A Maximum Entropy Approach
to Adaptive Statistical Language Modeling

Ronald Rosenfeld
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213 USA
roni@cs.cmu.edu

May 21, 1996

Abstract

An adaptive statistical language model is described, which successfully integrates long distance linguistic information with other knowledge sources. Most existing statistical language models exploit only the immediate history of a text. To extract information from further back in the document's history, we propose and use *trigger pairs* as the basic information bearing elements. This allows the model to adapt its expectations to the topic of discourse. Next, statistical evidence from multiple sources must be combined. Traditionally, linear interpolation and its variants have been used, but these are shown here to be seriously deficient. Instead, we apply the principle of Maximum Entropy (ME). Each information source gives rise to a set of constraints, to be imposed on the combined estimate. The intersection of these constraints is the set of probability functions which are consistent with all the information sources. The function with the highest entropy within that set is the ME solution. Given consistent statistical evidence, a unique ME solution is guaranteed to exist, and an iterative algorithm exists which is guaranteed to converge to it. The ME framework is extremely general: any phenomenon that can be described in terms of statistics of the text can be readily incorporated. An adaptive language model based on the ME approach was trained on the Wall Street Journal corpus, and showed 32%–39% perplexity reduction over the baseline. When interfaced to SPHINX-II, Carnegie Mellon's speech recognizer, it reduced its error rate by 10%–14%. This thus illustrates the feasibility of incorporating many diverse knowledge sources in a single, unified statistical framework.

1 Introduction

Language modeling is the attempt to characterize, capture and exploit regularities in natural language. In statistical language modeling, large amounts of text are used to automatically determine the model's parameters, in a process known as *training*. Language modeling is useful in automatic speech recognition, machine translation, and any other application that processes natural language with incomplete knowledge.

1.1 View from Bayes Law

Natural language can be viewed as a stochastic process. Every sentence, document, or other contextual unit of text is treated as a random variable with some probability distribution. For example, in speech recognition, an acoustic signal A is given, and the goal is to find the linguistic hypothesis L that is most likely to have given rise to it. Namely, we seek the L that maximizes $\Pr(L|A)$. Using Bayes Law:

$$\begin{aligned}\arg\max_L \Pr(L|A) &= \arg\max_L \frac{\Pr(A|L)\cdot\Pr(L)}{\Pr(A)} \\ &= \arg\max_L \Pr(A|L)\cdot\Pr(L) \qquad (1)\end{aligned}$$

For a given signal A, $\Pr(A|L)$ is estimated by the *acoustic matcher*, which compares A to its stored models of all speech units. Providing an estimate for $\Pr(L)$ is the responsibility of the language model.

Let $L = w_1^n \overset{\text{def}}{=} w_1, w_2, \ldots w_n$, where the w_i's are the words that make up the hypothesis. One way to estimate $\Pr(L)$ is to use the chain rule:

$$\Pr(L) = \prod_{i=1}^{n} \Pr(w_i|w_1^{i-1})$$

Indeed, most statistical language models try to estimate expressions of the form $\Pr(w_i|w_1^{i-1})$. The latter is often written as $\Pr(w|h)$, where $h \overset{\text{def}}{=} w_1^{i-1}$ is called the *history*.

1.2 View from Information Theory

Another view of statistical language modeling is grounded in information theory. Language is considered an information source L ([Abramson 63]), which emits a sequence of symbols w_i from a finite alphabet (the vocabulary). The distribution of the next symbol is highly dependent on the identity of the previous ones — the source L is a high-order Markov chain.

The information source L has a certain inherent entropy H. This is the amount of non-redundant information conveyed per word, on average, by L. According to Shannon's theorem ([Shannon 48]), any encoding of L must use at least H bits per word, on average.

The quality of a language model M can be judged by its *cross entropy* with regard to the distribution $P_T(\mathbf{x})$ of some hitherto unseen text T:

$$H'(P_T; P_M) = -\sum_{\mathbf{x}} P_T(\mathbf{x}) \cdot \log P_M(\mathbf{x}) \qquad (2)$$

$H'(P_T; P_M)$ has also been called the *logprob* ([Jelinek 89]). Often, the *perplexity* ([Jelinek *et al.* 77]) of the text with regard to the model is reported. It is defined as:

$$\mathrm{PP}_M(T) = 2^{H'(P_T;P_M)} \qquad (3)$$

Using an ideal model, which capitalizes on every conceivable correlation in the language, L's cross entropy would equal its true entropy H. In practice, however, all models fall far short of that goal. Worse, the quantity H is not directly measurable (though it can be bounded, see [Shannon 51, Cover and King 78, Jelinek 89]). On the other extreme, if the correlations among the w_i's were completely ignored, the cross entropy of the source L would be $\sum_w \Pr_{\text{PRIOR}}(w) \log \Pr_{\text{PRIOR}}(w)$, where $\Pr_{\text{PRIOR}}(w)$ is the prior probability of w. This quantity is typically much greater than H. All other language models fall within this range.

Under this view, the goal of statistical language modeling is to identify and exploit sources of information in the language stream, so as to bring the cross entropy down, as close as possible to the true entropy. This view of statistical language modeling is dominant in this work.

2 Information Sources in the Document's History

There are many potentially useful information sources in the history of a document. It is important to assess their potential before attempting to incorporate them into a model. In this work, several different methods were used for doing so, including mutual information ([Abramson 63]), training-set perplexity (perplexity of the training data, see [Huang *et al.* 93]) and Shannon-style games ([Shannon 51]). See [Rosenfeld 94b] for more details. In this section we describe several information sources and various indicators of their potential.

2.1 Context-Free Estimation (Unigram)

The most obvious information source for predicting the current word w_i is the prior distribution of words. Without this "source", entropy is $\log V$, where V is the vocabulary size. When the priors are estimated from the training data, a Maximum Likelihood based model will have training-set cross-entropy[1] of $H' = -\sum_{w \in V} P(w) \log P(w)$. Thus the information provided by the priors is

$$H(w_i) - H(w_i | \langle \text{PRIORS} \rangle) = \log V + \sum_{w \in V} P(w) \log P(w) \tag{4}$$

2.2 Short-Term History (Conventional N-gram)

An N-gram ([Bahl *et al.* 83]) uses the last N-1 words of the history as its sole information source. Thus a bigram predicts w_i from w_{i-1}, a trigram predicts it from (w_{i-2}, w_{i-1}), and so on. The N-gram family of models are easy to implement and easy to interface to the application (e.g. to the speech recognizer's search component). They are very powerful, and surprisingly difficult to improve on ([Jelinek 91]). They seem to capture well short-term dependencies. It is for these reasons that they have become the staple of statistical language modeling. Unfortunately, they are also seriously deficient:

- They are completely "blind" to any phenomenon, or constraint, that is outside their limited scope. As a result, nonsensical and even ungrammatical utterances may receive high scores as long as they don't violate local constraints.

- The predictors in N-gram models are defined by their ordinal place in the sentence, not by their linguistic role. The histories "GOLD PRICES FELL TO" and "GOLD PRICES FELL YESTERDAY TO" seem very different to a trigram, yet they are likely to have a very similar effect on the distribution of the next word.

2.3 Short-term Class History (Class-Based N-gram)

The parameter space spanned by N-gram models can be significantly reduced, and reliability of estimates consequently increased, by clustering the words into *classes*. This can be done at many different levels: one or more of the predictors may be clustered, as may the predicted word itself. See [Bahl *et al.* 83] for more details.

The decision as to which components to cluster, as well as the nature and extent of the clustering, are examples of the detail-vs.-reliability tradeoff which is central to all modeling. In addition, one must decide on the clustering itself. There are three general methods for doing so:

1. Clustering by Linguistic Knowledge ([Jelinek 89, Derouault and Merialdo 86]).

2. Clustering by Domain Knowledge ([Price 90]).

[1] A smoothed unigram will have a slightly higher cross-entropy

3. Data Driven Clustering ([Jelinek 89, appendix C], [Jelinek 89, appendix D], [Brown *et al.* 90b], [Kneser and Ney 91], [Suhm and Waibel 94]).

See [Rosenfeld 94b] for a more detailed exposition.

2.4 Intermediate Distance

Long-distance N-grams attempt to capture directly the dependence of the predicted word on N-1–grams which are some distance back. For example, a distance-2 trigram predicts w_i based on (w_{i-3}, w_{i-2}). As a special case, distance-1 N-grams are the familiar conventional N-grams.

In [Huang *et al.* 93] we attempted to estimate the amount of information in long-distance bigrams. A long-distance bigram was constructed for distance $d = 1, \ldots, 10, 1000$, using the 1 million word Brown Corpus as training data. The distance-1000 case was used as a control, since at that distance no significant information was expected. For each such bigram, the *training-set* perplexity was computed. The latter is an indication of the average mutual information between word w_i and word w_{i-d}. As expected, we found perplexity to be low for $d = 1$, and to increase significantly as we moved through $d = 2, 3, 4,$ and 5. For $d = 6, \ldots, 10$, training-set perplexity remained at about the same level[2]. See table 1. We concluded that significant information exists in the last 5 words of the history.

distance	1	2	3	4	5	6	7	8	9	10	1000
PP	83	119	124	135	139	138	138	139	139	139	141

Table 1: *Training-set* perplexity of long-distance bigrams for various distances, based on 1 million words of the Brown Corpus. The distance=1000 case was included as a control.

Long-distance N-grams are seriously deficient. Although they capture word-sequence correlations even when the sequences are separated by distance d, they fail to appropriately merge training instances that are based on different values of d. Thus they unnecessarily fragment the training data.

2.5 Long Distance (Triggers)

2.5.1 Evidence for Long Distance Information

Evidence for the significant amount of information present in the longer-distance history is found in the following two experiments:

Long-Distance Bigrams. The previous section discusses the experiment on long-distance bigrams reported in [Huang *et al.* 93]. As mentioned, training-set perplexity was found to be low for the conventional bigram ($d = 1$), and to increase significantly as one moved through $d = 2, 3, 4,$ and 5. For $d = 6, \ldots, 10$, training-set perplexity remained at about the same level. But interestingly, that level was slightly yet consistently below perplexity of the $d = 1000$ case (see table 1). We concluded that some information indeed exists in the more distant past, but it is spread thinly across the entire history.

*Shannon Game at IBM [Mercer and Roukos 92]***.** A "Shannon game" program was implemented at IBM, where a person tries to predict the next word in a document while given access to the entire history of the document. The performance of humans was compared to that of a trigram language model. In particular, the cases where humans outsmarted the model were examined. It was found that in 40% of these cases, the predicted word, or a word related to it, occurred in the history of the document.

[2] although below the perplexity of the $d = 1000$ case. See the following section.

2.5.2 The Concept of a Trigger Pair

Based on the above evidence, we chose the *trigger pair* as the basic information bearing element for extracting information from the long-distance document history ([Rosenfeld 92]). 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.

How should trigger pairs be selected for inclusion in a model? Even if we restrict our attention to trigger pairs where A and B are both single words, the number of such pairs is too large. Let V be the size of the vocabulary. Note that, unlike in a bigram model, where the number of different consecutive word pairs is much less than V^2, the number of word pairs where both words occurred in the same document is a significant fraction of V^2.

Our goal is to estimate probabilities of the form $P(h, w)$ or $P(w|h)$. We are thus interested in correlations between the current word w and features in the history h. For clarity of exposition, we will concentrate on trigger relationships between single words, although the ideas carry over to longer sequences. Let W be any given word. Define the events W and $W_\circ$ over the joint event space (h, w) as follows:

$$\begin{array}{lll} W & : & \{\text{W=w, i.e. } W \text{ is the next word. }\} \\ W_\circ & : & \{W \in h \text{ , i.e. } W \text{ occurred anywhere in the document's history }\} \end{array}$$

When considering a particular trigger pair $(A \rightarrow B)$, we are interested in the correlation between the event $A_\circ$ and the event B. We can assess the significance of the correlation between $A_\circ$ and B by measuring their cross product ratio. But significance or even extent of correlation are not enough in determining the utility of a proposed trigger pair. Consider a highly correlated trigger pair consisting of two rare words, such as (BREST→LITOVSK), and compare it to a less-well-correlated, but much more common pair[3], such as (STOCK→BOND). The occurrence of BREST provides much more information about LITOVSK than the occurrence of STOCK does about BOND. Therefore, an occurrence of BREST in the test data can be expected to benefit our modeling more than an occurrence of STOCK. But since STOCK is likely to be much more common in the test data, its *average utility* may very well be higher. If we can afford to incorporate only one of the two trigger pairs into our model, (STOCK→BOND) may be preferable.

A good measure of the expected benefit provided by $A_\circ$ in predicting B is the average mutual information between the two (see for example [Abramson 63, p.106]):

$$\begin{aligned} I(A_\circ{:}B) \quad = \quad & P(A_\circ, B) \log \frac{P(B|A_\circ)}{P(B)} \quad + \quad P(A_\circ, \overline{B}) \log \frac{P(\overline{B}|A_\circ)}{P(\overline{B})} \\ & + \; P(\overline{A_\circ}, B) \log \frac{P(B|\overline{A_\circ})}{P(B)} \quad + \quad P(\overline{A_\circ}, \overline{B}) \log \frac{P(\overline{B}|\overline{A_\circ})}{P(\overline{B})} \end{aligned} \tag{5}$$

In a related work, [Church and Hanks 90] uses a variant of the first term of equation 5 to automatically identify co-locational constraints.

2.5.3 Detailed Trigger Relations

In the trigger relations considered so far, each trigger pair partitioned the history into two classes, based on whether the trigger occurred or did not occur in it (call these triggers *binary*). One might wish to model long-distance relationships between word sequences in more detail. For example, one might wish to consider how far back in the history the trigger last occurred, or how many times it occurred. In the last case, for example, the space of all possible histories is partitioned into several (> 2) classes, each corresponding to a particular number of times a trigger occurred. Equation 5 can then be modified to measure the amount of information conveyed on average by this many-way classification.

[3] in the WSJ corpus, at least.

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 can significantly more be gained by considering how recently A occurred, or how many times?

We have studied these issues using the Wall Street Journal corpus of 38 million words. First, an index file was created that contained, for every word, a record of all of its occurrences. Then, for any candidate pair of words, we computed log cross product ratio, average mutual information (MI), and distance-based and count-based co-occurrence statistics. The latter were used to draw graphs depicting detailed trigger relations. Some illustrations are given in figs. 2 and 3. After using the program to manually browse through many hundreds of trigger pairs, we were able to draw the following general conclusions:

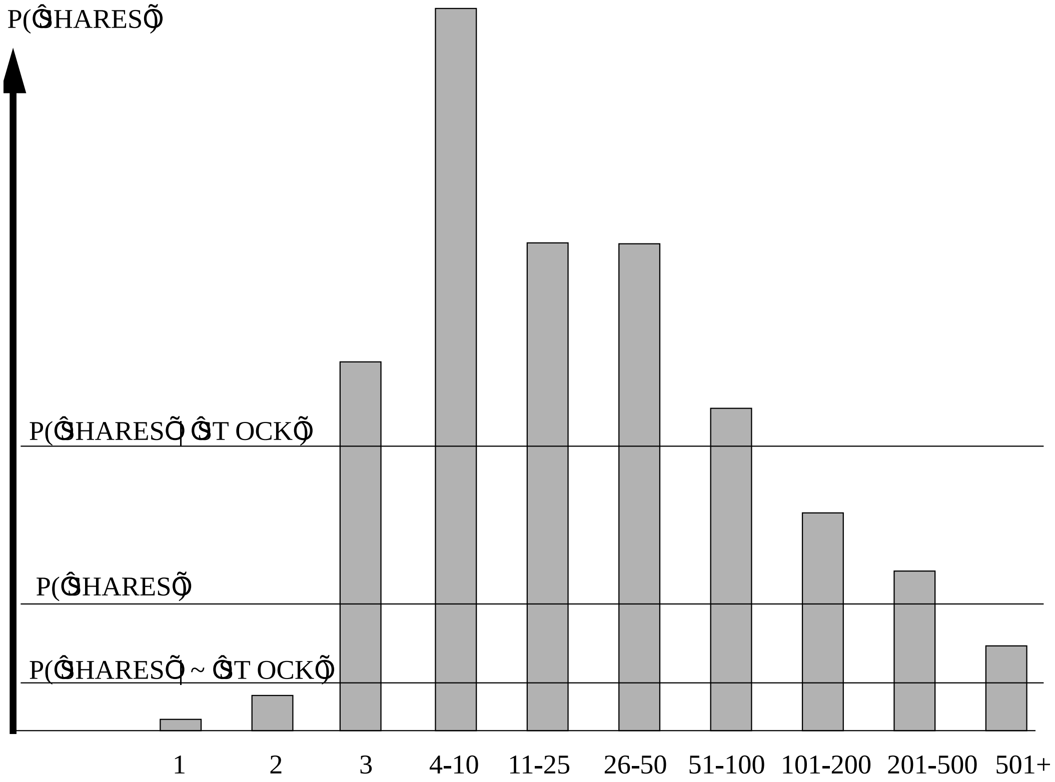

Figure 2: Probability of ’SHARES’ as a function of the distance from the last occurrence of ’STOCK’ in the same document. The middle horizontal line is the unconditional probability. The top (bottom) line is the probability of ’SHARES’ given that ’STOCK’ occurred (did not occur) before in the document.

1. Different trigger pairs display different behavior, and hence should be modeled differently. More detailed modeling should be used when the expected return is higher.

2. *Self triggers* (i.e. triggers of the form $(A \rightarrow A)$) are particularly powerful and robust. In fact, for more than two thirds of the words, the highest-MI trigger proved to be the word itself. For 90% of the words, the self-trigger was among the top 6 triggers.

3. Same-root triggers are also generally powerful, depending on the frequency of their inflection.

4. Most of the potential of triggers is concentrated in high-frequency words. (STOCK→BOND) is indeed much more useful than (BREST→LITOVSK).

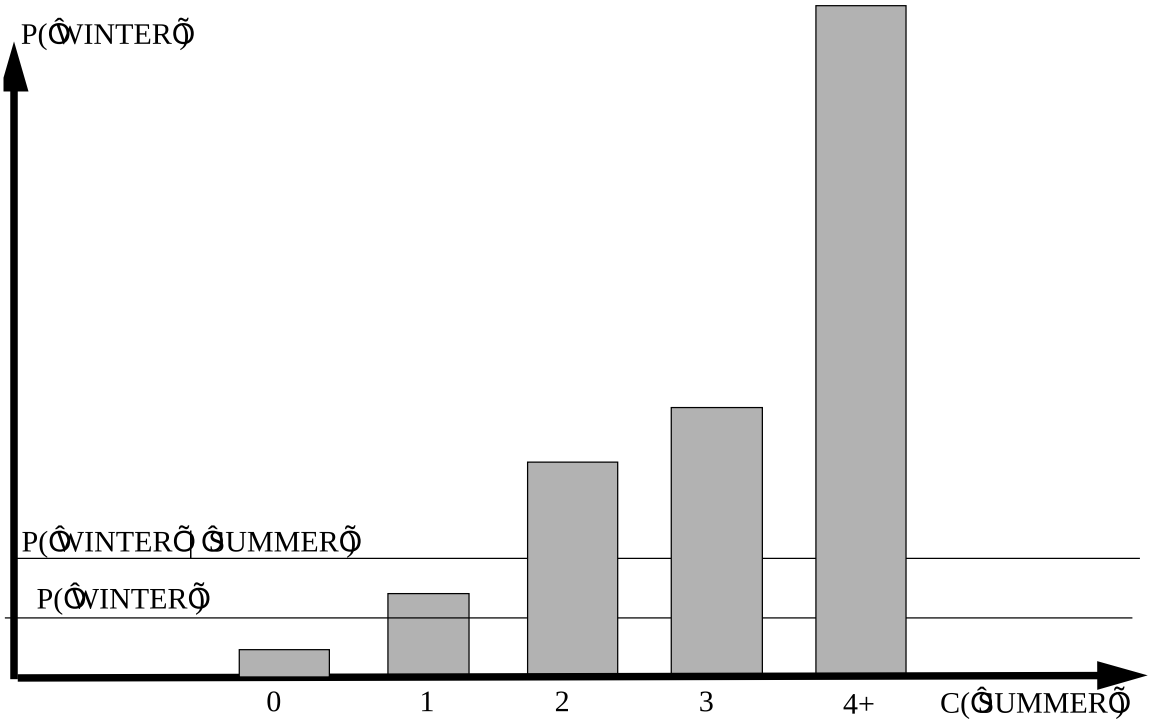

Figure 3: Probability of ’WINTER’ as a function of the number of times ’SUMMER’ occurred before it in the same document. Horizontal lines are as in fig. 2.

5. When the trigger and triggered words are from different domains of discourse, the trigger pair actually shows some slight mutual information. The occurrence of a word like ’STOCK’ signifies that the document is probably concerned with financial issues, thus reducing the probability of words characteristic of other domains. Such *negative triggers* can in principle be exploited in much the same way as regular, “positive” triggers. However, the amount of information they provide is typically very small.

2.6 Syntactic Constraints

Syntactic constraints are varied. They can be expressed as yes/no decisions about grammaticality, or, more cautiously, as scores, with very low scores assigned to ungrammatical utterances.

The extraction of syntactic information would typically involve a parser. Unfortunately, parsing of general English with reasonable coverage is not currently attainable. As an alternative, phrase parsing can be used. Another possibility is loose semantic parsing ([Ward 90, Ward 91]), extracting syntactic-semantic information.

The information content of syntactic constraints is hard to measure quantitatively. But they are likely to be very beneficial. This is because this knowledge source seems complementary to the statistical knowledge sources we can currently tame. Many of the speech recognizer’s errors are easily identified as such by humans because they violate basic syntactic constraints.

3 Combining Information Sources

Once the desired information sources are identified and the phenomena to be modeled are determined, 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 different knowledge sources. How does one combine them all to form one optimal estimate? We discuss existing solutions in this section, and propose a new one in the next.

3.1 Linear Interpolation

Given k models $\{P_i(w|h)\}_{i=1...k}$, we can combine them linearly with:

$$P_{\text{COMBINED}}(w|h) \overset{\text{def}}{=} \sum_{i=1}^{k} \lambda_i P_i(w|h) \tag{6}$$

where $0 < \lambda_i \leq 1$ and $\sum_i \lambda_i = 1$.

This method can be used both as a way of combining knowledge sources, and as a way of smoothing (when one of the component models is very "flat", such as a uniform distribution). An Estimation-Maximization (EM) type algorithm ([Dempster *et al.* 77]) is typically used to determine these weights. The result is a set of weights that is provably optimal with regard to the data used for its optimization. See [Jelinek and Mercer 80] for more details, and [Rosenfeld 94b] for further exposition.

Linear interpolation has very significant advantages, which make it the method of choice in many situations:

- **Linear Interpolation is extremely general.** Any language model can be used as a component. In fact, once a common set of heldout data is selected for weight optimization, the component models need no longer be maintained explicitly. Instead, they can be represented in terms of the probabilities they assign to the heldout data. Each model is represented as an array of probabilities. The EM algorithm simply looks for a linear combination of these arrays that would minimize perplexity, and is completely unaware of their origin.

- **Linear interpolation is easy to implement, experiment with, and analyze.** We have created an `interpolate` program that takes any number of probability streams, and an optional bin-partitioning stream, and runs the EM algorithm to convergence (see [Rosenfeld 94b, Appendix B]). We have used the program to experiment with many different component models and bin-classification schemes. Some of our general conclusions are:

  1. The exact value of the weights does not significantly affect perplexity. Weights need only be specified to within ~5% accuracy.
  2. Very little heldout data (several thousand words per weight or less) are enough to arrive at reasonable weights.

- **Linear interpolation cannot hurt.** The interpolated model is guaranteed to be no worse than any of its components. This is because each of the components can be viewed as a special case of the interpolation, with a weight of 1 for that component and 0 for all others. Strictly speaking, this is only guaranteed for the heldout data, not for new data. But if the heldout data set is large enough, the result will carry over. So, if we suspect that a new knowledge source can contribute to our current model, the quickest way to test it would be to build a simple model that uses that source, and to interpolate it with our current one. If the new source is not useful, it will simply be assigned a very small weight by the EM algorithm ([Jelinek 89]).

Linear interpolation is so advantageous because it reconciliates the different information sources in a straightforward and simple-minded way. But that simple-mindedness is also the source of its weaknesses:

- **Linearly interpolated models make suboptimal use of their components.** The different information sources are consulted "blindly", without regard to their strengths and weaknesses in particular contexts. Their weights are optimized globally, not locally (the “bucketing” scheme is an attempt to remedy this situation piece-meal). Thus the combined model does not make optimal use of the information at its disposal.

For example, in section 2.4 we discussed [Huang *et al.* 93], and reported our conclusion that a significant amount of information exists in long-distance bigrams, up to distance 4. We have tried to incorporate this information by combining these components using linear interpolation. But the combined model improved perplexity over the conventional (distance 1) bigram by an insignificant amount (2%). In section 5 we will see how a similar information source can contribute significantly to perplexity reduction, provided a better method of combining evidence is employed.

As another, more detailed, example, in [Rosenfeld and Huang 92] we report on our early work on trigger models. We used a trigger utility measure, closely related to mutual information, to select some 620,000 triggers. We combined evidence from multiple triggers using several variants of linear interpolation, then interpolated the result with a conventional backoff trigram. An example result is in table 4. The 10% reduction in perplexity, however gratifying, is well below the true potential of the triggers, as will be demonstrated in the following sections.

test set	trigram PP	trigram+triggers PP	improvement
70KW (WSJ)	170	153	10%

Table 4: Perplexity reduction by linearly interpolating the trigram with a trigger model. See [Rosenfeld and Huang 92] for details.

- **Linearly interpolated models are generally inconsistent with their components.** Each information source typically partitions the event space (h, w) and provides estimates based on the relative frequency of training data within each class of the partition. Therefore, within each of the component models, the estimates are consistent with the marginals of the training data. But this reasonable measure of consistency is in general violated by the interpolated model.

  For example, a bigram model partitions the event space according to the last word of the history. All histories that end in, say, “BANK” are associated with the same estimate, $P_{\text{BIGRAM}}(w|h)$. That estimate is consistent with the portion of the training data that ends in “BANK”, in the sense that, for every word w,

$$\sum_{\substack{h \in \text{ TRAINING-SET} \\ h \text{ ends in “BANK”}}} P_{\text{BIGRAM}}(w|h) = C(\text{BANK}, w) \tag{7}$$

  where $C(\text{BANK}, w)$ is the training-set count of the bigram (BANK, w). However, when the bigram component is linearly interpolated with another component, based on a different partitioning of the data, the combined model depends on the assigned weights. These weights are in turn optimized *globally*, and are thus influenced by the other marginals and by other partitions. As a result, equation 7 generally does not hold for the interpolated model.

3.2 Backoff

In the backoff method ([Katz 87]), the different information sources are ranked in order of detail or specificity. At runtime, the most detailed model is consulted first. If it is found to contain enough information about the predicted word in the current context, then that context is used exclusively to generate the estimate. Otherwise, the next model in line is consulted. As in the previous case, backoff can be used both as a way of combining information sources, and as a way of smoothing.

The backoff method does not actually reconcile multiple models. Instead, it chooses among them. One problem with this approach is that it exhibits a discontinuity around the point where the backoff decision is made. In spite of this problem, backing off is simple, compact, and often better than linear interpolation.

A problem common to both linear interpolation and backoff is that they give rise to systematic overestimation of some events. This problem was discussed and solved in [Rosenfeld and Huang 92], and the solution used in a speech recognition system in [Chase *et al.* 94].

4 The Maximum Entropy Principle

In this section we discuss an alternative method of combining knowledge sources, which is based on the Maximum Entropy approach first proposed by E. T. Jaynes in the 1950's ([Jaynes 57]). The Maximum Entropy principle was first applied to language modeling by [DellaPietra *et al.* 92].

In the methods described in the previous section, each knowledge source was used separately to construct a model, and the models were then combined. Under the Maximum Entropy approach, one does not construct separate models. Instead, one builds a single, combined model, which attempts to capture all the information provided by the various knowledge sources. Each such knowledge source gives rise to a set of *constraints*, to be imposed on the combined model. These constraints are typically expressed in terms of marginal distributions, as in the example at the end of section 3.1. This solves the inconsistency problem discussed in that section.

The intersection of all the constraints, if not empty, contains a (possibly infinite) set of probability functions, which are all consistent with the knowledge sources. The second step in the Maximum Entropy approach is to choose, from among the functions in that set, that function which has the highest entropy (i.e., the "flattest" function). In other words, once the desired knowledge sources have been incorporated, no other features of the data are assumed about the source. Instead, the "worst" (flattest) of the remaining possibilities is chosen.

Let us illustrate these ideas with a simple example.

4.1 An Example

Assume we wish to estimate $P(\text{“BANK}''|h)$, namely the probability of the word "BANK" given the document's history. One estimate may be provided by a conventional bigram. The bigram would partition the event space (h, w) based on the last word of the history. The partition is depicted graphically in figure 5. Each column is an equivalence class in this partition.

h ends in "THE"	*h* ends in "OF"	...	...
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

Table 5: The Event Space $\{(h, w)\}$ is partitioned by the bigram into equivalence classes (depicted here as columns). In each class, all histories end in the same word.

Consider one such equivalence class, say, the one where the history ends in "THE". The bigram assigns *the same probability estimate* to all events in that class:

$$P_{\text{BIGRAM}}(\text{BANK}|\text{THE}) = K_{\{\text{THE},\text{BANK}\}} \tag{8}$$

That estimate is derived from the distribution of the training data in that class. Specifically, it is derived as:

$$K_{\{\text{THE},\text{BANK}\}} \overset{\text{def}}{=} \frac{C(\text{THE, BANK})}{C(\text{THE})} \tag{9}$$

Another estimate may be provided by a particular trigger pair, say (LOAN→BANK). Assume we want to capture the dependency of "BANK" on whether or not "LOAN" occurred before it in the same document. Thus a different partition of the event space will be added, as in figure 6. Each of the two rows is an equivalence class in this partition[4].

	h ends in "THE"	h ends in "OF"	. . .	. . .
LOAN $\in h$	· . . . . . ·	· . . . . . ·	· . . . . . ·	· . . . . . ·
LOAN $\notin h$	· . . . . . ·	· . . . . . ·	· . . . . . ·	· . . . . . ·

Table 6: The Event Space $\{(h,w)\}$ is independently partitioned by the binary trigger word "LOAN" into another set of equivalence classes (depicted here as rows).

Similarly to the bigram case, consider now one such equivalence class, say, the one where "LOAN" did occur in the history. The trigger component assigns *the same probability estimate* to all events in that class:

$$P_{\text{LOAN}\rightarrow\text{BANK}}(\text{BANK}|\text{LOAN}\in h) \;=\; K_{\{\text{BANK},\text{LOAN}\in h\}} \tag{10}$$

That estimate is derived from the distribution of the training data in that class. Specifically, it is derived as:

$$K_{\{\text{BANK},\text{LOAN}\in h\}} \overset{\text{def}}{=} \frac{C(\text{BANK}, \text{LOAN}\in h)}{C(\text{LOAN}\in h)} \tag{11}$$

Thus the bigram component assigns the same estimate to all events in the same column, whereas the trigger component assigns the same estimate to all events in the same row. These estimates are clearly mutually inconsistent. How can they be reconciled?

Linear interpolation solves this problem by averaging the two answers. The backoff method solves it by choosing one of them. The Maximum Entropy approach, on the other hand, does away with the inconsistency by *relaxing the conditions imposed by the component sources.*

Consider the bigram. Under Maximum Entropy, we no longer insist that $P(\text{BANK}|h)$ always have the same value ($K_{\{\text{THE},\text{BANK}\}}$) whenever the history ends in "THE". Instead, we acknowledge that the history may have other features that affect the probability of "BANK". Rather, we only require that, in the combined estimate, $P(\text{BANK}|h)$ be equal to $K_{\{\text{THE},\text{BANK}\}}$ *on average in the training data*. Equation 8 is replaced by

$$\underset{h \text{ ends in "THE"}}{\mathbf{E}} \; [\; P_{\text{COMBINED}}(\text{BANK}|h) \;] \;=\; K_{\{\text{THE},\text{BANK}\}} \tag{12}$$

where **E** stands for an expectation, or average. Note that the constraint expressed by equation 12 is much weaker than that expressed by equation 8. There are many different functions P_{COMBINED} that would satisfy it. Only one degree of freedom was removed by imposing this new constraint, and many more remain.

[4] The equivalence classes are depicted graphically as rows and columns for clarity of exposition only. In reality, they need not be orthogonal.

Similarly, we require that $P_{\text{COMBINED}}(\text{BANK}|h)$ be equal to $K_{\{\text{BANK},\text{LOAN}\in h\}}$ *on average* over those histories that contain occurrences of "LOAN":

$$\mathop{\mathbf{E}}_{\text{“LOAN”}\in h} \; [\; P_{\text{COMBINED}}(\text{BANK}|h) \;] \;=\; K_{\{\text{BANK},\text{LOAN}\in h\}} \tag{13}$$

As in the bigram case, this constraint is much weaker than that imposed by equation 10.

Given the tremendous number of degrees of freedom left in the model, it is easy to see why the intersection of all such constraints would be non-empty. The next step in the Maximum Entropy approach is to find, among all the functions in that intersection, the one with the highest entropy. The search is carried out implicitly, as will be described in section 4.3.

4.2 Information Sources as Constraint Functions

Generalizing from the example above, we can view each information source as defining a subset (or many subsets) of the event space (h,w). For each subset, we impose a constraint on the combined estimate to be derived: that it agree on average with a certain statistic of the training data, defined over that subset. In the example above, the subsets were defined by a partition of the space, and the statistic was the marginal distribution of the training data in each one of the equivalence classes. But this need not be the case. We can define *any subset S* of the event space, and *any desired expectation K*, and impose the constraint:

$$\sum_{(h,w)\in S} [\; P(h,w) \;] \;=\; K \tag{14}$$

The subset S can be specified by an *index function*, also called *selector function*, f_S:

$$f_S(h,w) \;\stackrel{\text{def}}{=}\; \begin{cases} 1 & \text{if } (h,w)\in S \\ 0 & \text{otherwise} \end{cases}$$

so equation 14 becomes:

$$\sum_{(h,w)} [\; P(h,w) f_s(h,w) \;] \;=\; K \tag{15}$$

This notation suggests further generalization. We need not restrict ourselves to index functions. Any real-valued function $f(h,w)$ can be used. We call $f(h,w)$ *a constraint function*, and the associated K the *desired expectation*. Equation 15 now becomes:

$$\langle f\,,P\rangle \;=\; K \tag{16}$$

This generalized constraint suggests a new interpretation: $\langle f\,,P\rangle$ is the expectation of $f(h,w)$ under the desired distribution $P(h,w)$. We require of $P(h,w)$ to be such that the expectation of some given functions $\{f_i(h,w)\}_{i=1,2,\ldots}$ match some desired values $\{K_i\}_{i=1,2,\ldots}$, respectively.

The generalizations introduced above are extremely important, because they mean that any correlation, effect, or phenomenon that can be described in terms of statistics of (h,w) can be readily incorporated into the Maximum Entropy model. All information sources described in the previous section fall into this category, as do all other information sources that can be described by an algorithm.

Following is a general description of the Maximum Entropy model and its solution.

4.3 Maximum Entropy and the Generalized Iterative Scaling Algorithm

The Maximum Entropy (ME) Principle ([Jaynes 57, Kullback 59]) can be stated as follows:

1. Reformulate the different information sources as constraints 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.

Given a general event space $\{\mathbf{x}\}$, to derive a combined probability function $P(\mathbf{x})$, each constraint i is associated with a *constraint function* $f_i(\mathbf{x})$ and a *desired expectation* K_i. The constraint is then written as:

$$E_P f_i \overset{\text{def}}{=} \sum_{\mathbf{x}} P(\mathbf{x}) f_i(\mathbf{x}) = K_i \; . \tag{17}$$

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

$$P(\mathbf{x}) = \prod_i \mu_i^{\,f_i(\mathbf{x})} \; , \tag{18}$$

where the μ_i's are some unknown constants, to be found. To search the exponential family defined by (18) for the μ_i's that will make $P(\mathbf{x})$ satisfy all the constraints, an iterative algorithm, “Generalized Iterative Scaling” (GIS, [Darroch and Ratcliff 72]), exists, which is guaranteed to converge to the solution. GIS starts with some arbitrary $\mu_i^{(0)}$ values, which define the initial probability estimate:

$$P^{(0)}(\mathbf{x}) \overset{\text{def}}{=} \prod_i \mu_i^{(0)\,f_i(\mathbf{x})}$$

Each iteration creates a new estimate, which is improved in the sense that it matches the constraints better than its predecessor. Each iteration (say j) consists of the following steps:

1. Compute the expectations of all the f_i's under the current estimate function. Namely, compute $E_{P^{(j)}} f_i \overset{\text{def}}{=} \sum_{\mathbf{x}} P^{(j)}(\mathbf{x}) f_i(\mathbf{x})$.
2. Compare the *actual* values ($E_{P^{(j)}} f_i$'s) to the *desired* values (K_i's), and update the μ_i's according to the following formula:
$$\mu_i^{(j+1)} = \mu_i^{(j)} \cdot \frac{K_i}{E_{P^{(j)}} f_i} \tag{19}$$
3. Define the next estimate function based on the new μ_i's:
$$P^{(j+1)}(\mathbf{x}) \overset{\text{def}}{=} \prod_i \mu_i^{(j+1)\,f_i(\mathbf{x})} \tag{20}$$

Iterating is continued until convergence or near-convergence.

4.4 Estimating Conditional Distributions

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

One simple way is to estimate the joint, $P(h, w)$, from which the conditional, $P(w|h)$, can be readily derived. This has been tried, with moderate success only [Lau *et al.* 93b]. The likely reason is that the event space $\{(h, w)\}$ is of size $O(V^{L+1})$, where V is the vocabulary size and L is the history length. For any reasonable values of V and L, this is a huge space, and no feasible amount of training data is sufficient to train a model for it.

A better method was later proposed by [Brown *et al.*]. Let $P(h,w)$ be the desired probability estimate, and let $\tilde{\mathrm{P}}(h,w)$ be the empirical distribution of the training data. Let $f_i(h,w)$ be any constraint function, and let K_i be its desired expectation. Equation 17 can be rewritten as:

$$\sum_h P(h) \cdot \sum_w P(w|h) \cdot f_i(h,w) \ = \ K_i \tag{21}$$

We now modify the constraint to be:

$$\sum_h \tilde{\mathrm{P}}(h) \cdot \sum_w P(w|h) \cdot f_i(h,w) \ = \ K_i \tag{22}$$

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 $\tilde{\mathrm{P}}$. 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, $P(H_{f_i}) = \tilde{\mathrm{P}}(H_{f_i})$. Since typically H_{f_i} is a very small set, the assumption is reasonable. It has several significant benefits:

1. Although $Q(w|h) = P(w|h)$, modeling $Q(h,w)$ is much more feasible than modeling $P(h,w)$, since $Q(h,w) = 0$ for all but a minute fraction of the h's.
2. When applying the Generalized Iterative Scaling algorithm, we no longer need to sum over all possible histories (a very large space). Instead, we only sum over the histories that occur in the training data.
3. The unique ME solution that satisfies equations like (22) 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 training 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.

4.5 Maximum Entropy and Minimum Discrimination Information

The principle of Maximum Entropy can be viewed as a special case of the *Minimum Discrimination Information* (MDI) principle. Let $P_0(\mathbf{x})$ be a prior probability function, and let $\{Q_\alpha(\mathbf{x})\}_\alpha$ be a family of probability functions, where α varies over some set. As in the case of Maximum Entropy, $\{Q_\alpha(\mathbf{x})\}_\alpha$ might be defined by an intersection of constraints. One might wish to find the function $Q_0(\mathbf{x})$ in that family which is closest to the prior $P_0(\mathbf{x})$:

$$Q_0(\mathbf{x}) \ \overset{\text{def}}{=} \ \arg\min_\alpha D(Q_\alpha, P_0) \tag{23}$$

where the non-symmetric distance measure, $D(Q,P)$, is the Kullback-Liebler distance, also known as discrimination information or asymmetric divergence [Kullback 59]:

$$D(Q(\mathbf{x}), P(\mathbf{x})) \ \overset{\text{def}}{=} \ \sum_{\mathbf{x}} Q(\mathbf{x}) \log \frac{Q(\mathbf{x})}{P(\mathbf{x})} \tag{24}$$

In the special case when $P_0(\mathbf{x})$ is the uniform distribution, $Q_0(\mathbf{x})$ as defined by equation 23 is also the Maximum Entropy solution, namely the function with the highest entropy in the family $\{Q_\alpha(\mathbf{x})\}_\alpha$. We see thus that ME is a special case of MDI, where the distance is measured to the uniform distribution.

In a precursor to this work, [DellaPietra *et al.* 92] used the history of a document to construct a unigram. The latter was used to constrain the marginals of a bigram. The static bigram was used as the prior, and the MDI solution was sought among the family defined by the constrained marginals.

4.6 Assessing the Maximum Entropy Approach

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 [Good 63].

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. Many other knowledge sources can be incorporated, such as distance-dependent correlations and complicated higher-order effects. 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 ME model. Later on in this document we will show how this is done for the conventional *N*-gram model.

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 (For more on this problem, and on methods for coping with it, see [Rosenfeld 94b, section 5.7]).

2. While the algorithm is guaranteed to converge, we do not have a theoretical bound on its convergence rate (for all systems we tried, convergence was achieved within 10-20 iterations).

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 [Good 53] (as we have indeed done in this work), or else the constraints may be derived from other data, or be externally imposed. Under these circumstances, equivalence with the Maximum Likelihood principle no longer exists. More importantly, the constraints may no longer be consistent, and the theoretical results guaranteeing existence, uniqueness and convergence may not hold.

5 Using Maximum Entropy in Language Modeling

In this section, we describe how the Maximum Entropy framework was used to create a language model which tightly integrates varied knowledge sources.

5.1 Distance-1 N-grams

5.1.1 Conventional Formulation

In the conventional formulation of standard *N*-grams, the usual unigram, bigram and trigram Maximum Likelihood estimates are replaced by unigram, bigram and trigram constraints conveying the same information. Specifically, the constraint function for the unigram w_1 is:

$$f_{w_1}(h, w) = \begin{cases} 1 & \text{if } w = w_1 \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

The desired value, K_{w_1}, is set to $\tilde{\mathrm{E}}\,[f_{w_1}]$, the *empirical expectation* of f_{w_1}, i.e. its expectation in the training data:

$$\tilde{\mathrm{E}}\,[f_{w_1}] \overset{\text{def}}{=} \frac{1}{N} \sum_{(h,w)\in\text{TRAINING}} f_{w_1}(h,w), \tag{26}$$

and the associated constraint is:

$$\sum_h \tilde{\mathrm{P}}(h) \sum_w P(w|h) f_{w_1}(h,w) = \tilde{\mathrm{E}}\,[f_{w_1}]. \tag{27}$$

(As before, $\tilde{\mathrm{P}}()$ denotes the empirical distribution.) Similarly, the constraint function for the bigram $\{w_1, w_2\}$ is:

$$f_{\{w_1,w_2\}}(h,w) = \begin{cases} 1 & \text{if } h \text{ ends in } w_1 \text{ and } w = w_2 \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

and its associated constraint is:

$$\sum_h \tilde{\mathrm{P}}(h) \sum_w P(w|h) f_{\{w_1,w_2\}}(h,w) = \tilde{\mathrm{E}}\,[f_{\{w_1,w_2\}}]. \tag{29}$$

Finally, the constraint function for the trigram $\{w_1, w_2, w_3\}$ is:

$$f_{\{w_1,w_2,w_3\}}(h,w) = \begin{cases} 1 & \text{if } h \text{ ends in } (w_1,w_2) \text{ and } w = w_3 \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

and its associated constraint is:

$$\sum_h \tilde{\mathrm{P}}(h) \sum_w P(w|h) f_{\{w_1,w_2,w_3\}}(h,w) = \tilde{\mathrm{E}}\,[f_{\{w_1,w_2,w_3\}}]. \tag{31}$$

5.1.2 Complemented N-gram Formulation

Each constraint in a ME model induces a subset of the event space $\{(h,w)\}$. One can modify the N-gram constraints by modifying their respective subsets. In particular, the following set subtraction operations can be performed:

1. Modify each bigram constraint to exclude all events (h,w) that are part of an existing trigram constraint (call these "complemented bigrams").
2. Modify each unigram constraint to exclude all events (h,w) that are part of an existing bigram or trigram constraint (call these "complemented unigrams").

These changes are not merely notational — the resulting model differs from the original in significant ways. Neither are they applicable to ME models only. In fact, when applied to a conventional Backoff model, they yielded a modest reduction in perplexity. This is because at runtime, backoff conditions are better matched by the "complemented" events. Recently, [Kneser and Ney 95] used a similar observation to motivate their own modification to the backoff scheme, with similar results.

For the purpose of the ME model, though, the most important aspect of complemented N-grams is that their associated events do not overlap. Thus only one such constraint is active for any training datapoint (instead of up to three). This in turn results in faster convergence of the Generalized Iterative Scaling algorithm ([Rosenfeld 94b, p. 53]). For this reason we have chosen to use the complemented N-gram formulation in this work.

5.2 Triggers

5.2.1 Incorporating Triggers into ME

To formulate a (binary) trigger pair $A \rightarrow B$ as a constraint, define the constraint function $f_{A \rightarrow B}$ as:

$$f_{A \rightarrow B}(h, w) = \begin{cases} 1 & \text{if } A \in h, w = B \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

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

$$\sum_h \tilde{\mathrm{P}}(h) \sum_w P(w|h) f_{A \rightarrow B}(h, w) \ = \ \tilde{\mathrm{E}}\,[f_{A \rightarrow B}]. \tag{33}$$

5.2.2 Selecting Trigger Pairs

In section 2.5.2 , we discussed the use of mutual information as a measure of the utility of a trigger pair. Given the candidate trigger pair (BUENOS→AIRES), this proposed measure would be:

$$\begin{aligned} I(BUENOS_\circ{:}AIRES) \quad = \quad & P(BUENOS_\circ, AIRES) \log \frac{P(AIRES|BUENOS_\circ)}{P(AIRES)} \\ & + \ P(BUENOS_\circ, \overline{AIRES}) \log \frac{P(\overline{AIRES}|BUENOS_\circ)}{P(\overline{AIRES})} \\ & + \ P(\overline{BUENOS_\circ}, AIRES) \log \frac{P(AIRES|\overline{BUENOS_\circ})}{P(AIRES)} \\ & + \ P(\overline{BUENOS_\circ}, \overline{AIRES}) \log \frac{P(\overline{AIRES}|\overline{BUENOS_\circ})}{P(\overline{AIRES})} \end{aligned} \tag{34}$$

This measure is likely to result in a high utility score in this case. But is this trigger pair really that useful? Triggers are used in addition to N-grams. Therefore, trigger pairs are only useful to the extent that the information they provide supplements the information already provided by N-grams. In the example above, “AIRES” is almost always predicted by “BUENOS”, using a bigram constraint.

One possible fix is to modify the mutual information measure, so as to factor out triggering effects that fall within the range of the N-grams. Let $h = w_1^{i-1}$. Recall that

$$A_\circ \overset{\text{def}}{=} \{A \in w_1^{i-1}\}.$$

Then, in the context of trigram constraints, instead of using MI($A_\circ$:B) we can use MI($A_{\circ\text{-3g}} : B$), where:

$$A_{\circ\text{-3g}} \overset{\text{def}}{=} \{A \in w_1^{i-3}\}$$

We will designate this measure with MI-3g.

Using the WSJ occurrence file described in section 2.5.2, the 400 million possible (ordered) trigger pairs of the WSJ’s 20,000 word vocabulary were filtered. As a first step, only word pairs that co-occurred in at least 9 documents were maintained. This resulted in some 25 million (unordered) pairs. Next, MI($A_{\circ\text{-3g}} : B$) was computed for all these pairs. Only pairs that had at least 1 milibit (0.001 bit) of average mutual information were kept. This resulted in 1.4 million ordered trigger pairs, which were further sorted by MI-3g, separately for each B. A random sample is shown in table 7. A larger sample is provided in [Rosenfeld 94b, appendix C].

Browsing the complete list, several conclusions could be drawn:

HARVEST ⇐ CROP HARVEST CORN SOYBEAN SOYBEANS AGRICULTURE GRAIN DROUGHT GRAINS BUSHELS

HARVESTING ⇐ CROP HARVEST FORESTS FARMERS HARVESTING TIMBER TREES LOGGING ACRES FOREST

HASHEMI ⇐ IRAN IRANIAN TEHRAN IRAN’S IRANIANS LEBANON AYATOLLAH HOSTAGES KHOMEINI ISRAELI HOSTAGE SHIITE ISLAMIC IRAQ PERSIAN TERRORISM LEBANESE ARMS ISRAEL TERRORIST

HASTINGS ⇐ HASTINGS IMPEACHMENT ACQUITTED JUDGE TRIAL DISTRICT FLORIDA

HATE ⇐ HATE MY YOU HER MAN ME I LOVE

HAVANA ⇐ CUBAN CUBA CASTRO HAVANA FIDEL CASTRO’S CUBA’S CUBANS COMMUNIST MIAMI REVOLUTION

Table 7: The best triggers ”A” for some given words “B”, in descending order, as measured by $\mathrm{MI}(A_{o\text{-}3g} : B)$.

1. *Self-triggers*, namely words that trigger themselves ($A \rightarrow A$) are usually very good trigger pairs. In fact, in 68% of the cases, the best predictor for a word is the word itself. In 90% of the cases, the self-trigger is among the top 6 predictors.
2. Words based on the same stem are also good predictors.
3. In general, there is great similarity between same-stem words:
   - The strongest association is between nouns and their possessive, both for triggers (i.e. $B \Leftarrow$ . . . XYZ, . . . XYZ’S . . .) and for triggered words (i.e. the predictor sets of XYZ and XYZ’S are very similar).
   - Next is the association between nouns and their plurals.
   - Next is adjectivization (IRAN-IAN, ISRAEL-I).
4. Even when predictor sets are very similar, there is still a preference to self-triggers (i.e. ⟨XYZ⟩ predictor-set is biased towards ⟨XYZ⟩, ⟨XYZ⟩S predictor-set is biased towards ⟨XYZ⟩S, ⟨XYZ⟩’S predictor-set is biased towards ⟨XYZ⟩’S).
5. There is preference to more frequent words, as can be expected from the mutual information measure.

The MI-3g measure is still not optimal. Consider the sentence:

> “The district attorney’s office launched an investigation into loans made by several well connected banks.”

The MI-3g measure may suggest that (ATTORNEY→INVESTIGATION) is a good pair. And indeed, a model incorporating that pair may use “ATTORNEY” to trigger “INVESTIGATION” in the sentence above, raising its probability above the default value for the rest of the document. But when “INVESTIGATION” actually occurs, it is preceded by “LAUNCHED AN”, which allows the trigram component to predict it with a much higher probability. Raising the probability of “INVESTIGATION” incurs some cost, which is never justified in this example. This happens because MI-3g still measures “simple” mutual information, and not the *excess* mutual information beyond what is already supplied by the N-grams.

Similarly, trigger pairs affect each others’ usefulness. The utility of the trigger pair $A_1 \rightarrow B$ is diminished by the presence of the pair $A_2 \rightarrow B$, if the information they provide has some overlap. Also, the utility of a trigger pair depends on the way it will be used in the model. MI-3g fails to consider these factors as well.

For an optimal measure of the utility of a trigger pair, a procedure like the following could be used:

1. Train an ME model based on *N*-grams alone.

2. For every candidate trigger pair ($A \rightarrow B$), train a special instance of the base model that incorporates that pair (and that pair only).

3. Compute the excess information provided by each pair by comparing the entropy of predicting *B* with and without it.

4. For every *B*, choose the one trigger pair that maximizes the excess information.

5. Incorporate the new trigger pairs (one for each *B* in the vocabulary) into the base model, and repeat from step 2.

For a task as large as the WSJ (40 million words of training data, millions of constraints), this approach is clearly infeasible. But in much smaller tasks it could be employed (see for example [Ratnaparkhi and Roukos 94]).

5.2.3 A simple ME system

The difficulty in measuring the true utility of individual triggers means that, in general, one cannot directly compute how much information will be added to the system, and hence by how much entropy will be reduced. However, under special circumstances, this may still be possible. Consider the case where only unigram constraints are present, and only a single trigger is provided for each word in the vocabulary (one '*A*' for each '*B*'). Because there is no "crosstalk" between the *N*-gram constraints and the trigger constraints (nor among the trigger constraints themselves), it should be possible to calculate in advance the reduction in perplexity due to the introduction of the triggers.

To verify the theoretical arguments (as well as to test the code), the following experiment were conducted on the 38 million words of the WSJ corpus language training data (vocabulary=19,981, see appendix A). First, a ME model incorporating only the unigram constraints was created. Its training-set perplexity (PP) was 962 — exactly as calculated from simple Maximum Likelihood estimates. Next, for each word '*B*' in the vocabulary, the best predictor '*A*' (as measured by standard mutual information) was chosen. The 19,981 trigger pairs had a total mutual information of 0.37988 bits. Based on the argument above, the training-set perplexity of the model after incorporating these triggers should be:

$$962 \cdot 2^{-0.37988} \approx 739$$

The triggers were then added to the model, and the Generalized Iterative Scaling algorithm was run. It produced the following output:

iteration	training-PP	improvement
1	19981.0	
2	1919.6	90.4%
3	999.5	47.9%
4	821.5	17.8%
5	772.5	6.0%
6	755.0	2.3%
7	747.2	1.0%
8	743.1	0.5%
9	740.8	0.3%
10	739.4	0.2%

In complete agreement with the theoretical prediction.

5.3 A Model Combining N-grams and Triggers

As a first major test of the applicability of the ME approach, ME models were constructed which incorporated both *N*-gram and trigger constraints. One experiment was run with the best 3 triggers for each word (as judged by the MI-3g criterion), and another with the best 6 triggers per word.

In both *N*-gram and trigger constraints (as in all other constraints incorporated later), the desired value of each constraint (the right-hand side of equations 27, 29, 31 or 33) was replaced by its Good-Turing discounted value, since the latter is a better estimate of the true expectation of that constraint in new data[5].

A conventional backoff trigram model was used as a baseline. The Maximum Entropy models were also linearly interpolated with the conventional trigram, using a weight of 0.75 for the ME model and 0.25 for the trigram. 325,000 words of new data were used for testing[6]. Results are summarized in table 8.

vocabulary	top 20,000 words of WSJ corpus	
training set	5MW (WSJ)	
test set	325KW (WSJ)	
trigram perplexity (baseline)	173	173
ME experiment	top 3	top 6
ME constraints:		
unigrams	18400	18400
bigrams	240000	240000
trigrams	414000	414000
triggers	36000	65000
ME perplexity	134	130
perplexity reduction	23%	25%
$0.75\cdot$ME + $0.25\cdot$trigram perplexity	129	127
perplexity reduction	25%	27%

Table 8: Maximum Entropy models incorporating *N*-gram and trigger constraints.

Interpolation with the trigram model was done in order to test whether the ME model fully retained all the information provided by the *N*-grams, or whether part of it was somehow lost when trying to incorporate the trigger information. Since interpolation reduced perplexity by only 2%, we conclude that almost all the *N*-gram information was retained by the integrated ME model. This illustrates the ability of the ME framework to successfully accommodate multiple knowledge sources.

Similarly, there was little improvement in using 6 triggers per word vs. 3 triggers per word. This could be because little information was left after 3 triggers that could be exploited by trigger pairs. More likely it is a consequence of the suboptimal method we used for selecting triggers (see section 5.2.2). Many ’*A*’ triggers for the same word ‘*B* ’ are highly correlated, which means that much of the information they provide overlaps. Unfortunately, the MI-3g measure discussed in section 5.2.2 fails to account for this overlap.

The baseline trigram model used in this and all other experiments reported here was a “compact” backoff model: all trigrams occurring only once in the training set were ignored. This modification, which is the standard in the ARPA community, results in very slight degradation in perplexity (1% in this case), but realizes significant savings in memory requirements. All ME models described here also discarded this information.

[5] Note that this modification invalidates the equivalence with the Maximum Likelihood principle discussed in section 4.4. Furthermore, since the constraints no longer match the marginals of the training data, they are not guaranteed to be consistent, and hence a solution is not guaranteed to exist. Nevertheless, our intuition was that the large number of remaining degrees of freedom will practically guarantee a solution, and indeed this has always proven to be the case.

[6] We used a large test set to ensure the statistical significance of the results. At this size, perplexity of half the data set, randomly selected, is within $\sim$1% of the perplexity of the whole set.

5.4 Class Triggers

5.4.1 Motivation

In section 5.2.2 we mentioned that strong triggering relations exist among different inflections of the same stem, similar to the triggering relation a word has with itself. It is reasonable to hypothesize that the triggering relationship is really among the stems, not the inflections. This is further supported by our intuition (and observation) that triggers capture semantic correlations. One might assume, for example, that the stem "LOAN" triggers the stem "BANK". This relationship will hopefully capture, in a unified way, the affect that the occurrence of any of "LOAN", "LOANS", "LOAN'S", and "LOANED" might have on the probability of any of "BANK", "BANKS" and "BANKING" occurring next.

It should be noted that class triggers are not merely a notational shorthand. Even if one wrote down all possible combinations of word pairs from the above two lists, the result would not be the same as in using the single, class-based trigger. This is because, in a class trigger, the training data for all such word-pairs is clustered together. Which system is better is an empirical question. It depends on whether these words do indeed behave similarly with regard to long-distance prediction, which can only be decided by looking at the data.

5.4.2 ME Constraints for Class Trigger

Let $AA \overset{\text{def}}{=} \{A_1, A_2, \ldots A_n\}$ be some subset of the vocabulary, and let $BB \overset{\text{def}}{=} \{B_1, B_2, \ldots B_n\}$ be another subset. The ME constraint function for the class trigger ($AA \Rightarrow BB$) is:

$$f_{AA \rightarrow BB}(h, w) = \begin{cases} 1 & \text{if } (\exists A, A \in AA, A \in h) \wedge w \in BB \\ 0 & \text{otherwise} \end{cases} \tag{35}$$

Set $K_{AA \rightarrow BB}$ to $\tilde{\mathrm{E}}[f_{AA \rightarrow BB}]$, the empirical expectation of $f_{AA \rightarrow BB}$. Now impose on the desired probability estimate $P(h, w)$ the constraint:

$$\sum_h \tilde{\mathrm{P}}(h) \sum_w P(w|h) f_{AA \rightarrow BB}(h, w) \;=\; \tilde{\mathrm{E}}\,[f_{AA \rightarrow BB}] \tag{36}$$

5.4.3 Clustering Words for Class Triggers

Writing the ME constraints for class triggers is straightforward. The hard problem is finding useful classes. This is reminiscent of the case of class-based N-grams. Indeed, one could use any of the general methods discussed in section 2.3 : clustering by linguistic knowledge, clustering by domain knowledge, or data driven clustering.

To estimate the potential of class triggers, we chose to use the first of these methods. The choice was based on the strong conviction that some stem-based clustering is certainly "correct". This conviction was further supported by the observations made in section 5.2.2, after browsing the "best-predictors" list.

Using the 'morphe' program, developed at Carnegie Mellon[7], each word in the vocabulary was mapped to one or more stems. That mapping was then reversed to create word clusters. The ~20,000 words formed 13,171 clusters, 8,714 of which were singletons. Some words belonged to more than one cluster. A randomly selected sample is shown in table 9.

Next, two ME models were trained. The first included all "word self-triggers", one for each word in the vocabulary. The second included all "class self-triggers" ($f_{AA \rightarrow AA}$), one for each cluster AA. A threshold of 3 same-document occurrences was used for both types of triggers. Both models also included all the unigram constraints, with a threshold of 2 global occurrences. The use of only unigram constraints facilitated the quick estimation of the amount of information in the triggers, as was discussed in section 5.2.3. Both models were trained on the same 300,000 words of WSJ text. Results are summarized in table 10.

[7] We are grateful to David Evans and Steve Henderson for their generosity in providing us with this tool

[ACCRUAL]	:	ACCRUAL
[ACCRUE]	:	ACCRUE, ACCRUED, ACCRUING
[ACCUMULATE]	:	ACCUMULATE, ACCUMULATED, ACCUMULATING
[ACCUMULATION]	:	ACCUMULATION
[ACCURACY]	:	ACCURACY
[ACCURATE]	:	ACCURATE, ACCURATELY
[ACCURAY]	:	ACCURAY
[ACCUSATION]	:	ACCUSATION, ACCUSATIONS
[ACCUSE]	:	ACCUSE, ACCUSED, ACCUSES, ACCUSING
[ACCUSTOM]	:	ACCUSTOMED
[ACCUTANE]	:	ACCUTANE
[ACE]	:	ACE
[ACHIEVE]	:	ACHIEVE, ACHIEVED, ACHIEVES, ACHIEVING
[ACHIEVEMENT]	:	ACHIEVEMENT, ACHIEVEMENTS
[ACID]	:	ACID

Table 9: A randomly selected set of examples of stem-based clustering, using morphological analysis provided by the ’morphe’ program.

vocabulary	top 20,000 words of WSJ corpus	
training set	300KW (WSJ)	
test set	325KW (WSJ)	
unigram perplexity	903	
model	word self-triggers	class self-triggers
ME constraints:		
unigrams	9017	9017
word self-triggers	2658	—
class self-triggers	—	2409
training-set perplexity	745	740
test-set perplexity	888	870

Table 10: Word self-triggers vs. class self-triggers, in the presence of unigram constraints. Stem-based clustering does not help much.

Surprisingly, stem-based clustering resulted in only a 2% improvement in test-set perplexity in this context. One possible reason is the small amount of training data, which may not be sufficient to capture long-distance correlations among the less common members of the clusters. The experiment was therefore repeated, this time training on 5 million words. Results are summarized in table 11, and are even more disappointing. The class-based model is actually slightly worse than the word-based one (though the difference appears insignificant).

Why did stem-based clustering fail to improve perplexity? We did not find a satisfactory explanation. One possibility is as follows. Class triggers are allegedly superior to word triggers in that they also capture within-class, cross-word effects, such as the effect “ACCUSE” has on “ACCUSED”. But stem-based clusters often consist of one common word and several much less frequent variants. In these cases, all within-cluster cross-word effects include rare words, which means their impact is very small (recall that a trigger pair’s utility depends on the frequency of both its words).

vocabulary	top 20,000 words of WSJ corpus	
training set	5MW (WSJ)	
test set	325KW (WSJ)	
unigram perplexity	948	
model	word self-triggers	class self-triggers
ME constraints:		
unigrams	19490	19490
word self-triggers	10735	—
class self-triggers	—	12298
training-set perplexity	735	733
test-set perplexity	756	758

Table 11: Word self-triggers vs. class self-triggers, using more training data than in the previous experiment (table 10). Results are even more disappointing.

5.5 Long Distance N-grams

In section 2.4 We showed that there is quite a bit of information in bigrams of distance 2, 3 and 4. But in section 3.1, we reported that we were unable to benefit from this information using linear interpolation. With the Maximum Entropy approach, however, it might be possible to better integrate that knowledge.

5.5.1 Long Distance N-gram Constraints

Long distance N-gram constraints are incorporated into the ME formalism in much the same way as the conventional (distance 1) N-grams. For example, the constraint function for distance-j bigram $\{w_1, w_2\}$ is

$$f^{[j]}_{\{w_1,w_2\}}(h,w) = \begin{cases} 1 & \text{if } h = w_1^{i-1}, w_{i-j} = w_1 \text{ and } w = w_2 \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

and its associated constraint is

$$\sum_h \tilde{\mathrm{P}}(h) \sum_w P(w|h) f^{[j]}_{\{w_1,w_2\}}(h,w) \;=\; \tilde{\mathrm{E}}\,[f^{[j]}_{\{w_1,w_2\}}]. \tag{38}$$

where $\tilde{\mathrm{E}}\,[f^{[j]}_{\{w_1,w_2\}}]$ is the expectation of $f^{[j]}_{\{w_1,w_2\}}$ in the training data:

$$\tilde{\mathrm{E}}\,[f^{[j]}_{\{w_1,w_2\}}] \;\overset{\text{def}}{=}\; \frac{1}{N} \sum_{(h,w)\in \text{TRAINING}} f^{[j]}_{\{w_1,w_2\}}(h,w). \tag{39}$$

Similarly for the trigram constraints, and similarly for “complemented N-grams” (section 5.1.2).

5.6 Adding Distance-2 N-grams to the Model

The model described in section 5.3 was augmented to include distance-2 bigrams and trigrams. Three different systems were trained, on different amounts of training data: 1 million words, 5 million words, and 38 million words (the entire WSJ corpus). The systems and their performance are summarized in table 12. The trigram model used as baseline was described in section 5.3. Training time is reported in ’alpha-days’ which is the amount of computation done by a DEC/Alpha 3000/500 workstation in 24 hours.

The 38MW system was different than the others, in that it employed high thresholds (cutoffs) on the N-gram constraints: distance-1 bigrams and trigrams were included only if they occurred at least 9 times in

vocabulary	top 20,000 words of WSJ corpus		
test set	325KW		
training set	1MW	5MW	38MW*
trigram perplexity (baseline)	269	173	105
ME constraints:			
unigrams	13130	18421	19771
bigrams	65397	240256	327055
trigrams	79571	411646	427560
distance-2 bigrams	67186	280962	651418
distance-2 trigrams	65600	363095	794818
word triggers (max 3/word)	20209	35052	43066
training time (alpha-days)	< 1	12	~ 200
test-set perplexity	203	123	86
perplexity reduction	24%	29%	18%

Table 12: A Maximum Entropy model incorporating *N*-gram, distance-2 *N*-gram and trigger constraints. The 38MW system used far fewer parameters than the baseline, since it employed high *N*-gram thresholds to reduce training time.

the training data. Distance-2 bigrams and trigrams were included only if they occurred at least 5 times in the training data. This was done to reduce the computational load, which was quite severe for a system this size. The cutoffs used for the conventional *N*-grams were higher than those applied to the distance-2 *N*-grams because it was anticipated that the information lost from the former knowledge source will be re-introduced later, at least partially, by interpolation with the conventional trigram model. The actual values of the cutoffs were chosen so as to make it possible to finish the computation in 2-3 weeks.

As can be observed, the Maximum Entropy model is significantly better than the trigram 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 (2.26 million vs. 6.72 million).

To assess the relative contribution of the various information sources employed in the above experiments, Maximum Entropy models were constructed based on various subsets of these sources, using the 1MW system. Within each information source, the type and number of constraints are the same as in table 12. Results are summarized in table 13.

vocabulary	top 20,000 words of WSJ corpus	
training set	1MW	
test set	325KW	
	perplexity	%change
trigram (baseline)	269	—
ME models:		
dist.-1 *N*-grams + dist.-2 *N*-grams	249	-8%
dist.-1 *N*-grams + word triggers	208	-23%
dist.-1 *N*-grams + dist.-2 *N*-grams + word triggers	203	-24%

Table 13: Perplexity of Maximum Entropy models for various subsets of the information sources used in table 12. With 1MW of training data, information provided by distance-2 *N*-grams is largely overlapped by that provided by triggers.

The most notable result is that, in the 1MW system, distance-2 *N*-grams reduce perplexity by 8% by themselves, but only by 1–2% when added to the trigger constraints. Thus the information in distance-2 *N*-grams appears to be largely overlapped by that provided by the triggers. In contrast, distance-2 *N*-grams resulted in an additional 6% perplexity reduction in the 5MW system (see tables 8 and 12).

5.7 Maximum Entropy as a Knowledge Integrator

The experiments reported above clearly demonstrate our ability to significantly improve on the baseline trigram by integrating conventional *N*-grams, distance-2 *N*-grams and long distance triggers using a log-linear model and the Maximum Entropy principle. But how much of the reduction in perplexity is actually due to using the ME approach, as opposed to arising from the alternative knowledge sources themselves? How much improvement could have been achieved by integrating the same knowledge sources in a different, perhaps less computationally intensive way?

In section 3.1, we discussed two earlier attempts to do so. In the first, we used linear interpolation to combine the conventional *N*-gram with all long-distance *N*-grams up to distance 4. Each of the 4 *N*-gram component models was trained on the same data (1 million words), and the interpolation weights were optimized using heldout data. This resulted in a consistently trained model. And yet, perplexity was reduced by only 2% over the baseline, as compared to the 8% reduction in table 13.

In our second attempt ([Rosenfeld and Huang 92]), we combined evidence from multiple triggers using several variants of linear interpolation, then interpolated the result with a conventional backoff trigram. This resulted in some 10% reduction in perplexity, as compared to the respective 23% reduction using the ME framework. Admittedly, this last comparison is not as well controlled as the previous one, since the interactions among the various triggers were not consistently trained in the linear interpolation model (though the triggers themselves were). It is not clear how the triggers' interaction could have been modeled consistently without an exponential growth in the number of parameters. In any case, this only serves to highlight one of the biggest advantages of the ME method: that it facilitates the consistent and straightforward incorporation of diverse knowledge sources.

6 Adaptation in Language Modeling

6.1 Adaptation Vs. Long Distance Modeling

This work grew out of a desire to improve on the conventional trigram language model, by extracting information from the document's history. This approach is often termed "long-distance modeling". The *trigger pair* was chosen as the basic information bearing element for that purpose.

But triggers can be also viewed as vehicles of adaptation. As the topic of discourse becomes known, triggers capture and convey the semantic content of the document, and adjust the language model so that it better anticipates words that are more likely in that domain. Thus the models discussed so far can be considered adaptive as well.

This duality of long-distance modeling and adaptive modeling is quite strong. There is no clear distinction between the two. In one extreme, a trigger model based on the history of the current document can be viewed as a static (non-adaptive) probability function whose domain is the entire document history. In another extreme, a trigram model can be viewed as a bigram which is adapted at every step, based on the penultimate word of the history.

Fortunately, this type of distinction is not very important. More meaningful is classification based on the nature of the language source, and the relationship between the training and test data. In this section we propose such classification, and study the adaptive capabilities of Maximum Entropy and other modeling techniques.

6.2 Three Paradigms of Adaptation

The adaptation discussed 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 in the test data, 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 training data can be obtained. Thus a hybrid paradigm, *limited-data domain adaptation*, might be the most important one for real-world applications.

6.3 Within-Domain Adaptation

Maximum Entropy models are naturally suited for within-domain adaptation[8]. This is because constraints are typically derived from the training data. The ME model integrates the constraints, making the assumption that the same phenomena will hold in the test data as well.

But this last assumption is also a limitation. Of all the triggers selected by the mutual information measure, self-triggers were found to be particularly prevalent and strong (see section 5.2.2). This was true for very common, as well as moderately common words. It is reasonable to assume that it also holds for rare words. Unfortunately, Maximum Entropy triggers as described above can only capture self-correlations that are well represented in the training data. As long as the amount of training data is finite, self correlation among rare words is not likely to exceed the threshold. To capture these effects, the ME model was supplemented with a "rare words only" unigram cache, to be described in the next subsection.

Another source of adaptive information is self-correlations among word sequences. In principle, these can be captured by appropriate constraint functions, describing trigger relations among word sequences. But our implementation of triggers was limited to single word triggers. To capture these correlations, conditional bigram and trigram caches were added, to be described subsequently.

N-gram caches were first reported by [Kuhn 88] and [Kupiec 89]. [Kuhn and De Mori 90][Kuhn and De Mori 90b] employed a POS-based bigram cache to improve the performance of their static bigram. [Jelinek *et al.* 91] incorporated a trigram cache into a speech recognizer and reported reduced error rates.

6.3.1 Selective Unigram Cache

In a conventional 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.

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 a static model, and therefore requires a smaller modification to it.

Bayesian methods may be used to optimally combine the prior of a word with the new evidence provided by its occurrence. As a rough approximation, a selective unigram cache was implemented, 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. In the WSJ corpus, the optimal threshold was found to be in the range 10^{-3}–10^{-4}, with no significant differences within that range.

[8] Although they can be modified for cross-domain adaptation as well. See next subsection.

This scheme proved more useful for perplexity reduction than the conventional cache. This was especially true when the cache was combined with the ME model, since the latter captures well self correlations among more common words (see previous section).

6.3.2 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. However, Experimentation with such a trigram cache, constructed similarly to the conditional bigram cache, revealed 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.

6.3.3 Combining the Components

To maximize adaptive performance, the Maximum Entropy model was supplemented with the unigram and bigram caches described above. A conventional trigram (the one used as a baseline) was also added. This was especially important for the 38MW system, since it employed high cutoffs on *N*-gram constraints. These cutoffs effectively made the ME model "blind" to information from *N*-gram events that occurred eight or fewer times. The conventional trigram reintroduced some of that information.

The combined model was achieved by consulting an appropriate subset of the above four models. At any one time, the four component models 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.

The Maximum Entropy model was therefore started 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. Thus, in a formula:

$$\begin{aligned}
\lambda_{\mathrm{ME}} &= \min[0.3 + 0.4 \cdot [|h|/60]\ ,\ 0.7] \\
\lambda_{\mathrm{trigram}} &= \max[0.7 - 0.4 \cdot [|h|/60]\ ,\ 0.3] \\
\lambda_{\mathrm{unigram\_cache}} &= \min[0.001 \cdot |\mathrm{selective\_unigram\_cache}|\ ,\ 0.050] \\
\lambda_{\mathrm{bigram\_cache}} &= \begin{cases} 0.09 & \text{if last word in } h \text{ also occurred earlier in } h \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{40}$$

The threshold for words to enter the selective unigram cache was a static unigram probability of at least 0.001. The weights were always normalized to sum to 1.

While the general weighting scheme was chosen based on the considerations discussed above, the specific values of the weights were chosen by minimizing perplexity of unseen data.

6.3.4 Results and Analysis

Table 14 summarizes perplexity (PP) performance of various combinations of the trigram model, the Maximum Entropy model (ME), and the unigram and bigram caches, as follows:

vocabulary	top 20,000 words of WSJ corpus					
test set	325KW					
training set	1MW		5MW		38MW	
	PP	%change	PP	%change	PP	%change
trigram (baseline)	269	—	173	—	105	—
trigram + caches	193	-28%	133	-23%	88	-17%
Maximum Entropy (ME):	203	-24%	123	-29%	86	-18%
ME + trigram:	191	-29%	118	-32%	75	-28%
ME + trigram + caches:	163	-39%	108	-38%	71	-32%

Table 14: Best within domain adaptation perplexity (PP) results. Note that the adaptive model trained on 5 million words is almost as good as the baseline model trained on 38 million words.

trigram: This is the static perplexity, which serves as the baseline.

trigram + caches: These experiments represent the best adaptation achievable without the Maximum Entropy formalism (using a non-selective unigram cache results in a slightly higher perplexity). Note that improvement due to the caches is greater with less data. This 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 better, and therefore harder to improve on.

Maximum Entropy: These numbers are reproduced from table 12. The relative advantage of the "pure" Maximum Entropy model seems greater with more training data (except that the 38MW system is penalized by its high cutoffs). This is because ME uses constraint functions to capture correlations in the training data. The more data, the more N-gram and trigger correlations exist that are statistically reliable, and the more constraints are employed. This is also true with regard to the conventional N-grams in the baseline trigram model. The difference is thus in the number of distance-2 N-grams and trigger pairs.

ME + trigram: When the Maximum Entropy model is interpolated with the conventional trigram, the most significant perplexity reduction occurs in the 38MW system. This is because the 38MW ME model employed high N-gram cutoffs, and was thus "blind" to low count N-gram events. Interpolation with the conventional trigram reintroduced some of that information, although not in an optimal form (since linear interpolation is suboptimal) and not for the distance-2 N-grams.

ME + trigram + caches: These experiments represent the best adaptive scheme we achieved. As before, improvement due to the caches is smaller with more data. Compared with the trigram+caches experiment, the addition of the ME component improves perplexity by a relative 16% for the 1MW system, and by a relative 19% for the 5MW and 38MW systems.

To illustrate the success of our within-domain adaptation scheme, note that the best adaptive model trained on 1 million words is better than the baseline model trained on 5 million words, and the best adaptive model trained on 5 million words is almost as good as the baseline model trained on 38 million words. This is particularly noteworthy because the amount of training data available in various domains is often limited. In such cases, adaptation provides handy compensation.

6.4 Cross-Domain Adaptation

6.4.1 The Need for Cross-Domain Adaptation

Under the cross-domain adaptation paradigm, the training and test data are assumed to come from different sources. When this happens, the result is a significant degradation in language modeling quality. The further apart the two language sources are, the bigger the degradation. This effect can be quite strong even when the two sources are supposedly similar. Consider the example in table 15. Training data consists of articles from the Wall Street Journal (1987-1989). Test data is made of AP wire stories from the same period. The two sources can be considered very similar (especially relative to other sources such as technical literature, fine literature, broadcast etc.). And yet, perplexity of the AP data is *twice* that of WSJ data.

vocabulary	top 20,000 words of WSJ corpus	
training set	WSJ (38MW)	
test set	WSJ (325KW)	AP (420KW)
OOV rate	2.2%	3.9%
trigram hit rate	60%	50%
trigram perplexity	105	206

Table 15: Degradation in quality of language modeling when the test data is from a different domain than the training data. The trigram hit ratio is relative to a “compact” trigram .

A related phenomenon in cross-domain modeling is the increased rate of Out-Of-Vocabulary words. In the WSJ-AP example, cross-domain OOV rate is almost double the within-domain rate. Similarly, the rate of new bigrams and trigrams also increases (here reported by the complement measure, trigram hit rate, relative to a “compact” trigram, where training-set singletons were excluded).

Given these phenomena, it follows that the relative importance of caches is greater in cross-domain adaptation. This is because here one must rely less on correlations in the training-data, and more on correlations that are assumed to be universal (mostly self-correlations).

Table 16 shows the improvement achieved by the ME model and by the interpolated model under the cross-domain paradigm. As was predicted, the contribution of the ME component is slightly smaller than in the within-domain case, and the contribution of the caches is greater.

A note about triggers and adaptation: Triggers are generally more suitable for within-domain adaptation, because they rely on training-set correlations. But class triggers can still be used for cross domain adaptation. This is possible if correlations among classes is similar between the training and testing domains. If so, membership in the classes can be modified to better match the test domain. For example, (CEASEFIRE → SARAJEVO) may be a good trigger pair in 1995 data, whereas (CEASEFIRE → IRAQ) may be useful in 1991. Therefore, (CEASEFIRE → [embattled region]) can be adjusted appropriately and used for both. The same construct can be used for N-gram constraints ([Rudnicky 94]). Automatically defining useful concepts such as [embattled region] is, of course, a difficult and open problem.

6.5 Limited-Data Domain Adaptation

Under the limited-data domain adaptation paradigm, moderate amounts of training data are available from the test domain. Larger amounts of data may be available from other, “outside”, domains. This situation is

vocabulary	top 20,000 words of WSJ corpus
training data	38MW (WSJ)
test data	420KW (AP)
trigram (baseline)	
perplexity	206
Maximum Entropy	
perplexity	170
perplexity reduction	17%
ME + trigram + caches	
perplexity	130
perplexity reduction	37%

Table 16: Perplexity improvement of Maximum Entropy and interpolated adaptive models under the cross-domain adaptation paradigm. Compared to the within-domain adaptation experiment, the impact of the ME component is slightly smaller, while that of the caches is greater.

often encountered in real-world applications.

How best to integrate the more detailed knowledge from the outside domain with the less detailed knowledge in the test domain is still an open question. Some form of interpolation seems reasonable. Other ideas are also being pursued ([Rudnicky 94]. Here we would only like to establish a baseline for future work. In the following model, the only information to come from the outside domain (WSJ) is the list of triggers. This is the same list used in all the ME models reported above. All training, including training of these triggers, was done using 5 million words of AP wire data.

Table 17 shows the results. Compared with the within-domain case, the impact of the ME component is somewhat diminished, although it is still strong.

vocabulary	top 20,000 words of WSJ corpus
trigger derivation data	38MW (WSJ)
training data	5MW (AP)
test data	420KW (AP)
trigram (baseline)	
perplexity	170
Maximum Entropy	
perplexity	135
perplexity reduction	21%
ME + trigram + caches	
perplexity	114
perplexity reduction	33%

Table 17: Perplexity improvement of Maximum Entropy and interpolated adaptive models under the limited-data domain adaptation paradigm. Compared with the within-domain case, the impact of the ME component is somewhat diminished.

7 Adaptive Modeling and Speech Recognition Accuracy

Perhaps the most prominent use of language modeling is in automatic speech recognition. In this section, we report on the effect of our improved models on the performance of SPHINX-II, Carnegie Mellon's speech recognition system. A more detailed exposition, including a discussion of LM interface issues, can be found in [Rosenfeld 94b, chapter 7].

7.1 Within-Domain Adaptation

To evaluate recognition error rate reduction under the within-domain adaptation paradigm, we used the ARPA CSR (Continuous Speech Recognition) S1 evaluation set of November 1993 ([Kubala *et al.* 94], [Pallet *et al.* 94], [Hwang *et al.* 94]). It consisted of 424 utterances produced in the context of complete long documents by two male and two female speakers. The version of SPHINX-II ([Huang *et al.* 93]) used for this experiment had gender-dependent 10K senone acoustic models (see [Huang *et al.* 93c]). In addition to the ∼20,000 words in the standard WSJ lexicon, 178 out-of-vocabulary words and their correct phonetic transcriptions were added in order to create closed vocabulary conditions. The forward and backward passes of SPHINX II were first run to create word lattices, which were then used by three independent best-first passes. The first such pass used the 38MW static trigram language model, and served as the baseline. The other two passes used the interpolated adaptive language model, which was based on the same 38 million words of training data. The first of these two adaptive runs was for unsupervised word-by-word adaptation, in which the recognizer's output was used to update the language model. The other run used supervised adaptation, in which the recognizer's output was used for within-sentence adaptation, while the correct sentence transcription was used for across-sentence adaptation. Results are summarized in table 18.

language model	word error rate	% change
trigram (baseline)	19.9%	—
unsupervised adaptation	17.9%	-10%
supervised adaptation	17.1%	-14%

Table 18: Word error rate reduction of the adaptive language model over a conventional trigram model.

7.2 Cross-Domain Adaptation

To test error rate reduction under the cross-domain adaptation paradigm, we used the cross-domain system reported in section 6.4. 206 sentences, recorded by 3 male and 3 female speakers, were used as test data. Results are reported in table 19. As was expected from the perplexity experiments, relative improvement is smaller than that achieved under the within-domain adaptation paradigm.

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

Table 19: Word error rate reduction of the adaptive language model over a conventional trigram model, under the cross-domain adaptation paradigm.

For a more detailed discussion of recognition experiments, see [Rosenfeld 94b].

7.3 Perplexity and Recognition Error Rate

The ME-based adaptive language model that was trained on the full WSJ corpus (38 million words) reduced perplexity by 32% over the baseline trigram. The associated reduction in recognition word error rate was 14% under the most favorable circumstances. This does indeed conform to the empirically observed “square-root” law, which states that improvement in error rate is often approximately the square root of the improvement in perplexity ($\sqrt{0.68} = 0.82 \approx 0.86$). Still, why is the impact on error rate not any greater?

Perplexity does not take into account acoustic confusability, and does not pay special attention to outliers (tails of the distribution), where more recognition errors occur. But in addition to these deficiencies another factor is to blame. A language model affects recognition error rate through its *discriminative* power, namely its ability to assign higher scores to hypotheses that are more likely, and lower scores to those that are less likely. But perplexity is affected only by the scores assigned by the language model to *likely* hypotheses – those that are part of a test set, which typically consists of “true” sentences. Thus a language model that *overestimates* probabilities of unlikely hypotheses is not directly penalized by perplexity. The only penalty is indirect, since assigning high probabilities to some hypotheses means a commensurate reduction in the total probability assigned to all other hypotheses. If overestimation is confined to a small portion of the probability space, the effect on perplexity would be negligible. Yet such a model can give rise to significant recognition errors, because the high scores it assign to some unlikely hypotheses may cause the latter to be selected by the recognizer.

8 Acknowledgements

I am grateful to Peter Brown, Stephen Della Pietra, Vincent Della Pietra, Bob Mercer and Salim Roukos for introducing me to the Maximum Entropy principle and encouraging me to make use of their latest developments; to Raymond Lau and Salim Roukos for collaboration on the first implementation of the ME training procedure; to David Evans and Steve Henderson for providing me with the ’morphe’ program; to the speech group and others at Carnegie Mellon for varied logistical help, and to Raj Reddy and Xuedong Huang for much appreciated support and encouragement. Finally, I am grateful to three anonymous reviewers for very useful comments and suggestions.

This research was supported 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 author and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

A The ARPA WSJ Language Corpus

The first ARPA CSR Wall Street Journal corpus consists of articles published in the Wall Street Journal from December 1986 through November 1989. The original data was obtained, conditioned and processed for linguistic research by the Association for Computational Linguistics’ Data Collection Initiative (ACL/DCI). The corpus was chosen by the ARPA speech recognition community to be the basis for its CSR (Continuous Speech Recognition) common evaluation project. Subsequently, most of the data was further processed by Doug Paul at MIT’s Lincoln Labs [Paul and Baker 92], and conditioned for use in speech recognition. This included transforming many common text constructs to the way they are likely to be said when read aloud (e.g. “$123.45” might be transformed into “A hundred and twenty three dollars and forty five cents”), some quality filtering, preparation of various standard vocabularies, and much more. We refer to this data set as the “WSJ” corpus.

The version of this corpus used in the experiments described in this paper is the one where punctuation marks were assumed not to be verbalized, and were thus removed from the data. This was known as the “nvp” (non-verbalized-punctuation) condition. In this form, the WSJ corpus contained some 41.5 million words.

All our experiments (except where stated otherwise) used the ’20o’ vocabulary, which was derived as the most frequent 19,979 non-vp words in the data. It includes all words that occurred at least 60 times in that corpus (and 5 that occurred 59 times). All other words were mapped to a unique symbol, “<UNK>”, which was made part of the vocabulary, and had a frequency of about 2.2%. The pseudo word “</s>” was added to the vocabulary to designate end-of-sentence. The pseudo word “<s>” was used to designate beginning-of-sentence, but was not made part of the vocabulary. Following are the top and bottom of the vocabulary, in order of descending frequency, together with the words’ count in the corpus:

```
THE     2322098
</s>    1842029
OF      1096268
TO      1060667
A       962706
AND     870573
IN      801787
THAT    415956
FOR     408726
ONE     335366
IS      318271
SAID    301506
DOLLARS 271557
IT      256913
...
...
...
ARROW’S  60
ARDUOUS  60
APPETITES       60
ANNAPOLIS       60
ANGST    60
ANARCHY  60
AMASS    60
ALTERATIONS     60
AGGRAVATE       60
AGENDAS  60
ADAGE    60
ACQUAINTED      60
ACCREDITED      60
ACCELERATOR     60
ABUSERS  60
WRACKED  59
WOLTERS  59
WIMP     59
WESTINGHOUSE’S  59
WAIST    59
```

A fraction of the WSJ corpus (about 10%), in paragraph units, was set aside for acoustic training and for system development and evaluation. The rest of the data was designated for language model development by the ARPA sites. It consisted of some 38.5 million words.

From this set, we set aside about 0.5 million words for language model testing, taken from two separate time periods well within the global time period (July 1987 and January-February 1988). The remaining data are the 38 million words used in the large models. Smaller models were trained on appropriate subsets.

Our language training set had the following statistics:

- $\sim$ 87,000 article.

- ∼ 750,000 paragraphs.
- ∼ 1.8 million sentences (only 2 sentences/paragraph, on average).
- ∼ 38 million words (some 450 words/article, on average).

Most of the data were well-behaved, but there were some extremes:

- maximum number of paragraphs per article: 193.
- maximum number of sentences per paragraph: 51.
- maximum number of words per sentence: 257.
- maximum number of words per paragraph: 1483.
- maximum number of words per article: 6738.

Following are all the bigrams which occurred more than 65,535 times in the corpus:

```
318432 <UNK> </s>
669736 <UNK> <UNK>
83416 <UNK> A
192159 <UNK> AND
111521 <UNK> IN
174512 <UNK> OF
139056 <UNK> THE
119338 <UNK> TO
170200 <s> <UNK>
66212 <s> BUT
75614 <s> IN
281852 <s> THE
161514 A <UNK>
148801 AND <UNK>
76187 FOR THE
72880 IN <UNK>
173797 IN THE
110289 MILLION DOLLARS
144923 MR. <UNK>
83799 NINETEEN EIGHTY
153740 OF <UNK>
217427 OF THE
65565 ON THE
366931 THE <UNK>
127259 TO <UNK>
72312 TO THE
89184 U. S.
```

The most frequent trigram in the training data occurred 14,283 times. It was:

```
<s> IN THE
```

References

[Abramson 63] Norman Abramson. *Information Theory and Coding*, McGraw-Hill, New-York, 1963.

[Bahl *et al.* 83] Lalit Bahl, Fred Jelinek and Robert Mercer. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-5, number 2, pages 179–190, March 1983.

[Brown *et al.* 90b] Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai and Robert L. Mercer. Class-Based N-gram Models of Natural Language. In *Proceedings of the IBM Natural Language ITL*, March 1990. Paris, France.

[Brown *et al.*] Peter Brown, Stephen DellaPietra, Vincent DellaPietra, Robert Mercer, Arthur Nadas and Salim Roukos. A Maximum Penalized Entropy Construction of Conditional Log-Linear Language and Translation Models Using Learned Features and a Generalized Csiszar Algorithm. *Unpublished IBM research report*.

[Chase *et al.* 94] Lin Chase, Ron Rosenfeld, and Wayne Ward, “Error-Responsive Modifications to Speech Recognizers: Negative N-grams”, in *Proc. International Conference on Spoken Language Processing*, Yokohama, Japan, September 1994.

[Church and Hanks 90] Ken Church and Patrick Hanks. Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics*, Volume 16, number 1, pages 22–29, March 1990.

[Cover and King 78] Thomas M. Cover and Roger C. King. A Convergent Gambling Estimate of the Entropy of English. *IEEE Transactions on Information Theory*, Volume IT-24, number 4, pages 413–421, July 1978.

[Darroch and Ratcliff 72] J. N. Darroch and D. Ratcliff. Generalized Iterative Scaling for Log-Linear Models. *The Annals of Mathematical Statistics*, Volume 43, pages 1470–1480, 1972.

[DellaPietra *et al.* 92] Stephen Della Pietra, Vincent Della Pietra, Robert Mercer and Salim Roukos. Adaptive Language Modeling Using Minimum Discriminant Estimation. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages I-633–636, San Francisco, March 1992. Also published in *Proceedings of the DARPA Workshop on Speech and Natural Language*, Morgan Kaufmann, pages 103–106, February 1992.

[Dempster *et al.* 77] A. P. Dempster, N. M. Laird and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, volume 39, number 1, pages 1–38, 1977.

[Derouault and Merialdo 86] Anne-Marie Derouault and Bernard Merialdo. Natural Language Modeling for Phoneme-to-Text Transcription. *IEEE Transactions on Pattern Analysis and Machine Translation*, Volume PAMI-8, number 6, pages 742–749, November 1986.

[Good 53] I. J. Good. The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika*, Volume 40, parts 3,4, pages 237–264, 1953.

[Good 63] I. J. Good. Maximum Entropy for Hypothesis Formulation, Especially for Multidimensional Contingency Tables. *Annals of Mathematical Statistics*, Volume 34, pages 911–934, 1963.

[Huang *et al.* 93] Xuedong Huang, Fileno Alleva, Hsiao-wuen Hon, Mei-Yuh Hwang, Kai-Fu Lee and Ronald Rosenfeld. The SPHINX-II Speech Recognition System: An Overview. *Computer, Speech and Language*, volume 2, pages 137–148, 1993.

[Huang *et al.* 93c] Xuedong Huang, Fileno Alleva, Mei-Yuh Hwang and Ronald Rosenfeld. An Overview of the SPHINX-II Speech Recognition System. In *Proceedings of the ARPA Human Language Technology Workshop*, published as *Human Language Technology*, pages 81–86. Morgan Kaufmann, March 1993.

[Hwang *et al.* 94] Mei-Yuh Hwang, Ronald Rosenfeld, Eric Thayer, Ravi Mosur, Lin Chase, Robert Weide, Xuedong Huang and Fil Alleva. Improved Acoustic and Adaptive Language Models for Continuous Speech Recognition. In *Proceedings of the ARPA Spoken Language Technologies Workshop*, March 1994.

[Jaynes 57] E. T. Jaynes. Information Theory and Statistical Mechanics. *Physics Reviews* **106**, pages 620–630, 1957.

[Jelinek 89] Fred Jelinek. Self-Organized Language Modeling for Speech Recognition. in *Readings in Speech Recognition*, Alex Waibel and Kai-Fu Lee (Editors). Morgan Kaufmann, 1989.

[Jelinek 91] Fred Jelinek. Up From Trigrams! Eurospeech 1991.

[Jelinek *et al.* 77] Fred Jelinek, Robert L. Mercer, Lalit R. Bahl and James K. Baker. Perplexity — A Measure of Difficulty of Speech Recognition Tasks. *94th Meeting of the Acoustic Society of America*, Miami Beach, Florida, December 1977.

[Jelinek and Mercer 80] Fred Jelinek and Robert Mercer. Interpolated Estimation of Markov Source Parameters from Sparse Data. In *Pattern Recognition in Practice*, E. S. Gelsema and L. N. Kanal (editors), pages 381–402. North Holland, Amsterdam, 1980.

[Jelinek *et al.* 91] F. Jelinek, B. Merialdo, S. Roukos and M. Strauss. A Dynamic Language Model for Speech Recognition. In *Proceedings of the DARPA Workshop on Speech and Natural Language*, pages 293–295, February 1991.

[Katz 87] Slava M. Katz. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, volume ASSP-35, pages 400–401, March 1987.

[Kneser and Ney 91] Reinhard Kneser and Hermann Ney. Forming Word Classes by Statistical Clustering for Statistical Language Modeling. In *Proceedings of the 1st QUALICO Conference*, Trier, Germany, September 1991.

[Kneser and Ney 95] Reinhard Kneser and Hermann Ney. Improved Smoothing for M-gram Language Modeling. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Detroit, MI, May 1995.

[Kubala *et al.* 94] Francis Kubala and members of the CSR Corpus Coordinating Committee (CCCC). The Hub and Spoke Paradigm for CSR Evaluation. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 40–44, March 1994. Morgan Kaufmann.

[Kuhn 88] Roland Kuhn. 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.

[Kuhn and De Mori 90] Roland Kuhn and Renato De Mori. A Cache-Based Natural Language Model for Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-12, number 6, pages 570–583, June 1990.

[Kuhn and De Mori 90b] Roland Kuhn and Renato De Mori. *Correction to* A Cache-Based Natural Language Model for Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-14, number 6, pages 691–692, June 1992.

[Kullback 59] S. Kullback. *Information Theory in Statistics*. Wiley, New York, 1959.

[Kupiec 89] J. Kupiec. Probabilistic Models of Short and Long Distance Word Dependencies in Running Text. In *Proceedings of the DARPA Workshop on Speech and Natural Language*, pages 290–295, February 1989.

[Lau 93] Raymond Lau. Maximum Likelihood Maximum Entropy Trigger Language Model. *Bachelor's Thesis*, Massachusetts Institute of Technology, May 1993.

[Lau *et al*. 93a] Raymond Lau, Ronald Rosenfeld and Salim Roukos. Trigger-Based Language Models: a Maximum Entropy Approach. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages II 45–48, Minneapolis, MN, April 1993.

[Lau *et al*. 93b] Raymond Lau, Ronald Rosenfeld and Salim Roukos. Adaptive Language Modeling Using the Maximum Entropy Principle. In *Proceedings of the ARPA Human Language Technology Workshop*, published as *Human Language Technology*, pages 108–113. Morgan Kaufmann, March 1993.

[Mercer and Roukos 92] Robert L. Mercer and Salim Roukos. Personal communication. 1992.

[Pallet *et al*. 94] D. S. Pallett, J. G. Fiscus, W. M. Fisher, J. S. Garofolo, B. Lund and M. Pryzbocki. 1993 Benchmark Tests for the ARPA spoken Language Program. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 51–73, March 1994. Morgan Kaufmann.

[Paul and Baker 92] Doug B. Paul and Janet M. Baker. The Design for the Wall Street Journal-based CSR Corpus. In *Proceedings of the DARPA SLS Workshop*, February 1992.

[Price 90] Patti Price. Evaluation of Spoken Language Systems: the ATIS Domain. In *Proceedings of the Third DARPA Speech and Natural Language Workshop*, Richard Stern (editor), Morgan Kaufmann, June 1990.

[Ratnaparkhi and Roukos 94] A. Ratnaparkhi and S. Roukos. A Maximum Entropy Model for Prepositional Phrase Attachment. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 242–242e, March 1994. Morgan Kaufmann.

[Rosenfeld 92] Ronald Rosenfeld, Adaptive Statistical Language Modeling: a Maximum Entropy Approach. *Ph.D. Thesis Proposal, Carnegie Mellon University*, September 1992.

[Rosenfeld 94a] Ronald Rosenfeld, A Hybrid Approach to Adaptive Statistical Language Modeling. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 76–81, March 1994. Morgan Kaufmann.

[Rosenfeld 94b] Ronald Rosenfeld, Adaptive Statistical Language Modeling: A Maximum Entropy Approach. *Ph.D. thesis, Carnegie Mellon University*, April 1994. Also published as *Technical Report CMU-CS-94-138*, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, April 1994.

[Rosenfeld and Huang 92] Ronald Rosenfeld and Xuedong Huang. Improvements in Stochastic Language Modeling. In *Proceedings of the DARPA Workshop on Speech and Natural Language*, published by Morgan Kaufmann, pages 107–111, February 1992.

[Rudnicky 94] Alexander Rudnicky. Personal communication. 1994.

[Shannon 48] C. E. Shannon. A Mathematical Theory of Communication. *Bell Systems Technical Journal*, Volume 27, pages 379–423 (Part I), pages 623-656 (Part II), 1948.

[Shannon 51] C. E. Shannon. Prediction and Entropy of Printed English. *Bell Systems Technical Journal*, Volume 30, pages 50–64, 1951.

[Suhm and Waibel 94] Bernhard Suhm and Alex Waibel. Towards Better Language Models for Spontaneous Speech. *ICSLP 94*, Yokohama, Vol. 2, pp. 831-834.

[Ward 90] Wayne Ward. The CMU Air Travel Information Service: Understanding Spontaneous Speech. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages, 127–129, June 1990.

[Ward 91] Wayne Ward. Evaluation of the CMU ATIS System. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages, 101–105, February 1991.