ADAPTIVE LANGUAGE MODELING USING THE MAXIMUM ENTROPY PRINCIPLE

Raymond Lau, Ronald Rosenfeld[*] *Salim Roukos*

IBM Research Division
Thomas J. Watson Research Center
Yorktown Heights, NY 10598

ABSTRACT

We describe our ongoing efforts at adaptive statistical language modeling. Central to our approach is the Maximum Entropy (ME) Principle, allowing us to combine evidence from multiple sources, such as long-distance triggers and conventional short-distance trigrams. Given consistent statistical evidence, a unique ME solution is guaranteed to exist, and an iterative algorithm exists which is guaranteed to converge to it. Among the advantages of this approach are its simplicity, its generality, and its incremental nature. Among its disadvantages are its computational requirements. We describe a succession of ME models, culminating in our current Maximum Likelihood / Maximum Entropy (ML/ME) model. Preliminary results with the latter show a 27% perplexity reduction as compared to a conventional trigram model.

1. STATE OF THE ART

Until recently, the most successful language model (given enough training data) was the trigram [1], where the probability of a word is estimated based solely on the two words preceding it. The trigram model is simple yet powerful [2]. However, since it does not use anything but the very immediate history, it is incapable of adapting to the style or topic of the document, and is therefore considered a *static* model.

In contrast, a dynamic or *adaptive* model is one that changes its estimates as a result of "seeing" some of the text. An adaptive model may, for example, rely on the history of the current document in estimating the probability of a word. Adaptive models are superior to static ones in that they are able to improve their performance after seeing some of the data. This is particularly useful in two situations. First, when a large heterogeneous language source is composed of smaller, more homogeneous segments, such as newspaper articles. An adaptive model trained on the heterogeneous source will be able to hone in on the particular "sublanguage" used in each of the articles. Secondly, when a model trained on data from one domain is used in another domain. Again, an adaptive model will be able to adjust to the new language, thus improving its performance.

The most successful adaptive LM to date is described in [3]. A cache of the last few hundred words is maintained, and is used to derive a "cache trigram". The latter is then interpolated with the static trigram. This results in a 23% reduction in perplexity, and a 5%–24% reduction in the error rate of a speech recognizer.

In what follows, we describe our efforts at improving our adaptive statistical language models by capitalizing on the information present in the document history.

2. TRIGGER-BASED MODELING

To extract information from the document history, we propose the idea of ***a trigger pair as the basic information bearing element.*** If a word sequence A is significantly correlated with another word sequence B, then $(A \rightarrow B)$ is considered a "trigger pair", with A being the *trigger* and B the *triggered sequence.* When A occurs in the document, it triggers B, causing its probability estimate to change.

Before attempting to design a trigger-based model, one should study what long distance factors have significant effects on word probabilities. Obviously, some information about $P(B)$ can be gained simply by knowing that A had occurred. But exactly how much? And can we gain significantly more by considering how recently A occurred, or how many times?

We have studied these issues using the a Wall Street Journal corpus of 38 million words. Some illustrations are given in figs. 1 and 2. As can be expected, different trigger pairs give different answers, and hence should be modeled differently. More detailed modeling should be used when the expected return is higher.

Once we determined the phenomena to be modeled, one main issue still needs to be addressed. Given the part of the document processed so far (h), and a word w considered for the next position, there are many different estimates of $P(w|h)$. These estimates are derived from the various triggers of w, from the static trigram model, and possibly from other sources. how do we combine them all to form one optimal estimate? We propose a solution to this problem in the next section.

[*] This work is now continued by Ron Rosenfeld at Carnegie Mellon University.

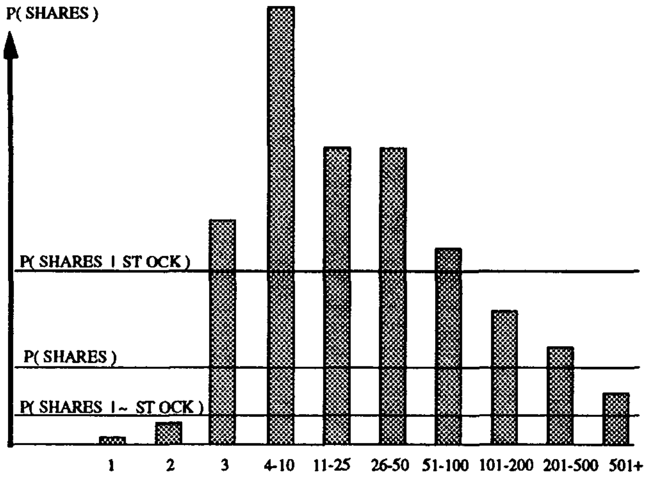

Figure 1: Probability of 'SHARES' as a function of the distance from the last occurrence of 'STOCK' in the same document. The middle horizontal line is the unconditional probability. The top (bottom) line is the probability of 'SHARES' given that 'STOCK' occurred (did not occur) before in the document.

3. MAXIMUM ENTROPY SOLUTIONS

Using several different probability estimates to arrive at one combined estimate is a general problem that arises in many tasks. We use the maximum entropy (ME) principle ([4, 5]), which can be summarized as follows:

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.

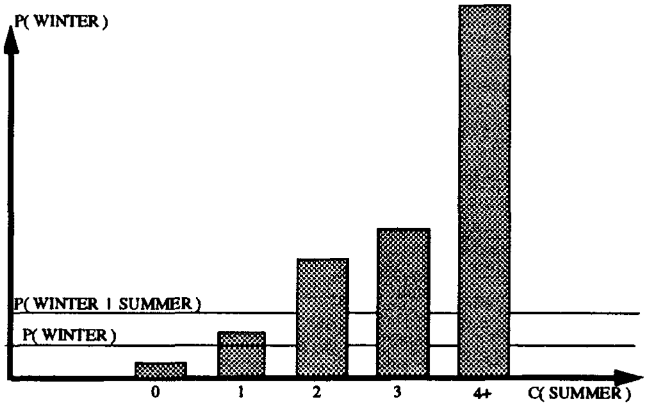

Figure 2: Probability of 'WINTER' as a function of the number of times 'SUMMER' occurred before it in the same document. Horizontal lines are as in fig. 1.

In the next 3 sections, we describe a succession of models we developed, all based on the ME principle. We then expand on the last model, describe possible future extensions to it, and report current results. More details can be found in [6, 7].

4. MODEL I: EARLY ATTEMPTS

Assume we have identified for each word w in a vocabulary, V, a set of n_w trigger words $t_{w1} t_{w2} \ldots t_{wn_w}$; we further assume that we have the relative frequency of observing a trigger word, t, occurring somewhere in the history, h, (in our case we have used a history length, K, of either 25, 50, 200, or 1000 words) and the word w just occurs after the history from some training text; denote the observed relative frequency of a trigger and a word w by

$$d(t, w) = \frac{c(t \in h \; and \, w \; immediatelyf \; ollows \; h)}{N}$$

where $c(.)$ is the count in the training data. We use $\{t, w\}$ to indicate the event that trigger t occurred in the history and word w occurs next; the term *long-distance* bigram has been used for this event.

Assume we have a joint distribution $p(h, w)$ of the history of K words and the next word w. We require this joint model to assign to the events $\{t, w\}$ a probability that matches the observed relative frequencies. Assuming we have R such constraints we find a model that has Maximum Entropy:

$$p^*(h, w) = \arg\max - \sum_{h,w} p(h, w) \lg p(h, w)$$

subject to the R trigger constraints[1]:

$$p(t, w) = \sum_{h:t \in h} p(h, w) = d(t, w)$$

We also include the case that none of the triggers of word w occur in the history (we denote this event by $\{t_0, w\}$.) Using Lagrange multipliers, one can easily show that the Maximum Entropy model is given by:

$$p(h, w) = \prod_{t:t \in h} \mu_{wt}$$

i.e., the joint probability is the product of $l_h(w)$ factors one factor for each trigger t_{wi} of word w that occurs in the history h (or one factor if none of the triggers occur.) The Maximum Entropy joint distribution over a space of $|V|^{K+1}$ is given by R parameters, one for each constraint. In our case, we used a maximum of 20 triggers per word for a 20k vocabulary with an average of 10 resulting in 200,000 constraints.

[1] we also imposed unigram constraints to match the unigram distribution of the vocabulary

4.1. How to determine the factors?

One can use the "Brown" algorithm to determine the set of factors. At each iteration, one updates the factor of one constraint and as long as one cycles through all constraints repeatedly the factors will converge to the optimal value. At the *i-th* iteration, assume we are updating the factor that corresponds to the $\{t, w\}$-constraint. Then the update is given by:

$$\mu_{wt}^{new} = \mu_{wt}^{old} \frac{d(t,w)}{m(t,w)}$$

where the model predicted value $m(t, w)$ is given by:

$$m(t,w) = \sum_{h:t\in h} p^{old}(h,w) \tag{1}$$

where p^{old} uses the old factor values.

Using the ME joint model, we define a conditional *unigram* model by:

$$p(w|h) = \frac{p^*(h,w)}{\sum_w p^*(h,w)}$$

This is a "time-varying" unigram model where the previous K words determine the relative probability that w would occur next. The perplexity of the resulting model was about 2000 much higher than the perplexity of a static unigram model. In particular, the model underestimated the probability of the frequent words. To ease that problem we disallowed any triggers for the most frequent L words. We experimented with L ranging from 100 to 500 words. The resulting model was better though its perplexity was still about 1100 which is 43% higher than the static unigram perplexity of 772. One reason that we conjecture was that the ME model gives a rather high probability for histories that are quite unlikely in reality and the trigger constraints are matched using those unrealistic histories. We tried an ad hoc computation where the summation over the histories in Equation 1 was weighed by a crude estimate, $w(h)$, of the probability of the history i.e. we used

$$m(t,w) = \sum_{h:t\in h} w(h) p^{old}(h,w)$$

The resulting model had a much lower perplexity of 559, about 27% lower than the static unigram model on a test set of (1927 words). This ad hoc computation indicates that we need to model the histories more realistically. The model we propose in the next section is derived from the viewpoint that ME indicates that R factors define a conditional model that captures the "long-distance" bigram constraints and that using this parametric form with Maximum Likelihood estimation may allow us to concentrate on typical histories that occur in the data.

5. MODEL II: ML OF CONDITIONAL ME

The ME viewpoint results in a conditional model that belongs to the exponential family with K parameters when K constraints are contemplated. We can use Maximum Likelihood estimation to estimate the K factors of the model. The log likelihood of a training set is given by:

$$\begin{aligned} L &= \sum_{t=0}^{N-1} \lg p(w_{t+1}|h_t) \\ &= \sum_{t=0}^{N-1} \lg \frac{\prod_{i\in I_{h_t}(w_{t+1})} \mu_i}{\sum_w \prod_{j\in J_{h_t}(w)} \mu_j} \end{aligned}$$

where $I_h(w)$ is the set of triggers for word w that occur in h. The convexity of the log likelihood guarantees that any hill climbing method will converge to the global optimum. The gradient can be shown to be:

$$\frac{\partial}{\partial \mu_{wt}} L = \frac{1}{\mu_{wt}} (d(t,w) - \sum_{h:t\in h} p(w|h))$$

one can use the gradient to iteratively re-estimate the factors by:

$$\mu_{wt}^{new} = \mu_{wt}^{old} + \frac{1}{\mu_{wt}^{old}} (d(t,w) - m'(t,w))$$

where the model predicted value $m'(t, w)$ for a constraint is:

$$m'(t,w) = \sum_{h:t\in h} p(w|h))$$

The training data is used to estimate the gradient given the current estimate of the factors. The size of the gradient step can be optimized by a line search on a small amount of training data.

Given the "time-varying" unigram estimate, we use the methods of [8] to obtain a bigram LM whose unigram matches the time-varying unigram using a window of the most recent L words.

6. CURRENT MODEL: ML/ME

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:

$$E_P f_i \overset{\text{def}}{=} \sum_{\mathbf{x}} P(\mathbf{x}) f_i(\mathbf{x}) = c_i . \tag{2}$$

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

$$P(\mathbf{x}) = \prod_i \mu_i^{f_i(\mathbf{x})} , \tag{3}$$

where the μ_i's are some unknown constants, to be found. Probability functions of the form (3) 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 exponential family defined by (3) for the μ_i's that will make $P(\mathbf{x})$ satisfy all the constraints, an iterative algorithm, "Generalized Iterative Scaling", exists, which is guaranteed to converge to the solution ([9]).

6.1. Formulating Triggers as Constraints

To reformulate a trigger pair $A \rightarrow B$ as a constraint, define the constraint function $f_{A \rightarrow B}$ as:

$$f_{A \rightarrow B}(h, w) = \begin{cases} 1 & \text{if } A \in h, w = B \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

Set $c_{A \rightarrow B}$ to $\tilde{E}[f_{A \rightarrow B}]$, the *empirical expectation* of $f_{A \rightarrow B}$ (ie its expectation in the training data). Now impose on the desired probability estimate $P(h, w)$ the constraint:

$$E_P\left[f_{A \rightarrow B}\right] = \tilde{E}\left[f_{A \rightarrow B}\right] \qquad (5)$$

6.2. Estimating Conditionals: The ML/ME Solution

Generalized Iterative Scaling can be used to find the ME estimate of a simple (non-conditional) probability distribution over some event space. But in our case, we need to estimate conditional probabilities of the form $P(w|h)$. How should this be done more efficiently than in the previous models?

An elegant solution was proposed by [10]. Let $P(h, w)$ 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 5 can be rewritten as:

$$\sum_h P(h) \cdot \sum_w P(w|h) \cdot f_i(h, w) = c_i \qquad (6)$$

We now modify the constraint to be:

$$\sum_h \tilde{P}(h) \cdot \sum_w P(w|h) \cdot f_i(h, w) = c_i \qquad (7)$$

One possible interpretation of this modification is as follows. Instead of constraining the expectation of $f_i(h, w)$ with regard to $P(h, w)$, we constrain its expectation with regard to a different probability distribution, say $Q(h, w)$, whose conditional $Q(w|h)$ is the same as that of P, but whose marginal $Q(h)$ is the same as that of $\tilde{P}$. To better understand the effect of this change, define H as the set of all possible histories h, and define H_{f_i} as the partition of H induced by f_i. Then the modification is equivalent to assuming that, for every constraint f_i, $P(H_{f_i}) = \tilde{P}(H_{f_i})$. Since typically H_{f_i} is a very small set, the assumption is reasonable.

The unique ME solution that satisfies equations like (7) or (6) can be shown to also be the Maximum Likelihood (ML) solution, namely that function which, among the exponential family defined by the constraints, has the maximum likelihood of generating the data. The identity of the ML and ME solutions, apart from being aesthetically pleasing, is extremely useful when estimating the conditional $P(w|h)$. It means that hillclimbing methods can be used in conjunction with Generalized Iterative Scaling to speed up the search. Since the likelihood objective function is convex, hillclimbing will not get stuck in local minima.

6.3. Incorporating the trigram model

We combine the trigger based model with the currently best static model, the N-Gram, by reformulating the latter to fit into the ML/ME paradigm. The usual unigram, bigram and trigram ML estimates are replaced by unigram, bigram and trigram constraints conveying the same information. Specifically, the constraint function for the unigram w_1 is:

$$f_{w_1}(h, w) = \begin{cases} 1 & \text{if } w = w1 \\ 0 & \text{otherwise} \end{cases} \qquad (8)$$

and its associated constraint is:

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

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

$$f_{w_1,w_2}(h, w) = \begin{cases} 1 & \text{if } h \text{ ends in } w_1 \text{ and } w = w2 \\ 0 & \text{otherwise} \end{cases} \qquad (10)$$

and its associated constraint is

$$\sum_h \tilde{P}(h) \sum_w P(w|h) f_{w_1,w_2}(h, w) = \tilde{E} f_{w_1,w_2}(h, w). \qquad (11)$$

and similarly for higher-order ngrams.

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 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 dominates the total time cost of the algorithm.

7. ME: PROS AND CONS

The ME principle and the Generalized Iterative Scaling algorithm have several important advantages:

1. The ME principle is simple and intuitively appealing. It imposes all of the constituent constraints, but assumes nothing else. For the special case of constraints derived from marginal probabilities, it is equivalent to assuming a lack of higher-order interactions [11].
2. ME is extremely general. Any probability estimate of any subset of the event space can be used, including estimates that were not derived from the data or that are

inconsistent with it. The distance dependence and count dependence illustrated in figs. 1 and 2 can be readily accommodated. Many other knowledge sources, including higher-order effects. can be incorporated. Note that constraints need not be independent of nor uncorrelated with each other.

3. The information captured by existing language models can be absorbed into the ML/ME model. We have shown how this is done for the conventional N-gram model. Later on we will show, how it can be done for the cache model of [3].

4. Generalized Iterative Scaling lends itself to incremental adaptation. New constraints can be added at any time. Old constraints can be maintained or else allowed to relax.

5. A unique ME solution is guaranteed to exist for consistent constraints. The Generalized Iterative Scaling algorithm is guaranteed to converge to it.

This approach also has the following weaknesses:

1. Generalized Iterative Scaling is computationally very expensive. When the complete system is trained on the entire 50 million words of Wall Street Journal data, it is expected to require many thousands of MIPS-hours to run to completion.

2. While the algorithm is guaranteed to converge, we do not have a theoretical bound on its convergence rate.

3. It is sometimes useful to impose constraints that are not satisfied by the training data. For example, we may choose to use Good-Turing discounting [12], or else the constraints may be derived from other data, or be externally imposed. Under these circumstances, the constraints may no longer be consistent, and the theoretical results guaranteeing existence, uniqueness and convergence may not hold.

8. INCORPORATING THE CACHE MODEL

It seems that the power of the cache model, described in section 1, comes from the "bursty" nature of language. Namely, infrequent words tend to occur in "bursts", and once a word occurred in a document, its probability of recurrence is significantly elevated.

Of course, this phenomena can be captured by a trigger pair of the form $A \rightarrow A$, which we call a "self trigger". We have done exactly that in [13]. We found that self triggers are responsible for a disproportionately large part of the reduction in perplexity. Furthermore, self triggers proved particularly robust: when tested in new domains, they maintained the correlations found in the training data better than the "regular" triggers did.

Thus self triggers are particularly important, and should be modeled separately and in more detail. The trigger model we currently use does not distinguish between one or more occurrences of a given word in the history, whereas the cache model does. For self-triggers, the additional information can be significant (see fig. 3).

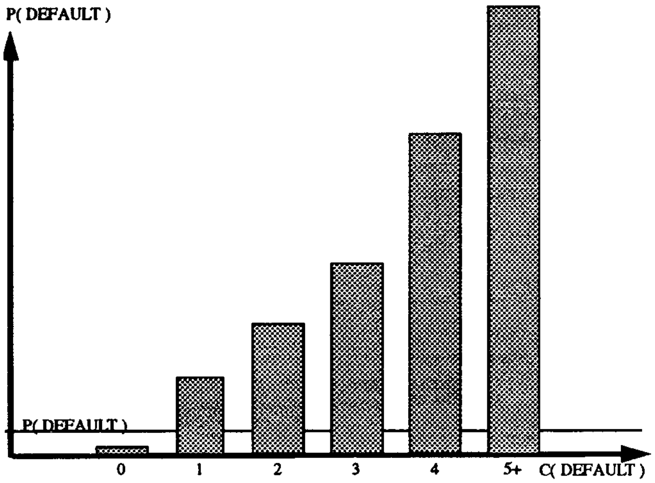

Figure 3: Behavior of a self-trigger: Probability of 'DEFAULT' as a function of the number of times it already occurred in the document. The horizontal line is the unconditional probability.

We plan to model self triggers in more detail. We will consider explicit modeling of frequency of occurrence, distance from last occurrence, and other factors. All of these aspects can easily be formulated as constraints and incorporated into the ME formalism.

9. RESULTS

The ML/ME model described above was trained on 5 million words of Wall Street Journal text, using DARPA's official "20o" vocabulary of some 20,000 words. A conventional trigram model was used as a baseline. The constraints used by the ML/ME model were: 18,400 unigram constraints, 240,000 bigram constraints, and 414,000 trigram constraints. One experiment was run with 36,000 trigger constraints (best 3 triggers for each word), and another with 65,000 trigger constraints (best 6 triggers per word). All models were trained on the same data, and evaluated on 325,000 words on independent data. The Maximum Entropy models were also interpolated with the conventional trigram, using yet unseen data for interpolation. Results are summarized in table 1.

model	Test-set Perplexity	% improvement over baseline
trigram	173	—
ML/ME-top3	134	23%
+trigram	129	25%
ML/ME-top6	130	25%
+trigram	127	27%

Table 1: Improvement of Maximum Likelihood / Maximum Entropy model over a conventional trigram model. Training is on 5 million words of WSJ text. Vocabulary is 20,000 words.

The trigger constraints used in this run were selected very crudely, and their number was not optimized. We believe much more improvement can be achieved. Special modeling of self triggers has not been implemented yet. Similarly, we expect it to yield further improvement.

10. ACKNOWLEDGEMENTS

We are grateful to Peter Brown, Stephen Della Pietra, Vincent Della Pietra and Bob Mercer for many suggestions and discussions.

Research by Ron Rosenfeld was sponsored by the Defense Advanced Research Projects Agency and monitored by the Space and Naval Warfare Systems Command under Contract N00039-91-C-0158, ARPA Order No. 7239. 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.

References

1. Bahl, L., Jelinek, F., Mercer, R.L., "A Statistical Approach to Continuous Speech Recognition," *IEEE Trans. on PAMI,* 1983.
2. Jelinek, F., "Up From Trigrams!" Eurospeech 1991.
3. Jelinek, F., Merialdo, B., Roukos, S., and Strauss, M., "A Dynamic Language Model for Speech Recognition." *Proceedings of the Speech and Natural Language DARPA Workshop,* pp.293–295, Feb. 1991.
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, 7].
6. Rosenfeld, R., "Adaptive Statistical Language Modeling: a Maximum Entropy Approach," *Ph.D. Thesis Proposal, Carnegie Mellon University,* September 1992.
7. Lau, R., Rosenfeld, R., Roukos, S., "Trigger-Based Language Models: a Maximum Entropy Approach," *Proceedings of ICASSP-93,* April 1993.
8. Della Pietra,S., Della Pietra, V., Mercer, R. L., Roukos, S., "Adaptive Language Modeling Using Minimum Discriminant Estimation," *Proceedings of ICASSP-92,* pp. I-633-636, San Francisco, March 1992.
9. Darroch, J.N. and Ratcliff, D., "Generalized Iterative Scaling for Log-Linear Models", *The Annals of Mathematical Statistics,* Vol. 43, pp 1470–1480, 1972.
10. 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.*
11. Good, I. J., "Maximum Entropy for Hypothesis Formulation, Especially for Multidimensional Contingency Tables." *Annals of Mathematical Statistics,* Vol. 34, pp. 911–934, 1963.
12. Good, I. J., "The Population Frequencies of Species and the Estimation of Population Parameters." *Biometrika,* Vol. 40, no. 3, 4, pp. 237–264, 1953.
13. Rosenfeld, R., and Huang, X. D., "Improvements in Stochastic Language Modeling." *Proceedings of the Speech and Natural Language DARPA Workshop,* Feb. 1992.