde{v}_p$ will not be adequate. Similarly, if the addition of $\tilde{d}_p$ causes the major structural associations in W to shift in some substantial manner, then (14) will not properly apply. If, on the other hand, the new document generally conforms to the rest of the corpus $\mathcal{T}$, then $\tilde{v}_p$ in (14) will be a reasonable representation for $\tilde{d}_p$. Such pseudodocuments can then be folded into $\mathcal{T}$, leading to an extended corpus denoted by $\mathcal{T}^+$.

D. Computational Effort

Let us first note that classical methods for determining the SVD of dense matrices (see, e.g., [24]) are not optimal for large sparse matrices such as W. Because these methods apply orthogonal transformations (Householder or Givens) directly to the input matrix, they incur excessive fill-in and thereby require tremendous amounts of memory. In addition, they compute all the singular values of W; but here $R \ll M(\ll N)$, so determining all M singular values is computationally wasteful.

Instead, it is more appropriate to solve a sparse symmetric eigenvalue problem, which can then be used to indirectly compute the sparse singular value decomposition. Several suitable iterative algorithms have been proposed by Berry, based on either the subspace iteration or the Lanczos recursion method [25]. The primary cost of these algorithms lies in the total number of sparse matrix–vector multiplications required. Let us denote by μ_r and μ_c the average number of nonzero entries per row and column of W, respectively. Then the total cost in floating point operations (flops) per iteration is given by [25]

$$\mathcal{N}_{svd} = R[2(1+\mu_r)M + 2(1+\mu_c)N]. \tag{15}$$

In a typical case, the density of W (defined as the ratio of the number of nonzero entries over MN) is about 0.25% (cf. [23]), and the value of R is roughly 100. This expression can therefore be approximated by

$$\mathcal{N}_{svd} \approx MN. \tag{16}$$

For the values of M and N mentioned earlier, this corresponds to a few billion flops per iteration. On any midrange desktop machine, such as the Apple Power Macintosh G3/266 (rated at approximately 50 Mflops), this translates into (up to) a few minutes of CPU time. As convergence is typically achieved after 100 or so iterations, the entire decomposition is usually completed within a matter of hours.

This takes care of the off-line cost of the approach. As for the on-line cost, it centers around (14), i.e., the construction of the pseudodocument. For the proposed paradigm to be useful, this ultimately must be done in real time. Assuming that the quantity US^{-1} is precomputed, the cost in flops per pseudodocument is seen to be

$$\mathcal{N}_{pd} = \mu_c M \tag{17}$$

which, under the above conditions, reduces to:

$$\mathcal{N}_{pd} \approx 0.0025 M^2. \tag{18}$$

Thus, for usual values of the vocabulary size, a pseudodocument can be constructed in a fraction of a second of CPU time. In addition, caching can be used to take advantage of any redundancy across similar (or overlapping) pseudodocuments. These observations bode well for the real-time implementation of the LSA framework.

IV. Clustering

In the vector space $\mathcal{S}$ obtained above, each word $w_i \in \mathcal{V}$ is represented by the associated left singular vector of dimension R, u_i, and each document $d_j \in \mathcal{T}$ is represented by the associated right singular vector of dimension R, v_j. Clearly, this opens up the opportunity to apply familiar clustering techniques in $\mathcal{S}$, as long as a distance measure consistent with the SVD formalism is defined on the vector space. The nice thing about this form of clustering is that it takes the global context into account, as opposed to conventional n-gram-based clustering methods which only consider collocational effects.

Since the matrix W embodies, by construction, all structural associations between words and documents, it follows that, for a given training corpus, WW^T characterizes all co-occurrences between words, and W^TW characterizes all co-occurrences between documents. Thus, the extent to which words u_i and u_j have a similar pattern of occurrence across the entire set of documents can be inferred from the (i,j) cell of WW^T, and the extent to which documents v_i and v_j contain a similar pattern of words from the entire vocabulary can be inferred from the (i,j) cell of W^TW.

A. Word Clustering

Expanding WW^T using the SVD expression (9), we obtain

$$WW^T = US^2U^T. \tag{19}$$

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. In other words, how "close" u_i is to u_j in the space $\mathcal{S}$ can be characterized by the dot product between u_iS and u_jS. As a result, a natural metric to consider for the "closeness" between u_i and u_j is the cosine of the angle between u_iS and u_jS. Thus:

$$K(u_i, u_j) = \cos(u_iS, u_jS) = \frac{u_iS^2u_j^T}{\|u_iS\| \, \|u_jS\|} \tag{20}$$

for any $1 \le i, j \le M$. A value of $K(u_i, u_j) = 1$ means the two words always occur in the same semantic context, while a value of $K(u_i, u_j) < 1$ means the two words are used in increasingly different semantic contexts. While (20) does not define a *bona fide* distance measure in the space $\mathcal{S}$, it easy leads to one. For example, over the interval $[0, 2\pi]$, the measure

$$\mathcal{D}(u_i, u_j) = \cos^{-1} K(u_i, u_j) \tag{21}$$

can be readily verified to satisfy the properties of a distance on $\mathcal{S}$.

Once (21) is specified, it is straightforward to proceed with the clustering of the vectors u_i, using any of a variety of algorithms [26]. Since the number of such vectors is relatively large, it is advisable to perform this clustering in stages, using, for example, K-means and bottom-up clustering sequentially. In that case, K-means clustering is used to obtain a coarse partition of the vocabulary $\mathcal{V}$ in to a small set of superclusters. Each supercluster is then itself partitioned using bottom-up clustering. The result of this process is a set of clusters C_k, $1 \le k \le K$, which partitions the space $\mathcal{S}$.

B. Document Clustering

Similarly, expanding W^TW using the SVD expression (9) yields

$$W^TW = VS^2V^T. \tag{22}$$

As before, this means that the (i,j) cell of W^TW can be obtained by taking the dot product between the ith and jth columns of the matrix VS, namely v_iS and v_jS. As a result, a natural metric to consider for the "closeness" between v_i and v_j is the cosine of the angle between v_iS and v_jS. Thus

$$K(v_i, v_j) = \cos(v_iS, v_jS) = \frac{v_iS^2v_j^T}{\|v_iS\| \, \|v_jS\|} \tag{23}$$

for any $1 \le i, j \le N$. This is the same functional as (20), and therefore the distance (21) is equally valid for both word and document clusterings.

Earlier comments regarding clustering implementation apply here as well. The end result is a set of clusters D_ℓ, $1 \le \ell \le L$.

C. Qualitative Evaluation

At this point it might be useful, for concept validation purposes, to illustrate the above clustering framework through a simple experiment. For the sake of brevity, we will only treat the case of word clustering, on a small subset of the corpus

we originally used in [27]. (For an illustration of document clustering, we refer the reader to some recent work by Gotoh and Renals [28]. This work was conducted on a different data base, the British National Corpus, which contains a greater variety of topics.)

We considered a subset $\mathcal{T}$ of the *NAB News* corpus [19], composed of about $N = 21\,000$ documents, comprising approximately ten million words. These articles were selected randomly from the *Wall Street Journal (WSJ)* portion of the corpus. The vocabulary $\mathcal{V}$ was constructed by taking the 20 000 most frequent words of the *NAB News* corpus, augmented by some words from an earlier release of the *WSJ* corpus, for a total of $M = 23\,000$ words. Note that, in contrast with [27], here no attempt was made to remove the noncontent words ("function," or "stop," words). Although such words are uninformative in applications like query analysis, removing them from $\mathcal{V}$ may affect the probability of the unknown word in statistical language modeling.

This led to a ($23\,000 \times 21\,000$) word-document matrix of co-occurrences, stored in sparse fashion. We performed the SVD of this matrix using the single vector Lanczos method [25]. Over the course of this decomposition, we experimented with different numbers of singular values retained (i.e., different dimensions of the associated vector space). Of the values $R = 500$, $R = 250$, $R = 125$, and $R = 75$, we found that $R = 125$ seemed to achieve an adequate balance between reconstruction error (as measured by the difference in the Frobenius norms of W and $\hat{W}$) and noise suppression (as measured by the ratio of the traces of W and $\hat{W}$).

We clustered the (word) vectors in this space into 100 superclasses of approximately 200 vectors each using simple K-means clustering. We then refined each of the superclasses into 20 classes each using bottom-up clustering [26]. This produced a set of 2000 classes, each comprising about ten words on average. Finally, we merged related classes from different superclasses back together to avoid excessive fragmentation. This resulted in a cluster set of size 500.

To show what these word classes look like, we selected two examples of the clusters so obtained.

- **Word Class 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.*
- **Word Class 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.*

The first thing to note is that these word classes comprise words with different part of speech, a marked difference with conventional class n-gram techniques (cf. [4]–[7]). This is a direct consequence of the semantic nature of the derivation. Second, some obvious words seem to be missing from the classes: for example, the singular noun "drawing" from Class 1 and the present tense verb "rule" from Class 2. This is one of the effects of polyzemy: "drawing" and "rule" are more likely to appear in the training text with their alternative meanings (as in "drawing a conclusion" and "breaking a rule," respectively), thus resulting in different class assignments. Finally, some words seem to contribute only marginally to the classes: for example, "hysteria" from Class 1 and "here" from Class 2. These are the unavoidable outliers at the periphery of the clusters.

V. Language Modeling

We are now ready to exploit the framework developed so far in the space $\mathcal{S}$ for the purpose of language modeling. Let w_q denote the word about to be predicted, H_{q-1} the admissible history (context) for this particular word, and $\Pr(w_q|H_{q-1})$ the associated language model probability. In the case of an n-gram language model, for example, $\Pr(w_q|H_{q-1}) = \Pr(w_q|w_{q-1}\,w_{q-2}\cdots w_{q-n+1})$, since the relevant history comprises the last $n-1$ words.

To take the LSA framework into account, we have to consider the slightly modified expression

$$\Pr(w_q|H_{q-1}) = \Pr(w_q|H_{q-1}, \mathcal{S}) \tag{24}$$

where the conditioning on $\mathcal{S}$ reflects the fact that in the proposed derivation the probability depends on the particular vector space arising from the SVD representation. As usual, the quality of this modeling can be measured by the perplexity of (24) on some test text. If Q denotes the total number of words in the test text, this measure is given by

$$PP = \exp\left(-\frac{1}{Q}\sum_{q=1}^{Q}\log \Pr(w_q|H_{q-1}, \mathcal{S})\right). \tag{25}$$

Thus, to construct a semantic language model, there are two issues that need to be addressed: i) specify what the history is in the case of LSA, and ii) find a suitable way to compute (24).

Since the SVD operates on a matrix of co-occurrences between words and documents, the nominal history is, as pointed out before, the document in which w_q appears. However, to be admissible, the context must be causal, and therefore be truncated at word w_{q-1}. Thus, in practice, we have to define H_{q-1} to be the current document up to word w_{q-1}. Note, however, that the method described in the previous sections could be trivially modified to accommodate other admissible histories. For example, H_{q-1} could be anything from the last $n-1$ words, to the current sentence, to the current document, to the past m documents (the latter three, of course, up to word w_{q-1}). The choice only depends on what information is available on the dynamics of the relevant parameters, to enable the selection of the largest semantically consistent text unit. This is a major benefit of the large-span approach.

Without loss of generality, let us therefore continue to assume that H_{q-1} consists of the current document up to word w_{q-1}. There are several way of proceeding, depending on what expansion of $\Pr(w_q|H_{q-1}, \mathcal{S})$ is considered. The choice of this expansion is directly related to the amount of smoothing

desired in the space $\mathcal{S}$, and hence will be most likely dictated by training corpus structure and coverage considerations.

A. Direct Modeling

The simplest choice is to model $\Pr(w_q|H_{q-1}, \mathcal{S})$ directly, in which case no smoothing applies. Obviously, the current document will not (normally) have been seen in $\mathcal{T}$, therefore qualifying as a pseudodocument in the terminology of Section III. If we denote this pseudodocument by $H_{q-1} = \tilde{d}_{q-1}$, then we will be able to use (14) to derive a vector representation $\tilde{v}_{q-1} \in \mathcal{S}$ associated with this pseudodocument. The language model thus becomes

$$\Pr(w_q|H_{q-1}, \mathcal{S}) = \Pr(w_q|\tilde{d}_{q-1}) \tag{26}$$

where $\Pr(w_q|\tilde{d}_{q-1})$ is computed directly from the representations of w_q and $\tilde{d}_{q-1}$ in the space $\mathcal{S}$. In other words, this expression can be directly inferred from the "closeness" between u_q and $\tilde{v}_{q-1}$ in $\mathcal{S}$. We now follow a reasoning similar to that of the previous section to specify that relationship.

Since the matrix W embodies structural associations between words and documents, the extent to which word u_i and document d_j co-occur in the training corpus can be inferred from the (i, j) cell of W. From the SVD formalism, it follows that this can be characterized by taking the dot product between the ith row of the matrix $US^{1/2}$ and the jth row of the matrix $VS^{1/2}$, namely $u_i S^{1/2}$ and $v_j S^{1/2}$. In other words, this dot product reflects how "close" u_i is to v_j in the space $\mathcal{S}$. As a result, a natural metric to consider for the "closeness" between u_q and $\tilde{v}_{q-1}$ is the cosine of the angle between $u_q S^{1/2}$ and $v_{q-1} S^{1/2}$. Thus

$$\begin{aligned} K(u_q, \tilde{v}_{q-1}) &= \cos(u_q S^{1/2}, \tilde{v}_{q-1} S^{1/2}) \\ &= \frac{u_q S \tilde{v}_{q-1}^T}{\|u_q S^{1/2}\| \, \|\tilde{v}_{q-1} S^{1/2}\|} \end{aligned} \tag{27}$$

for any q indexing a word in the text data. A value of $K(u_q, \tilde{v}_{q-1}) = 1$ means that $\tilde{d}_{q-1}$ is a strong semantic predictor of w_q, while a value of $K(u_q, \tilde{v}_{q-1}) < 1$ means that the history carries increasingly less information about the current word. Note that (27) is functionally equivalent to (20) and (23), but involves scaling by $S^{1/2}$ instead of S. Thus, a transformation similar to (21) can be used to infer from (27) a *bona fide* distance in the space $\mathcal{S}$.

To enable the computation of $\Pr(w_q|\tilde{d}_{q-1})$, it remains to go from this distance measure to an actual probability measure. This can be done through simple induction (see, e.g., [26]), by just normalizing appropriately to ensure that the total probability mass is equal to one. In this manner, the distance measure naturally induces an empirical multivariate distribution in the space $\mathcal{S}$.[1] Since this is a joint distribution on words and documents, it is suitable to look up the quantity $\Pr(w_q, \tilde{d}_{q-1})$, for every word $w_q \in \mathcal{V}$ and every (pseudo-)document $\tilde{d}_{q-1} \in \mathcal{T}^+$. Applying marginal probability expansion, $\Pr(w_q|\tilde{d}_{q-1})$ can thus be obtained as

$$\Pr(w_q|\tilde{d}_{q-1}) = \frac{\Pr(w_q, \tilde{d}_{q-1})}{\sum_{w_i \in \mathcal{V}} \Pr(w_i, \tilde{d}_{q-1})} \tag{28}$$

where the summation in the denominator extends over all words in $\mathcal{V}$.

Note that $\Pr(w_q|\tilde{d}_{q-1})$ reflects the "relevance" of word w_q to the admissible history, as observed through $\tilde{d}_{q-1}$. As such, it will be highest for words whose meaning aligns most closely with the semantic fabric of $\tilde{d}_{q-1}$ (i.e., relevant "content" words), and lowest for words which do not convey any particular information about this fabric (e.g., "function" words like "the"). Since content words tend to be rare and function words tend to be frequent, this will translate into a relatively high value for (25). Thus, even though this model appears to have the same order as a standard unigram, it will likely exhibit a significantly weaker predictive power.

B. Smoothing via Word Clustering

Alternatively, we can take advantage of the additional layer of knowledge uncovered in the previous section through word clustering. This clustering essentially acts as a smoothing mechanism on top of the vector space representation derived from LSA. By exploiting it, we can expect words related to the current document to contribute with more synergy, and unrelated words to be better discounted. Along those lines, the right-hand side of (26) is expanded as:

$$\Pr(w_q|\tilde{d}_{q-1}) = \sum_{k=1}^{K} \Pr(w_q|C_k) \Pr(C_k|\tilde{d}_{q-1}) \tag{29}$$

where the clusters C_k result from the word clustering of Section IV-A. In (29), the probability $\Pr(C_k|\tilde{d}_{q-1})$ is qualitatively similar to (26) and can therefore be obtained with the help of (27), by simply replacing the representation of the word w_q by that of the centroid of word cluster C_k. In contrast, the probability $\Pr(w_q|C_k)$ depends on the "closeness" of w_q relative to this (word) centroid. To derive it, we therefore have to rely on the empirical multivariate distribution induced not by the distance obtained from (27), but by that obtained from the measure (20) mentioned in Section IV-A. Note that a distinct distribution can be inferred on each of the clusters C_k, thus allowing us to compute all quantities $\Pr(w_i|C_k)$ for $1 \leq i \leq M$ and $1 \leq k \leq K$.

The behavior of the model (29) depends on the number of word clusters defined in the space $\mathcal{S}$. If there are as many classes as words in the vocabulary ($K = M$), then (29) reduces to (26), thus introducing no smoothing compared to direct modeling. Conversely, if all the words are in a single class ($K = 1$), the model becomes maximally smooth: the influence of specific semantic events disappears, leaving only a broad (and therefore weak) vocabulary effect to take into account. This may in turn degrade the predictive power of the model.

Generally speaking, as the number of word classes C_k increases, the contribution of $\Pr(w_q|C_k)$ tends to increase, because the clusters become more and more semantically meaningful. By the same token, however, the contribution of

[1] Alternatively, it is also possible to induce a family of exponential distributions with pertinent marginality constraints, which is potentially optimal [29]. In practice, we have not found problematic to rely on the empirical distribution instead.

$\Pr(C_k|\tilde{d}_{q-1})$ for a given $\tilde{d}_{q-1}$ tends to decrease, because the clusters eventually become too specific and fail to reflect the overall semantic fabric of $\tilde{d}_{q-1}$. These two trends have the net effect to decrease perplexity at first, and then increase it as the number of classes continues to increase. Thus, there exists an optimal cluster set size where perplexity is minimized.

C. Smoothing via Document Clustering

Another possibility is exploit document clusters as opposed to word clusters. This amounts to a different kind of smoothing, in which we express the right-hand side of (26) as

$$\Pr(w_q|\tilde{d}_{q-1}) = \sum_{\ell=1}^{L} \Pr(w_q|D_\ell)\ \Pr(D_\ell|\tilde{d}_{q-1}) \tag{30}$$

where the clusters D_ℓ result from the document clustering of the previous section. This time, it is the probability $\Pr(w_q|D_\ell)$ that is qualitatively similar to (26), and can therefore be obtained with the help of (27). As for the probability $\Pr(D_\ell|\tilde{d}_{q-1})$, it depends on the "closeness" of $\tilde{d}_{q-1}$ relative to the centroid of document cluster D_ℓ. Thus, it can be obtained through the empirical multivariate distribution induced by the distance derived from (23) in Section IV-B. As before, a distinct distribution can be inferred on each of the clusters D_ℓ, thereby allowing us to compute all quantities $\Pr(w_j|D_\ell)$ for $1 \le j \le N$ and $1 \le \ell \le L$.

Again, the behavior of the model (30) depends on the number of document clusters defined in the space $\mathcal{S}$. Compared to (29), however, (30) is more difficult to interpret in the limits (i.e., $L = 1$ and $L = N$). If $L = N$, for example, (30) does not reduce to (26), because $\tilde{d}_{q-1}$ has not been seen in the training data, and therefore cannot be identified with any of the existing clusters. Similarly, the fact that all the documents are in a single cluster ($L = 1$) does not necessarily imply the degree of degenerescence observed in Section V-B, because the cluster itself is strongly indicative of the general discourse domain (which was less true of the "vocabulary cluster" in Section V-B). Hence, depending on the size and structure of the corpus, the model may still be adequate to capture general discourse effects.

So what happens as the number of document classes D_ℓ increases? The contribution of $\Pr(w_q|D_\ell)$ tends to increase, to the extent that a more homogeneous topic boosts the effects of any related content words. On the other hand, the contribution of $\Pr(D_\ell|\tilde{d}_{q-1})$ tends to decrease, because the clusters represent more and more specific topics, which increases the chance that the pseudodocument $\tilde{d}_{q-1}$ becomes an outlier. These two trends have the same net effect as above. Thus, again there exists an optimal cluster set size where perplexity is minimized.

D. Smoothing via Joint Clustering

Finally, the above two alternatives can be merged. This leads to a mixture language model specified by

$$\Pr(w_q|\tilde{d}_{q-1}) = \sum_{k=1}^{K}\sum_{\ell=1}^{L} \Pr(w_q|C_k,\, D_\ell)\ \Pr(C_k, D_\ell|\tilde{d}_{q-1}) \tag{31}$$

which, for tractability, can be approximated as

$$\Pr(w_q|\tilde{d}_{q-1}) = \sum_{k=1}^{K}\sum_{\ell=1}^{L} \Pr(w_q|C_k)\ \Pr(C_k|D_\ell)\ \Pr(D_\ell|\tilde{d}_{q-1}). \tag{32}$$

In this expression, the clusters C_k and D_ℓ are as previously, as are the quantities $\Pr(w_q|C_k)$ and $\Pr(D_\ell|\tilde{d}_{q-1})$. As for the probability $\Pr(C_k|D_\ell)$, it is qualitatively similar to (26), and can therefore be obtained accordingly. Note that the simplification from (31) to (32) enables us to derive the probabilities in exactly the same way as above.

As before, the behavior of the model (32) depends on the number of word clusters and document clusters defined in the space $\mathcal{S}$. Most of the earlier comments can be extended to this case in a straightforward fashion. For example, if there are as many word classes as words in the vocabulary ($K = M$), then (32) reduces to (30), thus introducing no further smoothing compared to the modeling based on document clusters. Generally speaking, for a given number of word classes, we can expect the model to follow the behavior of (29), and for a given number of document classes, we can expect the model to follow the behavior of (30). Consequently, there exist an optimal set of word clusters and an optimal set of document clusters which are associated with minimal perplexity.

VI. Integration with N-Grams

As pointed out earlier, the LSA framework just proposed does not exploit positional information at all. Hence, it is inherently unable to adequately capture the (local) branching properties of the language. But this is precisely what is normally assessed through perplexity, of course. Thus, in terms of predictive power, as measured by perplexity, LSA models should not be expected to match n-grams. On the other hand, it is clearly desirable to combine global (document level) constraints such as provided by the LSA paradigm with local (immediate context) constraints such as provided by the n-gram paradigm. This amounts to leveraging multispan information to derive an integrated language model combining the benefits of both short- and large-span contexts.

This integration could occur in a number of ways, such as straightforward interpolation, or within the maximum entropy framework [13]. In the following, we develop an alternative formulation for the combination of the two paradigms. The underlying premise of this formulation is that it makes most sense for the recognition process to proceed locally while taking global constraints into account. Consequently, the n-gram paradigm should assume a primary role and the LSA framework a secondary role. The end result, in effect, is a modified n-gram language model incorporating large-span semantic information.

To achieve this goal, we need to compute:

$$\Pr(w_q|H_{q-1}) = \Pr(w_q|H_{q-1}^{(n)},\, H_{q-1}^{(l)}) \tag{33}$$

where the history H_{q-1} now comprises an n-gram component $[H_{q-1}^{(n)} = w_{q-1}\, w_{q-2} \cdots w_{q-n+1}]$ as well as an LSA com-

ponent ($H_{q-1}^{(l)} = \tilde{d}_{q-1}$). Following the same reasoning as for (28), this expression can be rewritten as

$$\Pr(w_q|H_{q-1}) = \frac{\Pr(w_q,\, H_{q-1}^{(l)}|H_{q-1}^{(n)})}{\sum_{w_i\in\mathcal{V}} \Pr(w_i,\, H_{q-1}^{(l)}|H_{q-1}^{(n)})} \tag{34}$$

where the summation in the denominator extends over all words in $\mathcal{V}$. Expanding and rearranging, the numerator of (34) is seen to be:

$$\begin{aligned}&\Pr(w_q,\, H_{q-1}^{(l)}|H_{q-1}^{(n)})\\ &\quad= \Pr(w_q|H_{q-1}^{(n)}) \cdot \Pr(H_{q-1}^{(l)}|w_q,\, H_{q-1}^{(n)})\\ &\quad= \Pr(w_q|w_{q-1}\, w_{q-2}\, \cdots\, w_{q-n+1})\\ &\qquad\cdot \Pr(\tilde{d}_{q-1}|w_q\, w_{q-1}\, w_{q-2}\, \cdots\, w_{q-n+1}).\end{aligned} \tag{35}$$

Now 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, and probably does not matter very much for function words. As a result, the integrated probability becomes

$$\begin{aligned}&\Pr(w_q|H_{q-1})\\ &\quad= \frac{\Pr(w_q|w_{q-1}\, w_{q-2}\, \cdots\, w_{q-n+1})\, \Pr(\tilde{d}_{q-1}|w_q)}{\sum_{w_i\in\mathcal{V}} \Pr(w_i|w_{q-1}\, w_{q-2}\, \cdots\, w_{q-n+1})\, \Pr(\tilde{d}_{q-1}|w_i)}.\end{aligned} \tag{36}$$

Interestingly, this expression has a quasi-Bayesian interpretation. If $\Pr(\tilde{d}_{q-1}|w_q)$ is viewed as a prior probability on the current document history, then (36) simply translates the classical Bayesian estimation of the n-gram (local) probability using a prior distribution obtained from (global) LSA. This provides additional evidence to justify the above assumption.

As a final remark, note that the above derivation does not assume any particular form of $\Pr(\tilde{d}_{q-1}|w_q)$. Thus, any of the expressions (26), (29), (30), or (32) can be used to compute (36), resulting in four families of combined n-gram/LSA language models.

VII. Performance

To evaluate the performance of the language models proposed in the previous section, it was desirable to train on a larger, more typical corpus than that used in Section IV-C. We considered the so-called *WSJ0* part of the *NAB News* corpus [19]. This was convenient for comparison purposes since conventional bigram and trigram language models are readily available, trained on exactly the same data [11], [19]. Thus, the training text corpus $\mathcal{T}$ was composed of about $N = 87\,000$ documents spanning the years 1987 to 1989, comprising approximately 42 million words. In addition, about 2 million words from 1992 and 1994 were set aside for test purposes. The vocabulary $\mathcal{V}$ was the same as in Section IV-C, and comprised a total of $M = 23\,000$ words.

TABLE I
Perplexity Figures for Integrated Bigram and Trigram Language Models (36), Compared to Standard Bigram and Trigram

Language Model	Test Set Perplexity
Standard Bigram	215
Standard Trigram	142
Combined Bigram/LSA	147
Combined Trigram/LSA	115

A. Direct Model

We performed the SVD of the matrix of co-occurrences between words and documents in the same manner as described in Section IV-C. This led to a vector space $\mathcal{S}$ of dimension $R = 125$. We then constructed the direct model (26) and combined it as in (36), either with the standard bigram (yielding the integrated bigram/LSA, or bi-LSA, language model), or with the standard trigram (yielding the integrated trigram/LSA, or tri-LSA, language model). Finally, we measured the resulting perplexity on the test data previously set aside. A summary is provided in Table I.

We found a value of 147 for the bi-LSA model and 115 for the tri-LSA model. These results are to be compared with the baseline results obtained with the standard bigram and trigram language models, found to be 215 and 142, respectively. Thus, the bi-LSA language model (36) leads to a 32% reduction in perplexity compared to the standard bigram, which brings it to the same level of performance as the standard trigram. The tri-LSA language model leads to a somewhat smaller relative improvement compared to the standard trigram; however, the reduction in perplexity still reaches almost 20%.

To investigate scalability issues, we also randomly separated the documents into five bins of approximately 17 000 documents each. We then performed five distinct SVD's of the resulting matrices, again using $R = 125$ throughout for the order of the decomposition. This allowed us to measure perplexity for each bin. We then took the average to obtain a single perplexity value. The results, reported in Table II for the bi-LSA case, show that the binning process does not significantly degrade performance. This in turn indicates that, if necessary, the computational load can be alleviated by using a random sample of documents in lieu of the entire corpus. This strategy might also be required to avoid numerical or convergence problems in the case of very large corpora. With the full *NAB News* corpus (comprising about half a million documents), for example, the matrix W would have been too large for our current implementation of the SVD algorithm.

B. Smoothed Models

To take advantage of smoothing in the integrated language models, word and/or document clustering had to be done. We used the same two-level procedure (using K-means and bottom-up clustering) as described in Section IV-C to cluster

TABLE II
PERPLEXITY FIGURES FOR SCALABILITY INVESTIGATION. "BIGRAM/LSA FULL" REFERS TO INTEGRATED BIGRAM LANGUAGE MODEL DERIVED ON ENTIRE CORPUS; "BIGRAM/LSA BIN n" ($n = 1, \cdots, 5$) REFERS TO MODEL DERIVED ON BIN n; "BIGRAM/LSA AVERAGE 1–5" REFERS TO AVERAGE PERPLEXITY OVER ALL BINS

Language Model	Test Set Perplexity
Bigram/LSA Full	147
Bigram/LSA Bin 1	134
Bigram/LSA Bin 2	159
Bigram/LSA Bin 3	151
Bigram/LSA Bin 4	146
Bigram/LSA Bin 5	155
Bigram/LSA Average 1-5	149

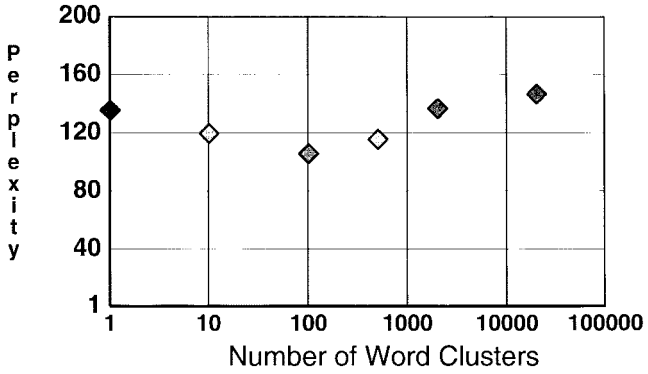

Fig. 1. Perplexity versus number of word clusters for bigram/LSA language model (36) with expansion (29).

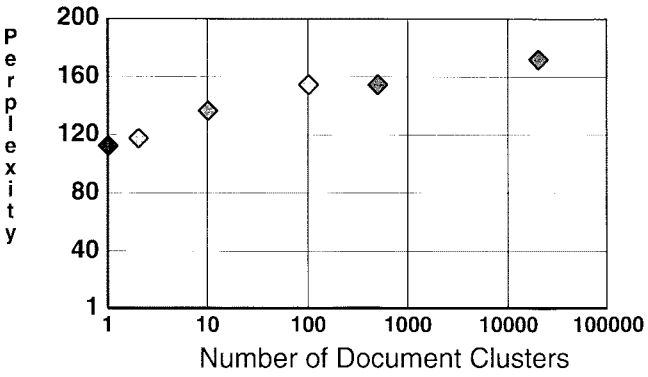

Fig. 2. Perplexity versus number of document clusters for bigram/LSA language model (36) with expansion (30).

the word vectors obtained above, and merged related classes to create cluster sets of different size. We then independently repeated this procedure to cluster the document vectors, and again merged related classes to create cluster sets of different size. Finally, for each combination of cluster set sizes, we measured perplexity as before.

The bi-LSA results are illustrated in Figs. 1–5 for different sizes of the word and document cluster sets, as appropriate.

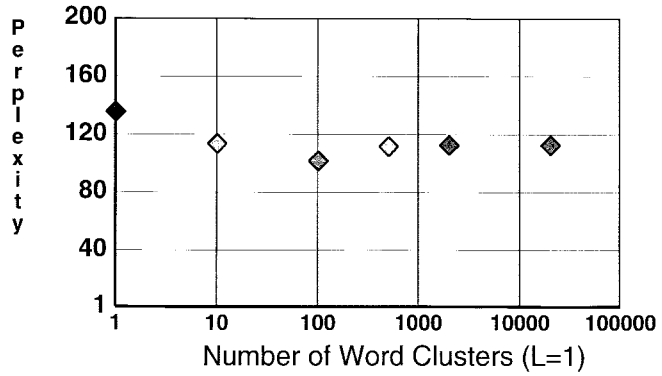

Fig. 3. Perplexity versus number of word clusters for bigram/LSA language model (36) with expansion (32), in case of single document cluster.

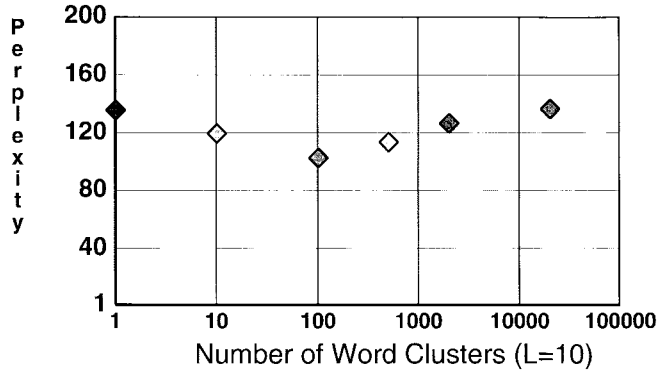

Fig. 4. Perplexity versus number of word clusters for bigram/LSA language model (36) with expansion (32), in case of ten document clusters.

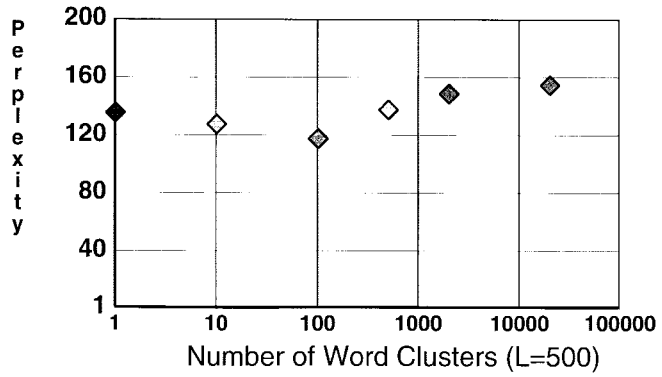

Fig. 5. Perplexity versus number of word clusters for bigram/LSA language model (36) with expansion (32), in case of 500 document clusters.

Figs. 1 and 2 correspond to using word classes only and document classes only, respectively. Figs. 3–5 corresponds to using both word and document classes. In all cases perplexity is plotted (on a linear scale) against the relevant number of clusters (on a log scale). All plots are seen to go through a perplexity minimum for a particular size of the cluster set.

On Fig. 1, this minimum is equal to 106 and is reached for a word cluster set size $K = 100$. This is to be compared with the perplexity associated with $K = 23\,000$ clusters, which, as predicted earlier, is 147, i.e., the same value as

obtained using direct modeling. This important difference in perplexity illustrates the smoothing benefits brought about by clustering. Words related to the current document contribute with more synergy, while unrelated words are better discounted, This, in turn, causes perplexity to drop. Conversely, when K is too small, too much smoothing is introduced and information gets lost in the process, causing perplexity to edge up.

Fig. 2 exhibits the same general behavior as Fig. 1, with two notable differences. First, the minimum perplexity is somewhat higher (116) than in Fig. 1. This indicates that clustering documents is not as powerful as clustering words, in the sense just described. Second, the minimum is attained for a size of the document cluster set smaller (1) than the optimal size of the word cluster set observed in Fig. 1, and perplexity increases faster away from this value. This may perhaps reflect the fact that it is more difficult to achieve semantic homogeneity at the document level than at the word level, an intuitively reasonable proposition. Alternatively, it may be an artifact of the document collection considered, which arguably is already quite homogeneous to begin with.

Figs. 3–5 plot the perplexity obtained against the number of word clusters, for three values of the document cluster set size ($L = 1$, $L = 10$, and $L = 500$, respectively). The three curves exhibit the same general convex shape observed in Fig. 1, but reach different minimum values. The minimum is equal to 102 in Fig. 3, 107 in Fig. 4, and 118 in Fig. 5. Thus, the best curve is the one of Fig. 3, obtained with a document cluster set size $L = 1$. In this case the minimum is reached for a word cluster set size $K = 100$. Note, however, that the curve is fairly flat, with perplexity values virtually identical over a wide range of word cluster set sizes.

A qualitatively similar behavior was observed in the case of the corresponding tri-LSA language models, and the best results obtained in each case have been grouped in Table III. To summarize, the best smoothed bi-LSA perplexity values (102–106) are about 50% better than that obtained using the standard bigram, while the best smoothed tri-LSA perplexity values (95–98) are about 30% better than that obtained using the standard trigram. We conclude that the new integrated language models are quite effective in combining global semantic prediction with the usual local predictive power of n-grams.

TABLE III
PERPLEXITY FIGURES FOR BEST INTEGRATED BIGRAM AND TRIGRAM LANGUAGE MODELS (36) UNDER VARIOUS SMOOTHING SCHEMES

Language Model	Test Set Perplexity
Bigram/LSA, No Smoothing	147
With Document Smoothing	116
With Word Smoothing	106
With Joint Smoothing	102
Trigram/LSA, No Smoothing	115
With Document Smoothing	103
With Word Smoothing	98
With Joint Smoothing	95

VIII. CONCLUSION

We have described a language modeling approach based on the LSA paradigm. In this approach, hidden (semantic) redundancies are tracked across documents, where a document is defined as a semantically homogeneous set of sentences embodying a given storyline. One of the advantages of this framework is that it results in a vector representation of each word and document in a space of relatively modest dimension. This makes it possible to specify suitable metrics for word–document, word–word, and document–document comparisons. In addition, well-known clustering algorithms can be applied efficiently, which allows for a variety of smoothing schemes.

The vector representation resulting from the LSA approach embodies the major structural associations of the corpus as determined by the overall pattern of the language. Hence, the new language models are semantic in nature and capture large span relationships between words. This stands in marked contrast with conventional n-grams, which inherently rely on more syntactically-oriented, short-span relationships. This means that one paradigm is better suited to account for the local constraints in the language, while the other one is more adept at handling global constraints.

As a result, the two approaches complement each other. To harness this synergy, we have derived an integrative formulation to combine the standard n-gram formalism with the LSA paradigm. By taking advantage of the various kinds of smoothing available, several families of integrated n-gram/LSA models have been obtained. The resulting multispan language models were shown to substantially outperform the associated standard n-grams on a subset of the *NAB News* corpus.

ACKNOWLEDGMENT

The author would like to thank N. B. Coccaro, University of Colorado at Boulder, for many profitable discussions on information retrieval and latent semantic indexing during his internship at Apple Computer, Inc., in the summer of 1995. Without these early exchanges, this work would not have been possible. He is also grateful to T. K. Landauer, from the same institution, for pointing out a number of insightful references on the latent semantic model of knowledge acquisition, and to the anonymous reviewers for their constructive comments and helpful suggestions that greatly improved the manuscript.

REFERENCES

[1] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 179–190, Mar. 1983.

[2] F. Jelinek, "Self-organized language modeling for speech recognition," in *Readings in Speech Recognition*, A. Waibel and K. F. Lee, Eds. San Mateo, CA: Morgan Kaufmann, 1990, pp. 450–506.

[3] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 400–401, Mar. 1987.

[4] U. Essen and V. Steinbiss, "Co-occurrence smoothing for stochastic language modeling," in *Proc. 1992 Int. Conf. Acoustics, Speech, Signal Processing*, San Francisco, CA, Mar. 1992, pp. 161–164.
[5] G. Maltese and F. Mancini, "An automatic technique to include grammatical and morphological information in a trigram-based statistical language model," in *Proc. 1992 Int. Conf. Acoustics, Speech, Signal Processing*, San Francisco, CA, Mar. 1992, pp. 157–160.
[6] M. Jardino and G. Adda, "Automatic word classification using simulated annealing," in *Proc. 1993 Int. Conf. Acoustics, Speech, Signal Processing*, Minneapolis, MN, May 1993, pp. 41–44.
[7] M. Tamoto and T. Kawabata, "Clustering word category based on binomial *posteriori* co-occurrence distribution," in *Proc. 1995 Int. Conf. Acoustics, Speech, Signal Processing*, Detroit, MI, May 1995, pp. 165–168.
[8] M. Jardino, "Multilingual stochastic n-gram class language models," in *Proc. 1996 Int. Conf. Acoustics, Speech, Signal Processing*, Atlanta, GA, May 1996, pp. I161–I163.
[9] T. Niesler and P. Woodland, "A variable-length category-based N-gram language model," in *Proc. 1996 Int. Conf. Acoustics, Speech, Signal Processing*, Atlanta, GA, May 1996, pp. I164–I167.
[10] A. Farhat, J. Isabelle, and D. O'Shaughnessy, "Clustering words for statistical language models based on contextual word similarity," in *Proc. 1996 Int. Conf. Acoustics, Speech, Signal Processing*, Atlanta, GA, May 1996, pp. I180–I183.
[11] R. Rosenfeld, "The CMU statistical language modeling toolkit and its use in the 1994 ARPA CSR evaluation," in *Proc. ARPA Speech and Natural Language Workshop*, Mar. 1994.
[12] R. Lau, R. Rosenfeld, and S. Roukos, "Trigger-based language models: A maximum entropy approach," in *Proc. 1993 Int. Conf. Acoustics, Speech, Signal Processing*, Minneapolis, MN, May 1993, pp. II45–II48.
[13] R. Rosenfeld, "A maximum entropy approach to adaptive statistical language modeling," in *Computer Speech and Language.* New York: Academic Press, 1996, vol. 10, pp. 187–228.
[14] S. Deerwester *et al.*, "Indexing by latent semantic analysis," *J. Amer. Soc. Inform. Sci.*, vol. 41, pp. 391–407, 1990.
[15] P. W. Foltz and S. T. Dumais, "Personalized information delivery: An analysis of information filtering methods," *Commun. ACM*, vol. 35, pp. 51–60, 1992.
[16] M. W. Berry, S. T. Dumais, and G. W. O'Brien, "Using linear algebra for intelligent information retrieval," *SIAM Rev.*, vol. 37, pp. 573–595, 1995.
[17] R. E. Story, "An explanation of the effectiveness of latent semantic indexing by means of a Bayesian regression model," *Inform. Process. Manage.*, vol. 32, pp. 329–344, 1996.
[18] T. K. Landauer and S. T. Dumais, "Solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge," *Psych. Rev.*, vol. 104, pp. 211–240, 1997.
[19] F. Kubala *et al.*, "The hub and spoke paradigm for CSR evaluation," in *Proc. ARPA Speech and Natural Language Workshop*, Mar. 1994, pp. 40–44.
[20] 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. Cogn. Sci. Soc.*, 1997.
[21] S. T. Dumais, "Improving the retrieval of information from external sources," *Behavior Res. Methods, Instrum., Comput.*, vol. 23, pp. 229–236, 1991.
[22] J. K. Cullum and R. A. Willoughby, "Real rectangular matrices" in *Lanczos Algorithms for Large Symmetric Eigenvalue Computations—Vol. 1, Theory.* Boston, MA: Brickhauser, 1985.
[23] S. T. Dumais, "Latent semantic indexing (LSI) and TREC-2," in *Proc. Second Text Retrieval Conf. (TREC-2)*, D. Harman, Ed., 1994, NIST Pub. 500-215, pp. 105–116, .
[24] M. Berry and A. Sameh, "An overview of parallel algorithms for the singular value and dense symmetric eigenvalue problems," *J. Computat. Appl. Math.*, vol. 27, pp. 191–213, 1989.
[25] M. W. Berry, "Large-scale sparse singular value computations," *Int. J. Supercomp. Appl.*, vol. 6, pp. 13–49, 1992.
[26] J. R. Bellegarda, "Context-dependent vector clustering for speech recognition," *Speech and Speaker Recognition: Advanced Topics*, C.-H. Lee, F. K. Soong, and K. K. Paliwal, Eds. Boston, MA: Kluwer, Mar. 1996, ch. 6, pp. 133–157.
[27] J. R. Bellegarda *et al.*, "A novel word clustering algorithm based on latent semantic analysis," in *Proc. 1996 Int. Conf. Acoustics, Speech, Signal Processing*, Atlanta, GA, May 1996, pp. I172–I175.
[28] Y. Gotoh and S. Renals, "Document space models using latent semantic analysis," in *Proc. EuroSpeech'97*, Rhodes, Greece, vol. 3, pp. 1443–1448.
[29] W. Byrne, personal communication, Nov. 1997.

Jerome R. Bellegarda (M'87–SM'98) was born in Freiburg-in-Breisgau, West Germany, on November 30, 1961. He received the Diplome d'Ingénieur degree (summa cum laude) 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, Rochester, NY, in 1984 and 1987, respectively.

In 1987, he was a Research Associate in the Department of Electrical Engineering, University of Rochester, developing multiple access coding techniques. From 1988 to 1994, he was a Research Staff Member at the IBM T. J. Watson Research Center, Yorktown Heights, NY, working on various improvements to the modeling component of the IBM continuous speech recognition system, and developing advanced feature extraction and recognition modeling algorithms for cursive on-line handwriting. In 1994, he joined Apple Computer, Cupertino, CA, where he is currently Principal Scientist in the Spoken Language Group. At Apple, he has worked on speaker adaptation, Asian dictation, statistical language modeling, and advanced dialog interactions. His research interests include voice-driven man–machine communications, multiple input/output modalities, and multimedia knowledge management. He has also contributed chapters to several edited books, including *Advances in Handwriting and Drawing: A Multidisciplinary Approach* (Paris, France: Europia, 1994) and *Automatic Speech and Speaker Recognition—Advanced Topics* (Boston, MA: Kluwer, 1996). He has written more than 60 journal and conference papers, and holds a dozen patents.

Dr. Bellegarda was a member of the ARPA CSR Corpus Coordination Committee between 1992 and 1994.