A HYBRID APPROACH TO ADAPTIVE STATISTICAL LANGUAGE MODELING

Ronald Rosenfeld

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 12513

ABSTRACT

We describe our latest attempt at adaptive language modeling. At the heart of our approach is a Maximum Entropy (ME) model, which incorporates many knowledge sources in a consistent manner. The other components are a selective unigram cache, a conditional bigram cache, and a conventional static trigram. We describe the knowledge sources used to build such a model with ARPA's official WSJ corpus, and report on perplexity and word error rate results obtained with it. Then, three different adaptation paradigms are discussed, and an additional experiment, based on AP wire data, is used to compare them.

1. OVERVIEW OF ME FRAMEWORK

Using several different probability estimates to arrive at one combined estimate is a general problem that arises in many tasks. The Maximum Entropy (ME) principle has recently been demonstrated as a powerful tool for combining statistical estimates from diverse sources[1, 2, 3]. The ME principle ([4, 5]) proposes the following:

1. Reformulate the different estimates as constraints on the expectation of various functions, to be satisfied by the target (combined) estimate.
2. Among all probability distributions that satisfy these constraints, choose the one that has the highest entropy.

More specifically, for estimating a probability function $P(\mathbf{x})$, each constraint i is associated with a *constraint function* $f_i(\mathbf{x})$ and a *desired expectation* c_i. The constraint is then written as:

$$E_P f_i \stackrel{\text{def}}{=} \sum_{\mathbf{x}} P(\mathbf{x}) f_i(\mathbf{x}) = c_i \,. \qquad (1)$$

Given consistent constraints, a unique ME solutions is guaranteed to exist, and to be of the form:

$$P(\mathbf{x}) = \prod_i \mu_i^{f_i(\mathbf{x})} \,, \qquad (2)$$

where the μ_i's are some unknown constants, to be found. Probability functions of the form (2) are called *log-linear*, and the family of functions defined by holding the f_i's fixed and varying the μ_i's is called *an exponential family*.

To search the family defined by (2) for the μ_i's that will make $P(\mathbf{x})$ satisfy all the constraints, an iterative algorithm, "Generalized Iterative Scaling" (GIS), exists, which is guaranteed to converge to the solution ([6]), as long as the constraints are mutually consistent. GIS starts with arbitrary μ_i values. At each iteration, it computes the expectations $E_P f_i$ over the training data, compares them to the desired values c_i's, and then adjusts the μ_i's by an amount proportional to the ratio of the two.

Generalized Iterative Scaling can be used to find the ME estimate of a simple (non-conditional) probability distribution over some event space. An adaptation of GIS to conditional probabilities was proposed by [7], as follows. Let $P(w|h)$ be the desired probability estimate, and let $\tilde{P}(h,w)$ be the empirical distribution of the training data. Let $f_i(h,w)$ be any constraint function, and let c_i be its desired expectation. Equation 1 is now modified to:

$$\sum_h \tilde{P}(h) \cdot \sum_w P(w|h) \cdot f_i(h,w) = c_i \qquad (3)$$

See also [1, 2].

2. CAPTURING LONG-DISTANCE LINGUISTIC PHENOMENA

The ME framework is very general, freeing the modeler to concentrate on searching for significant information sources and choosing the phenomena to be modeled. In statistical language modeling, we are interested in information about the identity of the next word, w_i, given the *history* h, namely the part of the document that was already processed by the system. We have so far considered the following information sources, all contained within the history:

Conventional N-grams: the immediately preceding few words, say (w_{i-2}, w_{i-1}).

Long distance N-grams[8]: N-grams preceding w_i by j positions.

triggers[9]: the appearance in the history of words related to w_i.

class triggers: trigger relations among word clusters.

count-based cache: the number of times w_i already occurred in the history.

distance-based cache: the last time w_i occurred in the history.

linguistically defined constraints: number agreement, tense agreement, etc.

Any potential source can be considered separately, and the amount of information in it estimated. For example, in estimating the potential of count-based caches, we might measure dependencies of the form depicted in figure 1, and calculate the amount of information they may provide. See also [3].

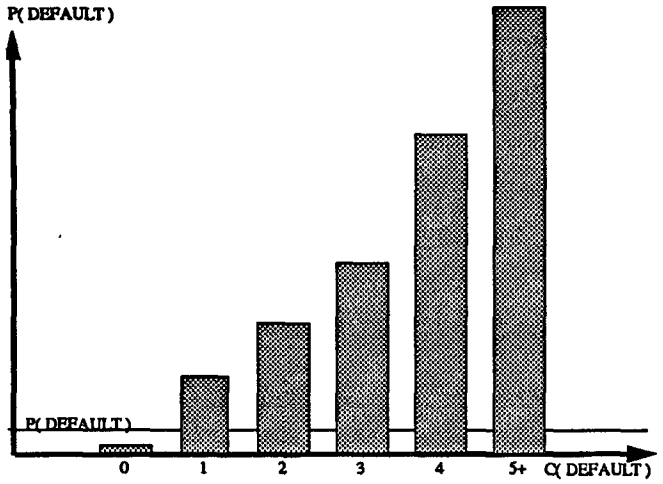

Figure 1: Count-based cache information: Probability of 'DEFAULT' as a function of the number of times it already occurred in the document. The horizontal line is the unconditional probability.

Perhaps the most important feature of the Maximum Entropy framework is its extreme generality. *For any conceivable linguistic or statistical phenomena, appropriate constraint functions can readily be written.* We will demonstrate this process for several of the knowledge sources listed above.

2.1. Formulating N-grams as Constraints

The usual unigram, bigram and trigram Maximum Likelihood estimates can be replaced by unigram, bigram and trigram constraints conveying the same information. Specifically, the constraint function for the unigram w_1 is:

$$f_{w_1}(h,w) = \begin{cases} 1 & \text{if } w = w1 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

and its associated constraint is:

$$\sum_h \tilde{P}(h) \sum_w P(w|h) f_{w_1}(h,w) = \tilde{E} f_{w_1}(h,w). \quad (5)$$

Similarly, the constraint function for the bigram w_1, w_2 is

$$f_{w_1,w_2}(h,w) = \begin{cases} 1 & \text{if } h \text{ ends in } w_1 \text{ and } w = w2 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

and its associated constraint is

$$\sum_h \tilde{P}(h) \sum_w P(w|h) f_{w_1,w_2}(h,w) = \tilde{E} f_{w_1,w_2}(h,w). \quad (7)$$

and similarly for higher-order N-grams.

2.2. Formulating long-distance N-grams as Constraints

The constraint functions for long distance N-grams are very similar to those for conventional (distance 1) N-gram. For example, the constrain function for the distance-2 trigram $\{w_1, w_2, w_3\}$ is:

$$f_{w_1,w_2,w_3}(h,w) = \begin{cases} 1 & \text{if } h \text{ ends in } \{w_1, w_2, w^*\} \text{ for some } w^*, \\ & \text{and } w = w3 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

and its associated constraint is

$$\sum_h \tilde{P}(h) \sum_w P(w|h) f_{w_1,w_2,w_3}(h,w) = \tilde{E} f_{w_1,w_2,w_3}(h,w). \quad (9)$$

and similarly for other long distance N-grams.

2.3. Formulating Triggers as Constraints

For class triggers, let A, B be two related word clusters. Define the constraint function $f_{A \to B}$ as:

$$f_{A \to B}(h,w) = \begin{cases} 1 & \text{if } \exists w_j \in A, w_j \in h, w \in B \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Set $c_{A \to B}$ to $\tilde{E}[f_{A \to B}]$, the *empirical expectation* of $f_{A \to B}$ (i.e. its expectation in the training data). Now the constraint on $P(h,w)$ is:

$$E_P[f_{A \to B}] = \tilde{E}[f_{A \to B}] \quad (11)$$

3. SELECTIVE UNIGRAM CACHE

In a document-based unigram cache, all words that occurred in the history of the document are stored, and are used to dynamically generate a unigram, which is in turn combined with other language model components. N-gram caches were first reported by [10].

The motivation behind a unigram cache is that, once a word occurs in a document, its probability of re-occurring is typically greatly elevated. But the extent of this phenomenon

depends on the prior frequency of the word, and is most pronounced for rare words. The occurrence of a common word like 'THE' provides little new information. Put another way, the occurrence of a rare word is more surprising, and hence provides more information, whereas the occurrence of a more common word deviates less from the expectations of the static model, and therefore requires a smaller modification to it.

Bayesian analysis may be used to optimally combine the prior of a word with the new evidence provided by its occurrence. As a rough first approximation, we implemented a selective unigram cache, where only rare words are stored in the cache. A word is defined as rare relative to a threshold of static unigram frequency. The exact value of the threshold was determined by optimizing perplexity on unseen data. This scheme proved more useful for perplexity reduction than the conventional cache.

4. CONDITIONAL BIGRAM AND TRIGRAM CACHES

In a document-based bigram cache, all consecutive word pairs that occurred in the history of the document are stored, and are used to dynamically generate a bigram, which is in turn combined with other language model components. A trigram cache is similar but is based on all consecutive word triples.

An alternative way of viewing a bigram cache is as a set of unigram caches, one for each word in the history. At most one such unigram is consulted at any one time, depending on the identity of the last word of the history. Viewed this way, it is clear that the bigram cache should contribute to the combined model only if the last word of the history is a (non-selective) unigram "cache hit". In all other cases, the uniform distribution of the bigram cache would only serve to flatten, hence degrade, the combined estimate.

We therefore chose to use a conditional bigram cache, which has a non-zero weight only during such a "hit".

A similar argument can be applied to the trigram cache. Such a cache should only be consulted if the last two words of the history occurred before, i.e. the trigram cache should contribute only immediately following a bigram cache hit. We experimented with such a trigram cache, constructed similarly to the conditional bigram cache. However, we found that it contributed little to perplexity reduction. This is to be expected: every bigram cache hit is also a unigram cache hit. Therefore, the trigram cache can only refine the distinctions already provided by the bigram cache. A document's history is typically small (225 words on average in the WSJ corpus). For such a modest cache, the refinement provided by the trigram is small and statistically unreliable.

Another way of viewing the selective bigram and trigram caches is as regular (i.e. non-selective) caches, which are later interpolated using weights that depend on the count of their context. Then, zero context-counts force respective zero weights.

5. THE WSJ SYSTEM

As a testbed for the above ideas, we used ARPA's CSR task. The training data was 38 million words of Wall Street Journal (WSJ) text from 1987–1989. The vocabulary used was ARPA's official "20o.nvp" (20,000 most common WSJ words, non-verbalized punctuation).

To measure the impact of the amount of training data on language model adaptation, we experimented with systems based on varying amounts of training data. The largest model we built was based on the entire 38M words of WSJ training data, and is described below.

5.1. The Component Models

The adaptive language model was based on four component language models:

1. A conventional "compact" backoff trigram model. "Compact" here means that singleton trigrams (word triplets that occurred only once in the training data) were excluded from the model. It consisted of 3.2 million trigrams and 3.5 million bigrams. This model also served as the baseline for comparisons, and was dubbed "the static model".

2. A Maximum Entropy model trained on the same data as the trigram, and consisting of the following knowledge sources:

   - High cutoff, distance-1 (conventional) N-grams:
     - All trigrams that occurred 9 or more times in the training data (428,000 in all).
     - All bigrams that occurred 9 or more times in the training data (327,000).
     - all unigrams.

     The high cutoffs were necessary in order to reduce the heavy computational requirements of the training procedure.
   - High cutoff, distance-2 bigrams and trigrams:
     - All distance-2 trigrams that occurred 5 or more times in the training data (795,000 in all).
     - All distance-2 bigrams that occurred 5 or more times in the training data (651,000).

     The cutoffs used for the conventional N-grams were higher than those applied to the distance-2 N-grams. This was done because we expected that the information lost from the former knowledge

source will be re-introduced, at least partially, by interpolation with the static model.

- Word Trigger Pairs: For every word in the vocabulary, the top 3 triggers were selected based on their mutual information with that word as computed from the training data[1, 2]. This resulted in some 43,000 word trigger pairs.

3. A selective unigram cache, as described earlier, using a unigram threshold of 0.001.

4. A conditional bigram cache, as described earlier.

5.2. Combining the LM Components

The combined model was achieved by consulting an appropriate subset of the above four models. At any one time, the four component LMs were combined linearly. But the weights used were not fixed, nor did they follow a linear pattern over time.

Since the Maximum Entropy model incorporated information from trigger pairs, its relative weight should be increased with the length of the history. But since it also incorporated new information from distance-2 N-grams, it is useful even at the very beginning of a document, and its weight should not start at zero.

We therefore started the Maximum Entropy model with a weight of ~0.3, which was gradually increased over the first 60 words of the document, to ~0.7. The conventional trigram started with a weight of ~0.7, and was decreased concurrently to ~0.3. The conditional bigram cache had a non-zero weight only during a cache hit, which allowed for a relatively high weight of ~0.09. The selective unigram cache had a weight proportional to the size of the cache, saturating at ~0.05. The weights were always normalized to sum to 1.

While the general weighting scheme was chosen based on considerations discussed above, the specific values of the weights were chosen by minimizing perplexity of unseen data. It became clear later that this did not always correspond with minimizing error rate. Subsequently, further weight modifications were determined by direct trial-and-error measurements of word error rate on development data.

5.3. Varying the Training Data

As mentioned before, we also experimented with systems based on less training data. We built two such systems, one based on 5 million words, and the other based on 1 million words. Both systems were identical to the larger systems described above, except that the Maximum Entropy model did not employ high cutoffs, but was instead based on the same N-gram information as the conventional trigram model.

5.4. Computational Costs

The computational bottleneck of the Generalized Iterative Scaling algorithm is in constraints which, for typical histories h, are non-zero for a large number of words w's. This means that bigram constraints are more expensive than trigram constraints. Implicit computation can be used for unigram constraints. Therefore, the time cost of bigram and trigger constraints dominated the total time cost of the algorithm.

The computational burden of training the Maximum Entropy model for the large system (38MW) was quite severe. Fortunately, the training procedure is highly parallelizable (see [1]). Training was run in parallel on 10-25 high performance workstations, with an average of perhaps 15 machines. Even so, it took 3 weeks to complete.

In comparison, training the 5MW system took only a few machine-days, and training the 1MW system was trivial.

5.5. Perplexity Reduction

We used 325,000 words of unseen WSJ data to measure perplexities of the baseline trigram model, the Maximum Entropy component, and the interpolated adaptive model (the latter consisting of the first two together with the unigram and bigram caches). This was done for each of the three systems (38MW, 5MW and 1MW). Results are summarized in table 1.

amt. of training data	1M	5M	38M
trigram (baseline) perplexity	269	173	105
Maximum Entropy perplexity	203	123	86
PP reduction	24%	29%	18%
interpolated model perplexity	163	108	71
PP reduction	39%	38%	32%

Table 1: Perplexity (PP) improvement of Maximum Entropy and interpolated adaptive models over a conventional trigram model, for varying amounts of training data. The 38MW ME model used far fewer parameters than the baseline, since it employed high N-gram cutoffs. See text.

As can be observed, the Maximum Entropy model, even when used alone, was significantly better than the static model. Its relative advantage seems greater with more training data. With the large (38MW) system, practical consideration required imposing high cutoffs on the ME model, and yet its perplexity is still significantly better than that of the baseline. This is particularly notable because the ME model uses only *one third* the number of parameters used by the trigram model (2.26M vs. 6.72M).

When the Maximum Entropy model is supplemented with the other three components, perplexity is again reduced significantly. Here the relationship with the amount of training data is reversed: the less training data, the greater the improvement. This effect is due to the caches, and can be explained as follows: The amount of information provided by the caches is independent of the amount of training data, and is therefore fixed across the three systems. However, the 1MW system has higher perplexity, and therefore the relative improvement provided by the caches is greater. Put another way, models based on more data are stronger, and therefore harder to improve on.

5.6. Error Rate Reduction

To evaluate error rate reduction, we used the Nov93 ARPA S1 evaluation set[11, 12, 13]. It consisted of 424 utterances produced in the context of complete long documents by two male and two female speakers. We used the SPHINX-II recognizer([14, 15, 16]) with sex-dependent non-PD 10K senone acoustic models. In addition to the 20K words in the lexicon, 178 OOV words and their correct phonetic transcriptions were added in order to create closed vocabulary conditions. We first ran the forward and backward passes of SPHINX II to create word lattices, which were then used by three independent A* passes. The first such pass used the 38MW static trigram language model. The other two passes used the 38MW interpolated adaptive LM. The first of these two adaptive runs was for unsupervised word-by-word adaptation, in which the decoder output was used to update the language model. The other run used supervised adaptation, in which the decoder output was used for within-sentence adaptation, while the correct sentence transcription was used for across-sentence adaptation. Results are summarized in table 2.

language model	word error rate	% reduction
static trigram (baseline)	19.9%	—
unsupervised adaptation	17.8%	10%
supervised adaptation	17.0%	14%

Table 2: Word error rate reduction of adaptive language models over a conventional trigram model.

6. THREE PARADIGMS OF ADAPTATION

The adaptation we concentrated on so far was the kind we call *within-domain adaptation*. In this paradigm, a heterogeneous language source (such as WSJ) is treated as a complex product of multiple domains-of-discourse ("sublanguages"). The goal is then to produce a continuously modified model that tracks sublanguage mixtures, sublanguage shifts, style shifts, etc.

In contrast, a *cross-domain adaptation* paradigm is one in which the test data comes from a source to which the language model has never been exposed. The most salient aspect of this case is the large number of out-of-vocabulary words, as well as the high proportion of new bigrams and trigrams.

Cross-domain adaptation is most important in cases where no data from the test domain is available for training the system. But in practice this rarely happens. More likely, a limited amount of LM training can be obtained. Thus a hybrid paradigm, *limited-data domain*, might be the most important one for real-world applications.

The main disadvantage of the Maximum Entropy framework is the computational requirements of training the ME model. But these are not severe for modest amounts of training data (up to, say, 5M words, with current CPUs). The approach is thus particularly attractive in limited-data domains.

7. THE AP WIRE EXPERIMENT

We have already seen the effect of the amount of training data on perplexity reduction in the WSJ system. To test our adaptation mechanisms under both the cross-domain and limited-data paradigms, we constructed another experiment, this time using AP wire data for testing.

For measuring cross-domain adaptation, we used the 38MW WSJ models described above. For measuring limited-data adaptation, we used 5M words of AP wire to train a conventional compact backoff trigram, and a Maximum Entropy model, similar to the ones used by the WSJ system, except that the trigger pair list was copied from the WSJ system.

All models were tested on 420,000 words of unseen AP data. We chose the same "20o" vocabulary used in the WSJ experiments, to facilitate cross comparisons. As before, we measured perplexities of the baseline trigram model, the maximum Entropy component, and the interpolated adaptive model. Results are summarized in table 3.

To test error rate reduction under the cross-domain adaptation paradigm, we used 206 sentences, recorded by 3 male and 3 female speakers, under the same system configuration described in section . Results are reported in table 4.

8. SUMMARY

We described our latest attempt at adaptive language modeling. At the heart of our approach is a Maximum Entropy (ME) model, which incorporates many knowledge sources in a consistent manner. We have demonstrated that the ME model significantly improves on the conventional static trigram, a challenge which has evaded many past attempts([17, 18]). The approach is particularly applicable in domains with a modest amount of LM training data.

paradigm	cross-domain	limited-data
training data	38MW (WSJ)	5M (AP)
trigram (baseline) perplexity	206	170
Maximum Entropy perplexity	170	135
PP reduction	17%	21%
interpolated model perplexity	130	114
PP reduction	37%	33%

Table 3: Perplexity improvement of Maximum Entropy and interpolated adaptive models, for both cross-domain and limited-data adaptation, testing on 420KW of unseen AP wire data.

9. ACKNOWLEDGEMENTS

I am grateful to the entire CMU speech group, and many other individuals at CMU, for generously allowing me to monopolize their machines for weeks on end. I am particularly grateful to Lin Chase and Ravishankar Mosur for much needed help in designing and implementing the interface to SPHINX-II, to Alex Rudnicky for conditioning tools for the AP wire data, and to Raj Reddy for his support and encouragement.

The ideas for this work were developed during my 1992 summer visit with the Speech and Natural Language group at IBM Watson Research Center. I am grateful to Peter Brown, Stephen Della Pietra, Vincent Della Pietra, Raymond Lau, Bob Mercer and Salim Roukos for their very significant participation.

This research was sponsored by the Department of the Navy, Naval Research Laboratory under Grant No. N00014-93-1-2005. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

training data test data language model	38MW (WSJ) 206 sentences (AP) word error rate	% change
trigram (baseline)	22.1%	—
supervised adaptation	19.8%	-10%

Table 4: Word error rate reduction of the adaptive language model over a conventional trigram model, under the cross-domain adaptation paradigm.

References

1. Rosenfeld, R., "Adaptive Statistical Language Modeling: a Maximum Entropy Approach," *Ph.D. Thesis, Carnegie Mellon University*, April 1994.
2. Lau, R., Rosenfeld, R., Roukos, S., "Trigger-Based Language Models: a Maximum Entropy Approach," *Proceedings of ICASSP-93*, April 1993.
3. Lau, R., Rosenfeld, R., Roukos, S., "Adaptive Language Modeling Using the Maximum Entropy Principle", in *Proc. ARPA Human Language Technology Workshop*, March 1993.
4. Jaines, E. T., "Information Theory and Statistical Mechanics." *Phys. Rev.* **106**, pp. 620–630, 1957.
5. Kullback, S., *Information Theory in Statistics*. Wiley, New York, 1959.
6. Darroch, J.N. and Ratcliff, D., "Generalized Iterative Scaling for Log-Linear Models", *The Annals of Mathematical Statistics*, Vol. 43, pp 1470–1480, 1972.
7. Brown, P., Della Pietra, S., Della Pietra, V., Mercer, R., Nadas, A., and Roukos, S., "Maximum Entropy Methods and Their Applications to Maximum Likelihood Parameter Estimation of Conditional Exponential Models," *A forthcoming IBM technical report.*
8. Huang, X.D., Alleva, F., Hon, H.W., Hwang, M.Y., Lee, K.F. and Rosenfeld, R., "The SPHINX-II Speech Recognition System: An Overview." *Computer, Speech and Language*, 1992.
9. Rosenfeld, R., and Huang, X. D., "Improvements in Stochastic Language Modeling." *Proc. DARPA Speech and Natural Language Workshop*, February 1992.
10. Kuhn, R., "Speech Recognition and the Frequency of Recently Used Words: A Modified Markov Model for Natural Language." *12th International Conference on Computational Linguistics [COLING 88]*, pages 348-350, Budapest, August 1988.
11. Kubala, F. et al., "The Hub and Spoke Paradigm for CSR Evaluation," in *Proc. ARPA Human Language Technology Workshop*, March 1994.
12. Pallett, D.S., Fiscus, J.G., Fisher, W.M., Garofolo, J.S., Lund, B., and Pryzbocki, M., "1993 Benchmark Tests for the ARPA spoken Language Program", in *Proc. ARPA Human Language Technology Workshop*, March 1994.
13. Rosenfeld, R., "Language Model Adaptation in ARPA's CSR Evaluation", *ARPA Spoken Language Systems Workshop*, March 1994.
14. Huang, X.D., Alleva, F., Hon, H.W., Hwang, M.Y., Lee, K.F., and Rosenfeld, R., "The SPHINX-II Speech Recognition System: An Overview", *Computer, Speech and Language*, 1993.
15. Huang, X., Alleva, F., Hwang, M-Y, and Rosenfeld, R., "An Overview of the SPHINX-II Speech Recognition System", in *Proc. ARPA Human Language Technology Workshop*, March 1993.
16. Hwang, M., Rosenfeld, R., Thayer, E., Mosur, R., Chase, L., Weide, R., Huang, X., and Alleva, F., "Improving Speech-Recognition Performance Via Phone-Dependent VQ Codebooks, Multiple Speaker Clusters And Adaptive Language Models", *ARPA Spoken Language Systems Workshop*, March 1994.
17. Bahl, L., Brown, P., DeSouza, P., and Mercer, R., "A Tree-Based Statistical Language Model for natural Language Speech Recognition", *IEEE Transactions on Acustics, Speech and Signal Processing*, 37, pp. 1001-1008, 1989.
18. Jelinek, F., "Up From Trigrams!" Eurospeech 1991.