Large Vocabulary Speech Recognition with Multispan Statistical Language Models

Jerome R. Bellegarda, *Senior Member, IEEE*

***Abstract*—Multispan language modeling refers to the integration of the various constraints, both local and global, present in the language. It was recently proposed to capture global constraints through the use of latent semantic analysis, while taking local constraints into account via the usual n-gram approach. This has led to several families of data-driven, multispan language models for large vocabulary speech recognition. Because of the inherent complementarity in the two types of constraints, the multispan performance, as measured by perplexity, has been shown to compare favorably with the corresponding n-gram performance. The objective of this work is to characterize the behavior of such multispan modeling in actual recognition. Major implementation issues are addressed, including search integration and context scope selection. Experiments are conducted on a subset of the *Wall Street Journal* (WSJ) speaker-independent, 20 000-word vocabulary, continuous speech task. Results show that, compared to standard n-gram, the multispan framework can lead to a reduction in average word error rate of over 20%. The paper concludes with a discussion of intrinsic multi-span tradeoffs, such as the influence of training data selection on the resulting performance.**

***Index Terms*—Latent semantic analysis, multispan integration, n-grams, speech recognition, statistical language modeling.**

I. Introduction

OVER the past decade, n-gram language modeling has steadily emerged as the formalism of choice for large vocabulary continuous speech recognition in a wide range of domains. Concerns regarding parameter reliability, however, restrict current implementations to low values of n (cf., e.g., [1]), which in turn imposes an artificially local horizon to the language model. As a result, n-grams are inherently unable to capture large-span relationships in the language.

Consider, for instance, predicting the word "*fell*" from the word "*stocks*" in the two equivalent phrases:

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

and

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

In (1), the prediction can be done with the help of a bigram language model ($n = 2$). With the kind of resources currently available, this is rather straightforward [2]. 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 take an overall view of the entire sentence, as opposed to just the n preceding words. This requires a paradigm shift toward parsing and rule-based grammars, such as are routinely and successfully employed in small and medium vocabulary recognition applications. This approach solves the locality problem, since it takes sentence-level constraints into account. Unfortunately, it is still too restrictive for large vocabulary recognition: parsing-based methods do not (yet) scale well to general discourse, which is precisely the reason why the n-gram framework was so widely adopted in the first place.

This has sparked interest in statistical large-span modeling, which is concerned with alternative ways to extract suitable long distance information (other than resorting to a formal parsing mechanism). Broadly speaking, the goal of statistical large-span modeling is to relate to one another those words that 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.

One early attempt along these lines was based on the concept of word triggers [3]. 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. Thus, the trigger approach solves the problem, at least for those trigger pairs that have been selected by the algorithm [4].

Unfortunately, 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 cooccur 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, which limits the potential of low frequency word triggers [4]. Still, self-triggers have been shown to be particularly powerful and robust [3], which underscores the desirability of exploiting correlations between the current word and features of the document history. What seems to be needed is a somewhat more flexible framework to exploit the long distance information present in this history.

Manuscript received November 11, 1998; revised August 23, 1999. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. James R. Glass.

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

Publisher Item Identifier S 1063-6676(00)00321-7.

This observation led the author to explore the use of latent semantic analysis for such purpose [5]. Latent semantic analysis (LSA) was originally formulated in the context of information retrieval, where it proved to be a very effective indexing mechanism [6]–[10]. In latent semantic indexing, co-occurrence analysis takes place across much larger spans than with a traditional n-gram approach, and on a much larger scale than with the trigger approach. The span of choice is a *document*, which can be defined as a semantically homogeneous set of sentences embodying a given storyline. As for scale, every combination of words from the vocabulary is viewed as a potential trigger combination. Thus, to a large extent, the LSA paradigm can be viewed as an extension of the word trigger concept, where trigger pair selection is addressed as part of the analysis, rather than as a postprocessing step. This extension (in both span and scale) leads 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* corpus (NAB) [11]. Once this is done, the LSA paradigm can be used for word and document clustering (cf. [12] and [13]), as well as for language modeling [14]. In all cases, it was found to be suitable to capture some of the global constraints present in the language. In fact, hybrid n-gram + LSA language models, where LSA is embedded into the standard n-gram formulation, were shown to result in a substantial reduction in perplexity [15].

The objective of this paper is to assess the behavior of such multispan language models in actual speech recognition experiments. Specifically, we examine the achievable reduction in average word error rate, and discuss a number of factors which influence performance. The paper is organized as follows. In the next section, we review the salient properties of LSA-based statistical language modeling. Section III addresses the major implementation issues involved in using the resulting multispan models for large vocabulary recognition. In Section IV, we illustrate some of the benefits associated with multispan modeling on a subset of the *Wall Street Journal* (WSJ) task. Finally, Section V discusses the inherent tradeoffs associated with the approach, as evidenced by the influence of the data selected to train the LSA component of the multispan model.

II. N-Gram + LSA Language Modeling

Let $\mathcal{V}$, $|\mathcal{V}| = M$, be some underlying vocabulary and $\mathcal{T}$ a training text corpus, comprising N articles (documents) relevant to some domain of interest (like business news, for example, in the case of the NAB corpus [11]). Typically, M and N are on the order of ten thousand and hundred thousand, respectively; $\mathcal{T}$ might comprise a hundred million words or so.

The LSA approach defines a mapping between the discrete sets $\mathcal{V}$, $\mathcal{T}$ and a continuous vector space $\mathcal{S}$, whereby each word w_i in $\mathcal{V}$ is represented by a vector u_i in $\mathcal{S}$, and each document d_j in $\mathcal{T}$ is represented by a vector v_j in $\mathcal{S}$. For the sake of brevity, we refer the reader to [5] for further details on the mechanics of LSA and n-gram + LSA language modeling, and just briefly summarize here.

A. Feature Representation

The first step is the construction of a matrix (W) of co-occurrences between words and documents. In marked contrast with n-gram modeling, word order is ignored, which is of course in line with the semantic nature of the approach [16]. Thus, the matrix W 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.

Among other possibilities, a suitable expression for the $(i,\ j)$th element of W is given by (cf. [12])

$$w_{i,j} = (1 - \varepsilon_i)\,\frac{c_{i,j}}{n_j} \tag{3}$$

where $c_{i,j}$ is the number of times w_i occurs in d_j, n_j is the total number of words present in d_j, and ε_i is the normalized entropy of w_i in the corpus $\mathcal{T}$. The expression for ε_i is easily seen to be:

$$\varepsilon_i = -\frac{1}{\log N}\sum_{j=1}^{N}\frac{c_{i,j}}{t_i}\log\frac{c_{i,j}}{t_i} \tag{4}$$

where $t_i = \sum_j c_{i,j}$ is the total number of times w_i occurs in $\mathcal{T}$. Note that a value of ε_i close to 1 indicates a word distributed across many documents throughout the corpus, while a value of ε_i close to zero means that the word is present only in a few specific documents. Hence, $1 - \varepsilon_i$ represents a global indexing weight for the word w_i.

B. Singular Value Decomposition

The second step, after the word-document matrix of co-occurrences is constructed, is to compute the singular value decomposition (SVD) of W as

$$W \approx \hat{W} = U\,S\,V^T \tag{5}$$

where U is the $(M \times R)$ matrix of left singular vectors u_i $(1 \le i \le M)$, S is the $(R \times R)$ diagonal matrix of singular values, V is the $(N \times R)$ matrix of right singular vectors v_j $(1 \le j \le N)$, $R \ll M (\ll N)$ is the order of the decomposition, and superscript T denotes matrix transposition. The role of the SVD, intrinsically, is to establish a one-to-one mapping between words/documents and left/right singular vectors. The left singular vectors represent the words in the given vocabulary, and the right singular vectors represent the documents in the given corpus. Thus, the (continuous) vector space $\mathcal{S}$ sought is the one spanned by U and V.

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 the LSA space $\mathcal{S}$.

C. LSA Language Modeling

The third step is to leverage this property for language modeling purposes. Let w_q denote the word about to be predicted, and H_{q-1} the admissible LSA history (context) for this particular word, i.e., the current document up to word w_{q-1}, denoted by $\tilde{d}_{q-1}$. Then the associated LSA language model probability is given by

$$\Pr(w_q|H_{q-1}, \mathcal{S}) = \Pr(w_q|\tilde{d}_{q-1}) \tag{6}$$

where the conditioning on $\mathcal{S}$ reflects the fact that the probability depends on the particular vector space arising from the SVD representation.

The context $\tilde{d}_{q-1}$ can be thought of as an additional column of the matrix W, and therefore has a representation in the space $\mathcal{S}$ given by

$$\tilde{v}_{q-1} = \tilde{d}_{q-1}^T U S^{-1} \tag{7}$$

after some straightforward algebraic manipulation of (5). This vector representation for $\tilde{d}_{q-1}$ is adequate under some consistency conditions on the general patterns present in the domain considered; see [5] for a complete discussion.

Intuitively, $\Pr(w_q|\tilde{d}_{q-1})$ 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*"). This behavior is exactly the opposite of that observed with the conventional n-gram formalism, which assigns higher probabilities to (frequent) function words than to (rarer) content words. Hence, the attractive synergy potential between the two paradigms.

D. Integration with N-grams

Finally, the fourth step is to exploit this potential by integrating the two together. This integration can occur in a number of ways, such as simple interpolation, or within the maximum entropy framework [4]. Alternatively, if we denote by $\overline{H}_{q-1}$ the overall available history (comprising an n-gram component as well as the LSA component mentioned above), then a suitable expression for the integrated probability is given by [5]

$$\Pr(w_q|\overline{H}_{q-1}) = \frac{\Pr(w_q|w_{q-1}w_{q-2}\cdots w_{q-n+1})\Pr(\tilde{d}_{q-1}|w_q)}{\sum_{w_i\in\mathcal{V}} \Pr(w_i|w_{q-1}w_{q-2}\cdots w_{q-n+1})\Pr(\tilde{d}_{q-1}|w_i)}. \tag{8}$$

Note that, if $\Pr(\tilde{d}_{q-1}|w_q)$ is viewed as a prior probability on the current document history, then (8) simply translates the classical Bayesian estimation of the n-gram (local) probability using a prior distribution obtained from (global) LSA. The end result, in effect, is a modified n-gram language model incorporating large-span semantic information.

In practice, expressions like (8) are often slightly modified so that a relative weight can be placed on each contribution (here, the n-gram and LSA probabilities). Usually, this is done via empirically determined weighting coefficients. In the present case, such weighting is motivated by the fact that the "prior" probability $\Pr(\tilde{d}_{q-1}|w_q)$ could change substantially as the current document unfolds. Thus, rather than using arbitrary weights, an alternative solution is to dynamically tailor the document history $\tilde{d}_{q-1}$ so that the n-gram and LSA contributions remain empirically balanced. We refer to this procedure as context scope selection, whose details are discussed in Section III-C.

E. Clustering

Before addressing implementation details, however, let us briefly review how to exploit the above framework to generate additional families of multispan language models. Because the LSA space $\mathcal{S}$ is a continuous vector space, it is easy to perform clustering of words and/or documents in $\mathcal{S}$. The nice thing about such clustering is that, fundamentally, it takes the global context into account, as opposed to conventional n-gram-based clustering methods which only consider collocational effects. This in turn results in a number of smoothing benefits (cf. [5], [15]).

To illustrate, assume that a set of word clusters C_k, $1 \le k \le K$, has been produced in $\mathcal{S}$, for example through a combination of K-means and bottom-up clustering [20]. Then

$$\Pr(w_q|\tilde{d}_{q-1}) = \sum_{k=1}^{K} \Pr(w_q|C_k)\Pr(C_k|\tilde{d}_{q-1}) \tag{9}$$

represents an appropriate expansion of (6), which carries over to (8) in a straighforward manner. Exploiting document clusters instead of word clusters leads to a similar expansion. Finally, an expression analogous to (9) can also be derived to take advantage of both word and document clusters. Associated with these different families are various tradeoffs discussed in detail in [5].

III. Implementation Issues

This section addresses the computational complexity of the n-gram + LSA approach, as well as three implementation issues of particular interest: 1) how to efficiently integrate the hybrid n-gram + LSA language model into the search; 2) how to dynamically perform adequate context scope selection; and 3) how to initialize a suitable representation of this context.

A. Computational Effort

Of particular concern here is the—on-line—cost of computing the hybrid probability (8), assuming the LSA space $\mathcal{S}$ is already in place. (For a discussion of the—off-line—cost of deriving $\mathcal{S}$, see [5].) Disregarding any clustering for simplicity, this online cost has three components: 1) the construction of the pseudo-document representation in $\mathcal{S}$, as done via (7); 2) the computation of the LSA probability $\Pr(w_q|\tilde{d}_{q-1})$ in (6); and 3) the integration proper, in (8). For the proposed paradigm to be useful, all of this ultimately must be done in real-time.

Clearly, the cost of constructing the pseudo-document representation in $\mathcal{S}$ depends on the number of nonzero entries in $\tilde{d}_{q-1}$. Let us denote by μ_{q-1} the fraction of the total vocabulary size M associated with $\tilde{d}_{q-1}$ at instant q. This fraction is guaranteed to increase monotonically with q. In fact, μ_{q-1} could potentially span the entire [0, 1] range, depending on the underlying vocabulary as well as the characteristics of the document

currently being created. On the other hand, the typical density of W, defined as the ratio of the number of nonzero entries over MN, is about 0.25% (cf. [17]). So, on the average, μ_{q-1} would be expected to hover around that value.

Assuming that the $(M \times R)$ matrix US^{-1} is precomputed, the cost of (7) in floating-point operations (flops) is seen to be $(2\mu_{q-1}M-1)R$ flops per pseudo-document. Similarly, the cost of computing $\Pr(w_q|\tilde{d}_{q-1})$ can be shown to be $(2R-1)R$ flops per word [5]. As for (8), the normalizing factor is needed when computing perplexity numbers, but can be ignored when deriving pseudo-likelihood scores. This yields a cost of just one additional multiplication for the integration of LSA into the n-gram formalism. The total cost to compute $\Pr(w_q|\overline{H}_{q-1})$, per word and pseudo-document, is thus obtained as

$$\mathcal{N}_{tot} = 2(\mu_{q-1}M + R - 1)R + 1 = \mathcal{O}(MR). \tag{10}$$

For sufficient values of q, this is guaranteed to be dominated by the pseudo-document calculation.

B. Search Integration

There are two ways to take advantage of multispan modeling for large vocabulary speech recognition. One is to rescore previously produced N-best lists using the integrated models. (This was the scenario implicitly assumed in [15] and [5].) The other is to use the multispan models directly in the search itself. The latter is preferable, since it allows incremental pruning based on the best knowledge source available.

Compared to N-best rescoring, however, integrating multispan modeling into the search entails a much higher computational cost, because of the large number of partial hypothesis paths to score. The problem is not so much in the computation of the LSA probabilities (6), which can be classically alleviated through appropriate thresholding and caching. More troublesome is the calculation of each pseudo-document vector representation in (7), which, as just shown, requires $\mathcal{O}(MR)$ flops.

As it turns out, this cost can be reduced by exploiting the sequential nature of pseudo-documents. Clearly, as each active theory is expanded, the associated document context remains largely unchanged, with only the most recent candidate word added. Assume further that the training corpus $\mathcal{T}$ is large enough, so that the normalized entropy ε_i $(1 \leq i \leq M)$ does not change appreciably with the addition of each pseudo-document. Then it is possible to express the new pseudo-document vector directly in terms of the old pseudo-document vector, instead of each time recomputing the entire mapping from scratch.

To see that, consider $\tilde{d}_q$, and assume, without loss of generality, that word w_i is observed at time q. Then, from (3), we will have, for $k = i$:

$$w_{i,q} = (1-\varepsilon_i)\frac{c_{i,q-1}+1}{n_q} = \frac{n_q - 1}{n_q} w_{i,q-1} + \frac{1-\varepsilon_i}{n_q} \tag{11}$$

while, for $1 \leq k \leq M$, $k \neq i$

$$w_{k,q} = \frac{n_q - 1}{n_q} w_{k,q-1}. \tag{12}$$

Hence, we can express $\tilde{d}_q$ as

$$\tilde{d}_q = \frac{n_q - 1}{n_q}\tilde{d}_{q-1} + [0 \cdots 0 \ \frac{1-\varepsilon_i}{n_q} \ 0 \cdots 0]^T \tag{13}$$

which in turn implies, from (7)

$$\tilde{v}_q = \frac{n_q - 1}{n_q}\tilde{v}_{q-1} + \frac{1-\varepsilon_i}{n_q} u_i S^{-1}. \tag{14}$$

It is easily verified that (14) requires only $5R + 1 = \mathcal{O}(R)$ floating point operations. Thus, we can update the pseudo-document vector directly in the LSA space at a fraction of the cost previously required to map the sparse representation to the space $\mathcal{S}$. With this strategy, the total cost of the hybrid n-gram + LSA language model, in terms of computing $\Pr(w_q|\overline{H}_{q-1})$, becomes

$$\mathcal{N}_{tot} = 2(R+1)^2 = \mathcal{O}(R^2). \tag{15}$$

For typical values of R, this amounts to less than 0.05 Mflops. While this is definitely more expensive than the usual table look-up required in conventional n-gram language modeling, the total cost (15) arguably represents a relatively modest overhead. This allows multispan language modeling to be taken advantage of in early stages of the search.

C. Context Scope Selection

Another major implementation issue has to do with the dynamic selection of the context scope. During training, this scope is fixed to be the current document. During recognition, however, the concept of "current document" is ill-defined, because 1) its length grows with each new word, and 2) it is not necessarily clear at which point completion occurs. As a result, a decision has to be made regarding what to consider "current," versus what to consider part of an earlier (presumably less relevant) document.

The simplest solution is to postulate that all utterances spoken since the beginning of the session are part of the current document. This is adequate only if the user starts a new session each time she/he wants to work on a new document. (Again, this was the scenario implicitly assumed in [15] and [5].) If, however, the user needs to dictate in a heterogeneous manner, this solution might fail, because the (single, cumulative) pseudo-document built under this assumption might not be sufficiently representative of each individual topic. Note, from (14), that this approach corresponds to the following closed form expression for $\tilde{v}_q$:

$$\tilde{v}_q = \frac{1}{n_q}\sum_{p=1}^{q}(1-\varepsilon_{i_p})u_{i_p}S^{-1} \tag{16}$$

where i_p is the index of the word observed at time p, and the initial pseudo-document vector is taken to be identically zero.

An alternative solution is to limit the size of the history considered, so as to avoid relying on old, possibly obsolete fragments to construct the current context. The size limit could be expressed in anything from words to paragraphs. If, for example, only the last P words are assumed to belong to the current document, this approach corresponds to computing the latest pseudo-document vector using a truncated version of (16), namely

$$\tilde{v}_q = \frac{1}{P} \sum_{p=q-P+1}^{q} (1 - \varepsilon_{i_p}) u_{i_p} S^{-1}. \tag{17}$$

The problem here is the difficulty of determining the constant P *a priori*, since it is highly dependent on the kind of documents spoken by the user. Also note that this requires a slight modification to (14), so that the oldest factor in the summation (i.e., associated with time $q - P + 1$) is properly subtracted when incrementing q.

It is also possible to adopt an intermediate solution, which does not require a hard decision to be made on the size of the caching window. This solution uses exponential forgetting to progressively discount older utterances. Assuming $0 < \lambda \leq 1$, this approach corresponds to the closed form solution given by

$$\tilde{v}_q = \frac{1}{n_q} \sum_{p=1}^{q} \lambda^{(n_q - n_p)} (1 - \varepsilon_{i_p}) u_{i_p} S^{-1} \tag{18}$$

where the parameter λ is chosen according to the expected heterogeneity of the session. As before, this requires a slight modification to (14), so that the first term in the right hand side is now multiplied by λ. In addition, it is straightforward, if desired, to concurrently place a hard limit on the size of the history, in the same vein as (17).

D. Initialization

What remains to be specified is how to initialize the pseudo-document vector $\tilde{v}_0$. One possibility is to take $\tilde{v}_0$ to be identically zero, as in the previous discussion. At the other extreme, we could initialize it to be the centroid vector of all training documents. Or, alternatively, $\tilde{v}_0$ could be set to the centroid of a specific region of the LSA space, if some information is available regarding the expected subdomain of the session.

Clearly, this decision does not make a great deal of difference when forgetting is used, due to data discounting (in the case of exponential forgetting), or elimination (in the case of a rectangular window). It is only relevant when a homogeneous session is expected, in which case it makes most sense to initialize the pseudo-document as close as possible to the main topic of the session.

IV. Recognition Results

With the basic implementation framework in place, one can now proceed with actual recognition experiments. Following [5], we have trained the LSA component on the WSJ0 part of the NAB corpus. This was convenient for comparison purposes since conventional n-gram language models are readily available, trained on exactly the same data [11].

A. Experimental Conditions

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. The vocabulary $\mathcal{V}$ was constructed by taking the 20 000 most frequent words of the NAB corpus, augmented by some words from an earlier release of the WSJ corpus, for a total of $M = 23\,000$ words.

We performed the singular value decomposition of the matrix of co-occurrences between words and documents using the single vector Lanczos method [18]. Over the course of this decomposition, we experimented with different numbers of singular values retained, and found that $R = 125$ seemed to achieve an adequate balance between reconstruction error (as measured by Frobenius norm differences) and noise suppression (as measured by trace ratios). This led to a vector space $\mathcal{S}$ of dimension 125, which we used to construct the direct LSA model (6).

Drawing on the results of [15], we then applied the combination of K-means and bottom-up clustering described in [5] to derive $K = 100$ word clusters in $\mathcal{S}$. This enabled us to construct the word-clustered LSA model (9). Finally, using (8), we combined each of these models with the standard bigram, as well as the word-clustered model with the standard trigram.

The resulting multispan language models, dubbed (direct or word-clustered) bi-LSA and tri-LSA models, respectively, were then used in lieu of the standard WSJ0 bigram and trigram models in a series of speaker-independent, continuous speech recognition experiments, detailed below. These experiments were conducted on a subset of the WSJ 20 000 word-vocabulary task. The acoustic training corpus consisted of 7200 sentences of data uttered by 84 different native speakers of English (WSJ0 SI-84). The test corpus consisted of 496 sentences uttered by 12 additional native speakers of English.

All experiments relied on the same set of continuous parameter hidden Markov models with tied mixture diagonal Gaussian output distributions (see, e.g., [19]). Since the focus was on measuring language modeling improvements, we selected a fairly basic set up for acoustic modeling. We used decision trees (cf., e.g., [20]) to cluster the observed triphones into 2000 allophones. Each allophone was then assigned a mixture of distributions from a total of about 20 000 distributions tied at the state cluster level. Training was carried out on a speaker-independent basis using maximum likelihood estimation. No speaker adaptation was performed.

The recognition system used a two-pass decoding strategy [21], with a Viterbi beam search in the forward pass and an A^* stack search in the backward pass. In the forward pass, scores for optimal partial paths from the beginning node to each within-beam language model node were stored at each frame. These scores were then used as the heuristics in evaluating incomplete paths in the backward pass. On the test data considered, this system produced a baseline error rate of 16.7% across the 12 speakers, using the standard bigram language model (the corresponding perplexity was 215).

TABLE I
PERPLEXITY AND WORD ERROR RATE REDUCTION USING Bi-LSA LANGUAGE MODELING

Speaker	Reduction in Perplexity	Reduction in Word Error Rate
001	22.8 %	8.4 %
002	28.5 %	21.5 %
00a	30.6 %	17.5 %
00b	27.4 %	10.1 %
00c	33.6 %	10.0 %
00d	26.2 %	17.3 %
00f	33.3 %	11.5 %
203	35.3 %	16.1 %
400	15.4 %	14.8 %
430	19.7 %	19.3 %
431	20.0 %	12.2 %
432	24.7 %	7.8 %
Overall	24.7 %	13.7 %

TABLE II
WORD ERROR RATE REDUCTION (WER) USING N-LSA LANGUAGE MODELING WITH WORD CLUSTERING IN LSA COMPONENT

Speaker	WER Reduction, Bi-LSA Model	WER Reduction, Tri-LSA Model
001	11.2 %	8.8 %
002	35.0 %	24.6 %
00a	25.9 %	19.1 %
00b	7.8 %	8.5 %
00c	17.6 %	12.9 %
00d	35.4 %	22.4 %
00f	16.9 %	12.5 %
203	34.2 %	21.2 %
400	19.8 %	15.5 %
430	20.2 %	18.1 %
431	18.3 %	13.4 %
432	27.9 %	14.3 %
Overall	22.5 %	15.8 %

B. Error Rate versus Perplexity

It is important to note that the task chosen represents a severe test of the LSA component of the multispan language model. By design, the test corpus is constructed with no more than three or four consecutive sentences extracted from a single article. Overall, it comprises 140 distinct document fragments, which means that each speaker speaks, on the average, about 12 different "mini-documents." As a result, the context effectively changes every 60 words or so, which makes it somewhat challenging to build a very accurate pseudo-document representation. This is a situation where it is critical for the multispan model to appropriately forget the context as it unfolds, to avoid relying on an obsolete representation. Throughout, we used the exponential forgetting approach described in the last section, with a value $\lambda = 0.975$. (For the sake of illustration, this means that the word which occurred 60 words ago is discounted through a weight of about 0.2.)

Table I summarizes the performance achieved using the (direct) bi-LSA language model, as compared with that achieved using the baseline bigram. The comparison is made in terms of both reduction in test data perplexity (first column) and reduction in actual word error rate (second column). It can be seen that all speakers substantially benefit from multispan modeling, displaying a reduction in perplexity ranging from about 15% to more than 35%, and a reduction in word error rate ranging from about 8% to 21.5%. Overall, we observed a perplexity reduction of about 25%, and an average word error rate reduction on the order of 15%.

As usual, the reduction in average error rate is less than the corresponding reduction in perplexity, due to the influence of the acoustic component in actual recognition, and the resulting "ripple effect" of each recognition error. In the case of n-LSA language modeling, this effect can be expected to be more pronounced than in the standard n-gram case. This is because recognition errors are potentially able to affect the LSA context well into the future, through the estimation of a flawed representation of the pseudo-document in the LSA space. This lingering behavior, which can obviously reduce the effectiveness of the LSA component, is a direct by-product of large-span modeling. Clearly, the more accurate the recognition system, the less problematic this unsupervised context construction becomes.

In terms of CPU performance, we observed an increase in decoding time of about 30% when using the bi-LSA language model, as compared to the decoding time obtained when using the conventional bigram. This, of course, can be traced to the overhead calculated in (15). For our recognition system, this translates into a CPU load roughly comparable to that of a conventional trigram.

C. Tri-LSA versus Bi-LSA Modeling

To illustrate the performance improvement achievable through clustering, we then repeated the experiments corresponding to the last column of Table I, but this time expanding (6) using (9), i.e., using the word-clustered bi-LSA model. The results are reported in the first column of Table II.With clustering, all speakers again show marked improvement, with a reduction in word error rate ranging from around 8% to more than 35%. The reduction in average error rate increases to 22.5%. Comments similar to those made regarding Table I apply here as well.

To assess whether the LSA component still helps to the same extent when a larger order n-gram is used, we also combined the word-clustered LSA model with the standard trigram, and measured the performance of the resulting (word-clustered) tri-LSA model against the baseline trigram performance. The results are reported in the second column of Table II.

The qualitative behavior of the two n-LSA language models appears to be quite similar. Quantitatively, the average reduction achieved by tri-LSA is about 30% less than that achieved by bi-LSA. This is most likely related to the greater predictive power of the trigram compared to the bigram, which makes the LSA contribution of the hybrid language model comparatively smaller. (Interestingly, this contribution seems to vary substantially from speaker to speaker, reflecting the varying role played

TABLE III
INFLUENCE OF CONTEXT SCOPE SELECTION ON WORD ERROR RATE REDUCTION, THROUGH DIFFERENT VALUES OF EXPONENTIAL FORGETTING FACTOR λ

Speaker	$\lambda = 1.0$	$\lambda = 0.99$	$\lambda = 0.98$	$\lambda = 0.97$	$\lambda = 0.96$	$\lambda = 0.95$
001	7.7 %	11.9 %	11.2 %	4.9 %	−2.1 %	−3.5 %
002	27.7 %	33.3 %	33.9 %	35.0 %	37.9 %	36.2 %
00a	15.7 %	25.2 %	21.2 %	25.9 %	23.0 %	20.8 %
00b	8.2 %	9.7 %	7.8 %	9.7 %	7.8 %	7.8 %
00c	10.3 %	12.9 %	17.6 %	16.5 %	16.5 %	16.2 %
00d	16.1 %	27.8 %	33.6 %	35.4 %	39.2 %	33.0 %
00f	10.7 %	11.1 %	15.3 %	16.9 %	16.5 %	16.9 %
203	15.4 %	21.5 %	32.2 %	34.2 %	33.6 %	28.9 %
400	15.9 %	17.0 %	18.1 %	19.8 %	19.2 %	16.5 %
430	12.6 %	19.3 %	20.2 %	17.6 %	14.3 %	10.9 %
431	8.9 %	15.0 %	18.3 %	18.3 %	17.8 %	13.6 %
432	11.2 %	16.2 %	23.5 %	27.9 %	27.9 %	26.3 %
Overall	13.2 %	18.4 %	21.1 %	21.9 %	21.6 %	19.3 %

by global constraints from one set of spoken utterances to another.) For the sake of simplicity, we will adopt the bi-LSA framework for the remainder of this paper.

D. Context Scope Selection

One way to measure the influence of context scope selection is to vary the value of the parameter λ in the exponential forgetting framework. Recall from Section III-C that the value $\lambda = 1$ corresponds to an unbounded context (as would be appropriate for a very homogeneous session), while decreasing values of λ correspond to increasingly more restrictive contexts (as required for a more heterogeneous session). Said another way, the gap between λ and 1 tracks the expected heterogeneity of the current session.

Table III presents recognition results for values of λ ranging from $\lambda = 1$ to $\lambda = 0.95$, in decrements of 0.01. In all cases we considered the same word-clustered bi-LSA framework as just used above, so the results can be compared to those of the first column of Table II (where, as mentioned before, $\lambda = 0.975$). It can be seen that, with no forgetting, the overall performance is substantially less than the comparable one observed in Table II (approximately 13% compared to 22.5% reduction in word error rate). This is consistent with the characteristics of the task, and underscores the role of discounting as a suitable counterbalance to frequent context changes.

Performance rapidly improves as λ decreases from $\lambda = 1$ to $\lambda = 0.97$, presumably because the pseudo-document representation gets less and less contaminated with obsolete data. If forgetting becomes too aggressive, however, the performance starts degrading, as the effective context no longer has an equivalent length which is sufficient for the task at hand. In the present case, this happens for $\lambda < 0.97$. Not surprisingly, this degradation is more or less severe depending on the actual article fragments uttered. For example, speaker 00b seems to be considerably less affected than, say, speaker 001.

V. INHERENT TRADEOFFS

In the previous section, the LSA component of the multi-span language model was trained on exactly the same data as its n-gram component. This is not a requirement, however, which raises the question of how critical the selection of the LSA training data is to the performance of the recognizer. This is particularly interesting since LSA is known to be weaker on heterogeneous corpora (cf., e.g., [13]).

A. Cross-Domain Training

To ascertain the matter, we went back to the original expression (8) with the direct model (6), so the results could be compared to those of Table I. We kept the same underlying vocabulary $\mathcal{V}$, left the bigram component unchanged, and repeated the LSA training on non-WSJ data from the same general period. Three corpora of increasing size were considered, all corresponding to Associated Press (AP) data:

1) $\mathcal{T}_1$, composed of $N_1 = 84,000$ documents from 1989, comprising approximately 44 million words;
2) $\mathcal{T}_2$, composed of $N_2 = 155\,000$ documents from 1988 and 1989, comprising approximately 80 million words;
3) $\mathcal{T}_3$, composed of $N_3 = 224\,000$ documents from 1988–1990, comprising approximately 117 million words.

In each case we proceeded with the LSA training as described in Section II. The resulting word error rate reductions are reported in Table IV.

Two things are immediately apparent. First, the performance improvement in all cases is much smaller than in Table I. Larger training set sizes notwithstanding, on the average the multispan model trained on AP data is about four times less effective than that trained on WSJ data. This suggests a relatively high sensitivity of the LSA component to the domain considered. To put this observation into perspective, recall that: 1) by definition, content words are what characterize a domain; and 2) LSA inherently relies on content words, since, in contrast with n-grams, it cannot take advantage of the structural aspects of the sentence. It therefore makes sense to expect a higher sensitivity for the LSA component than for the usual n-gram.

Second, the overall performance does not improve appreciably with more training data, a fact already observed in [5] using a perplexity measure. This supports the conjecture that, no matter the amount of data involved, LSA still detects a

TABLE IV
MULTISPAN SENSITIVITY TO LSA TRAINING DATA: CROSS-DOMAIN STUDY

Speaker	AP 84 K Docs	AP 155 K Docs	AP 224 K Docs
001	0.0 %	6.3 %	7.0 %
002	0.0 %	4.0 %	5.1 %
00a	8.4 %	9.5 %	11.3 %
00b	−3.1 %	−3.1 %	−3.1 %
00c	2.1 %	2.0 %	2.4 %
00d	2.6 %	2.4 %	2.9 %
00f	2.7 %	2.7 %	3.8 %
203	3.4 %	3.1 %	4.7 %
400	7.1 %	7.3 %	7.1 %
430	5.0 %	3.4 %	0.0 %
431	−0.5 %	4.2 %	3.3 %
432	1.7 %	2.2 %	4.5 %
Overall	2.4 %	3.3 %	4.0 %

substantial mismatch between AP and WSJ data from the same general period. This, in turn, suggests that the LSA component is sensitive not just to the general training domain, but also to the particular style of composition, as might be reflected, for example, in the choice of content words and/or word co-occurrences. On the positive side, this bodes well for rapid adaptation to cross-domain data, provided a suitable adaptation framework can be derived.

B. Within-Domain Targeted Training

Knowing the results obtained using out-of-domain training data, it is tempting to go the other way and investigate the performance that can be achieved using perfectly within-domain training data. In addition, this might be useful to establish an upper bound on multispan performance. So, we opted to retrain the LSA parameters on just the test set, which we refer to as targeted training. We therefore defined a (much smaller) corpus $\mathcal{T}_4$, composed of only the $N_4 = 140$ test documents. This corpus comprised approximately 8500 words, which effectively reduced the vocabulary $\mathcal{V}$ to about 2500 words. We then repeated the above experiments, again with the bigram component left unchanged. The resulting error rate reductions are presented in the first column of Table V, labeled "targeted direct model." As before, this label refers to the expression (8) with the direct model (6).

A couple of points can be made. First, there is a limit to the performance that can be gained by applying LSA constraints. With the direct model (6), this limit is seen to be around 17%. However, this improvement may not be indicative of the best possible achievable with the multispan language model, due again to the atypical document fragmentation existing in the test data. Second, this overall performance improvement is only about 25% better than that observed in Table I (13.7%). This may, in part, be due to the importance of composition style mentioned earlier. Indeed, targeted data may not offer much value-add if we presume that "style" can be appropriately captured using general data *from the same source* in the same domain. This, in turn, suggests that within-domain adaptation may not generally be compelling.

TABLE V
MULTISPAN SENSITIVITY TO LSA TRAINING DATA: WITHIN-DOMAIN TARGETED STUDY

Speaker	Targeted Direct Model	Targeted Clustered Model
001	13.3 %	18.9 %
002	28.2 %	50.3 %
00a	21.5 %	40.9 %
00b	15.2 %	26.1 %
00c	14.1 %	27.4 %
00d	15.5 %	43.3 %
00f	18.0 %	28.4 %
203	17.4 %	40.9 %
400	14.8 %	29.1 %
430	23.5 %	31.1 %
431	17.4 %	30.0 %
432	10.6 %	34.1 %
Overall	17.1 %	33.6 %

Finally, to gauge the effect of clustering with such a narrow training set, we repeated the experiments once more with the word-clustered model (9), using the same clustering set up as in Section IV. We postulated that most clusters would be sharply defined, given the relatively small amount of training data. The resulting error rate reductions are presented in the last column of Table V, labeled "targeted clustered model." The overall performance improvement (33.6%) is seen to be almost 50% better than the comparable one observed in the first column of Table II (22.5%). We believe, however, that this is partly a consequence of the artificially limited task at hand. In a way, it simply translates the power of clustering when clear-cut regions of the LSA space can be isolated.

C. Discussion

The results so far suggest that the hybrid n-gram + LSA approach studied in this paper is a promising avenue for multispan language modeling. Clearly, one has to be cognizant of some of the limitations of the method, as evidenced by the sensitivity to LSA training data demonstrated above, as well as the earlier discussion on context scope selection. As already mentioned, these limitations can be mitigated through careful attention to the expected domain of use, and a judicious choice of the exponential forgetting factor λ.

There is another limitation, however, which has only been alluded to briefly in Section V-A. We pointed out earlier that LSA is inherently more adept at handling content words than function words. But, as is well-known, a substantial proportion of speech recognition errors come from function words, because of their tendency to be shorter, not well articulated, and acoustically confusable. In general, the LSA component will not be able to help fix such errors. Thus, the benefits of the multispan approach will not extend to this particular class of large vocabulary recognition errors.

VI. CONCLUSION

We have investigated the behavior of multispan (hybrid n-gram + LSA) language models, constructed by embedding

latent semantic analysis into the standard n-gram formulation, in actual recognition experiments. When compared to the associated standard n-gram on a subset of the WSJ large vocabulary task, the multispan approach resulted in a substantial improvement in performance, as measured by both perplexity and average word error rate

Compared to the standard bigram, the bi-LSA language model achieved a reduction in perplexity of about 25%, and a reduction in average word error rate of about 15%. The latter figure improved to 22.5% after performing word clustering in the LSA space. Compared to the standard trigram, the word-clustered tri-LSA language model achieved a reduction in average word error rate of about 16%.

The experimental task chosen showed marked document fragmentation, which underscored the importance of dynamic context scope selection. We have experimented with different parameters within an exponential forgetting framework, and found that appropriate discounting of obsolete data could make a substantial difference when several "mini-documents" were uttered in quick succession. This is likely to have practical implications in product implementations incorporating the kind of multispan language modeling proposed in this paper.

We have also looked at the influence of the LSA training data on the resulting performance. The multispan approach showed much more sensitivity to the training domain than to the size of the training data. This suggests that cross-domain adaptation has greater potential than within-domain adaptation for adaptive multispan language modeling. Future efforts will concentrate on the derivation of an adaptation framework suitable for this purpose.

ACKNOWLEDGMENT

The author would like to thank the associate editor, Dr. J. Glass, Massachusetts Institute of Technology, for many helpful suggestions that contributed to a much improved manuscript, and to the anonymous reviewers for their constructive feedback and insightful comments.

REFERENCES

[1] T. Niesler and P. Woodland, "A variable-length category-based N-gram language model," in *Proc. 1996 Int. Conf. Acoustics, Speech, and Signal Processing*, Atlanta, GA, May 1996, pp. I164–I167.

[2] R. Rosenfeld, "The CMU statistical language modeing toolkit and its use in the 1994 ARPA CSR evaluation," in *Proc. ARPA Speech and Natural Language Workshop*, Mar. 1994.

[3] R. Lau, R. Rosenfeld, and S. Roukos, "Trigger-based language models: A maximum entropy approach," in *Proc. 1993 Int. Conf. Acoustics, Speech, and Signal Processing*, Minneapolis, MN, May 1993, pp. II45–II48.

[4] R. Rosenfeld, "A maximum entropy approach to adaptive statistical language modeling," in *Computer Speech and Language*. New York: Academic, July 1996, vol. 10, pp. 187–228.

[5] J. R. Bellegarda, "A multi-span language modeling framework for large vocabulary speech recognition," in *IEEE Trans. Speech Audio Processing*, vol. 6, Sept. 1998, pp. 456–467.

[6] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Inform. Sci.*, vol. 41, pp. 391–407, 1990.

[7] P. W. Foltz and S. T. Dumais, "Personalized information delivery: An analysis of information filtering methods," *Commun. ACM*, vol. 35, pp. 51–60, 1992.

[8] 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.

[9] 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.

[10] 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, pp. 211–240, 1997.

[11] 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.

[12] J. R. Bellegarda *et al.*, "A novel word clustering algorithm based on latent semantic analysis," in *Proc. 1996 Int. Conf. Acoustics, Speech, and Signal Processing*, Atlanta, GA, May 1996, pp. I172–I175.

[13] Y. Gotoh and S. Renals, "Document space models using latent semantic analysis," in *Proc. EuroSpeech'97*, Rhodes, Greece, Sept. 1997, pp. 1433–1448.

[14] J. R. Bellegarda, "A latent semantic analysis for large-span language modeling," in *Proc. EuroSpeech'97*, Rhodes, Greece, Sept. 1997, pp. 1451–1454.

[15] ——, "Exploiting both local and global constraints for multi-span statistical language modeling," in *Proc. 1998 Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, Seattle, WA, May 1998, pp. 677–680.

[16] 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. Sci. Soc.*, 1998.

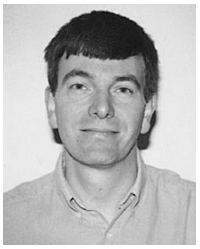
[17] S. T. Dumais, "Latent semantic indexing (LSI) and TREC-2," in *Proc. 2nd Text Retrieval Conf.(TREC-2)*, D. Harman, Ed., 1994, pp. 105–116.

[18] M. W. Berry, "Large-scale sparse singular value computations," *Int. J. Supercomput. Appl.*, vol. 6, pp. 13–49, 1992.

[19] J. R. Bellegarda and D. Nahamoo, "Tied mixture continuous parameter modeling for speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 2033–2045, Dec. 1990.

[20] J. R. Bellegarda, "Context-dependent vector clustering for speech recognition," in *Speech and Speaker Recognition: Advanced Topics*, C.-H. Lee, F. K. Soong, and K. K. Paliwal, Eds. Boston, MA: Kluwer, 1996, ch. 6, pp. 133–157.

[21] R. Schwartz, L. Nguyen, and J. Makhoul, "Multiple-pass search strategy," in *Automatic Speech and Speaker Recognition: Advanced Topics*, C.-H. Lee, F. K. Soong, and K. K. Paliwal, Eds. Boston, MA: Kluwer , 1996, ch. 18, pp. 429–456.



Jerome R. Bellegarda (M'87–SM'98) received the Diplôme 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 speech and handwriting recognition, particularly acoustic and chirographic modeling. 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, advanced dialog interactions, and voice authentication. His research interests include voice-driven man-machine communications, multiple input/output modalities, and multimedia knowledge management. He has written over 70 journal and conference papers, and holds 13 patents. He has also contributed chapters to several edited books, including *Advances in Handwriting and Drawing: A Multidisciplinary Approach* (Paris, France: Europia, 1994), *Automatic Speech and Speaker Recognition: Advanced Topics* (Boston, MA: Kluwer, 1996), and *Robustness in Language and Speech Technology* (Dordrecht, The Netherlands: Kluwer, in press).

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