Language Modeling with Sentence-Level Mixtures

Rukmini Iyer† *Mari Ostendorf*† *J. Robin Rohlicek*‡

† Boston University
Boston, MA 02215

‡ BBN Inc.
Cambridge, MA 02138

ABSTRACT

This paper introduces a simple mixture language model that attempts to capture long distance constraints in a sentence or paragraph. The model is an m-component mixture of trigram models. The models were constructed using a 5K vocabulary and trained using a 76 million word Wall Street Journal text corpus. Using the BU recognition system, experiments show a 7% improvement in recognition accuracy with the mixture trigram models as compared to using a trigram model.

1. INTRODUCTION

The overall performance of a large vocabulary continuous speech recognizer is greatly impacted by the constraints imposed by a language model, or the effective constraints of a stochastic language model that provides the a priori probability estimates of the word sequence $P(w_1, \ldots, w_T)$. The most commonly used statistical language model assumes that the word sequence can be described as a high order Markov process, typically referred to as an n-gram model, where the probability of a word sequence is given by:

$$P(w_1, \ldots, w_T) = \prod_{i=1}^{T} P(w_i | w_{i-1}, \ldots, w_{i-n+1}). \quad (1)$$

The standard n-gram models that are commonly used are the bigram ($n = 2$) and the trigram ($n = 3$) models, where n is limited primarily because of insufficient training data. However, with such low order dependencies, these models fail to take advantage of 'long distance constraints' over the sentence or paragraph. Such long distance dependencies may be grammatical, as in verb tense agreement or singular/plural quantifier-noun agreement. Or, they may also be a consequence of the inhomogeneous nature of language; different words or word sequences are more likely for particular broad topics or tasks. Consider, for example, the following responses made by the combined BU-BBN recognition system on the 1993 Wall Street Journal (WSJ) benchmark H1-C1 (20K) test:

> REF: the first recipient joseph webster junior ** ****** a PHI BETA KAPPA chemistry GRAD who plans to take courses this fall in ART RELIGION **** music and political science
> HYP: the first recipient joseph webster junior HE FRIDAY a CAP OF ***** chemistry GRANT who plans to take courses this fall in AREN'T REALLY CHIN music and political science
>
> REF: *** COCAINE doesn't require a SYRINGE THE symbol of drug abuse and CURRENT aids risk YET can be just as ADDICTIVE and deadly as HEROIN
> HYP: THE KING doesn't require a STRANGE A symbol of drug abuse and TRADE aids risk IT can be just as ADDICTED and deadly as CHAIRMAN

In the first example, "art" and "religion" make more sense in the context of "courses" than "aren't really chin", and similarly "heroin" should be more likely than "chairman" in the context of "drug abuse".

The problem of representing long-distance dependencies has been explored in other stochastic language models, though they tend to address only one or the other of the two issues raised here, i.e. either sentence-level or task-level dependence. Language model adaptation (e.g. [1, 2, 3]) addresses the problem of inhomogeneity of n-gram statistics, but mainly represents task level dependencies and does little to account for dependencies within a sentence. A context-free grammar could account for sentence level dependencies, but it is costly to build task-specific grammars as well as costly to implement in recognition. A few automatic learning techniques, which are straight-forward to apply to new tasks, have been investigated for designing static models of long term dependence. For example, Bahl *et al.* [4] used decision tree clustering to reduce the number of n-grams while keeping n large. Other efforts include models that integrate n-grams with context-free grammars (e.g., [5, 6, 7]).

Our approach to representing long term dependence attempts to address both issues, while still using a very simple model. We propose to use a mixture of n-gram language models, but unlike previous work in mixture language modeling our mixture components are combined at the sentence level. The component n-grams enable us to capture topic dependence, while using mixtures at the sentence level captures the notion that topics do not change mid-sentence. Like the model proposed by Kneser and Steinbiss [8], our language model uses m component language models, each of which can be identified with the n-gram statistics of a specific topic or broad class

of sentences. However, unlike [8], which uses mixtures at the n-gram level with dynamically adapted mixture coefficients, we use sentence-level mixtures to capture within-sentence dependencies. Thus, the probability of a word sequence is the weighted combination:

$$P(w_1, \ldots, w_T) = \sum_{k=1}^{m} \lambda_k \prod_{i=1}^{T} P_k(w_i | w_{i-1}, \ldots, w_{i-n+1}). \quad (2)$$

Our approach has the advantage that it can be used either as a static or a dynamic model, and can easily leverage the techniques that have been developed for adaptive language modeling, particularly cache [1, 9] and trigger [2, 3] models. One might raise the issue of recognition search cost for a model of mixtures at the sentence level, but in the N-best rescoring framework [10] the additional cost of the mixture language model is minimal.

The general framework and mechanism for designing the mixture language model will be described in the next section, including descriptions of automatic topic clustering and robust estimation techniques. Following this discussion, we will present some experimental results on mixture language modeling obtained using the BU recognition system. Finally, the paper will conclude with a discussion of the possible extensions of mixture language models, to dynamic language modeling and to applications other than speech transcription.

2. MIXTURE LANGUAGE MODEL

2.1. General Framework

The sentence-level mixture language model was originally motivated by an observation that news stories (and certainly other domains as well) often reflect the characteristics of different topics or sub-domains, such as sports, finance, national news and local news. The likelihood of different words or n-grams could be very different in different sub-domains, and it is unlikely that one would switch sub-domains mid-sentence. A model with sentence-level mixtures of topic-dependent component models would address this problem, but the model would be more general if it also allowed for n-gram level mixtures within the components (e.g. for robust estimation). Thus, we propose a model using mixtures at two levels: the sentence and the n-gram level. Using trigram components, this model is described by

$$\begin{aligned} P(w_1, \ldots, w_T) &= \sum_{k=1}^{m} \lambda_k \prod_{i=1}^{T} [\theta_k P_k(w_i | w_{i-1}, w_{i-2}) \\ &\quad + (1 - \theta_k) P_I(w_i | w_{i-1}, w_{i-2})], \quad (3) \end{aligned}$$

where k is an index to the particular topic described by the component language model $P_k(\cdot|\cdot)$, $P_I(\cdot|\cdot)$ is a topic-independent model that is interpolated with the topic-dependent model for purposes of robust estimation or dynamic language model adaptation, and λ_k and θ_k are the sentence-level and n-gram level mixture weights, respectively. (Note that the component-dependent term $P_k(w_i|w_{i-1}, w_{i-2})$ could itself be a mixture.)

Two important aspects of the model are the definition of "topics" and robust parameter estimation. The m component distributions of the language model correspond to different "topics", where topic can mean any broad class of sentences, such as subject area (as in the examples given above) or verb tense. Topics can be specified by hand, according to text labels if they are available, or by heuristic rules associated with known characteristics of a task domain. Topics can also be determined automatically, which is the approach taken here, using any of a variety of clustering methods to initialize the component models. Robust parameter estimation is another important issue in mixture language modeling, because the process of partitioning the data into topic-dependent subsets reduces the amount of training available to estimate each component language model. These two issues, automatic clustering for topic initialization and robust parameter estimation, are described further in the next two subsections.

2.2. Clustering Algorithm

Since the standard WSJ language model training data does not have topic labels associated with the text, it was necessary to use automatic clustering to identify natural groupings of the data into "topics". Because of its conceptual simplicity, agglomerative clustering is used to partition the training data into the desired number of clusters. The clustering is at the paragraph level, relying on the assumption that an entire paragraph comes from a single topic. Each paragraph begins as a singleton cluster. Paragraph pairs are then progressively grouped into clusters by computing the similarity between clusters and grouping the two most similar clusters. The basic clustering algorithm is as follows:

1. Let the desired number of clusters be C^* and the initial number of clusters C be the number of singleton data samples, or paragraphs.
2. Find the best matched clusters, say A_i and A_j, to minimize the similarity criterion S_{ij}.
3. Merge A_i and A_j and decrement C.
4. If current number of clusters $C = C^*$, then stop; otherwise go to Step 2.

At the end of this stage, we have the desired number of partitions of the training data. To save computation, we run agglomerative clustering first on subsets of the data, and then continue by agglomerating resulting clusters into a final set of m clusters.

A variety of similarity measures can be envisioned. We use a normalized measure of the number of content words in common between the two clusters. (Paragraphs comprise both function words (e.g. is, that, but) and content words (e.g stocks, theater, trading), but the function words do not contribute towards the identification of a paragraph as belonging to a particular topic so they are ignored in the similarity criterion.) Letting A_i be the set of unique content words in cluster i, $|A_i|$ the number of elements in A_i, and N_i the number of paragraphs in cluster i, then the specific measure of similarity of two clusters i and j is

$$S_{ij} = \frac{N_{ij}|A_i \cap A_j|}{|A_i \cup A_j|}, \quad (4)$$

where

$$N_{ij} = \sqrt{\frac{N_i + N_j}{N_i \times N_j}} \quad (5)$$

is a normalization factor used to avoid the tendency for small clusters to group with one large cluster rather than other small clusters.

At this point, we have only experimented with a small number of clusters, so it is difficult to see coherent topics in them. However, it appears that the current models are putting news related to foreign affairs (politics, as well as travel) into one cluster and news relating to finance (stocks, prices, loans) in another.

2.3. Parameter Estimation

Each component model is a conventional n-gram model. Initial n-gram estimates for the component models are based on the partitions of the training data, obtained by using the above clustering algorithm. The initial component models are estimated separately for each cluster, where the Witten-Bell back-off [11] is used to compute the probabilities of n-grams not observed in training, based on distributing a certain amount of the total probability mass among unseen n-grams. This method was chosen based on the results of [12] and our own comparative experiments with different back-off methods for WSJ n-gram language models. The parameters of the component models can be re-estimated using the Expectation-Maximization (EM) algorithm [13]. However, since the EM algorithm was computationally intensive, an iterative re-labeling re-estimation technique was used. At each iteration, the training data is re-partitioned, by re-labeling each utterance according to which component model maximizes the likelihood of that utterance. Then, the component n-gram statistics are re-computed using the new subsets of the training data, again using the Witten-Bell back-off technique. The iterations continue until a steady state size for the clusters is reached.

Since the component models are built on partitioned training data, there is a danger of them being undertrained. There are two main mechanisms we have explored for robust parameter estimation, in addition to using standard back-off techniques. One approach is to include a general model P_G trained on all the data as one of the mixture components. This approach has the advantage that the general model will be more appropriate for recognizing sentences that do not fall clearly into any of the topic-dependent components, but the possible disadvantage that the component models may be underutilized because they are relatively undertrained. An alternative is to interpolate the general model with each component model at the n-gram level, but this may force the component models to be too general in order to allow for unforeseen topics. Given these trade-offs, we chose to implement a compromise between the two approaches, i.e. to include a general model as one of the components, as well as some component level smoothing via interpolation with a general model. Specifically, the model is given by

$$P(w_1, \ldots, w_T) = \sum_{k=1,\ldots,C,G} \lambda_k \prod_{i=1}^{T} (\theta_k P_k(w_i|w_{i-1}, w_{i-2}) + (1-\theta_k) P_{G'}(w_i|w_{i-1}, w_{i-2})) \quad (6)$$

where $P_{G'}$ is a general model (which may or may not be the same as P_G), $\{\lambda_k\}$ provide weights for the different topics, and $\{\theta_k\}$ serve to smooth the component models.

Both sets of mixture weights are estimated on a separate data set, using a maximum likelihood criterion and initializing with uniform weights. To simplify the initial implementation, we did not estimate the two sets of weights $\{\lambda_k\}$ and $\{\theta_k\}$ jointly. Rather, we first labeled the sentences in the mixture weight estimation data set according to their most likely component models, and then separately estimated the weight θ_k to maximize the likelihood of the data assigned to its cluster. For a single set of data, the mixture weight estimation algorithm involves iteratively updating

$$\theta_k^{new} = \frac{1}{N} \sum_{i=1}^{N} \frac{\theta_k^{old} P_k(w_1, .., w_{n_i})}{\sum_j \theta_j^{old} P_j(w_1, \ldots, w_{n_i})} \quad (7)$$

where n_i is the number of words in sentence i and N is the total number of sentences in cluster k. After the component models have been estimated, the sentence-level mixture weights $\{\lambda_k\}$ are estimated using an analogous algorithm.

3. EXPERIMENTS

The corpus used for training the different component models comprised the 38 million WSJ0 data, as well as the 38 million word augmented LM data obtained from BBN Inc. The vocabulary is the standard 5K non-verbalized pronunciation (NVP) data augmented with the verbalized punctuation words and a few additional words. In order to compute the mixture weights, both at the trigram-level as well as the sentence-level, the WSJ1 speaker-independent transcriptions serve as

the "held out" data set. Because we felt that the training data may not accurately represent the optional verbalized punctuation frequency in the WSJ1 data, we chose to train models on two data sets. The general model P_G and the component models P_k were trained on the WSJ0 NVP data augmented by the BBN data. The general model $P_{G'}$ was trained on the WSJ0 verbalized pronunciation data, so that using $P_{G'}$ in smoothing the component models also provides a simple means of allowing for verbalized pronunciation.

The experiments compare a single trigram language model to a five-component mixture of trigram models. To explore the trade-offs of using different numbers of clusters, we also consider an eight-component trigram mixture. Perplexity and recognition results are reported on the Nov. '93 ARPA development and evaluation 5k vocabulary WSJ test sets.

3.1. Recognition Paradigm

The BU Stochastic Segment Model recognition system is combined with the BBN BYBLOS system and uses the N-best rescoring formalism [10]. In this formalism, the BYBLOS system, a speaker-independent Hidden Markov Model System [14][1], is used to compute the top N sentence hypotheses of which the top 100 are subsequently rescored by the SSM. A five-pass search strategy is used to generate the N-best hypotheses, and these are rescored with thirteen state HMMs. A weighted combination of scores from different knowledge sources is used to re-rank the hypotheses and the top ranking hypothesis is used as the recognized output. The weights for recombination are estimated on one test set (in this case the 93 H2 development test data) and held fixed for all other test sets.

We conducted a series of experiments in the rescoring paradigm to assess the usefulness of the mixture model. Unless otherwise noted, the only acoustic model score used was based on the stochastic segment model. The language model scores used varied with the experiments. For the best-case system, we used all scores, which included the SSM and the BBN Byblos HMM and SNN acoustic scores, and both the BBN trigram and BU mixture language model scores.

3.2. Results

The results reported in Table 1 compare three different language models in terms of perplexity and recognition performance: a simple trigram, and five- and eight-component mixtures. The mixture models reduce the perplexity only by a small amount, but there is a reduction in word-error with the five-component mixture model. We hypothesize that there is not enough training data to effectively use more mixture components.

[1]For an indication of the performance of this system, see the benchmark summary in [17].

No. of components	% Word error	Perplexity
1	7.3	118
5	7.1	116
8	7.2	114

Table 1: Dependence on number of components: evaluation on the '93 ARPA 5k WSJ development test set.

The next series of experiments, summarized in Table 2[2], compared recognition performance for the BBN trigram language model [15], the BU 5-component mixture model, and the case where both language model scores are used in the N-best reranking. All language models were estimated from the same training data. The results show a 7% reduction in error rate on the evaluation test set, comparing the combined language models to the BBN trigram. It is interesting that the combination of the trigram and the mixture model yielded a small improvement in performance (not significant, but consistent across test sets), since the trigram is a component of the mixture model. The difference between the mixture model and the two combined models corresponds to a linear vs. non-linear combination of component probabilities, respectively.

For reference, we also include the best case system performance, which corresponds the the case where all acoustic and language model scores. Even with all the acoustic model scores, adding the mixture language model improves performance, giving a best case result of 5.3% word error on the '93 5k WSJ evaluation test set.

4. DISCUSSION

In summary, this paper presents a new approach to language modeling, which offers the potential for capturing both topic-dependent effects and long-range sentence level effects in

[2]The performance figures quoted here are better than those reported in the official November 1993 WSJ benchmark results, because more language model training data was available in the experiment reported here.

KSs used		% Word Error	
AM	LM	Dev	Eval
SSM	trigram	7.4	6.1
	mixture	7.1	5.8
	both	7.0	5.7
all	both	6.3	5.3

Table 2: Summary of results on'93 ARPA 5k WSJ test sets for different acoustic model (AM) and language model (LM) knowledge sources (KSs).

a conceptually simple variation of statistical n-gram models. The model is actually a two-level mixture model, with separate mixture weights at the n-gram and sentence levels. Training involves either automatic clustering or heuristic rules to determine the initial topic-dependent models, and an iterative algorithm for estimating mixture-weights at the different levels. Recognition experiments on the WSJ task showed a significant improvement in the accuracy for the BU-SSM recognition system.

This work can be extended in several ways. First, time limitations did not permit us to explore the use of the complete EM algorithm for estimating mixture components and weights jointly, and we hope to investigate that approach in the future. In addition, it may be useful to consider other metrics for automatic topic clustering, such as a word count weighted by inverse document frequencies or a multinomial distribution assumption with a likelihood clustering criterion. Of course, it would also be interesting to see if further performance gains could be achieved with more clusters. Much more could also be done in the area of robust parameter estimation. For example, one could use an n-gram part-of-speech sequence model as the base for all component models and topic-dependent word likelihoods given the part-of-speech label, a natural extension of [16].

Dynamic language model adaptation, which makes use of the previous document history to tune the language model to that particular topic, can easily fit into the mixture model framework in two ways. First, the sentence-level mixture weights can be adapted according to the likelihood of the respective mixture components in the previous utterance, as in [8] for n-gram level mixture weights. Second, the dynamic n-gram cache model [1, 9] can easily be incorporated into the mixture language model. However, in the mixture model, it is possible to have component-dependent cache models, where each component cache would be updated after each sentence according to the likelihood of that component given the recognized word string. Trigger models [2, 3] could also be component dependent.

The simple static mixture language model can also be useful in applications other than continuous speech transcription. For example, topic-dependent models could be used for topic spotting. In addition, as mentioned earlier, the notion of topic need not be related to subject area, it can be related to speaking style or speaker goal. In the ATIS task, for example, the goal of the speaker (e.g. flight information request, response clarification, error correction) is likely to be reflected in the language of the utterance. Representing this structure explicitly has the double benefit of improving recognition performance and providing information for a dialog model.

From a cursory look at our recognition errors from the recent WSJ benchmark tests, it is clear that topic-dependent models will not be enough to dramatically reduce word error rate. Out-of-vocabulary words and function words also represent a major source of errors. However, an important advantage of this framework is that it is a simple extension of existing language modeling techniques that can easily be integrated with other language modeling advances.

Acknowledgments

This work was supported jointly by ARPA and ONR on grant ONR ONR-N00014-92-J-1778. We gratefully acknowledge the cooperation of several researchers at BBN, who provided the N-best hypotheses used in our recognition experiments, as well as additional language model training data.

References

1. F. Jelinek, B. Merialdo, S. Roukos and M. Strauss, "A Dynamic LM for Speech Recognition," *Proc. DARPA Workshop on Speech and Natural Language*, pp. 293-295, 1991.
2. R. Lau, R. Rosenfeld and S. Roukos, "Trigger-Based Language Models: a Maximum Entropy Approach," *Proc. Int'l. Conf. on Acoust., Speech and Signal Proc.*, Vol. II, pp. 45-48, 1993.
3. R. Rosenfeld, "A Hybrid Approach to Adaptive Statistical Language Modeling," this proceedings.
4. L. R. Bahl, P. F. Brown, P. V. deSouza and R. L. Mercer, "A Tree-Based Statistical Language Model for Natural Language Speech Recognition," *IEEE Trans. on Acoust., Speech, and Signal Proc.*, Vol. 37, No. 7, pp. 1001-1008, 1989.
5. J. H. Wright, G. J. F. Jones and H. Lloyd-Thomas, "A Consolidated Language Model For Speech Recognition," *Proc. EuroSpeech*, Vol. 2, pp. 977-980, 1993.
6. M. Meteer and J. R. Rohlicek, "Statistical Language Modeling Combining n-gram and Context Free Grammars," *Proc. Int'l. Conf. on Acoust., Speech and Signal Proc.*, Vol. 2, pp. 37-40, 1993.
7. J. Lafferty, "Integrating Probabilistic Finite-State and Context-Free Models of Language," presentation at the IEEE ASR Workshop, December 1993.
8. R. Kneser and V. Steinbiss, "On the Dynamic Adaptation Of Stochastic LM," *Proc. Int'l. Conf. on Acoust., Speech and Signal Proc.*, Vol. 2, pp. 586-589, 1993.
9. R. Kuhn and R. de Mori, "A Cache Based Natural Language Model for Speech Recognition," *IEEE Trans. PAMI*, Vol. 14, pp. 570-583, 1992.
10. M. Ostendorf, A. Kannan, S. Austin, O. Kimball, R. Schwartz and J. R. Rohlicek, "Integration of Diverse Recognition Methodologies Through Reevaluation of N-Best Sentence Hypotheses," *Proc. DARPA Workshop on Speech and Natural Language*, pp. 83-87, February 1991.
11. H.Witten and T. C. Bell, "The Zero Frequency Estimation of Probabilities of Novel Events in Adaptive Text Compression," *IEEE Trans. Information Theory*, Vol. IT-37, No. 4, pp. 1085-1094, 1991.
12. P. Placeway and R. Schwartz, "Estimation Of Powerful LM from Small and Large Corpora," *Proc. Int'l. Conf. on Acoust., Speech and Signal Proc.*, Vol. 2, pp. 33-36, 1993.
13. A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood Estimation from Incomplete Data," *Journal of the Royal Statistical Society (B)*, Vol. 39, No. 1, pp. 1–38, 1977.

14. BBN Byblos November 1993 WSJ Benchmark system.

15. R. Schwartz *et al.*, "On Using Written Language Training Data for Spoken Language Modeling," this proceedings.

16. M. Elbeze and A.-M. Derouault, "A Morphological Model for Large Vocabulary Speech Recognition," *Proc. Int'l. Conf. on Acoust., Speech and Signal Proc.*, Vol. 1, pp. 577-580, 1990.

17. D. Pallett, J. Fiscus, W. Fisher, J. Garofolo, B. Lund and M. Pryzbocki, "1993 Benchmark Tests for the ARPA spoken Language Program," this proceedings.