Computational Linguistics and Chinese Language Processing
Vol. 9, No. 2 , August 2004, pp. 29-44

Latent Semantic Language Modeling and Smoothing

Jen-Tzung Chien*, Meng-Sung Wu* and Hua-Jui Peng*

Abstract

Language modeling plays a critical role for automatic speech recognition. Typically, the *n*-gram language models suffer from the lack of a good representation of historical words and an inability to estimate unseen parameters due to insufficient training data. In this study, we explore the application of latent semantic information (LSI) to language modeling and parameter smoothing. Our approach adopts latent semantic analysis to transform all words and documents into a common semantic space. The word-to-word, word-to-document and document-to-document relations are, accordingly, exploited for language modeling and smoothing. For language modeling, we present a new representation of historical words based on retrieval of the most relevant document. We also develop a novel parameter smoothing method, where the language models of seen and unseen words are estimated by interpolating the k nearest seen words in the training corpus. The interpolation coefficients are determined according to the closeness of words in the semantic space. As shown by experiments, the proposed modeling and smoothing methods can significantly reduce the perplexity of language models with moderate computational cost.

Keywords: language modeling, parameter smoothing, speech recognition, and latent semantic analysis.

1. Introduction

Language models have been successfully developed for speech recognition, optical character recognition, machine translation, information retrieval, etc. Many studies in the field of speech recognition have focused on this topic [Jelinek 1990, Jelinek 1991]. As shown in Figure 1, a speech recognition system is composed of syllable-level and word-level matching processes, in which the acoustic model l and language model t are applied, respectively. In theory, the speech recognition procedure combines the acoustic model and language model according to the Bayes rule. Let O denote the acoustic data, and let $W = \{w_1, \mathbf{L}, w_l\} = w_1^l$ denote a string of l

* Department of Computer Science and Information Engineering , National Cheng Kung University, Tainan, Taiwan, ROC
E-mail: jtchien@mail.ncku.edu.tw

words. The speech recognition task aims to find the most likely word string $\hat{W}$ by maximizing the *a posteriori* probability given the observed acoustic data *O*:

$$\hat{W} = \arg\max_{W} P(W|O) = \arg\max_{W} P_l(O|W)P_t(W), \tag{1}$$

where $P_t(W)$ is the *a priori* probability of the occurring word string *W*, and $P_l(O|W)$ is the probability of observing data *O* given the word string *W*. The parameters t and l are the language model and speech hidden Markov models (HMM's), respectively. Hereafter, we will neglect the notation t in $P_t(W)$. The language model $\Pr(W)$ aims to measure the probability of word occurrence. This model is employed to predict the word occurrence given the history words. In an *n*-gram model, we assume that the probability of a word depends only on the preceding *n*-1 words. The *N*-gram model $\Pr(W)$ is written as

$$\Pr(W) = \Pr(w_1,..., w_l) = \prod_{q=1}^{l} \Pr(w_q|w_1, w_2,..., w_{q-1}) \cong \prod_{q=1}^{l} \Pr(w_q|w_{q-n+1}^{q-1}). \tag{2}$$

The sequence $H_q = \{w_1, \mathbf{L}, w_{q-1}\}$ is referred to as the *history* H_q for word w_q. To estimate $\Pr(w_q|w_{q-n+1}^{q-1})$, we can count the number of words w_q following the history words w_{q-n+1}^{q-1} and divide it by the total number of occurring history words w_{q-n+1}^{q-1}, i.e.,

$$\Pr(w_q|w_{q-n+1}^{q-1}) = \frac{c(w_{q-n+1}^{q})}{\sum_{w_i} c(w_{q-n+1}^{q})}. \tag{3}$$

This probability estimation is called the *maximum likelihood estimation* (MLE). The bigram model $\Pr(W) \cong \prod_{q=1}^{l} \Pr(w_q|w_{q-1})$ and trigram model $\Pr(W) \cong \prod_{q=1}^{l} \Pr(w_q|w_{q-2}, w_{q-1})$ are employed in most speech recognition systems. However, when a word sequence (w_{q-2}, w_{q-1}, w_q) is not occurs in the training data, the trigram model $\Pr(w_q|w_{q-2}, w_{q-1})$ could not be estimated. We may apply parameter smoothing to find the unseen trigram model. In the literature, several smoothing methods have been proposed to deal with the data sparseness problem [Katz 1987, Kawabata and Tamoto 1996, Lau *et al*. 1993, Zhai and Lafferty 2001]. Also, *maximum a posteriori* adaptation of the language model has been presented to resolve the problem of domain mismatch between training and test corpora [Bellegarda 2000a, Federico 1996, Masataki *et al*. 1997]. Besides the problems of data sparseness and domain mismatch, the *n*-gram model is inferior in terms of characterizing long-distance word relationships. For example, the trigram model is unable to characterize word dependence beyond the span of three successive words. In [Lau *et al*. 1993, Zhou and Lua 1999], the trigram model was improved by extracting word relationships from the document history. This approach was exploited to search the trigger pair, $w_A \rightarrow w_B$, where the appearance of w_A in the document history significantly affects the probability of occurring w_B. The trigger pairs provide long distance information because the triggering and triggered words might be separated by several words. However, trigger pair selection neglects the possibility of

low-frequency word triggers, which might contain useful semantic information. The LSA method was developed to resolve this problem.

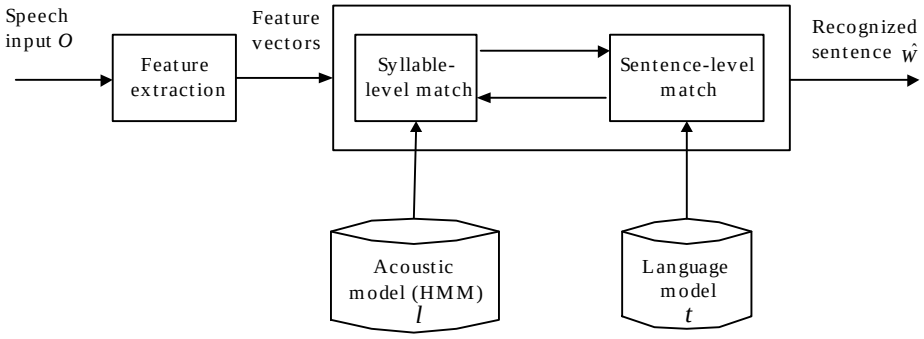

Figure 1. A schematic diagram of a speech recognition system.

In this paper, a new language modeling and smoothing method is proposed based on the framework of latent semantic analysis (LSA). The traditional ***n***-gram model is weak in terms of characterizing the information in historical words. This weakness is compensated for herein by using the LSA framework, where word-to-word, word-to-document and document-to-document similarities are found in the semantic space. With the use of LSA, all the words are mapped to a common semantic space, which is constructed via the ***singular value decomposition*** (SVD) of a word-by-document matrix. Bellegarda [1998, 2000a, 2000b] applied the LSA framework to the ***n***-gram model such that the resulting word error rate and perplexity were substantially reduced. The LSA representation of the history suffers from a drawback in that the representation of the history carries insufficient information at the beginning of a text document. To overcome this problem, we propose a relevance retrieval framework to represent the history. For language model smoothing, we estimate unseen language models by using the seen models corresponding to the ***k*** nearest neighbor words. Because this smoothing method extracts synonym and semantic information, it can be also referred to as "semantic smoothing." In the following section, we briefly introduce the framework of LSA. Section 3 addresses the proposed language modeling and smoothing approaches. The LSA framework is applied to relevance feedback language modeling and k nearest neighbor language smoothing. Section 4 describes the experimental setup and reports the results for the perplexity and computational cost. Finally, we draw conclusions in Section 5.

2. Latent semantic analysis

In the literature [Berry *et al.* 1995, Deerwester *et al.* 1990, Ricardo and Berthier 2000], latent semantic analysis (LSA) has been widely applied to vector space based information retrieval. During the past few years, LSA has also been applied to language model adaptation [Bellegarda 1998, Bellegarda 2000a, Novak and Mammone 2001]. Latent semantic analysis is a dimension reduction technique that projects the query and document into a common semantic space [Deerwester *et al.* 1990, Ding 1999]. This projection reduces the document vector from a high dimensional space to a low dimensional space, which is referred as the latent semantic space. The goal is to represent similar documents as close points in the latent semantic space, based on an appropriate metric. This metric can capture the significant associations between words and documents. Given an $M \times N$ matrix **A**, with M terms and N documents, $M \geq N$ and rank (**A**) = R. The weighted count $a_{i,j}$ of matrix **A** is the number of occurrences of each word w_i in a document d_j, calculated as follows:

$$a_{i,j} = (1 - e_i)\frac{c_{i,j}}{n_j}. \tag{4}$$

Here, $c_{i,j}$ is the number of terms w_i occurring in document d_j, n_j is the total number of words in d_j, and e_i is the normalized entropy of w_i in the collection of data consisting of N documents, i.e.,

$$e_i = -\frac{1}{\log N}\sum_{j=1}^{N}\frac{c_{i,j}}{t_i}\log\frac{c_{i,j}}{t_i}, \tag{5}$$

where $t_i = \sum_j c_{i.j}$ is the total number of times w_i occurs in the collection of data. A value of e_i that is close to one occurs in case of $c_{i,j} = t_i / N$. This means that the word w_i is distributed across many documents throughout the corpus. A value of e_i that is close to zero, i.e., the case in which $c_{i,j} = t_i$, indicates that the word w_i is present in only a few documents. Hence, in (4), $1 - e_i$ represents a global indexing weight for the word w_i, and $c_{i,j}/n_j$ indicates that the word w_i occurs in frequently in document d_j.

Latent semantic analysis is a conceptual-indexing method, which uses singular value decomposition (SVD) [Berry *et al.* 1995, Golub and Van Loan 1989] to find the latent semantic structure of word to document association. SVD decomposes the matrix **A** into three sub-matrices:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T, \tag{6}$$

where **U** and **V** are orthogonal matrices, $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}_R$, and Σ is a diagonal matrix. As shown in Figure 2, the first R columns of **U** and **V,** and the first R diagonal elements of Σ can be used to approach **A** with $\text{rank}(\mathbf{A}) = R$ by means of $\mathbf{A}_R = \mathbf{U}_R\Sigma_R\mathbf{V}_R^T$, where $\mathbf{A}_R$ is a representative matrix **A**. The result of SVD is a set of vectors representing the location of each term and document in the reduced R-dimensional LSA space [Berry 1992]. For a given training

corpus, $\mathbf{AA}^T$ characterizes all the co-occurrences between words, and $\mathbf{A}^T\mathbf{A}$ characterizes all the co-occurrences between documents. That is, a similar pattern of occurring words w_i and w_j can be inferred from the (i, j) cell of $\mathbf{AA}^T$, and a similar pattern of words contained in documents d_i and d_j can be inferred from the (i, j) cell of $\mathbf{A}^T\mathbf{A}$ [Bellegarda 1998, Bellegarda 1997, Bellegarda 2000a, Chen and Goodman 1999]. This LSA approach performs well when a major portion of the meaningful semantic structure [Deerwester *et al.* 1990] is captured.

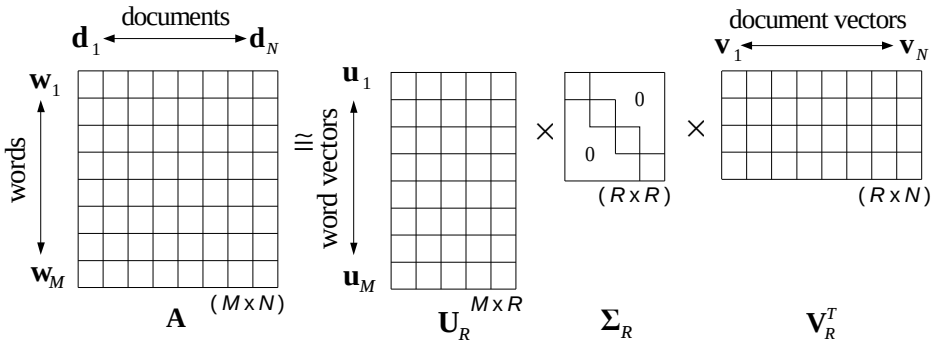

Figure 2. A diagram of the truncated SVD.

3. New language modeling and smoothing techniques

3.1 LSA Parameter Modeling

N-gram language models are useful for modeling the local dependencies of word occurrences but not for capturing global word dependencies. The modeling process leads to the estimation of the conditional probability $\Pr(w_q | w_{q-n+1}^{q-1})$, which characterizes the linguistic regularity in a span of *n* words. When the window size *n* is limited, the *n*-gram is weak in terms of capturing long distance dependencies. Long distance correlation between words is commonly found in language and is caused by closeness in meaning; e.g., the words "stock" and "fund" are both likely to occur in financial news. To deal with long distance modeling, the LSA approach can be applied to extract large span semantic knowledge. Our motivation lies in the fact that there exists some latent structure in the occurrence patterns of words across documents. Hence, the *n*-gram language model can be improved by employing LSA to perform large span prediction of word occurrence.

Let the word w_q denote the predicted word, let H_{q-1} denote the history for w_q, and let $\Pr(w_q | H_{q-1})$ be the associated language model probability. Using the *n*-gram language model, we find that $H_{q-1} = \{w_{q-1}, w_{q-2}, \mathbf{L}, w_{q-n+1}\}$ is the relevant history composed of the preceding *n*-1 words. The LSA language model is expressed by

$$\Pr(w_q|H_{q-1}) = \Pr(w_q|H_{q-1}, S) = \Pr(w_q|\mathbf{d}_{q-1}), \tag{7}$$

where the conditioning on S reflects the fact that the probability depends on the particular vector space arising from the SVD representation, and where $\Pr(w_q|\mathbf{d}_{q-1})$ is computed directly based on the closeness of w_q and $\mathbf{d}_{q-1}$ in the semantic space S. The vector $\mathbf{d}_{q-1}$ can be viewed as an additional *pseudodocument* vector for matrix $\mathbf{A}$ [Bellegarda 1998, Bellegarda 2000a, Bellegarda 2000b]. The representation $\mathbf{v}_{q-1}$ for the pseudodocument vector $\mathbf{d}_{q-1}$ in the space S is given by

$$\mathbf{v}_{q-1} = \mathbf{d}_{q-1}^T \mathbf{U}\Sigma^{-1}. \tag{8}$$

By referring to (4), we can obtain the pseudodocument vector $\mathbf{d}_q$ recursively in the LSA space via [Bellegarda 2000a, Bellegarda 2000b]

$$\mathbf{d}_q = \frac{n_q - 1}{n_q}\mathbf{d}_{q-1} + [0\,\mathbf{L}\,0\,\frac{1-e_q}{n_q}\,0\,\mathbf{L}\,0]^T. \tag{9}$$

To clarify (8) and (9), we provide their derivations in the Appendix.

However, at the beginning of a text document, it is difficult to capture long distance word dependencies for calculating $\Pr(w_q|\mathbf{d}_{q-1})$ due to the shortness of the history H_{q-1}. To overcome this weakness, we present here a new method for estimating the pseudodocument vector $\mathbf{d}_{q-1}$. *Our method aims to retrieve the most likely relevance document* $\hat{\mathbf{d}}_{q-1}$ *from the training documents* $\mathbf{d}_1, \mathbf{L}, \mathbf{d}_N$ *so as to represent the pseudodocument vector* $\mathbf{d}_{q-1}$. The LSA probability $\Pr(w_q|\mathbf{d}_{q-1})$ is replaced by $\Pr(w_q|\hat{\mathbf{d}}_{q-1})$. Accordingly, the pseudodocument $\hat{\mathbf{d}}_{q-1}$ is estimated by

$$\hat{\mathbf{d}}_{q-1} = \arg\max_{\mathbf{d}_i} \Pr(\mathbf{d}_i|\mathbf{d}_{q-1}), \qquad i = 1, \mathbf{L}, N. \tag{10}$$

Here, $\mathbf{d}_{q-1}$ is obtained recursively from (9). The probability $\Pr(\mathbf{d}_i|\mathbf{d}_{q-1})$ is determined by finding the cosine of the angle between the vectors $\mathbf{d}_i$ and $\mathbf{d}_{q-1}$ in the latent semantic space; i.e., by using the vectors $\mathbf{v}_i\Sigma$ and $\mathbf{v}_{q-1}\Sigma$ in

$$\Pr(\mathbf{d}_i|\mathbf{d}_{q-1}) = \cos(\mathbf{v}_i\Sigma, \mathbf{v}_{q-1}\Sigma) = \frac{\mathbf{v}_i\Sigma^2\mathbf{v}_{q-1}^T}{\|\mathbf{v}_i\Sigma\|\,\|\mathbf{v}_{q-1}\Sigma\|}. \tag{11}$$

When q is increased, the most likely document vector $\hat{\mathbf{d}}_{q-1}$ moves around in the LSA space. Assuming that $\hat{\mathbf{d}}_{q-1}$ is semantically homogeneous, we can expect the resulting trajectory to eventually settle down in the vicinity of the document cluster corresponding to the closest semantic content.

In this study, the LSA language model is exploited by integrating the effects of histories obtained from the conventional n-gram component $H_{q-1}^{(n)} = \{w_{q-1}, w_{q-2}, \mathbf{L}, w_{q-n+1}\}$ and the LSA component $H_{q-1}^{(l)} = \hat{\mathbf{d}}_{q-1}$ [Bellegarda 1998, Bellegarda 2000a]. The new language model

is written as

$$\begin{aligned}\Pr(w_q|H_{q-1}) &= \Pr(w_q|H_{q-1}^{(n)}, H_{q-1}^{(l)}) = \frac{\Pr(w_q, H_{q-1}^{(l)}|H_{q-1}^{(n)})}{\sum_{w_i}\Pr(w_i, H_{q-1}^{(l)}|H_{q-1}^{(n)})} \\ &= \frac{\Pr(w_q|H_{q-1}^{(n)})\Pr(H_{q-1}^{(l)}|w_q, H_{q-1}^{(n)})}{\sum_{w_i}\Pr(w_i|H_{q-1}^{(n)})\Pr(H_{q-1}^{(l)}|w_i, H_{q-1}^{(n)})} \\ &= \frac{\Pr(w_q|w_{q-1}, w_{q-1}, \mathbf{L}, w_{q-n+1})\Pr(\hat{\mathbf{d}}_{q-1}|w_q)}{\sum_{w_i}\Pr(w_i|w_{q-1}, w_{q-1}, \mathbf{L}, w_{q-n+1})\Pr(\hat{\mathbf{d}}_{q-1}|w_i)} \\ &= \frac{\Pr(w_q|w_{q-1}, w_{q-1}, \mathbf{L}, w_{q-n+1})\dfrac{\Pr(w_q|\hat{\mathbf{d}}_{q-1})}{\Pr(w_q)}}{\sum_{w_i}\Pr(w_i|w_{q-1}, w_{q-1}, \mathbf{L}, w_{q-n+1})\dfrac{\Pr(w_i|\hat{\mathbf{d}}_{q-1})}{\Pr(w_i)}}.\end{aligned} \tag{12}$$

In (12), we assume that $\Pr(H_{q-1}^{(l)}|w_q, H_{q-1}^{(n)}) = \Pr(\hat{\mathbf{d}}_{q-1}|w_q)$. The probability $\Pr(w_q|\hat{\mathbf{d}}_{q-1})$ is computed based on the representations of w_q and $\hat{\mathbf{d}}_{q-1}$ in the semantic space *S*, which are provided by $\mathbf{u}_q\Sigma^{1/2}$ and $\hat{\mathbf{v}}_{q-1}\Sigma^{1/2}$, respectively. The LSA probability is calculated as follows:

$$\Pr(w_q|\hat{\mathbf{d}}_{q-1}) = \cos(\mathbf{u}_q\Sigma^{1/2}, \hat{\mathbf{v}}_{q-1}\Sigma^{1/2}) = \frac{\mathbf{u}_q\Sigma\hat{\mathbf{v}}_{q-1}^T}{\left\|\mathbf{u}_q\Sigma^{1/2}\right\|\left\|\hat{\mathbf{v}}_{q-1}\Sigma^{1/2}\right\|}. \tag{13}$$

3.2 LSA Parameter Smoothing

In the real world, a training corpus is not sufficient to estimate the *n*-gram model for all word occurrences $\{w_{q-n+1}, \mathbf{L}, w_{q-1}, w_q\}$. To overcome the problem of insufficient data, the parameter smoothing method can be used to estimate the joint probabilities of unseen word occurrences and, simultaneously, smooth those of seen word occurrences in the training corpus. It is common to interpolate the *n*-gram and (*n*-1)-gram for the purpose of language model smoothing. Jelinek-Mercer smoothing [Jelinek and Mercer 1980] is represented as follows:

$$\Pr_{JM}(w_q|w_{q-n+1}^{q-1}) = l_q\Pr(w_q|w_{q-n+1}^{q-1}) + (1-l_q)\Pr_{JM}(w_q|w_{q-n+2}^{q-1}). \tag{14}$$

The smoothed *n*-gram model $\Pr_{JM}(w_q|w_{q-n+1}^{q-1})$ is defined recursively as a linear interpolation between the maximum likelihood *n*-gram model $\Pr(w_q|w_{q-n+1}^{q-1})$ and the smoothed (*n*-1)-gram model $\Pr_{JM}(w_q|w_{q-n+2}^{q-1})$. This smoothing process is intended to flatten the probability distribution. Let N_q denote the number of occurrences for word w_q preceding w_{q-n+1}^{q-1}:

$$N_q = \left|\left\{w_q : c(w_{q-n+1}^{q-1}w_q) > 0\right\}\right|. \tag{15}$$

The well-known Witten-Bell smoothing approach [Written and Bell 1991] incorporates the

interpolation coefficient

$$1-l_q=\frac{N_q}{N_q+\sum_{w_q}c(w_{q-n+1}^q)}, \tag{16}$$

into (14) of Jelinek-Mercer smoothing to generate

$$\Pr_{WB}(w_q|w_{q-n+1}^{q-1})=\frac{\sum_{w_q}c(w_{q-n+1}^q)+N_q\Pr_{WB}(w_q|w_{q-n+2}^{q-1})}{N_q+\sum_{w_q}c(w_{q-n+1}^q)}. \tag{17}$$

In this paper, we will present a novel smoothing method in which the language models of seen and unseen word occurrences are estimated by interpolating the LSA language model of a word occurrence and of the k nearest word occurrences. Let us consider the words "car," "automobile," "driver," and "elephant". "Car" and "automobile" are synonyms. "Driver" is related and "elephant" is unrelated to "car" and "automobile." If the words "car" and "automobile" do not appear in the given documents, we may collect many documents containing related words, e.g., the motor, vehicle, engine, etc. The statistics of these nearest seen words can be used to estimate the language model of the unseen words. When the bigram model is used, the smoothed model $\tilde{\Pr}(w_q|w_{q-1})$ is estimated by interpolating the LSA bigram $\Pr(w_q|w_{q-1})$ of the word pair occurrence (w_q,w_{q-1}) and those of the other k occurrences $(w_q,\hat{w}_j^q)$, $1\le j\le k$, where the k nearest words $\hat{w}_j^q$ to word w_q are determined according to the LSA probabilities:

$$\Pr(w_q|w_j)=\cos(\mathbf{u}_q\Sigma,\mathbf{u}_j\Sigma)=\frac{\mathbf{u}_q\Sigma^2\mathbf{u}_j^T}{\|\mathbf{u}_q\Sigma\|\|\mathbf{u}_j\Sigma\|}, \qquad 1\le j\le M. \tag{18}$$

The interpolation is performed as follows:

$$\tilde{\Pr}(w_q|w_{q-1})=a_q\Pr(w_q|w_{q-1})+(1-a_q)\sum_{j=1}^{k}b_j^q\Pr(w_q|\hat{w}_j^q), \tag{19}$$

where the weighting coefficients $\{b_j^q,1\le j\le k\}$ and the interpolation coefficient a_q are estimated by

$$b_j^q=\frac{\Pr(w_q|\hat{w}_j^q)}{\sum_{j=1}^{k}\Pr(w_q|\hat{w}_j^q)} \tag{20}$$

and

$$a_q=\frac{\Pr(w_q|w_{q-1})}{\Pr(w_q|w_{q-1})+\sum_{j=1}^{k}b_j^q\Pr(w_q|\hat{w}_j^q)}, \tag{21}$$

respectively. As seen in (20), the weighting coefficient b_j^q is proportional to the LSA probability of the word pair $(w_q,\hat{w}_j^q)$ and has the property $\sum_{j=1}^{k}b_j^q=1$. That is, the closer

the word $\hat{w}_j^q$ is to the current word w_q, the higher is the weighting coefficient that b_j^q produces. Also, it is reasonable to adopt the interpolation coefficient a_q in (21), which is proportional to the closeness between w_q and w_{q-1} in the semantic space. The smoothing method proposed in (19) should be performed when the current word w_q is trained using LSA. Different from the Jelinek-Mercer and Witten-Bell smoothing methods that adopt the maximum likelihood language model, the proposed smoothing technique is combined with the LSA framework, and the probabilities $\Pr(w_q|w_{q-1})$ and $\Pr(w_q|\hat{w}_j^q)$ are computed via the LSA procedure.

4. Experiments

We evaluated the performance of the proposed language model through experiments. Two databases were employed. The first database was the *CKIP* balanced corpus of Modern Chinese (http://godel.iis.sinica.edu.tw), which was collected by Academia Sinica in Taiwan, ROC. Totally, this database has twenty-five million Chinese characters and a vocabulary size of 80,000 words. In addition, we collected 9,372 news documents during 2001 and 2002 from the news websites of CNA (http://www.cna.com.tw), ChinaTimes (http://news.chinatimes.com) and UDNnews (http://www.udnnews.com.tw). We randomly sampled 9,148 documents for training and the remaining 224 documents for testing. The news documents were divided into eight categories, including technology, society, international, leisure, politics, finance, entertainment, and sports news. The numbers of training and testing documents in the eight news categories are listed in Table 1. We chose the most frequent 32,941 words to construct our dictionary. Using the LSA procedure, we built a 32,941*9,148 word by document matrix $\mathbf{A}$ using training data. The SVD algorithm was applied with different numbers of singular values. In this study, we used MATLAB for the SVD operation and compared the performance of LSA language modeling, with the number of singular values R set at 25, 50, 75, and 100.

The measure of perplexity was adopted to evaluate the different language models. The computational costs were reported for comparison. Here, the computation time was measured in minutes by testing 224 documents using a personal computer with a Pentium IV-1.6GHz processor and 256 MB RAM. The bigram model was employed in the experiments.

Table 1. Numbers of training and testing documents for the eight news categories.

	Technology	Social	International	Leisure	Politics	Financial	Entertain	Sports
Training data	289	2,658	330	1,106	1,299	2,605	430	431
Testing data	30	24	23	24	24	49	23	27

4.1 Perplexity

Perplexity is an important parameter used to evaluate the performance of language models. Consider an information source containing of word sequence, $w_1, w_2, \mathbf{K}, w_l$, each of which is chosen from a vocabulary V. The entropy of a source emitting the words $w_1, w_2, \mathbf{K}, w_l$ is defined as

$$E = -\lim_{l\to\infty}\frac{1}{l}\sum_{w_1 w_2 \mathbf{K} w_l}\Pr(w_1, w_2, \mathbf{K}, w_l)\cdot\log\Pr(w_1, w_2, \mathbf{K}, w_l)\cdot \tag{22}$$

If the source is ergodic, the entropy in (22) is equivalent to

$$E = -\lim_{l\to\infty}\frac{1}{l}\log\Pr(w_1, w_2, \mathbf{K}, w_l)\cdot \tag{23}$$

Since the *n*-gram language model is used, *E* can be estimated as follows:

$$\tilde{E} = -\frac{1}{l}\sum_{q=1}^{l}\log\Pr(w_q\,|\,w_{q-n+1}^{q-1})\cdot \tag{24}$$

Given testing documents with *l* words, the perplexity is calculated as follows:

$$\textit{Perplexity} = 2^{\tilde{E}}. \tag{25}$$

In general, the entropy $\tilde{E}$ is the average difficulty or uncertainty of each word using the language model. The lower measured the perplexity, the better the speech recognition accuracy that can be achieved.

4.2 Evaluation of Different Language Modeling and Smoothing Methods

In the experiments, we evaluated different language modeling and smoothing methods in terms of perplexity and computation time. First of all, we investigated the effect of the SVD dimension in the proposed LSA bigram model. No parameter smoothing was performed. In Figures 3 and 4, we compare the perplexity and computation time for different SVD dimensions. Here, the computation time was a measure of the SVD operation of a 32,941*9,148 word by document matrix $\mathbf{A}$. We found that an SVD dimension of 25 was appropriate for constructing the semantic space. In the subsequent evaluation, the SVD dimension was fixed at 25 for the proposed LSA bigram and LSA smoothing. Next, we examined the effect of the parameter k in the proposed LSA smoothing method. LSA smoothing of seen and unseen bigrams was performed by combining the bigrams corresponding to the k nearest words. In Figure 5, we show the results for perplexity versus the *k* nearest neighbor words when LSA smoothing was applied to the standard bigram and proposed LSA bigram. The values $k = 5$, 10, 30 and 50 were examined. When proposed LSA modeling and smoothing was used, the lowest perplexity of 81 was achieved by using $k = 5$. The perplexity of the standard bigram with LSA smoothing was calculated as 102. We then fixed $k = 5$ in the subsequent comparison experiment.

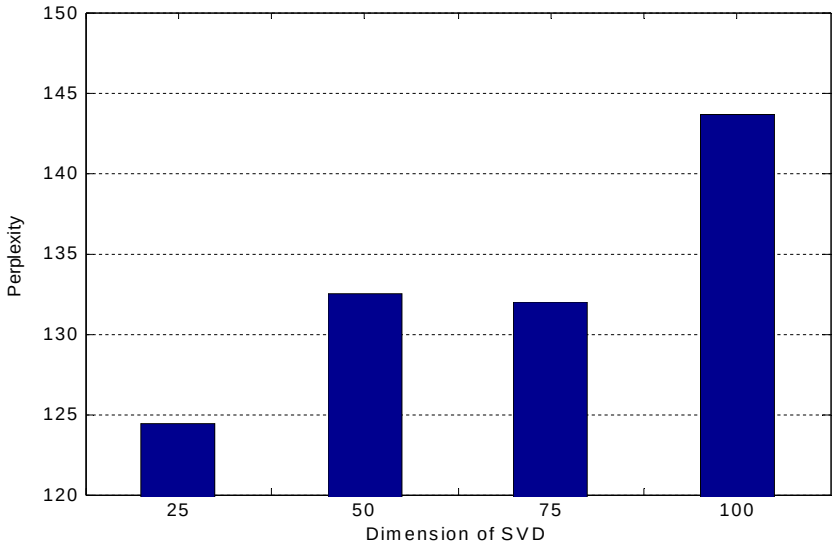

Figure 3. Comparison of perplexity results for different SVD dimensions.

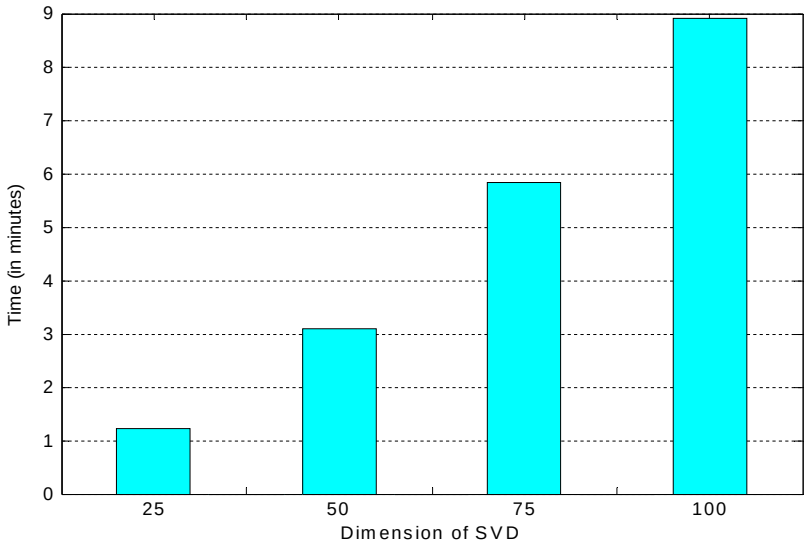

Figure 4. Comparison of computation times for different SVD dimensions.

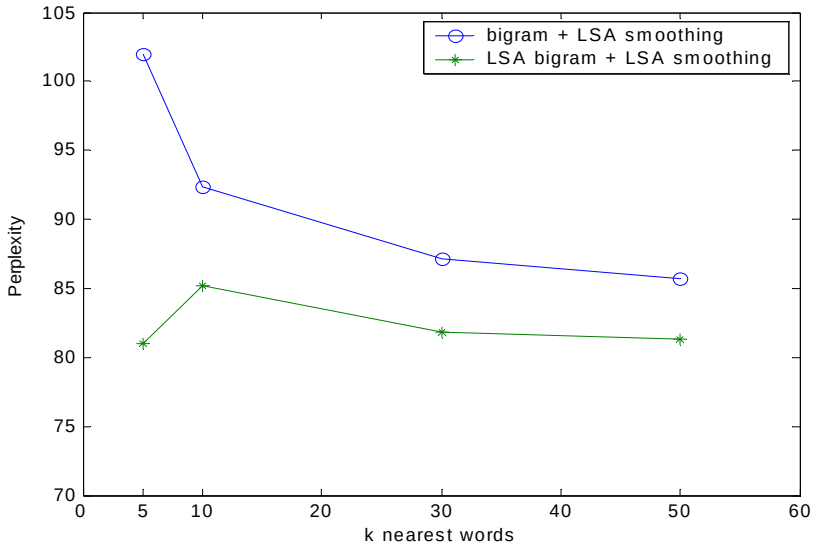

Figure 5. Perplexity versus the k nearest neighbor words when LSA smoothing was applied to the standard bigram and proposed LSA bigram.

Furthermore, different language modeling and smoothing methods were compared, and the results are shown in Table 2. Besides the standard bigram, we implemented Bellegarda's LSA bigram [Bellegarda 1998] and the proposed LSA bigram to evaluate the effect of language modeling. The main difference is that proposed LSA bigram aims to retrieve the most likely relevance document vector in order to represent the historical words. In addition, the language models with and without parameter smoothing were examined. The algorithms of Witten-Bell smoothing and the proposed LSA smoothing were also used for the purpose of comparison. The Witten-Bell smoothed bigram is estimated by interpolating with the corresponding unigram. The proposed LSA smoothing combines the bigrams corresponding to the k nearest seen words in the training corpus. We can see that the baseline bigram model has a perplexity of 158.3. The perplexity was reduced to 128.7 and 124.4 by applying Bellegarda's LSA bigram and proposed LSA bigram, respectively. However, when Witten-Bell smoothing was incorporated, the perplexity is greatly reduced from 158.3 without smoothing to 122.6 with smoothing. When the proposed LSA bigram with Witten-Bell smoothing were used, the perplexity could be improved to 108.7. This indicates the importance of adopting a smoothing algorithm in the language model. Furthermore, when the proposed LSA smoothing was used, the perplexity was reduced to 102, which is better than the perplexity of 122.6 obtained using Witten-Bell smoothing. This is because the Witten-Bell smoothing method estimates the n-gram model by using the (n-1)-gram, while the proposed LSA smoothing approach always adopts nearest n-gram models

without using the (*n*-1)-gram. Among the different combinations, the lowest perplexity of 81 was achieved by applying the proposed LSA bigram with LSA smoothing. Compared to baseline system, the perplexity could be improved by up to 48.8%. The computation times of the different methods were also compared. The results show that the computation overhead of using a smoothing algorithm is slight. The computation load of the LSA bigram is much higher than that of the standard bigram. This result indicates that the smoothing algorithm can lead to greater improvement in perplexity with a lower computation cost than can be achieved by modifying the language model.

Table 2. Comparison of perplexity and computation time for different language modeling and smoothing methods.

Language Model		Perplexity	Reduction Rate (%)	Computation Time (minutes)
Modeling Method	*Smoothing Method*			
Bigram	N/A	158.3	N/A	48.3
Bigram	Witten-Bell Smoothing	122.6	22.6	51.3
Bellegarda's LSA Bigram	N/A	128.7	18.7	176.7
Proposed LSA Bigram	N/A	124.4	21.4	161.2
Proposed LSA Bigram	Witten-Bell Smoothing	108.7	31.3	163.3
Bigram	LSA Smoothing	102	35.6	52.2
Proposed LSA Bigram	LSA Smoothing	81	48.8	163.4

5. Conclusion

Statistical *n*-gram modeling is limited in terms of its ability to represent the historical words and estimate the unseen parameters of an inadequate training corpus. In this paper, we have presented new language modeling and smoothing methods that are based on the framework of latent semantic analysis. The concept of relevance retrieval has been adopted in order to exploit a new language modeling approach, where the most likely pseudodocument is retrieved to represent the historical words. The language model is estimated according to the closeness of the current word vector and the historical pseudodocument vector in the common LSA space. To overcome the problem of insufficient training data, we perform LSA smoothing, where the bigram of the current word is computed by interpolating with the bigrams corresponding to the k nearest words. The weighting coefficients of the k nearest words are proportional to the

closeness to the current word in the LSA space. From the results of experiments in which Chinese news documents were evaluated, we found that the language modeling performance could be greatly improved by applying the proposed LSA parameter modeling and smoothing algorithms. The proposed methods outperformed Bellegarda's LSA bigram and Witten-Bell smoothing. Compared to the baseline bigram model, the perplexity was reduced by up to 48.8%. Also, the perplexity improvement and computation efficiency that could be achieved through parameter smoothing were better than that which could be achieved through parameter modeling. This approach can be easily extended to the trigram model and other languages. In the future, we will explore theoretical rules for determining the SVD dimension for LSA. We will also investigate the effect of the amount of training data on the LSA framework. We are currently applying the proposed language model to information retrieval and large vocabulary continuous speech recognition.

Appendix

Derivations of Equations (8) and (9)

In (8), the pseudodocument vector $\mathbf{d}_{q-1}$ is the (q-1)th column vector of matrix $\mathbf{A}$. From SVD, we know $\mathbf{d}_{q-1} = \mathbf{U}\Sigma\mathbf{v}_{q-1}^T$. Because $\mathbf{U}$ is orthogonal and Σ is diagonal, the representation $\mathbf{v}_{q-1}$ in semantic space S is obtained by

$$\mathbf{d}_{q-1} = \mathbf{U}\Sigma\mathbf{v}_{q-1}^T \Rightarrow \mathbf{v}_{q-1}^T = \Sigma^{-1}\mathbf{U}^T\mathbf{d}_{q-1} \Rightarrow \mathbf{v}_{q-1} = (\Sigma^{-1}\mathbf{U}^T\mathbf{d}_{q-1})^T = \mathbf{d}_{q-1}^T\mathbf{U}\Sigma^{-1}. \tag{26}$$

Also, from (4), we can derive the recursive formula for $a_{i,q}$ corresponding to word w_i and document d_q

$$\begin{aligned} a_{i,q} &= (1-e_i)\frac{c_{i,q}}{n_q} = (1-e_i)\frac{c_{i,q-1}+1}{n_q} = (1-e_i)\frac{c_{i,q-1}}{n_q} + \frac{1-e_i}{n_q} \\ &= (1-e_i)\frac{c_{i,q-1}}{n_{q-1}}\cdot\frac{n_{q-1}}{n_q} + \frac{1-e_i}{n_q} = a_{i,q-1}\cdot\frac{n_{q-1}}{n_q} + \frac{1-e_i}{n_q} \end{aligned}. \tag{27}$$

By extending this formula using vector representation, we obtain (9) by

$$\mathbf{d}_q = \frac{n_q-1}{n_q}\mathbf{d}_{q-1} + \frac{1-e_q}{n_q}\cdot[0\,\mathbf{L}\,0\,1\,0\,\mathbf{L}\,0]^T = \frac{n_q-1}{n_q}\mathbf{d}_{q-1} + [0\,\mathbf{L}\,0\,\frac{1-e_q}{n_q}\,0\,\mathbf{L}\,0]^T, \tag{28}$$

where the "1" appears at coordinate i in the above vector.

Acknowledgment

The authors thank the anonymous reviewers for providing valuable comments, which considerably improved the quality of this paper.

References

Bellegarda, J. R., "A Multi-span Language Modeling Framework for Large Vocabulary Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, 6(5) 1998, pp. 456-467.

Bellegarda, J. R., "A statistical language modeling approach integrating local and global constraints," *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, 1997, pp. 262-269.

Bellegarda, J. R., "Exploiting latent semantic information in statistical language modeling," *Proceeding of IEEE*, 88(8) 2000a, pp. 1279-1296.

Bellegarda, J. R., "Large vocabulary speech recognition with multi-span statistical language models," *IEEE Transactions on Speech and Audio Processing*, 8(1) 2000b, pp. 76-84.

Berry, M. W., S. T. Dumais and G. W. O'Brien, "Using Linear algebra for Intelligent Information Retrieval," *Society for Industrial and Applied Mathematics (SIAM): Review*, 37(4) 1995, pp. 573-595.

Berry, M. W., "Large scale singular value computations," *International Journal of Supercomputer Applications*, vol. 6, 1992, pp. 13-49.

Chen, S. and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," *Computer Speech and Language*, 13(4) 1999, pp. 359-394.

Deerwester, S., S. T. Dumais, T. K. Landauer, G. W. Furnas and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, 41(6) 1990, pp. 391-407.

Ding, C. H. Q., "A similarity-based probability model for latent semantic indexing," *Proc. 22nd Annual International ACM SIGIR Conference*, 1999, pp. 58-65.

Federico, M., "Bayesian estimation methods for n-gram language model adaptation," *Proc. of the International Conference on Spoken Language Processing*, vol. 1, 1996, pp. 240-243.

Golub, G. and C. Van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD: Johns Hopkins, 1989.

Jelinek, F., "Self-Organized Language Modeling for Speech Recognition," *Readings in Speech Recognition*, Morgan-Kaufmann Publishers, 1990, pp. 450-506.

Jelinek, F., "Up From Trigrams," *Proc. European Conference on Speech communication and Technology*, vol. 3, 1991, pp. 1037-1040.

Jelinek, F. and R. Mercer, "Interpolated estimation of Markov source parameters from sparse data," *Pattern Recognition in Practice*, 1980, pp. 381-397.

Katz, S.M., "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3) 1987, pp. 400-401.

Kawabata, T. and M. Tamoto, "Back-off Method for N-gram Smoothing based on Binomial Posteriori Distribution," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, 1996, pp.192-195.

Lau, R., R. Rosenfeld and S. Roukos, "Trigger-based language models: A maximum entropy approach," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, 1993, pp. 45-48.

Masataki, H., Y. Sagisaka, K. Hisaki and T. Kawahara, "Task adaptation using MAP estimation in *n*-gram language modeling," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, 1997, pp.783-786.

Novak, M. and R. Mammone, "Use of non-negative matrix factorization for language model adaptation in a lecture transcription task," *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, 2001, pp. 541-544.

Ricardo, B.-Y. and R. -N. Berthier, *Modern information retrieval*, Addison-Wesley, 2000.

Written, I. H. and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Transaction on Information Theory*, 37(4) 1991, pp. 1085-1094.

Zhai, C. and J. Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," *Proc. 24th Annual International ACM SIGIR Conference*, 2001, pp. 334-342.

Zhou, G. D. and K. T. Lua, "Interpolation of n-gram and mutual-information based trigger pair language models for Mandarin speech recognition," *Computer Speech and Language*, 13(2) 1999, pp.125-141.