EXPLOITING BOTH LOCAL AND GLOBAL CONSTRAINTS FOR MULTI-SPAN STATISTICAL LANGUAGE MODELING

Jerome R. Bellegarda

Spoken Language Group
Apple Computer, Inc.
Cupertino, California 95014

ABSTRACT

A new framework is proposed to integrate the various constraints, both local and global, that are present in the language. Local constraints are captured via n-gram language modeling, while global constraints are taken into account through the use of latent semantic analysis. An integrative formulation is derived for the combination of these two paradigms, resulting in several families of multi-span language models for large vocabulary speech recognition. Because of the inherent complementarity in the two types of constraints, the performance of the integrated language models, as measured by perplexity, compares favorably with the corresponding n-gram performance.

1. INTRODUCTION

As is well known, the performance of a large vocabulary speech recognition system is heavily influenced by the predictive power of its language modeling component. In the past decade, the n-gram paradigm has steadily grown in popularity, and its various implementations are now applied as a matter of course to discriminate between different strings of n words. Still, it remains extremely challenging to go beyond, say $n \leq 4$, with currently available databases and processing power [1]. This imposes an artificially local horizon to the language model and thereby curtails its contribution to the recognition process. Fundamentally, of all the constraints present in the language, the n-gram approach is able to capture only the local ones.

Taking more global constraints into account has traditionally involved a paradigm shift toward parsing and rule-based grammars, such as are routinely and successfully employed in small vocabulary recognition applications. This approach solves the locality problem, since it typically operates at the level of an entire sentence. Unfortunately, it is not (yet) practical for large vocabulary recognition, which is precisely why the n-gram framework was so widely adopted in the first place. This has motivated further investigation into the extraction of suitable long distance information without resorting to a formal parsing mechanism.

One such attempt was based on the concept of word triggers [2]. Unfortunately, trigger pair selection is a complex issue: different pairs display markedly different behavior, which limits the potential of low frequency triggers [3]. Still, self-triggers seem to be particularly powerful and robust [2], which underscores the desirability of exploiting correlations between the current word and features of the document history.

This observation led the author to explore the use of latent semantic analysis (LSA) for such purpose [4], [5]. In some respect, the LSA paradigm can be viewed as an extension of the word trigger concept, where a more systematic framework is used to handle the trigger pair selection. In [4], LSA was used for word clustering, and in [5], for language modeling. In both cases, it was found to be suitable to capture some of the global constraints present in the language.

This paper, building on the results of [4] and [5], proposes several families of multi-span language models which leverage both the n-gram paradigm and the LSA framework. The paper is organized as follows. In the next section we review the salient properties of LSA-based statistical language modeling. In Section 3, we discuss various smoothing schemes based on clustering in the LSA space, along with the resulting trade-offs in predictive power. Section 4 addresses the integration of this framework with conventional n-gram language modeling. Finally, in Section 5 a series of experimental results illustrates some of the benefits associated with the integrated language models.

2. LSA LANGUAGE MODELING

Let $\mathcal{V}$, $|\mathcal{V}| = M$, be some vocabulary of interest and $\mathcal{T}$ a training text corpus, comprising N articles (documents) from a variety of sources. (Note that this implies that the 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 (NAB) News corpus [6].) 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 sets $\mathcal{V}$, $\mathcal{T}$ and a 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 [4], [5] for further details on the mechanics of LSA, and just briefly summarize here. The first step is the construction of a matrix of co-occurences between words and documents, W. This matrix W is accumulated from the available training data by simply keeping track of which word is found in what document. In marked contrast with n-gram modeling, word order is ignored.

The second step is to compute the singular value decomposition (SVD) of W. 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. Thus, the space $\mathcal{S}$ sought is the one 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 the LSA space $\mathcal{S}$.

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 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:

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

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

In this expression, $\Pr(w_q|\tilde{d}_{q-1})$ reflects the "relevance" of word w_q to the admissible history. 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 perplexity value. Hence, the model (1) will likely exhibit a rather weak predictive power. This provides some motivation for looking at various smoothing possibilities.

3. LSA SMOOTHING

The fairly low dimension of the space $\mathcal{S}$ opens up a variety of clustering opportunities. 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. Thereafter we illustrāte smoothing based mostly on word clustering; see also [7] for an illustration of document clustering.

Since the matrix W embodies, by construction, all structural associations between words and documents, it follows that, for a given training corpus, $W\,W^T$ (where ^{T} denotes matrix transpose) characterizes all co-occurrences between words. Thus, the extent to which words w_i and w_j have a similar pattern of occurrence across the entire set of documents can be inferred from the (i,j) cell of $W\,W^T$. From the SVD formalism, it follows that this can be characterized by taking the dot product between the ith and the jth row of the matrix US, namely u_iS and u_jS [4].

In other words, how "close" u_i is to u_j in the space $\mathcal{S}$ can be characterized by the natural metric:

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

for any $1 \le i,j \le M$. Once this metric is specified, it is straightforward to proceed with the clustering of the vectors u_i using any of a variety of algorithms [8].

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 [4]. The result of this process is a set of word clusters C_k, $1 \le k \le K$. A similar reasoning leads to a set of document clusters D_ℓ, $1 \le \ell \le L$, which independently partitions the space $\mathcal{S}$. These sets embody two knowledge layers on top of the vector space representation derived from LSA. These layers characterize a number of semantically homogeneous regions in the space $\mathcal{S}$, corresponding to sub-vocabularies and sub-topics, respectively.

At this point, we can enhance the language model (1) by taking advantage of either or both of the knowledge layers just uncovered. In that sense, clustering essentially acts as a smoothing mechanism by leveraging a more flexible mixture framework. In the case of word clusters, for example, the right hand side of (1) can be expanded as:

$$\Pr(w_q|\tilde{d}_{q-1}) = \sum_{k=1}^{K} \Pr(w_q|C_k)\Pr(C_k|\tilde{d}_{q-1}). \qquad (3)$$

In the case of document clusters, this becomes:

$$\Pr(w_q|\tilde{d}_{q-1}) = \sum_{\ell=1}^{L} \Pr(w_q|D_\ell)\Pr(D_\ell|\tilde{d}_{q-1}). \qquad (4)$$

Finally, when the two layers are considered simultaneously, we get, after approximation for tractability:

$$\Pr(w_q|\tilde{d}_{q-1}) = \sum_{k=1}^{K}\sum_{\ell=1}^{L} \Pr(w_q|C_k)\Pr(C_k|D_\ell)\Pr(D_\ell|\tilde{d}_{q-1}). \qquad (5)$$

In these expressions, probabilities like $\Pr(w_q|C_k)$ depend on the "closeness" of w_q relative to the centroid of word cluster C_k, and can therefore be obtained with the help of (2). In contrast, probabilities like $\Pr(C_k|\tilde{d}_{q-1})$ are qualitatively similar to the right hand side of (1) and can therefore be obtained as in [5].

The behavior of the model (3) depends on the number of word clusters defined in the space $\mathcal{S}$. Generally speaking, as that number 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 $\Pr(C_k|\tilde{d}_{q-1})$ 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. Similar observations can be made for the models (4) and (5).

4. INTEGRATION WITH N-GRAMS

The above provides a way to handle some of the global constraints in the language. To obtain a multi-span language model, it remains to combine them with local constraints, such as provided by the n-gram paradigm. Obviously, the goal of the resulting integrated approach is to leverage the benefits of both.

The integration can occur in a number of ways, such as straightforward interpolation, or within the maximum entropy framework [3]. In the following, we develop an alternative formulation for the combination of the n-gram and LSA paradigms. 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:

$$\Pr(w_q|H_{q-1}) = \Pr(w_q|H_{q-1}^{(n)}, H_{q-1}^{(l)}), \qquad (6)$$

where the history H_{q-1} now comprises an n-gram component ($H_{q-1}^{(n)} = w_{q-1}w_{q-2}\ldots w_{q-n+1}$) as well as an LSA component ($H_{q-1}^{(l)} = \tilde{d}_{q-1}$). This expression can be rewritten as:

$$\Pr(w_q|H_{q-1}) = \frac{\Pr(w_q, H_{q-1}^{(l)}|H_{q-1}^{(n)})}{\sum_{w_i\in\mathcal{V}} \Pr(w_i, H_{q-1}^{(l)}|H_{q-1}^{(n)})}, \qquad (7)$$

where the summation in the denominator extends over all words in $\mathcal{V}$. Expanding and re-arranging, the numerator of (7) is seen to be:

$$\begin{aligned}\Pr(w_q, H_{q-1}^{(l)}|H_{q-1}^{(n)}) &= \Pr(w_q|H_{q-1}^{(n)})\Pr(H_{q-1}^{(l)}|w_q, H_{q-1}^{(n)}) \\ &= \Pr(w_q|w_{q-1}w_{q-2}\ldots w_{q-n+1}) \\ &\quad \cdot \Pr(\tilde{d}_{q-1}|w_q w_{q-1}w_{q-2}\ldots w_{q-n+1}). \qquad (8)\end{aligned}$$

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:

$$\Pr(w_q|H_{q-1}) = \frac{\Pr(w_q|w_{q-1}w_{q-2}\ldots w_{q-n+1})\Pr(\tilde{d}_{q-1}|w_q)}{\sum_{w_i\in\mathcal{V}} \Pr(w_i|w_{q-1}w_{q-2}\ldots w_{q-n+1})\Pr(\tilde{d}_{q-1}|w_i)}. \qquad (9)$$

Note that, if $\Pr(\tilde{d}_{q-1}|w_q)$ is viewed as a prior probability on the current document history, then (9) simply translates the classical Bayesian estimation of the n-gram (local) probability using a prior distribution obtained from (global) LSA.

5. PERFORMANCE

Performance was evaluated on the WSJ0 part of the NAB News corpus. This was convenient for comparison purposes since conventional bigram and trigram language models are readily available, trained on exactly the same data [6]. 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 used for test purposes. 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 Wall Street Journal 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 [9]. 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).

Using the resulting vector space S of dimension 125, we constructed the direct model (1) and combined it with the standard bigram, as in (9). We then measured the resulting perplexity on the test data, and found a value of 147. This result is to be compared with the baseline results obtained with the standard bigram and trigram language models of [6], found to be 215 and 142, respectively. Thus, compared to the standard bigram, we obtained a 32% reduction in perplexity with the direct integrated model [5], which brings it to the same level of performance as the standard trigram.

We then investigated the effect of smoothing. Word and document clustering were performed using the two-level procedure (K-means and bottom-up clustering) described in Section 3, and related classes were merged to create cluster sets of different size. For each cluster set size (or combination thereof), we measured perplexity as before. In all cases, a perplexity minimum was obtained for a particular size of the cluster set.

In the case of word clustering, the perplexity minimum was equal to 106 and was 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, was 147, i.e., the same value as obtained using (1). 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.

Document clustering exhibits the same general behavior, with two notable differences. First, the minimum perplexity was somewhat higher (116). This indicates that clustering documents may not be as powerful as clustering words, in the sense just described. Second, the minimum was attained for a smaller size ($L = 1$) of the document cluster set, and perplexity increased 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. Alternatively, it may be an artifact of the document collection, which arguably is already quite homogeneous to begin with.

With both word and document clustering, the best results were obtained for a word cluster set size $K = 100$ and a document cluster size $L = 1$. In that case the perplexity minimum was equal to 102. Similar comments apply here as well.

Thus, the perplexity values obtained with the best (smoothed) integrated bigram/LSA language models (102–106) are about 50% better (respectively, 25% better) than that obtained using the standard bigram (respectively, trigram) language model. We conclude that the new integrated language models are quite effective in combining global semantic prediction with the usual local predictive power of the bigram language model. In addition, we expect that much of the reduction in perplexity observed at the bigram level would carry over to a combined trigram/LSA language model.

6. CONCLUSION

We have described a new approach to the integration of the various constraints, both local and global, that are present in the language. This approach exploits the complementarity between n-grams, which inherently rely on syntactically-oriented, short-span relationships, and LSA language models, which tend to capture semantically oriented, large span relationships between words. To harness this synergy, we have derived an integrative formulation to combine the two paradigms. Clustering in the LSA space was explored both at the word level and at the document level, and was found to be beneficial for smoothing purposes. All of the resulting multi-span language models were shown to substantially outperform the associated standard n-grams on a subset of the NAB News corpus.

7. REFERENCES

[1] T. Niesler and P. Woodland, "A Variable-Length Category-Based N-Gram Language Model," in *Proc.* 1996 *ICASSP*, Atlanta, GA, pp. I164–I167.

[2] R. Lau, R. Rosenfeld, and S. Roukos, "Trigger-Based Language Models: A Maximum Entropy Approach," in *Proc.* 1993 *ICASSP*, Minneapolis, MN, pp. II45–48.

[3] R. Rosenfeld, "A Maximum Entropy Approach to Adaptive Statistical Language Modeling," *Computer Speech and Language*, Vol. 10, London: Academic Press, pp. 187–228, July 1996.

[4] J.R. Bellegarda *et al.*, "A Novel Word Clustering Algorithm Based on Latent Semantic Analysis," in *Proc.* 1996 *ICASSP*, Atlanta, GA, pp. I172–I175.

[5] J.R. Bellegarda, "A Latent Semantic Analysis for Large-Span Language Modeling," in *Proc. EuroSpeech*'97, Rhodes, GR, Vol. 3, pp. 1451–1454.

[6] F. Kubala *et al.*, "The Hub and Spoke Paradigm for CSR Evaluation", in *Proc.* ARPA *Speech and Natural Language Workshop*, Morgan Kaufmann, pp. 40–44, March 1994.

[7] Y. Gotoh and S. Renals, "Document Space Models Using Latent Semantic Analysis," in *Proc. EuroSpeech*'97, Rhodes, GR, Vol. 3, pp. 1443–1448.

[8] J.R. Bellegarda, "Context-Dependent Vector Clustering for Speech Recognition," Chapter 6 in *Speech and Speaker Recognition: Advanced Topics*, New York: Kluwer, pp. 133–157, March 1996.

[9] M.W. Berry, "Large-Scale Sparse Singular Value Computations," *Int. J. Supercomp. Appl.*, Vol. 6, No. 1, pp. 13–49, 1992.