Adaptive Statistical Language Modeling: A Maximum Entropy Approach

Ronald Rosenfeld

April 19, 1994
CMU-CS-94-138

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Thesis Committee:

Raj Reddy, co-chair
Xuedong Huang, co-chair, Carnegie Mellon and Microsoft Corporation
Jaime Carbonell
Alexander Rudnicky
Salim Roukos, IBM Corporation

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.

Keywords: language modeling, adaptive language modeling, statistical language modeling, maximum entropy, speech recognition.

Abstract

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. Language modeling is useful in automatic speech recognition, machine translation, and any other application that processes natural language with incomplete knowledge.

In this thesis, I view language as an information source which emits a stream of symbols from a finite alphabet (the vocabulary). The goal of language modeling is then to identify and exploit sources of information in the language stream, so as to minimize its perceived entropy.

Most existing statistical language models exploit the immediate past only. To extract information from further back in the document's history, I 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 many sources must be combined. Traditionally, linear interpolation and its variants have been used, but these are shown here to be seriously deficient. Instead, I 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%.

The significance of this thesis lies in improving language modeling, reducing speech recognition error rate, and in being the first large-scale test of the approach. It illustrates the feasibility of incorporating many diverse knowledge sources in a single, unified statistical framework.

To Lani, for making it all possible, enjoyable, and worthwhile,
and in loving memory of my mother, Ilana Kenner Rosenfeld.

Contents

List of Figures

Acknowledgements

My first debt and gratitude are to my parents. They raised me to love learning and to trust my ability. More recently, my infatuation with artificial intelligence is due in great part to Douglas Hofstadter's writings, and to an afternoon he generously spent with me ten years ago.

During my eight years at Carnegie Mellon I had the fortune of benefiting from several advisors. Dave Touretzky skillfully introduced me to the scientific world and instilled in me confidence in my research ability. Merrick Furst taught me the thrill of racing to crack the next big problem. Xuedong Huang introduced me to the fast-pace world of speech recognition research, and supported me every step of the way. His advice and friendship continue to guide me today. My deepest professional gratitude is to Raj Reddy. He has encouraged, supported and guided my scientific adventures with masterful strokes. I am continuously amazed at how much insight, perspective and vision I can absorb in a ten minute meeting with Raj.

Although never my official advisor, Geoff Hinton has been my mentor for many years. Geoff's scientific intuition is the continuous subject of my admiration. I am grateful to him for advice and friendship throughout the years, even when separated by many miles.

This thesis would literally not exist without the blessing of that powerhouse of statistical language modeling, the IBM speech and natural language group. I was first introduced to the Maximum Entropy principle by Peter Brown, Stephen Della Pietra, Vincent Della Pietra, Bob Mercer and Salim Roukos. I am grateful to Peter, Stephen, Vincent, Bob and Salim for sharing with me their latest developments and allowing me to make use of them in this thesis. I spent the summer of 1992 at the IBM Watson Research center, working with Salim Roukos and Raymond Lau on the first implementation of the Maximum Entropy training procedure. Their very significant contributions continued to affect my work well after I returned to Carnegie Mellon.

In addition to Raj, Xuedong and Salim, I am grateful to Jaime Carbonell and Alex Rudnicky for serving on my thesis committee. The last few weeks of marathon writing were quite grueling, and my committee's feedback and support were crucial for meeting the deadline.

I was first introduced to speech recognition by the excellent course created and taught by Alex Waibel and Kai-Fu Lee in the fall of 1990. Both Kai-Fu and Alex made themselves available later for answering questions, giving advice, and plain brainstorming. They continue to do so today. Two other lecturers in that course, Rich Stern and Wayne Ward, have been a source of much appreciated advice and feedback ever since.

Carnegie Mellon is a wonderful place. It is friendly and unassuming, yet challenging. My officemates over the years were always helpful, and fun to waste time with, perhaps too much so. I am especially indebted to Dan Julin. I shared two offices and seven years with Dan. His expertize and advice in so many computer areas allowed me to remain blissfully ignorant of them.

I spent the last year of my thesis tenure working remotely from Chicago. I am grateful to Roger Schank and his staff at the Institute for the Learning Sciences, Northwestern

University for generously allowing me to use their resources and making me feel at home during this period of exile.

Continuing to work in Carnegie Mellon's virtual environment while physically living in Chicago was a pioneering experiment. It was made possible, and successful, by Raj's support and vision. Throughout this period I relied heavily on help from many "proxies" in the speech group, especially Lin Chase, Ravi Mosur and Bob Weide. I am grateful to them all.

Running the experiments described herein required a prodigious amount of CPU. I thank the entire CMU speech group, as well as many other groups and individuals at CMU, for generously allowing me to monopolize their machines for weeks on end.

My biggest gratitude is undoubtedly to my wonderful wife, Lani. She has singlehandedly and patiently supported us throughout these long years. She has endured long stretches of spousal absence due to experiments, papers and deadlines. Most importantly, her love and unwavering support were my continuing sustenance. Now that I am finally graduating, I might as well turn over the diploma to her. She has earned it at least as much as I have.

Chapter 1

Introduction

1.1 Language Modeling: Motivation and Applications

Language modeling is the attempt to characterize, capture and exploit regularities in natural language.

Natural language is an immensely complicated phenomenon. It developed slowly and gradually over a long period, apparently to optimize human verbal communication. That optimization was carried out using organs and brain structures which developed much earlier, and for other purposes.

There is a great deal of variability and uncertainty in natural language. The most obvious source of variability is in the content of the intended message. But even for a given message, there is significant variability in the format chosen for conveying it. In addition, any medium for natural language is subject to noise, distortion and loss. The need to model language arises out of this uncertainty.

People use language models implicitly and subconsciously when processing natural language, because their knowledge is almost always partial. Similarly, every computer application that must process natural language with less than complete knowledge may benefit from language modeling.

The most prominent use of language modeling has been in *automatic speech recognition*, where a computer is used to transcribe spoken text into a written form. In the 1950's, speech recognition systems were built that could recognize vowels or digits, but they could not be successfully extended to handle more realistic language. This is because more knowledge, including linguistic knowledge, must be brought to bear on the recognition process ([Reddy 76, p. 503]). This new appreciation of the role of linguistic knowledge led to the development of sophisticated models, mostly statistical in nature (see next section).

A similar situation exists in the field of *machine translation*. Translating from one natural language to another involves a great deal of uncertainty. In addition to the sources listed above, variability also results from language-specific phenomena. These include multiple-sense words, idiomatic expressions, word order constraints, and others. Simple-minded machine translation efforts in the 50's proved incapable of handling real language. Tagging and parsing (which can be viewed as forms of language modeling) and related techniques had to be developed in order to handle the ambiguity, variability and idiosyncrasy of both the source and target languages. Recently, explicitly statistical models were introduced as well ([Brown+ 90]).

Two more applications that can benefit from language modeling are *optical character recognition* and *spelling correction*. In the first, the original text must be recovered from a potentially distorted image. Variability is particularly high if the text is hand-written. In the second, the "correct" or intended text is sought, and noise is introduced by human factors which are motoric (typos) or psycholinguistic (slips, misspellings) in nature. In both cases, exploiting linguistic knowledge will lead to improved performance.

1.2 Statistical Language Modeling

In statistical language modeling, large amounts of text are used to automatically determine the model's parameters, in a process known as *training*.

1.2.1 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. 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:

$$\begin{aligned} \arg\max_L \Pr(L|A) &= \arg\max_L \frac{\Pr(A|L)\cdot\Pr(L)}{\Pr(A)} \\ &= \arg\max_L \Pr(A|L)\cdot\Pr(L) \end{aligned} \tag{1.1}$$

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 $L = w_1^n \stackrel{\text{def}}{=} w_1, w_2, \ldots w_n$, 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 $h \stackrel{\text{def}}{=} w_1^{i-1}$ is called the *history*.

The event space (h, w) is very large, and no reasonable amount of data would be sufficient to span it. Some simplifying assumptions must be made. Typically, these come in the form of *clustering*: a partition of the event space h is defined, and histories that fall into the same equivalence class are assumed to have a similar effect on the probability distribution of the next word w. For example, in the trigram ([Bahl+ 83]) model, the partition is based on the last two words of the history, and the underlying assumption is:

$$\Pr(w_i|w_1^{i-1}) = \Pr(w_i|w_{i-2}, w_{i-1})$$

1.2.2 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. In this view, the trigram amounts to modeling the source as a second-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. Using an ideal model, which capitalizes on every conceivable correlation in the language, L would have a *perceived entropy* of H (see section 1.4.1 for exact quantification of this term). In practice, however, all models will fall far short of that goal. Worse, the quantity H is not directly measurable (though it can be bounded, see [Shannon 51, Jelinek 89, Cover+ 78]). On the other extreme, if the correlations among the w_i's were completely ignored, the perceived 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 perceived entropy down, as close as possible to its true value. This view of statistical language modeling is the dominant one in this thesis.

1.3 Statistical Models vs. Knowledge-Based Models

In an alternative type of language models, which I call "knowledge based", linguistic and domain knowledge are hand coded by experts. Often, these models provide only a "yes"/"no" answer regarding the grammaticality of candidate sentences. Other times, they may provide a ranking of candidates.

Statistical models enjoy the following advantages over knowledge-based ones:

- The probabilities produced by statistical models are more useful than the "yes"/"no" answers or even the rankings of the knowledge-based ones. Probabilities can be combined and otherwise manipulated. They convey a lot more information than a simple "accept"/"reject". Moreover, "yes"/"no" answers may actually prove harmful: actual use of natural language is often ungrammatical.

- The intuition of experts is often wrong. Overestimating our knowledge is a universal human trait.
- Once the statistical model has been developed and the training procedure implemented as a computer program, it can be run unsupervised on new data. Thus creating a model for a new domain can be done very fast.
- In practice, most knowledge-based models (e.g. parsers) are computationally intensive at runtime. Statistical models tend to run faster.

Statistical Models also have the following disadvantages:

- They do not capture the meaning of the text. As a result, nonsensical sentences may be deemed "reasonable" by these models (i.e. they may be assigned an unduly high probability).
- Statistical models require large amounts of training data, which are not always available. Porting the model to other languages or other domain is thus not always possible.
- Statistical models often do not make use of explicit linguistic and domain knowledge. Notwithstanding the comment above regarding overestimating experts' ability, *some* useful knowledge can and should be obtained from linguists or domain experts.

1.4 Measuring Model Quality

The ultimate measure of the quality of a language model is its impact on the performance of the application it was designed for. Thus, in speech recognition, we would evaluate a language model based on its effect on the recognition error rate. In practice, though, it is hard to always use this measure. Reliably measuring the error rate entails the processing of large amounts of data, which is very time consuming. More importantly, error rates are a result of complex and often non-linear interactions among many components. It is usually impossible to find analytical expression for the relationship between the error rate and the values of language model parameters. Consequently, automatic training that directly minimizes error rate is usually impossible.

In spite of the above, the "N-best" paradigm, which has been recently introduced to speech recognition ([Schwartz+ 90]), makes direct optimization of the error rate at least partially feasible. A short list of best-scoring hypotheses is produced by the recognizer, and a post-processor is used to rescore and re-rank them. Under these conditions, various parameters of a language model can be quickly tuned to optimize the re-ranking ([Huang+ 93b]).

1.4.1 Perplexity

A common alternative is to judge a statistical language model M by how well it predicts some hitherto unseen text T. This can be measured by the log-likelihood of M generating T, or, equivalently, by the *cross-entropy* of the distribution function $P_T(\mathbf{x})$ of the text, with regard to the probability function $P_M(\mathbf{x})$ of the model. Intuitively speaking, cross entropy is the entropy of T as "perceived" by the model M. Put another way, it is the amount of surprise in seeing T when using the model M to anticipate events. In an equation:

$$H(P_T; P_M) = -\sum_{\mathbf{x}} P_T(\mathbf{x}) \cdot \log P_M(\mathbf{x}) \tag{1.2}$$

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

$$\mathrm{PP}_M(T) = 2^{H(P_T; P_M)} \tag{1.3}$$

Perplexity can be roughly interpreted as the geometric mean of the branchout factor of the language: a language with perplexity X has roughly the same difficulty as another language in which every word can be followed by X different words with equal probabilities.

Perplexity is a function of both the model and the text. This fact must be borne in mind when comparing perplexity numbers for different texts and different models. A meaningful comparison can be made between perplexities of several models, all with respect to the same text and the same vocabulary. Comparing across texts or vocabularies is not well defined. Vocabularies must be the same, or else the smaller vocabulary will paradoxically bias the model towards lower perplexity (because it typically excludes rare words). Even if vocabularies are identical, different texts, with different out-of-vocabulary word rates, will render the comparison problematic at best.

1.4.2 Alternatives to Perplexity

Perplexity does not take into account acoustic confusability, and does not pay special attention to outliers (tail of the distribution), where most recognition errors occur. Lower perplexity does not always result in lower error rates, although it often does, especially when the reduction in perplexity is significant.

Several alternatives to perplexity have been suggested in the literature. Peter deSouza suggested *acoustic perplexity*, which takes into account acoustic confusability. However, a study showed that it is proportional to "regular" perplexity ([Jelinek 89]). [Ferretti+ 89] suggested the use of *Speech Decoder Entropy*, which measures the combined information provided by the acoustic and linguistic models together. This is typically less than the sum of their individual contributions, since some of the information overlaps. Using a sample text, it was found that the acoustic and linguistic sources are slightly more complementary than if they were completely orthogonal.

[Bahl+ 89] discusses the fact that recognition errors are strongly correlated with low-probability language model predictions ("surprises") . To capture this effect, it suggests measuring the fraction of surprises, namely the percentage of the text which was predicted with a probability below a certain threshold.

1.5 Smoothing

As was mentioned before, any reasonable amount of training data would be insufficient to adequately cover the event space (h, w). Even when clustering all events by $\{w_{i-2}, w_{i-1}, w_i\}$, as is done in the trigram model, coverage is still insufficient. For example, in a trigram model based on the Wall Street Journal corpus (WSJ, see appendix A), with 38 million words of training data, the rate of new trigrams in test data (taken from the same distribution as the training data) is 21% for a 5,000 vocabulary, and 32% for a 20,000 vocabulary.

Thus some method must be used to assign non-zero probabilities to events that have not been seen in the training data. This is known as *smoothing*. The simplest way to accomplish this is to assign each event a "floor" probability. This is equivalent to interpolating with the uniform probability model (see section 3.1).

Another method of smoothing was suggested by [Good 53]. He considers a sample from a large population, and uses the number of rare events to derive unbiased estimates of their probability. Based on this analysis, the best probability estimate for a trigram that occurred only once in a training set of size N is not $1/N$ but actually only a fraction of it. Thus the actual count of each rare event in the training set is "discounted" to arrive at an unbiased estimate, and the sum of the discounted mass is allocated to predicting unseen events (i.e. trigrams that did not occur at all in the training set). For other variants of smoothing, see [Nadas 84, Church+ 91, Ney+ 91, Placeway+ 93].

Chapter 2

Information Sources and Their Measures

2.1 Assessing the Potential of Information Sources

There are many potentially useful information sources in the history of a document. Assessing their potential before attempting to incorporate them into a model is important for several reasons. First, reasonable estimates can be used to rank several sources and thus help in deciding which ones to pursue first. Secondly, an upper bound on the utility of a source will help decide how much effort to allocate to its exploitation. Finally, once a source is being explored and incremental improvements are made to the model, an upper bound will help decide when to quit the process. Of course, the tighter the bound, the more informative and useful it will be.

2.1.1 Mutual Information

Mutual information (MI, [Abramson 63]) is a quantitative measure of the amount of information provided by one random variable (say X) on another (Y). Equivalently, mutual information is defined as the reduction in the entropy of Y when X is made known[1]:

$$\begin{aligned} I(X{:}Y) &\stackrel{\text{def}}{=} H(Y) - H(Y|X) \\ &= -\sum_y P(y)\log P(y) + \sum_x P(x)\sum_y P(y|x)\log P(y|x) \\ &= \sum_{x,y} P(x,y)\log\frac{P(x,y)}{P(x)P(y)} \end{aligned} \tag{2.1}$$

Several properties of mutual information are immediately apparent:

[1] Some authors refer to $I(X{:}Y)$ as the *average* mutual information.

- Mutual information is symmetric in its two variables: $I(X{:}Y) = I(Y{:}X)$, hence its name.
- If X, Y are independent (i.e. $P(x, y) = P(x)P(y)$), then $I(X{:}Y) = 0$.

Given a proposed information source, or *feature* of the history $f(h)$, we can estimate the amount of information it provides about the current word w by simply computing $I(f(h){:}w)$ over the training data. For example, to estimate the amount of information provided about the current word (w_i) by the last word of the history (w_{i-1}), we compute:

$$\begin{aligned} \tilde{\mathrm{I}}(w_{i-1}{:}w_i) &\stackrel{\text{def}}{=} \sum_{w_1}\sum_{w_2} \tilde{\mathrm{P}}(w_1, w_2) \log \frac{\tilde{\mathrm{P}}(w_1, w_2)}{\tilde{\mathrm{P}}(w_1)\tilde{\mathrm{P}}(w_1)} \\ &= \frac{1}{N}\sum_{w_1}\sum_{w_2} C(w_1, w_2) \log \frac{C(w_1, w_2)N}{C(w_1)C(w_1)} \end{aligned} \tag{2.2}$$

where $\tilde{\mathrm{I}}, \tilde{\mathrm{P}}$ denotes empirical estimates, measured over the training data, $C(w_1, w_2)$ is the bigram count, $C(w_1), C(w_2)$ are unigram counts, and N is the size of the training set.

The concept of mutual information can be extended in several ways. For example, it can be made conditional on the value of a third random variable ($I(X{:}Y \mid Z)$). It can also be used between *sets* of variables ($I(X, Z : Y, W)$). See [Abramson 63] for more details.

Mutual information measures how much direct information *exists* in feature $f(h)$ about the predicted word w_i. It does not say whether this information can be fully extracted, let alone how to extract it. It is therefore an *upper bound* estimate.

Mutual information has well defined properties, is easy to compute and analyze, and is intuitively appealing. However, it is not without its limitations:

- The amount of information that can actually be extracted from a variable depends on the framework and model being used. The model may be restricted to viewing the variable in a way that is "blind" to at least some of its information. For example, let X be an integer variable, and let

  $$Y = \begin{cases} 0 & \text{if } X \text{ is even} \\ 1 & \text{otherwise} \end{cases}$$

  be a binary variable. Then $I(X{:}Y) = H(Y)$, namely, X provides complete information about Y. But if the model under consideration is monotonic in X, it will not be able to capture this information.

- The upper bound provided by $I(X{:}Y)$ is often not tight. This is because MI does not take into account the information already provided about Y by other sources, say Z. If X and Z are mutually informative (i.e. $I(X{:}Z) > 0$), and if Z has already been incorporated into the model, then the most that X can add is the *excess information* it provides over Z, which is less than $I(X{:}Y)$. In the extreme case, if $X \equiv Z$, then $I(X{:}Y) = I(Z{:}Y) > 0$, yet X does not add any information to that already provided by Z.

- MI measures only *direct* information. It is possible that neither X nor Z provide any direct information about Y, yet their joint distribution does. For example, let X, Y, Z be binary variables. Let X, Y be independent, and let $Z = Y \oplus X$. Then $I(X{:}Y) = I(Z{:}Y) = 0$, yet $I(X, Z{:}Y) = H(Y)$ (i.e. X and Z together fully specify Y).

A possible solution to the last two problems is to directly measure $I(X, Z{:}Y)$ (the information provided about Y by the joint distribution of X and Z). But this must be extended to all candidate variables, resulting in an explosion of parameters, with a consequent drop in the reliability of the estimation. It also needs to be done separately for each candidate combination of variables.

2.1.2 Training Set Perplexity

An alternative method for assessing the potential of an information source is to measure its *training-set perplexity*. Namely, we train a simplified model, using the candidate information source the same way we intend to use it eventualy. We then measure the perplexity of the training data. For a large enough training set, the latter is usually a good indication of the amount of information conveyed by that source, under the current model: the lower the perplexity, the more information was conveyed. This is because the model captures as much as it can of that information, and whatever uncertainty remains shows up in the perplexity. it is important, though, that enough data be used relative to the number of parameters in the model, so that the model is not grossly over-fitted. Test set perplexity can be used as well, but interpretation may be more difficult. This is because complicating factors are involved: the difference between training and test data and the extent of smoothing.

Training set perplexity is not an accurate or easily interpretable measure. It cannot be meaningfully compared across different types of models, or even different data sets. Its main advantage is in that it uses the information source *in the manner in which it will eventually be used*, thus doing away with one of the problems with mutual information. This method is useful when comparing several similar features, all to be incorporated in the same manner. For example, in [Huang+ 93] we estimated the potential of w_{i-j} for various values of j by measuring the training set perplexity of *long-distance bigrams*, namely bigrams based on counts of the form $C(w_{i-j}, w_i)$ (see section 2.5).

2.1.3 Shannon-style Games

The potential of an information source can also be estimated by letting a human try to use it. A subject (either expert or layperson) is provided with the candidate information source, optionally other sources, and various tools for manipulating the data. She or he then attempt an appropriate task, such as guessing the identity of the next word. The game may be repeated with the exact same setup but without access to the candidate source. The difference in the success rate between the two trials indicates how much pertinent information exists in that source (of course different data, and/or different subjects, must be used the second time around). This is a generalization of the famous "Shannon game", proposed and used by C. E. Shannon to estimate the entropy of English[Shannon 51].

If the experiment is done correctly, the difference betwen the two trials can only be attributed to the additional source. On the other hand, it is possible that more information exists in the source that was not exploited by the subject(s). Therefore, strictly speaking, games of this type provide *lower* bounds on the information content of the source. However, current language modeling techniques usually fall far short of human capabilities. Human performance can seldom be reached. Therefore, improvement achieved by subjects in such games is often viewed as an *upper bound* on the practical potential of the source.

A "Shannon game" was implemented and tried at IBM ([Mercer+ 92]). Its results were used, among other things, to estimate the amount of information in the current sentence versus that in previous sentences, and also to justify research into triggers and other long distance sources. See section 2.6.

2.1.4 Summary

In this section, we discussed the importance of assessing the potential of candidate information sources, and several methods for doing so. We now turn to describing various such sources, and various indicators of their potential.

2.2 Context-Free Estimation (Unigram)

The most obvious information source for predicting the current word w_i is the prior distribution of words. This is the "information source" used by the unigram. Without this "source", entropy is $\log V$. When the priors are estimated from the training data, a Maximum Likelihood based unigram will have training-set entropy[2] of $H(\text{unigram}) = -\sum_{w \in V} P(w) \log P(w)$. Thus the information provided by the priors is

$$H(w_i) - H(w_i | \langle \text{PRIORS} \rangle) = \log V + \sum_{w \in V} P(w) \log P(w) \tag{2.3}$$

2.3 Short-Term History (Conventional N-gram)

An N-gram ([Bahl+ 83]) is a model that uses the last N-1 words of the history as its sole information source. The difference between the bigram, trigram, and other N-gram models is in the value of N.

Using an alternative view, that of equivalence classes, an N-gram model is one that partitions the data into equivalence classes, based on the last N-1 words of the history. Viewed this way, a bigram model induces a partition based on the last word of history. A

[2] A smoothed unigram will have a slightly higher entropy

trigram model further refines this partition by considering the next-to-last word. A 4-gram model further refines the trigram, and so on.

This hierarchy of refinements gives rise to the classic modeling tradeoff between detail and reliability. The bigram's equivalence classes are the largest, and hence the estimates they provide are the most reliable. The trigram's equivalence classes are smaller and more numerous. Many more of them contain only a few examples from the training data, and many more still are empty. On the other hand, the differentiating power of the trigram is greater, which, for a well-trained model, should result in lower perplexity. And similarly for higher-order *N*-grams.

Which *N*-gram model to use should be decided based on the amount of training data available, relative to the number of parameters. The latter is strongly affected by the vocabulary size. The nature of the data is also important. For example, the Wall Street Journal corpus, with 38 million words of training data and a 20,000 word vocabulary, is modeled much better by a trigram than by a bigram. In the ATIS task ([Price 90]), with $\sim$150,000 words of training and a 1400 word vocabulary, the benefit of a trigram model is less significant. With currently available amounts of training data, a 4-gram model does not seem to offer significant improvements over the trigram.

The tradeoff between the different *N*-gram models need not be decided on an all-or-none basis. Rather, it can be optimized separately for each context (see chapter 3).

Estimating mutual information is simple for the common events (i.e. common ngrams), but is unreliable for the uncommon ones, of which there are many. Training and test set perplexity measurement is more straightforward.

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.

[Mercer 92] tried, unsuccessfully, to create better predictors for an *N*-gram by optionally skipping some words in the history. The tree-based model described in [Bahl+ 89] also tried to solve these problems by allowing any question to be asked, in any order, about the last 20 words. This too had very limited success. Recently, [Isotani+ 94] took advantage of the clear partition of Japanese into function words and content words to create a *N*-gram model where predictors are determined by which of these two classes they belong to.

2.4 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. Let $g(w)$ denote the class that word w was assigned to. Then a word-based trigram:

$$P(w_i|h) = f(w_i|w_{i-1}, w_{i-2}) \tag{2.4}$$

may be turned into one of several different forms, including the following:

$$P(w_i|h) = f(w_i|w_{i-1}, g(w_{i-2})) \tag{2.5}$$

$$P(w_i|h) = f(w_i|g(w_{i-1}), g(w_{i-2})) \tag{2.6}$$

$$P(w_i|h) = f(g(w_i)|g(w_{i-1}), g(w_{i-2}))f(w_i|g(w_i)) \tag{2.7}$$

where $f()$ denotes an estimate based on relative frequency in the training data. See [Bahl+ 83] for more details.

The decision as to which components to cluster, as well as the nature and extent of the clustering, are another example of the detail-vs.-reliability tradeoff discussed in the previous section. Here too we must take into consideration the amount of training data, the number of parameters, and our beliefs about the way the language under consideration really behaves. For example, in the ATIS task, with ~150,000 words of training and a 1400 word vocabulary, clustering seems logical. Furthermore, the nature of the domain (airline travel reservation dialogs) is such that it can be reasonably assumed that, for example, all airline names behave similarly with regard to their right and left contexts. On the other hand, clustering in a large, varied, and data-rich domain like WSJ had limited usefulness so far.

In addition to deciding which components to cluster, we must decide on the clustering itself. There are three general methods for doing so:

2.4.1 Clustering by Linguistic Knowledge

The best known example of this method is clustering by part of speech (POS). POS clustering attempts to capture syntactic knowledge by throwing away the lexical identity of the words, and concentrating on the relationship between their syntactic roles. POS clustering was used in N-gram models by IBM ([Jelinek 89, Derouault+ 86]) and other sites, and was also incorporated into a cache by [Kuhn+ 90, Kuhn+ 90b].

There are several problems with this approach, though:

1. Some words can belong to more than one POS. Automatic tagging of these words is an open problem, with current systems achieving an error rate approaching 3% ([Jelinek 89, appendix C], [Derouault+ 86, Black+ 92],[Church 89]).

2. There are many different POS classifications used by linguists . Often one such system is neither a subset nor a superset of the others. This makes consistently tagged data even harder to obtain.

3. POS classification may make sense to the linguist, but is not necessarily optimal for language modeling ([Jelinek 89]). In section 2.4.3 we will discuss attempts to optimize the clustering based on the way the clusters will be used.

2.4.2 Clustering by Domain Knowledge

System designers almost always have some prior knowledge of the intended domain of their system. Unfortunately, a good part of assumed knowledge is often found to be wrong when checked against data. It is hard to tell when domain experts exceed their boundaries. When there is enough data to corroborate the purported knowledge, that prior knowledge is no longer necessary. But sometimes valid, useful knowledge does exist. One case in point is ATIS, the Airline Travel Information System ([Price 90]). Given the nature of the task, it seems reasonable to assume that airline names, airport names, city names etc. behave similarly with regard to their right and left contexts ([Ward 90, Ward 91]). This is not to say that they behave identically. They clearly don't. But given the limited amount of training data, there is more good than harm in clustering them together.

2.4.3 Data Driven Clustering

In data driven clustering, a large pool of data is used to automatically derive classes by statistical means. IBM has pioneered this approach, calling the resulted clusters NPOS (Nuclear Parts of Speech), and reporting on at least two such methods. In the first ([Jelinek 89, appendix C]), hidden Markov chains are used to automatically create word clusters that optimize a maximum likelihood criterion (HMMs were used similarly before by [Cave+ 80] to automatically cluster letters). In the second such method ([Jelinek 89, appendix D],[Brown+ 90b]), words are clustered using a greedy algorithm that tries to minimize the loss of mutual information incurred during a merge. This loss-of-MI criterion is then used to derive a bit-string representation of all the words in a vocabulary. This representation induces a tree structure, where the affinity between words is approximated by their relative positions in the tree.

Another related notion is that of a *synonym* ([Jelinek+ 90]). Predictions based on rare words are strengthened by consulting other words, dubbed *synonyms*, that are believed to behave somewhat similarly. The contribution of each synonym to the prediction depends on the extent of its similarity to the target word, which is judged by the similarity of the right and left contexts of their occurrences. Thus synonyms can be viewed as a "soft", "fuzzy", or weighted, form of clustering. Other attempts at data-driven clustering include [Kneser+ 91] and [Suhm+ 94].

Deriving the clustering automatically from the data overcomes the problem of over-reliance on intuition or suspect knowledge. It is also potentially superior in that the actual statistical method used for clustering can be tailored to the way clustering will be used. However, reliance on data instead of on external knowledge sources poses its own problems. For data-driven clustering to be useful, lots of data must be available. But there is a catch here. If there is enough data to statistically ascertain similarities between certain words, then there is probably enough data to model these words individually. This suggests that data-driven clustering has limited potential, and that some external knowledge (either linguistic or domain specific) is necessary to break that limit. Nevertheless, IBM reports better success with NPOS clustering than with POS clustering. [Kneser+ 91], with a similar method, reports mixed results.

2.5 Intermediate Distance

The sequential nature of the surface form of sentences does not reflect the deep structure of their meaning. In section 2.3 I mentioned that this weakens conventional N-gram models, because their predictors are defined by their ordinal place in the sentence, not by their linguistic role. Revisiting the example I used there, the histories "GOLD PRICES FELL" and "GOLD PRICES FELL YESTERDAY" are likely to have a somewhat similar effect on the probability that the next word is "TO". But this similarity is lost on an N-gram model. One might want to somehow make use of the predictive power of "PRICES FELL" in predicting "TO", even when these two phrases are separated by one or more words.

I have already mentioned several attempts to create better predictors than those based on ordinal word positions ([Mercer 92, Bahl+ 89, Isotani+ 94]). An alternative approach would be to use *long-distance N-grams*. These models 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.

[Huang+ 93] attempted to estimate the amount of information in long-distance bigrams. We constructed a long-distance backoff bigram for distance $d = 1, \ldots, 10, 1000$, using the 1 million word Brown Corpus as our training data. The distance-1000 case was used as a control, since at that distance we expected no significant information. For each such bigram, we computed *training-set perplexity*. As was discussed in section 2.1.2, 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[3]. See table 2.1. We concluded that significant information exists in the last 5 words of the history.

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.

[3] although below the perplexity of the $d = 1000$ case. See section 2.6.

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

Figure 2.1: *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.

2.6 Long Distance (Triggers)

2.6.1 Evidence for Long Distance Information

This thesis began as an attempt to capture some of the information present in the longer-distance history. I based my belief that there is a significant amount of information there on the following two experiments:

Long-Distance Bigrams. In section 2.5 I discussed the experiment on long-distance bigrams reported in [Huang+ 93]. As mentioned, we found training-set perplexity to be low for the conventional bigram ($d = 1$), 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. But interestingly, that level was slightly yet consistently below perplexity of the $d = 1000$ case (see table 2.1). 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+ 92]***.** A "Shannon game" program (see section 2.1.3) 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.

2.6.2 The Concept of a Trigger Pair

Based on the above evidence, I decided to use 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 $(A \to B)$ 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 we select trigger pairs 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, let's 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*. One usually measures the *log* of that quantity, which has units of bits, and is defined as:

$$\log \text{C.P.R.}(A_\circ, B) \stackrel{\text{def}}{=} \log \frac{C(A_\circ, B)C(\overline{A_\circ}, \overline{B})}{C(A_\circ, \overline{B})C(\overline{A_\circ}, B)} \tag{2.8}$$

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[4], 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:

$$\begin{aligned} I(A_\circ{:}B) = {} & P(A_\circ, B) \log \frac{P(B|A_\circ)}{P(B)} + P(A_\circ, \overline{B}) \log \frac{P(\overline{B}|A_\circ)}{P(\overline{B})} \\ & + P(\overline{A_\circ}, B) \log \frac{P(B|\overline{A_\circ})}{P(B)} + P(\overline{A_\circ}, \overline{B}) \log \frac{P(\overline{B}|\overline{A_\circ})}{P(\overline{B})} \end{aligned} \tag{2.9}$$

In a related work, [Church+ 90] uses a variant of the first term of equation 2.9 to automatically identify co-locational constraints.

2.6.3 Detailed Trigger Relations

In the trigger relations I considered so far, each trigger pair partitioned the history into two classes, based on whether the trigger occurred or did not occur in it. I call these triggers *binary*. One might wish to model long-distance relationships between word sequences in

[4] in the WSJ corpus, at least.

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 2.9 can then be modified to measure the amount of information conveyed on average by this many-way classification.

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 we gain significantly more by considering how recently A occurred, or how many times?

I have studied these issues using the Wall Street Journal corpus of 38 million words. First, I created an index file that contained, for every word, a record of all of its occurrences. Then, I created a program that, given a pair of words, computed their log cross product ratio, average mutual information, 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. 2.2 and 2.3. After using the program to manually browse through many

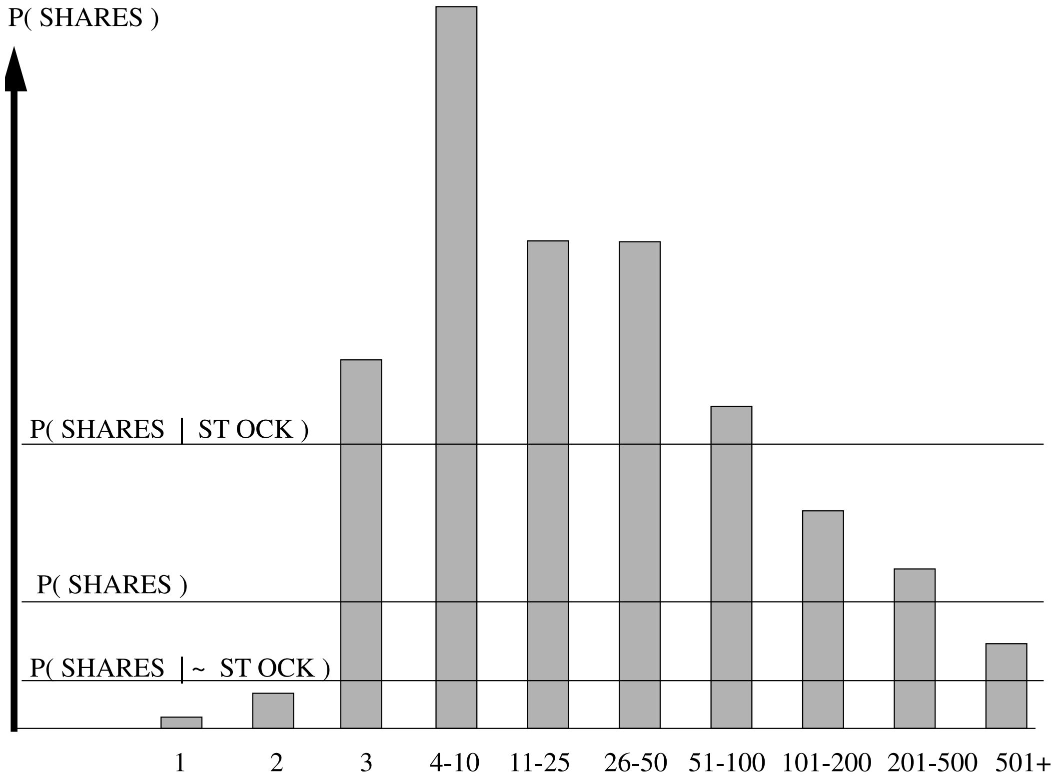

Figure 2.2: Probability of 'SHARES' as a function of the distance from the last occurrence of 'STOCK' in the same document. The middle horizontal line is the unconditional probability. The top (bottom) line is the probability of 'SHARES' given that 'STOCK' occurred (did not occur) before in the document.

hundreds of trigger pairs, I drew the following general conclusions:

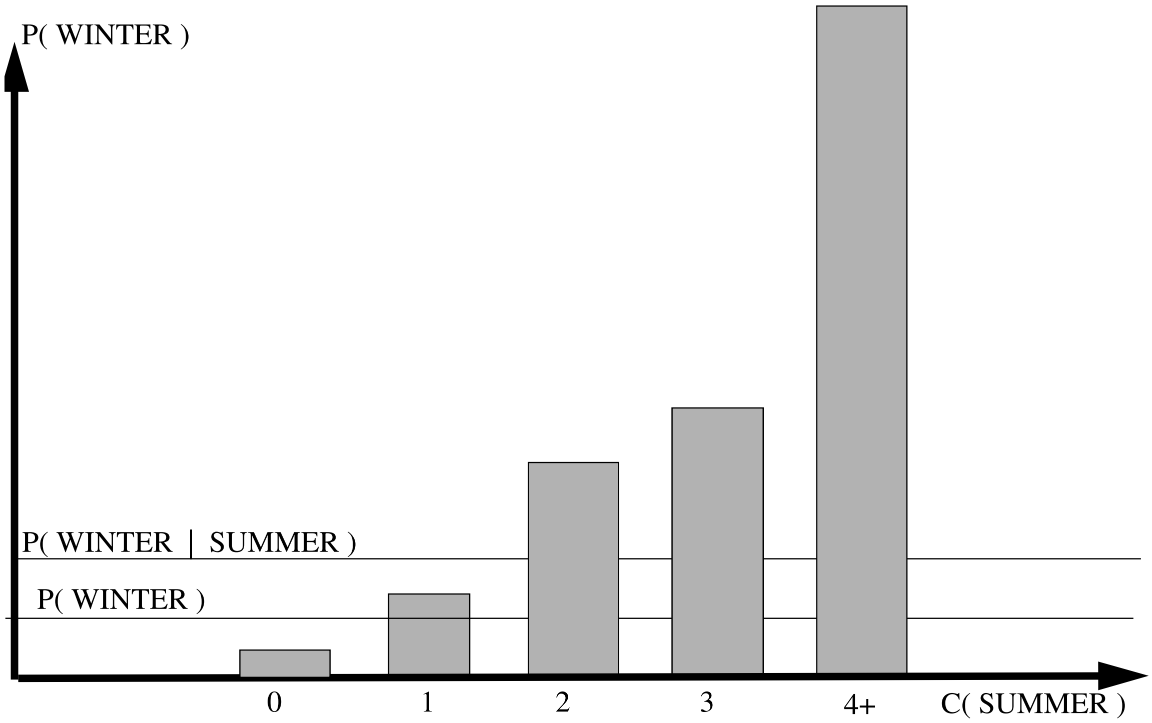

Figure 2.3: Probability of 'WINTER' as a function of the number of times 'SUMMER' occurred before it in the same document. Horizontal lines are as in fig. 2.2.

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$\rightarrow$ BOND) is indeed much more useful than (BREST$\rightarrow$LITOVSK).

5. When the trigger and triggered words are taken from different domains, 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.

2.7 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.

Chapter 3

Combining Information Sources

Once we identify the information sources we want to use and determine the phenomena to be modeled, one main issue still needs to be addressed. Given the part of the document processed so far (h), and a word w considered for the next position, there are many different estimates of $P(w|h)$. These estimates are derived from the different knowledge sources. How do we combine them all to form one optimal estimate? We discuss existing solutions in this chapter, and propose a new one in the next.

3.1 Linear Interpolation

3.1.1 Statement

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

$$P_{\text{COMBINED}}(w|h) \stackrel{\text{def}}{=} \sum_{i=1}^{k} \lambda_i P_i(w|h) \tag{3.1}$$

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).

3.1.2 The Weights

There are k-1 degrees of freedom in choosing k weights. To minimize perplexity of the combined model, an Estimation-Maximization (EM) type algorithm ([Dempster$^+$ 77]) is typically used to determine these weights (see [Jelinek 89] for details). The result is a set of weights that is provably optimal with regard to the data used for its optimization. If that data set is large enough and representative of the test data, the weights will be nearly optimal for the test data as well.

It should be noted that the magnitude of a weight does not always foretell the contribution of the associated knowledge source. The combined model is an *arithmetic* average of the component models. Perplexity, on the other hand, is a *geometric* average. It is thus possible that small linear contributions result in significant perplexity reduction. This would typically happen when these contributions are made to estimates that are otherwise very small.

3.1.3 Variations

As a generalization of linear interpolation, multiple sets of weights can be used. Which set to use can be determined on a case by case basis at run time. Typically, the data will be partitioned into "bins" or "buckets", based on the reliability of the estimates. Thus when interpolating a unigram, bigram and trigram, the bins could be determined by the count of the last two words. Each bin has a different set of weights, which are optimized based on held-out data belonging to that bin. When the count is large, the trigram will likely receive a large weight, and vice versa (see [Jelinek+ 80] for details).

Another variant of linear interpolation was used by [Bahl+ 89]. A classification tree was built based on the training data, and an estimate function was assigned to each node based on the part of the data rooted at that node. The deeper the node, the less data it contained, and hence the less reliable (though more pertinent) was its estimate. The final estimate for a given data point was a linear combination of the estimates along a path that started at the root and ended in the leaf containing that data point.

3.1.4 Pros and Cons

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.** I 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. An example output is given in appendix B. I have used the program to experiment with many different component models and bin-classification schemes. Some of my general conclusions are:

1. The exact value of the weights does not significantly affect perplexity. Weights need only be specified to within $\sim$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 2.5 I discussed [Huang+ 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 chapter 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+ 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 3.1. The 10% reduction in perplexity, however gratifying, is well below the true potential of the triggers, as will be demonstrated in the following chapters.

- **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.

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

Figure 3.1: Perplexity reduction by linearly interpolating the trigram with a trigger model. See [Rosenfeld+ 92] for details.

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,

$$\sum_{\substack{h \in \text{ TRAINING-SET} \\ h \text{ ends in "BANK"}}} P_{\text{BIGRAM}}(w|h) = C(\text{BANK}, w) \tag{3.2}$$

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 3.2 generally does not hold for the interpolated model.

3.2 Backoff

In the backoff method, 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 current context, it 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. Backoff can be seen as a special case of multi-bin linear interpolation: The bins are defined by which model is used to generate the answer. Within each bin, a weight of 1 is assigned to the active model, and 0 to all others. Viewed this way, it is clear that backing off is generally inferior to linear interpolation. Another problem with this method is that it exhibits a discontinuity around the point where the backoff decision is made. Nonetheless, backing off is simple, compact, and often almost as good as linear interpolation.

3.2.1 The Backoff N-gram Model

The best known example of a backoff scheme is the backoff N-gram ([Katz 87]. Let w_j^k stand for the sequence $(w_j, \ldots w_k)$. Then the backoff N-gram model is defined recursively as:

$$P_n(w_n|w_1^{n-1}) = \begin{cases} (1-d)C(w_1^n) \;/\; C(w_1^{n-1}) & \text{if } C(w_1^n) > 0 \\ \alpha(C(w_1^{n-1})) \cdot P_{n-1}(w_n|w_2^{n-1}) & \text{if } C(w_1^n) = 0 \end{cases} \tag{3.3}$$

where d, the discount ratio, is a function of $C(w_1^n)$, and the α's are the backoff weights, calculated to satisfy the sum-to-1 probability constraints.

3.2.2 Overestimation in the Backoff Scheme, and Its Correction

An important factor in the backoff N-gram model is its behavior on the backed-off cases, namely when a given n-gram w_1^n is found not to have occurred in the training data. In these cases, the model assumes that the probability is proportional to the estimate provided by the n-1–gram, $P_{n-1}(w_n|w_2^{n-1})$.

This last assumption is reasonable most of the time, since no other sources of information are available. But for frequent n-1–grams, there may exist sufficient statistical evidence to suggest that the backed-off probabilities should in fact be much lower. This phenomenon occurs at any value of n, but is easiest to demonstrate for the simple case of $n = 2$, i.e. a bigram. Consider the following fictitious but typical example:

N = 1,000,000
C("ON") = 10,000
C("AT") = 10,000
C("CALL") = 100
C("ON","AT") = 0
C("ON","CALL") = 0

N is the total number of words in the training set, and $C(w_i, w_j)$ is the number of (w_i, w_j) bigrams occurring in that set. The backoff model computes:

P("AT") = $\frac{1}{100}$
P("CALL") = $\frac{1}{10,000}$
P("AT"|"ON") = α("ON") $\cdot$ P("AT") = α("ON")$\cdot\frac{1}{100}$
P("CALL"|"ON") = α("ON") P("CALL") = α("ON") $\frac{1}{10000}$

Thus, according to this model, P("AT"|"ON") $\gg$ P("CALL"|"ON"). But this is clearly incorrect. In the case of "CALL", the expected number of ("ON","CALL") bigrams, assuming independence between "ON" and "CALL", is 1, so an actual count of 0 does not give much information, and may be ignored. However, in the case of "AT", the expected chance count of ("ON","AT") is 100, so an actual count of 0 means that the real probability of P("AT"|"ON") is in fact much lower than chance. The backoff model does not capture this information, and thus grossly overestimates P("AT"|"ON").

To solve this problem, I introduced a modification I dubbed *Confidence Interval Capping* ([Rosenfeld+ 92]). Let $C(w_1^n) = 0$. Given a global confidence level Q, to be determined empirically, we calculate a confidence interval in which the true value of $P(w_n|w_1^{n-1})$ should lie, using the constraint:

$$[1 - P(w_n|w_1^{n-1})]^{C(w_1^{n-1})} \geq Q \tag{3.4}$$

The confidence interval is therefore $[0 \ldots (1 - Q^{1/C(w_1^{n-1})})]$. We then provide another parameter, P $(0 < P \leq 1)$, and establish a ceiling, or a *cap*, at a point P within the confidence interval:

$$\mathrm{CAP}_{Q,P}(C(w_1^{n-1})) = P \cdot (1 - Q^{1/C(w_1^{n-1})}) \tag{3.5}$$

We now require that the estimated $P(w_n|w_1^{n-1})$ satisfy:

$$P(w_n|w_1^{n-1}) \leq \mathrm{CAP}_{Q,P}(C(w_1^{n-1})) \tag{3.6}$$

The backoff case of the standard model is therefore modified to:

$$P(w_n|w_1^{n-1}) \;=\; \min\,[\ \alpha(w_1^{n-1}) \cdot P_{n-1}(w_n|w_2^{n-1})\,,\ \mathrm{CAP}_{Q,P}(C(w_1^{n-1}))\ ] \tag{3.7}$$

This *capping off* of the estimates requires renormalization. But renormalization would increase the α's, which would in turn cause some backed-off probabilities to exceed the cap. An iterative reestimation of the α's is therefore required. The process was found to converge in 2-3 iterations.

Note that, although some computation is required to determine the new weights, once the model has been computed, it is no more complicated neither significantly more time consuming than the original one.

The test-set bigram perplexity reduction for various tasks is shown in table 3.2. Although the reduction is modest, as expected, it should be remembered that it is achieved with hardly any increase in the complexity of the model. As can be predicted from the statistical analysis, when the vocabulary is larger, the backoff rate is greater, and the improvement in perplexity can be expected to be greater too. This correction is thus more suitable for cases where training data is relatively sparse.

Corpus	training data	vocabulary	backoff rate	perplexity reduction
BC-48K	900,000	48,455	30%	6.3%
BC-5K	900,000	5,000	15%	2.5%
ATIS	100,000	914	5%	1.7%
WSJ-5K	42,000,000	5,000	2%	0.8%

Figure 3.2: Correcting Over-estimation in the Backoff *N*-gram model: Bigram perplexity reduction by Confidence Interval Capping. BC-48K is the brown corpus with the unabridged vocabulary of 48,455 words. BC-5K is the same corpus, restricted to the most frequent 5,000 words. ATIS is the class-based bigram developed at Carnegie Mellon [Ward 90] for the ATIS task [Price 90]. WSJ-5K is the WSJ corpus (see appendix A), in verbalized-punctuation mode, using the official “5c.vp” vocabulary.

Chapter 4

The Maximum Entropy Principle

In this chapter I discuss an alternative method of combining knowledge sources, which is based on an approach first proposed by E. T. Jaines in the 1950's ([Jaines 57]).

In the methods described in the previous chapter, each knowledge source was used separately to construct a model, and the models were then combined. Under the Maximum Entropy approach, we do not construct separate models. Instead, we build 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 in section 3.1.4. 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 knowledge sources have been incorporated, nothing else is assumed about the data.

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 4.1. Each column is an equivalence class in this partition.

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:

$$P_{\text{BIGRAM}}(\text{BANK}|\text{THE}) = K_{\{\text{THE,BANK}\}} \tag{4.1}$$

h ends in "THE"	h ends in "OF"	...	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Figure 4.1: 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.

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

$$K_{\{\text{THE},\text{BANK}\}} \stackrel{\text{def}}{=} \frac{C(\text{THE, BANK})}{C(\text{THE})} \tag{4.2}$$

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. We will thus add a different partition of the event space, as in figure 4.2. Each of the two rows is an equivalence class in this partition[1].

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

Figure 4.2: 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:

$$P_{\text{LOAN}\rightarrow\text{BANK}}(\text{BANK}|\text{LOAN}\in h) \ = \ K_{\{\text{BANK},\text{LOAN}\in h\}} \tag{4.3}$$

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

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

$$K_{\{\text{BANK},\text{LOAN}\in h\}} \stackrel{\text{def}}{=} \frac{C(\text{BANK}, \text{LOAN} \in h)}{C(\text{LOAN} \in h)} \tag{4.4}$$

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 4.1 is replaced by

$$\mathop{\mathbf{E}}_{h \text{ ends in "THE"}} [\ P_{\text{COMBINED}}(\text{BANK}|h)\ ] \ = \ K_{\{\text{THE},\text{BANK}\}} \tag{4.5}$$

where **E** stands for an expectation, or average. Note that the constraint expressed by equation 4.5 is much weaker than that expressed by equation 4.1. 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.

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":

$$\mathop{\mathbf{E}}_{\text{"LOAN"}\in h} [\ P_{\text{COMBINED}}(\text{BANK}|h)\ ] \ = \ K_{\{\text{BANK},\text{LOAN}\in h\}} \tag{4.6}$$

As in the bigram case, this constraint is much weaker than that imposed by equation 4.3.

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 4.4.

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:

$$\mathop{\mathbf{E}}_{(h,w)\in S} [\ P(h,w)\ ] \ = \ K \tag{4.7}$$

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 4.7 becomes:

$$\sum_{(h,w)} [\ P(h,w) f_s(h,w)\ ] \ = \ K \tag{4.8}$$

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 4.8 now becomes:

$$\langle f, P\rangle \ = \ K \tag{4.9}$$

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 chapter fall into this category, as do all other information sources that can be described by an algorithm.

In the following sections I present a general description of the Maximum Entropy model and its solution.

4.3 Maximum Entropy Formalism

The Maximum Entropy (ME) Principle ([Jaines 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:

$$E_P f_i \stackrel{\text{def}}{=} \sum_{\mathbf{x}} P(\mathbf{x}) f_i(\mathbf{x}) = K_i \ . \tag{4.10}$$

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

$$P(\mathbf{x}) = \prod_i \mu_i{}^{f_i(\mathbf{x})} \ , \tag{4.11}$$

where the μ_i's are some unknown constants, to be found([Jaines 57]). Probability functions of the form (4.11) are called *log-linear*, and the family of functions defined by holding the f_i's fixed and varying the μ_i's is called *an exponential family*.

To search the exponential family defined by (4.11) for the μ_i's that will make $P(\mathbf{x})$ satisfy all the constraints, an iterative algorithm, "Generalized Iterative Scaling" (GIS), exists, which is guaranteed to converge to the solution ([Darroch+ 72]). In the next section, I will briefly describe the workings of GIS.

4.4 The Generalized Iterative Scaling Algorithm

Generalized Iterative Scaling is an iterative algorithm. We start with some arbitrary μ_{i_1} values, which define the initial probability estimate:

$$P_0(\mathbf{x}) \stackrel{\text{def}}{=} \prod_i \mu_i^{(0)\, f_i(\mathbf{x})}$$

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

1. Compute the expectations of all the f_i's under the current estimate function. Namely, compute $E_{P_k} f_i \stackrel{\text{def}}{=} \sum_{\mathbf{x}} P_k(\mathbf{x}) f_i(\mathbf{x})$.

2. Compare the *actual* values ($E_{P^{(k)}} f_i$'s) to the *desired* values (K_i's), and update the μ_i's according to the following formula:

$$\mu_i^{(k+1)} = \mu_i^{(k)} \cdot \frac{K_i}{E_{P^{(k)}} f_i} \tag{4.12}$$

3. Define the next estimate function based on the new μ_i's:

$$P^{(k+1)}(\mathbf{x}) \stackrel{\text{def}}{=} \prod_i \mu_i^{(k+1)\, f_i(\mathbf{x})} \tag{4.13}$$

Iterating is continued until convergence or near-convergence.

4.5 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+ 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+]. Let $P(h,w)$ be the desired probability estimate, and let $\tilde{P}(h,w)$ be the empirical distribution of the training data. Let $f_i(h,w)$ be any constraint function, and let K_i be its desired expectation. Equation 4.10 can be rewritten as:

$$\sum_h P(h) \cdot \sum_w P(w|h) \cdot f_i(h,w) \;=\; K_i \tag{4.14}$$

We now modify the constraint to be:

$$\sum_h \tilde{P}(h) \cdot \sum_w P(w|h) \cdot f_i(h,w) \;=\; K_i \tag{4.15}$$

One possible interpretation of this modification is as follows. Instead of constraining the expectation of $f_i(h,w)$ with regard to $P(h,w)$, we constrain its expectation with regard to a different probability distribution, say $Q(h,w)$, whose conditional $Q(w|h)$ is the same as that of P, but whose marginal $Q(h)$ is the same as that of $\tilde{P}$. To better understand the effect of this change, define H as the set of all possible histories h, and define H_{f_i} as the partition of H induced by f_i. Then the modification is equivalent to assuming that, for every constraint f_i, $P(H_{f_i}) = \tilde{P}(H_{f_i})$. Since typically H_{f_i} is a very small set, the assumption is reasonable. 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 (4.15) 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.

4.6 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 $P_0(\mathbf{x})$:

$$Q_0(\mathbf{x}) \stackrel{\text{def}}{=} \arg\min_{\alpha} D(Q_\alpha, P_0) \tag{4.16}$$

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

$$D(Q(\mathbf{x}), P(\mathbf{x})) \stackrel{\text{def}}{=} \sum_{\mathbf{x}} Q(\mathbf{x}) \log \frac{Q(\mathbf{x})}{P(\mathbf{x})} \tag{4.17}$$

In the special case when $P_0(\mathbf{x})$ is the uniform distribution, $Q_0(\mathbf{x})$ as defined by equation 4.16 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+ 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.

4.7 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 I will show how this is done for the conventional N-gram model, and for the cache model of [Jelinek+ 91].

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 (more on this in 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 I 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], or else the constraints may be derived from other data, or be externally imposed. Under these circumstances, the constraints may no longer be consistent, and the theoretical results guaranteeing existence, uniqueness and convergence may not hold.

Chapter 5

Using Maximum Entropy in Language Modeling

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

5.1 Conventional N-grams

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

$$f_{w_1}(h, w) = \begin{cases} 1 & \text{if } w = w_1 \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

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:

$$\tilde{\mathrm{E}}[f_{w_1}] \stackrel{\text{def}}{=} \frac{1}{N} \sum_{(h,w)\in\text{TRAINING}} f_{w_1}(h, w), \tag{5.2}$$

and the associated constraint is:

$$\sum_{h} \tilde{\mathrm{P}}(h) \sum_{w} P(w|h) f_{w_1}(h, w) = \tilde{\mathrm{E}}[f_{w_1}]. \tag{5.3}$$

Similarly, the constraint function for the bigram $\{w_1, w_2\}$ is

$$f_{\{w_1,w_2\}}(h, w) = \begin{cases} 1 & \text{if } h \text{ ends in } w_1 \text{ and } w = w_2 \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

and its associated constraint is

$$\sum_h \tilde{\mathrm{P}}(h) \sum_w P(w|h) f_{\{w_1,w_2\}}(h,w) = \tilde{\mathrm{E}}\,[f_{\{w_1,w_2\}}]. \tag{5.5}$$

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

$$f_{\{w_1,w_2,w_3\}}(h,w) = \begin{cases} 1 & \text{if } h \text{ ends in } \{w_1, w_2\} \text{ and } w = w_3 \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

and its associated constraint is

$$\sum_h \tilde{\mathrm{P}}(h) \sum_w P(w|h) f_{\{w_1,w_2,w_3\}}(h,w) = \tilde{\mathrm{E}}\,[f_{\{w_1,w_2,w_3\}}]. \tag{5.7}$$

5.2 Triggers

5.2.1 Incorporating Triggers into ME

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

$$f_{A\to B}(h,w) = \begin{cases} 1 & \text{if } A \in h, w = B \\ 0 & \text{otherwise} \end{cases} \tag{5.8}$$

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

$$\sum_h \tilde{\mathrm{P}}(h) \sum_w P(w|h) f_{A\to B}(h,w) \;=\; \tilde{\mathrm{E}}\,[f_{A\to B}]. \tag{5.9}$$

5.2.2 Selecting Trigger Pairs

In section 2.6.2 , I 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:

$$\begin{aligned} I(BUENOS_\circ{:}AIRES) \;=\; & P(BUENOS_\circ, AIRES) \log \frac{P(AIRES|BUENOS_\circ)}{P(AIRES)} \\ & + P(BUENOS_\circ, \overline{AIRES}) \log \frac{P(\overline{AIRES}|BUENOS_\circ)}{P(\overline{AIRES})} \\ & + P(\overline{BUENOS_\circ}, AIRES) \log \frac{P(AIRES|\overline{BUENOS_\circ})}{P(AIRES)} \\ & + P(\overline{BUENOS_\circ}, \overline{AIRES}) \log \frac{P(\overline{AIRES}|\overline{BUENOS_\circ})}{P(\overline{AIRES})} \end{aligned} \tag{5.10}$$

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. This can be done by changing the definition of $A_\circ$. Let $h = w_1^{i-1}$. Then the old definition:

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

is changed, in the context of trigrams, to:

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

I designate this variant of mutual information with MI-3g$(A_\circ, B)$.

Using the WSJ occurrence file described in section 2.6.2 , I filtered the 400 million possible (ordered) trigger pairs of the WSJ’s 20,000 word vocabulary. 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-3g$(A_\circ, B)$ 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 5.1. A larger sample is provided in appendix C.

Browsing the complete list, several conclusions could be drawn:

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 baseform are also good predictors.

3. In general, there is great similarity between same-baseform words:

   - The strongest association is between nouns and their possessive, both for triggers (i.e. $B \Leftarrow \ldots$XYZ, $\ldots$XYZ’S $\ldots$) 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.

HARVEST $\Leftarrow$ CROP HARVEST CORN SOYBEAN SOYBEANS AGRICULTURE GRAIN DROUGHT GRAINS BUSHELS

HARVESTING $\Leftarrow$ CROP HARVEST FORESTS FARMERS HARVESTING TIMBER TREES LOGGING ACRES FOREST

HASHEMI $\Leftarrow$ IRAN IRANIAN TEHRAN IRAN'S IRANIANS LEBANON AYATOLLAH HOSTAGES KHOMEINI ISRAELI HOSTAGE SHIITE ISLAMIC IRAQ PERSIAN TERRORISM LEBANESE ARMS ISRAEL TERRORIST

HASTINGS $\Leftarrow$ HASTINGS IMPEACHMENT ACQUITTED JUDGE TRIAL DISTRICT FLORIDA

HATE $\Leftarrow$ HATE MY YOU HER MAN ME I LOVE

HAVANA $\Leftarrow$ CUBAN CUBA CASTRO HAVANA FIDEL CASTRO'S CUBA'S CUBANS COMMUNIST MIAMI REVOLUTION

Figure 5.1: The best triggers "A" for some given words "B", in descending order, as measured by the MI-3g$(A_\circ, B)$ variant of mutual information.

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$\rightarrow$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 (see section 2.1.1). 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$^+$ 94]).

5.2.3 A simple ME system

The difficulty in measuring the true utility of individual triggers means that, in general, we 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), I conducted the following experiment on the 38 million words of the WSJ corpus language training data (vocabulary=19,981, see appendix A). First, I created a ME model incorporating only the unigram constraints. Its training-set perplexity (PP) was 962 — exactly as calculated from simple Maximum Likelihood estimates. Next, for each word 'B' in the vocabulary, I chose the best predictor 'A' (as measured by standard mutual information). The 19,981 trigger pairs had a total MI 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, I constructed ME models incorporating 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. 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[1]. Results are summarized in table 5.2.

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%, I 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.

[1]I 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.

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·ME + 0.25·trigram perplexity	129	127
perplexity reduction	25%	27%

Figure 5.2: Maximum Entropy models incorporating *N*-gram and trigger constraints.

5.4 Class Triggers

5.4.1 Motivation

I mentioned in section 5.2.2 that strong triggering relations exist among different inflections of the same baseform, similar to the triggering relation a word has with itself. It is reasonable to hypothesize that the triggering relationship is really among the baseforms, not the surface variations. This is further supported by our intuition (and observation) that triggers capture semantic correlations. One might assume, for example, that the semantic baseform "LOAN" triggers the semantic baseform "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", "BANKING", "BANKER" and "BANKERS" occurring next.

It should be noted that class triggers are not merely a notational shorthand. Even if we 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 $AA \stackrel{\text{def}}{=} \{A_1, A_2, \ldots A_n\}$ be some subset of the vocabulary, and let $BB \stackrel{\text{def}}{=} \{B_1, B_2, \ldots B_n\}$ be another subset. The ME constraint function for the class trigger ($AA \Rightarrow BB$) is:

$$f_{AA \to BB}(h, w) = \begin{cases} 1 & \text{if } (\exists A, A \in AA, A \in h) \wedge w \in BB \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

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

$$\sum_h \tilde{\mathrm{P}}(h) \sum_w P(w|h) f_{AA \to BB}(h, w) \; = \; \tilde{\mathrm{E}}\,[f_{AA \to BB}] \tag{5.12}$$

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, we could use any of the general methods discussed in section 2.4 : clustering by linguistic knowledge, clustering by domain knowledge, or data driven clustering.

To estimate the potential of class triggers, I chose to use the first of these methods. The choice was based on the strong conviction that some baseform 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[2], I mapped each word in the vocabulary to one or more baseforms. I then reversed that mapping to create word clusters. The $\sim$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 5.3.

Next, I trained two ME models. The first included all "word self-triggers", one for each word in the vocabulary. The second included all "class self-triggers", one for each cluster. 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 allowed me to quickly estimating 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 5.4.

Surprisingly, baseform 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. I therefore repeated the experiment, this time training on 5 million words. Results are summarized in table 5.5, and are even more disappointing. The class-based model is actually slightly worse than the word-based one (though the difference appears insignificant).

---

[2] I am grateful to David Evans and Steve Henderson for their generosity in providing me 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

Figure 5.3: A randomly selected set of examples of baseform clustering, based on 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

Figure 5.4: Word self-triggers vs. class self-triggers, in the presence of unigram constraints. Baseform clustering does not help much.

Why did baseform clustering fail to improve perplexity? I 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 baseform 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

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

Figure 5.5: Word self-triggers vs. class self-triggers, using more training data than in the previous experiment (table 5.4). Results are even more disappointing.

depends on the frequency of both its words).

5.5 Long Distance N-grams

In section 2.5 I showed that there is quite a bit of information in bigrams of distance 2, 3 and 4. But in section 3.1.4, I 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

$$f^{[j]}_{\{w_1,w_2\}}(h, w) = \begin{cases} 1 & \text{if } h = w_1^{i-1}, w_{i-j} = w_1 \text{ and } w = w_2 \\ 0 & \text{otherwise} \end{cases} \tag{5.13}$$

and its associated constraint is

$$\sum_h \tilde{\mathrm{P}}(h) \sum_w P(w|h) f^{[j]}_{\{w_1,w_2\}}(h, w) \;=\; \tilde{\mathrm{E}}\,[f^{[j]}_{\{w_1,w_2\}}]. \tag{5.14}$$

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

$$\tilde{\mathrm{E}}\,[f^{[j]}_{\{w_1,w_2\}}] \;\stackrel{\text{def}}{=}\; \frac{1}{N} \sum_{(h,w)\in\text{TRAINING}} f^{[j]}_{\{w_1,w_2\}}(h, w). \tag{5.15}$$

And similarly for the unigram and trigram constraints.

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 5.6. 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-based workstation in 24 hours.

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%

Figure 5.6: 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 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 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 we anticipated that the information lost from the former knowledge source will be re-introduced later, at least partially, by interpolation with the conventional trigram. 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, I constructed Maximum Entropy models 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 5.6. Results are summarized in table 5.7.

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%

Figure 5.7: Perplexity of Maximum Entropy models for various subsets of the information sources used in table 5.6. With 1MW of training data, information provided by distance-2 *N*-grams largely overlaps 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 have a large overlap with that provided by the triggers. In contrast, distance-2 *N*-grams resulted in an additional 6% perplexity reduction in the 5MW system (see tables 5.2 and 5.6).

5.7 Handling the Computational Requirements

As can be seen in table 5.6, the computational burden of training a Maximum Entropy model for a large system is quite severe. The Generalized Iterative Scaling algorithm is highly CPU-intensive. Worse yet, for the 38MW system its working set exceeds 64MB, preventing the use of many currently available computers. In what follows I discuss several ideas for dealing with these problems. Where indicated, I have implemented these ideas in the current system.

5.7.1 Parallelization

The computational heart of the GIS algorithm is in accumulating the expectations, over the training data, of all the constraint functions. Since expectations are additive, this lends GIS to easy parallelization.

I have investigated the use of massively parallel machines. Collecting expectations is done in a very uniform manner, making both SIMD (single instruction stream, multiple data streams) and MIMD (multiple instruction streams, multiple data streams) architectures possible candidates. Unfortunately, collecting expectations requires a significant amount of "private", writable memory, proportional to the number of constraint functions. For the 38MW system, more than 20MB of writable memory were required. This ruled out the use of machines like MASSPAR or CM5, where every processor is assigned only a small amount of private memory. The large working set (more than 64MB for the 38MW system) also ruled out most available multi-processor workstations. I therefore decided to parallelize the training on conventional workstations.

The first step in the training procedure consists of computing the desired values for the expectations, by counting events in the training data. This step takes less than half the time of a single iteration and is therefore not a significant computational burden. I therefore implemented it in a single, non-parallelized process, dubbed `initiator`.

Following `initiator`, every iteration consists of two basic steps (see section 4.4):

1. Use the current model to collect expectation for all constraint functions, over the training data. This is where the bulk of the computation takes place.

2. Compare the expectations to their desired values, and update the model's parameters accordingly. This step is not computationally intensive.

Step 1 was parallelized as follows: the training data was partitioned into small segments. A master scheduler, `nudnik`, kept track of available machines, and scheduled `slave` processes that computed partial expectations based on the data in individual segments. `nudnik` also kept track of running `slaves`, restarting them if necessary. When `nudnik` detected that all segments were complete, it invoked `updator`, which added up the partial expectations, and performed step 2 above. Then `nudnik` started scheduling `slaves` for the next iteration. The segments could be of any size, but were typically chosen to require a few hours to process. This was a compromise between disk-space requirements (for storing partial expectations), and minimizing processor idle time. A typical portion of `nudnik`'s log is in table 5.8.

5.7.2 Reducing Computation per Iteration

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

Based on this analysis, computation can be reduced by judiciously pruning the bigram constraints. This must be carefully balanced against the information loss. Some of that loss can be recouped by later interpolating the ME model with a conventional trigram. The high cutoffs used for the 38MW system reduced computation in this manner.

```
Oct 10 14:55 iter #9, job 71001--72000 on alpha4 started
Oct 10 14:57 iter #9, job 45001--46000 on alf9 finished
Oct 10 14:57 iter #9, job 71001--72000 on alpha4 ABORTED  *****
Oct 10 15:00 iter #9, job 71001--72000 on alpha4 started
Oct 10 15:01 iter #9, job 81001--81500 on alf9 started
Oct 10 15:12 iter #9, job 62001--63000 on alpha1 finished
Oct 10 15:13 iter #9, job 71001--72000 on alpha4 ABORTED  *****
Oct 10 15:14 iter #9, job 71001--72000 on alpha1 started
Oct 10 15:17 iter #9, job 61001--62000 on alf10 finished
Oct 10 15:20 iter #9, job 79001--79500 on arrow.boltz finished
Oct 10 15:20 iter #9, job 79501--80000 on wing.boltz finished
Oct 10 15:22 iter #9, job 81501--82000 on alf10 started
Oct 10 15:25 iter #9, job 82001--82500 on wing.boltz started
Oct 10 15:27 iter #9, job 82501--83000 on arrow.boltz started
Oct 10 15:35 iter #9, job 63001--64000 on ubeda finished
Oct 10 15:36 iter #9, job 64001--65000 on alf7 finished
Oct 10 15:37 iter #9, job 83001--83500 on tink.boltz started
Oct 10 15:39 iter #9, job 70001--71000 on hp5 finished
Oct 10 15:39 iter #9, job 83001--83500 on tink.boltz ABORTED  *****
Oct 10 15:39 iter #9, job 83001--83500 on alf7 started
```

Figure 5.8: A portion of the log produced by `nudnik`, the scheduler used to parallelize the ME training procedure. Each ‘‘job” (segment) is identified by the range of WSJ articles it processes.

Another way to cut computation per iteration is to reduce the amount of training data. Expectations can be estimated from a fraction of the data. Common constraints (those that are non-zero frequently) can be estimated with very little data, whereas less common constraints would require a larger sample. Estimation can be used in the first few iterations, where exact values are not needed. This was implemented in the code but never used.

5.7.3 Speeding Up Convergence

Speeding up convergence means reducing the required number of iterations. GIS is a search algorithm, where every iteration moves us closer to the solution point. Unfortunately, that search takes place in a very high dimensional space. At each iteration, each parameter is moved towards its final position. The nature of the update rule is such that the magnitude of the move depends on the values of the associated constraint function. These values are, in turn, constrained to sum to unity at every point (h, w) in the training data (see [Darroch+ 72]. All this suggests that the magnitude of the steps taken at each iteration is roughly inversely proportional to the maximum number of active constraints at any training datapoint.

To increase the average step taken at each iteration (and hence speed up convergence), the following measures can be used:

1. *Limiting the maximum number of constraints which are active for each word w in the vocabulary*. This was done, for example, when we restricted the triggers to no more than 3 predictors per word. One can also modify the *N*-gram constraints such that only one of them is active for any training datapoint (instead of up to three). This is done by redefining the bigram constraints to exclude those datapoints that are part of a trigram constraint, and redefining unigram constraints to exclude those datapoints that are part of a bigram or trigram constraint. This too was implemented in our system ([Lau 93]).

   This argument works in the opposite direction as well: every new source of information will require new constraints, which may increase the maximum number of active constraints per word, thus slowing down convergence.

2. *Dynamically modifying the values of the constraint functions*. This allows us to emphasize some constraints over others during certain iterations, allowing the associated parameters to migrate faster towards their final position. Emphasis can then be reversed to help the other parameters along. The numerical values of the constraint functions can be modified between iterations, as long as the parameters μ_i are adjusted accordingly. These changes amount to a linear transformation of the search space ([Brown+]). This was implemented in a preliminary version of our system ([Lau 93]).

3. *Using alternative search methods*. As was mentioned in section 4.5, the ME solution to the conditional probability problem is also the Maximum Likelihood solution among the exponential family defined by the constraint functions. This 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 ([Brown+]).

5.7.4 Summary

There are many ways to combat the computational requirements of training a Maximum Entropy model. Only a few of them were explored here, and fewer yet were actually implemented. One should note, though, that ME computation is a concern only when dealing with problems of the size we attacked here: millions of constraint functions, and tens of millions of data points. But the Maximum Entropy approach can be used to tackle many problems that are much smaller in magnitude, yet no less daunting. One such example is [Ratnaparkhi+ 94]. Even within the narrow task of reducing perplexity, large amounts of training data are often unavailable, making ME modeling even more attractive.

Chapter 6

Adaptation in Language Modeling

6.1 Adaptation Vs. Long Distance Modeling

This thesis 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 a distinction based on the nature of the language source, and the relationship between the training and test data. In the next section I propose such a classification.

6.2 Three Paradigms of Adaptation

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

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

Cross-domain adaptation is most important in cases where no data from the test domain is available for training the system. But in practice this rarely happens. More likely, a limited amount of 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[1]. 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, I supplemented the ME model 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, I added conditional bigram and trigram caches, to be described subsequently.

N-gram caches were first reported by [Kuhn 88] and [Kupiec 89]. [Kuhn$^+$ 90, Kuhn$^+$ 90b] employed a POS-based bigram cache to improve the performance of their static bigram. [Jelinek$^+$ 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

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

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

I 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. I experimented with such a trigram cache, constructed similarly to the conditional bigram cache. However, I found that it contributed little to perplexity reduction. This is to be expected: every bigram cache hit is also a unigram cache hit. Therefore, the trigram cache can only refine the distinctions already provided by the bigram cache. A document's history is typically small (225 words on average in the WSJ corpus). For such a modest cache, the refinement provided by the trigram is small and statistically unreliable.

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

6.3.3 Combining the Components

To maximize adaptive performance, I supplemented the Maximum Entropy model with the unigram and bigram caches described above. I also added a conventional trigram (the one used as a baseline). 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.

I therefore started the Maximum Entropy model with a weight of $\sim$0.3, which was gradually increased over the first 60 words of the document, to $\sim$0.7. The conventional trigram started with a weight of $\sim$0.7, and was decreased concurrently to $\sim$0.3. The conditional bigram cache had a non-zero weight only during a cache hit, which allowed for a relatively high weight of $\sim$0.09. The selective unigram cache had a weight proportional to the size of the cache, saturating at $\sim$0.05. The threshold for words to enter the 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 6.1 summarizes perplexity (PP) performance of various combinations of the trigram model, the Maximum Entropy model (ME), and the unigram and bigram caches, as follows:

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.

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%

Figure 6.1: 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.

Maximum Entropy: These numbers are reproduced from table 5.6. 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 I 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 adaptove 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 6.2. 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

Figure 6.2: 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 phenomena 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 we rely less on correlations in the training-data, and more on correlations that are assumed to be universal (mostly self-correlations).

Table 6.3 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

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%

Figure 6.3: 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.

be a good trigger pair in 1993 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]).

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 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 I 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 6.4 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 perplexity reduction	 135 21%
ME + trigram + caches perplexity perplexity reduction	 114 33%

Figure 6.4: 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.

Chapter 7

Use of Language Modeling in Speech Recognition

Perhaps the most prominent use of language modeling is in automatic speech recognition. In this chapter, I discuss issues pertaining to the interface between a language model and a speech recognizer, and report on the effect of our improved models on the performance of SPHINX-II, Carnegie Mellon's speech recognition system.

7.1 Interfacing to the Recognizer

7.1.1 Different Types of Recognizers

Speech Recognizers are by and large search algorithms. The search takes place in the space of all possible utterances. An ideal outcome of the search is that utterance which best matches the incoming acoustic signal, given a linguistic and pragmatic context. Since the search space is very large, it is not tested exhaustively. Rather, some heuristics are used to guide the search process, and prune some possibilities. Consequently, *search errors* often result. These are utterances that (according to the combined acoustic-linguistic model) are a better match than the recognizer's output, yet were somehow skipped during the search.

The challenge in interfacing a language model to the recognizer is to minimize search errors while maintaining a feasible computational load.

A general rule in search strategies is to apply knowledge as early as possible, so as to constrain the search. But the stage at which linguistic knowledge can be applied, the extent of that knowledge, and the computational cost of applying it — all depend on the search method being used:

- In *beam search* ([Lee+ 90]), a time-synchronous search is performed. Forward or Viterbi ([Viterbi 67]) decoding is used to carry partial-utterance scores. Linguistic scores are applied at hypothesized word junctures. At that point, the identity of the last word and next word are known. These identities are sufficient for a bigram, which is the language model typically used with beam search. Any language model that looks further back into the history is harder to incorporate, because multiple word paths that end in the same word are usually merged, and all but one of the paths are lost. To use longer-distance models, at least some of these paths should be kept, at a potentially prohibitive rise in the number of states.

- A *Stack Decoder* ([Bahl+ 83]) is a best-first search through a tree of linguistic hypotheses. If the estimation function applied at the nodes never overestimates the remaining cost, the optimal path is guaranteed to be found ([Rich 83]). But to be efficient, the estimate function must not be too far off. Thus the success of this method often hinges on finding a good estimator. The big advantage of stack decoding for language modeling is that the tree structure allows access to the entire sentence history from any node. This facilitates the use of long-distance models.

- Recently, *multi-pass decoders* ([Soong+ 90, Austin+ 91, Huang+ 93c]) have been proposed and used. These consist of 2–5 sequential passes. There are many variations. In the most general scenario, search starts with one or two time-synchronous passes over the acoustic signal. This is the most computationally intensive stage, at which acoustic matching is performed. The result is a lattice of word hypothesis, with possibly multiple begin- and end-times. At this point, rescoring passes may be applied, which attempt to improve the tentative linguistic and perhaps acoustic scores assigned by the previous passes. Next, a search (typically best-first) is performed on the lattice, which may use a longer-distance language model, and which results in an ordered list of the highest-scoring *N* hypotheses ("N-best"). Finally, *N*-best rescoring may be applied to the list, potentially modifying any of the scores, then combining them to produce the top hypothesis.

  This gradual decision making process allows for a similarly gradual application of linguistic knowledge. Different language models can be used with the different passes. During the first, computationally intensive pass, the quick bigram may be best. Rescoring may then use a trigram to improve the linguistic scores. Progressively more complicated models can be used at the later stages. This is especially true during the last, *N*-best rescoring stage. At this point, only several hundred hypotheses are typically considered for each sentence, allowing time for very elaborate processing. On the other hand, applying superior knowledge may be too late at that point, since the correct hypothesis may have already been pruned.

7.1.2 The language model interface to SPHINX-II

The SPHINX-II version I used ([Huang+ 93c]) is a multi-pass decoder. The first two passes ("forward" and "backward") use a bigram language model and generate a word lattice. The third pass is a best-first search of that lattice. This is where the adaptive language model was introduced (during the contrastive, or baseline run, a conventional trigram was used instead). Every time the search process needed to expand a node, it called on the language model, providing it with the path to that node. The language model also had access to the previous sentences in the same document. Upon return, it produced an array of probabilities, one for each word in the vocabulary. A complete specification of the interface is given in appendix D. The output of this stage was a list of "*N*-best" hypotheses, which were then re-ranked using a potentially different set of acoustic-linguistic weights (see section 8.4 ahead).

Interfacing the adaptive model to a best-first search posed some problems. An adaptive model incrementally adjusts its parameters in response to each word it encounters along the path. Every time the language model is called, it must undo the effects of the path it was last called with before incorporating the effects of the current path. It does so by "rolling back" the incremental modifications introduced since the last common node, then "rolling forward" the modifications due to the new segment of the current path. But a best-first search always expands the most promising node. Therefore, the sequence of nodes that the language model is called upon to expand is irregular, and much computation is required to perform the "forward-backward" rolling.

One way to alleviate this burden would be to cache the state of the language model at various nodes. When expansion of a new node is requested, the nearest cached ancestor of that node is found, and rolling forward is performed from that point. This could reduce computation considerably. However, the best-first search, as currently implemented, required still less computation than the forward-backward passes. I therefore did not implement this cache.

7.2 Word Error Rate Reduction

7.2.1 The ARPA CSR Evaluation

To evaluate error rate reduction, I used the ARPA CSR (Continuous Speech Recognition) S1 evaluation set of November 1993 [Kubala+ 94, Pallet+ 94, Rosenfeld 94b]. It consisted of 424 utterances produced in the context of complete long documents by two male and two female speakers. I used a version of SPHINX-II ([Huang+ 93]) with sex-dependent non-PD 10K senone acoustic models (see [Huang+ 93c]). In addition to the $\sim$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. As described in section 7.1.2 above, 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 7.1[1].

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

Figure 7.1: Word error rate reduction of the adaptive language model over a conventional trigram model.

In the official ARPA evaluation described above, the OOV words were given special treatment: During the forward-backward passes, they were assigned uniform probabilities. During the best-first pass, they were assigned their average unigram probabilities. This was done for practical purposes, to avoid the need to train a special ME model for the new, extended vocabulary, while maintaining equitable comparison between the baseline and adaptive models.

I also ran a similar experiment, in which the baseline trigram was retrained to include the OOV's, such that the latter were treated as any other vocabulary word. This reduced the baseline word error rate. The ME model was *not* retrained, which penalized it somewhat. And yet, after interpolation, the relative improvement of the adaptive model was not significantly affected (see table 7.2).

language model	word error rate	% change
trigram (baseline)	18.4%	—
unsupervised adaptation	16.7%	-9%
supervised adaptation	16.1%	-13%

Figure 7.2: Word error rate reduction of the adaptive language model over a conventional trigram model, where the latter was retrained to include the OOVs.

7.2.2 Source of the Improvement

The adaptive language model developed in this work uses various sources of information[2]:

1. Information from the current sentence (conventional trigram, distance-2 trigram, triggers).

[1] The error rates in this experiment were higher than those achieved under comparable conditions in other evaluation runs. Upon analysis, this turned out to be due to two of the four speakers being "goats" (speakers which the recognizer performs poorly on).

[2] "source" is used here in a different sense than in the rest of this document

2. Information from previous hypothesized sentences (triggers and caches, unsupervised adaptation).

3. Information from previous correct sentences (triggers and caches, supervised adaptation).

To determine how the different sources contributed to the error rate reduction, I ran another experiment. An unsupervised adaptation system, identical to that reported in table 7.2 was run on the same data, but the context (history) was flushed after every sentence. This created a condition where every sentence was processed as if it were the first one in a document, namely without the benefit of adaptation from previous sentences. Results are presented in table 7.3 (the other results are reproduced from table 7.2). It seems that all three sources contributed significantly to error rate reduction.

	word error rate	% change	source of information
baseline	18.4	—	last two words
adaptive:			
isolated sents.	17.3	-6%	same sentence
unsupervised	16.7	-9%	+ previous hypothesized sentences
supervised	16.1	-13%	+ previous correct sentences

Figure 7.3: Word error rate reduction broken down by source of information.

7.2.3 Cross-Domain Adaptation

To test error rate reduction under the cross-domain adaptation paradigm, I 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 7.4.

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

Figure 7.4: Word error rate reduction of the adaptive language model over a conventional trigram model, under the cross-domain adaptation paradigm.

As was expected, relative improvement is smaller than that achieved under the within-domain adaptation paradigm. This underlines a fundamental limitation of the Maximum Entropy approach: it only models phenomena that are represented in the training data.

7.2.4 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. Yet the associated reduction in recognition word error rate was only 14% under the most favorable circumstances. This does confirm 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 ($\sqrt{0.68} = 0.82 \approx 0.86$).

Why is the impact on error rate not any greater? In addition to the deficiencies of perplexity mentioned in section 1.4.2, 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 our 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 underestimation 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.

Chapter 8

Future Directions

Throughout this document I mentioned possible future extensions of this work wherever it seemed appropriate. I summarize them here briefly, and add a few other topics that I consider for future work in language modeling.

8.1 Within The Maximum Entropy Paradigm

8.1.1 Incorporating New Knowledge Sources

One knowledge source that I identified in section 2.7 but never pursued is what I called "syntactic constraints". More generally, *linguistically derived constraints* can be a boon to existing models, because they seems complementary to the statistical knowledge sources we currently use. Recognizers often output hypotheses that violate basic grammatical constraints, such as tense, gender and plurality agreement. The latter can be easily captured by constraint functions of the type illustrated in chapter 5.

A methodical approach to reducing error rate must start with systematic analysis and classification of the errors currently made by the recognizer. This has recently been started at Carnegie Mellon ([Chase 93, Weide 94]). When the full results are in, more judicious choices can be made.

8.1.2 Speeding Up the Training

In section 5.7.3 I discussed several possible modifications of the Maximum Entropy training procedure, that will hopefully reduce the computational load:

- Dynamically modifying the constraint functions so as to temporarily emphasize some constraints over others, in order to speed their respective parameters along, towards their final position.
- Searching the parameter space using Gradient Descent (instead of the reestimation step of equation 4.12), with step size estimated from independent data.

- Estimating expectations of common constraints from a small subset of the data (in early iterations).

8.2 Improving the Predictors

Perhaps the worst weakness of current language models is that they capture short-term constraints by relying on variables that are defined by their ordinal position in the history. Defining better predictors has been attempted before ([Mercer 92, Bahl+ 89, Brown+ 90b]), but success seems hard to achieve. A recent attempt ([Lafferty+ 92]) seems promising. The framework of Link Grammar ([Sleator+ 91]) is used to capture linguistically meaningful short term dependencies. The latter can then be used to derive better estimates, either by standard Maximum Likelihood methods, or, better still, within the Maximum Entropy framework.

8.3 Adaptation

The need for adaptation has been demonstrated in section 6.4.1. One aspect of adaptation that has not been touched on is the difference between adaptation of style and adaptation of content. The difference between Wall Street Journal articles of different time periods is clearly in content, not in style. The difference between WSJ and, say, New York Times articles of the same period and on the same topic is in style, not in content. If we could somehow separate the two aspects of modeling, we could adapt one without necessarily adapting the other. Better yet, we could "mix and match" style-components and content-components to best model a new language source. However, it is not clear how this kind of separation can be achieved. Generally speaking, style may be captured by Part-Of-Speech modeling, whereas content resides mostly in the lexical identity of open class words. An attempt along these lines was made by [Elbeze+ 90, Cerf-Danon+ 91].

8.4 Combining Acoustic and Linguistic Evidence

As discussed in section 1.2, the quantity to be maximized in the search for the best hypothesis is:

$$\Pr(L|A) \propto \Pr(A|L) \cdot \Pr(L) \tag{8.1}$$

where A is the acoustic signal and L is the linguistic hypothesis. Let AS and LS be the acoustic and linguistic scores, respectively, as computed by the various components of the recognizer. If these scores were equally good, unbiased estimates of true probabilities, the correct way to combine them would be to multiply them together. In log form, we would seek to maximize $\log AS + \log LS$.

But in practice, these estimates are not directly combinable. For one, in some recognizers AS is not an estimate of a discrete probability, but rather of a *probability density* in a high-dimensional space. Worse yet, the dimensionality of that space depends on the number of speech frames in the utterance, and varies from sentence to sentence. Moreover, the acoustic and linguistic scores may be based on different amounts of data, and on models of differing strengths.

because of these factors, most speech recognition systems treat $\log AS$ and $\log LS$ as if they were scores supplied by a "black box", and look for the best empirical way to combine them. They also use additional scores (typically penalties) that factor in the number of words and phonemes in the hypothesis. Typically, the different scores are combined linearly using a single set of weights, which is optimized empirically. The Unified Stochastic Engine (USE) ([Huang$^+$ 93b]) introduced a more elaborate mechanism, where different weight sets were assigned based on the identity of the words in the hypothesis.

One way to further improve this situation is by choosing the weights judiciously, based on the linguistic and/or acoustic context. For example, if a given linguistic context occurred many times in the training data, a trigram's prediction is more reliable than if it occurred only once (or, worse yet, if the model had to back-off). In general, the reliability of a trigram's prediction can be estimated by statistical means. Estimating the reliability of the acoustic score is trickier, but still possible. Once we have these estimates, we can use them to better combine the scores: the more reliable an estimate, the higher its weight.

Appendix 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 until 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+ 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. I refer to this data set as the "WSJ" corpus.

The version of this corpus used in the experiments described in this thesis 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 my 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 included 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, I 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.

My language training set had the following statistics:

- $\sim$ 87,000 article.
- $\sim$ 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

Following is one of the articles in the training data:

<begin_document>

<begin_paragraph>

<s> ALLBRITTON COMMUNICATIONS COMPANY SOLD THE TIMES OF TRENTON TO NEWHOUSE NEWSPAPER GROUP AN ARM OF ADVANCE PUBLICATIONS INCORPORATED THE COMPANIES SAID </s>

<begin_paragraph>

<s> CLOSELY HELD ALLBRITTON BASED IN WASHINGTON BOUGHT THE PAPER IN NINETEEN EIGHTY ONE FROM WASHINGTON POST COMPANY A PUBLISHING AND BROADCASTING CONCERN </s>

<begin_paragraph>

<s> NEW YORK BASED ADVANCE PUBLICATIONS IS A CLOSELY HELD MEDIA COMPANY CONTROLLED BY THE NEWHOUSE FAMILY </s>

<begin_paragraph>

<s> THE COMPANIES WOULDN'T DISCLOSE TERMS OF THE SALE </s>

<s> BUT JOHN MORTON A NEWSPAPER INDUSTRY ANALYST FOR LYNCH JONES AND RYAN NEW YORK ESTIMATED THAT ALLBRITTON HAD PAID ABOUT TWELVE MILLION DOLLARS TO THIRTEEN MILLION DOLLARS FOR THE PAPER IN NINETEEN EIGHTY ONE </s>

<s> I'M SURE IT COMMANDED A MUCH HIGHER PRICE HE SAID </s>

<s> NOTING THAT THE TIMES PUBLISHES IN A COMPETITIVE MARKET HE ADDED IT WOULDN'T SURPRISE ME IF IT SOLD FOR TWENTY FIVE MILLION DOLLARS TO THIRTY MILLION DOLLARS </s>

<begin_paragraph>

<s> DONALD E. NEWHOUSE PRESIDENT OF ADVANCE PUBLICATIONS SAID THAT THE TIMES'S PUBLISHER RICHARD BILOTTI AND ITS EDITOR LINDA GRIST CUNNINGHAM WILL REMAIN IN THEIR POSTS </s>

<begin_paragraph>

<s> ASKED TO COMMENT ON THE SALE MR. BILOTTI SAID THAT <UNK> STRATEGIC PLANNING IS GOING IN A DIFFERENT DIRECTION </s>

<s> IT WAS STRICTLY A BUSINESS DECISION NOT AT ALL A LOSS OF CONFIDENCE IN THE PAPER </s>

<begin_paragraph>

<s> AS PREVIOUSLY REPORTED ALLBRITTON IS PURSUING FURTHER ACQUISITIONS OF NETWORK AFFILIATED TELEVISION STATIONS </s>

<s> AS OF SEPTEMBER THE COMPANY OWNED FIVE NETWORK AFFILIATES </s>

<begin_paragraph>

<s> MR. BILOTTI SAID THE PAPER IS OPERATING IN THE BLACK WITH TOTAL REVENUE GROWING TEN PERCENT TO FIFTEEN PERCENT A YEAR OVER THE PAST THREE YEARS </s>

<s> IN ITS MARKET MR. BILOTTI STATED THE PAPER CONTROLS SIXTY EIGHT PERCENT OF CLASSIFIED ADVERTISING AND SIXTY PERCENT OF RETAIL ADVERTISING </s>

<s> HE DECLINED TO PROVIDE SPECIFIC EARNINGS OR REVENUE FIGURES </s>

<begin_paragraph>

<s> THE MOST RECENT CIRCULATION REPORTS MR. BILOTTI SAID SHOW THE PAPER'S CIRCULATION AT SIXTY THREE THOUSAND EIGHT HUNDRED SEVENTY DAILY AND EIGHTY ONE THOUSAND THREE HUNDRED NINETY SIX SUNDAY </s>

Appendix B

The `interpolate` Program

The program `interpolate` takes as input any number of probability streams. These are assumed to be the output of several language models on a common set of data. The program then runs the Estimation-Maximization (EM) algorithm, to find the set of weights that, when used for linearly interpolating the models, will result in the lowest perplexity on that data.

The data can be partitioned into “bins” by an optional stream of tags. When present, separate sets of weights will be derived for each bin.

In the following, a Maximum Entropy model, a conventional trigram model, and a unigram cache model are interpolated, using a dataset of 325,000 words. The data is partitioned into bins based on the length of the history, namely the number of words since the beginning of the current document. Of the 325,000 words, the first 125,000 are used to find the optimal weights, and the last 200,000 are used to estimate the test-set perplexity of the new, interpolated model. Normally test-set perplexity would be higher than training-set perplexity. This is not the case in this example, which points out that the data set used is not homogeneous.

```
$ interpolate \
        + ME-test.fprobs \
        + x3gram-test.fprobs \
        + ucache-0.001-test.fprobs \
        -tag test-byhislen.tags \
        -cap byhislen.captions \
        -Test 200000

interpolate:
    3 models will be interpolated using the first 124655 data items
    The last 200000 data items will be used for testing
    tags will be taken from "test-byhislen.tags"
    interpolate: data is partitioned into 7 tags
    captions will be taken from "byhislen.captions"
```

```
by history length X:
  0< X <= 10         weights: 0.333  0.333  0.333  ( 2940 items) --> PP=89.24
 10< X <= 20         weights: 0.333  0.333  0.333  ( 2939 items) --> PP=76.61
 20< X <= 50         weights: 0.333  0.333  0.333  ( 8638 items) --> PP=69.15
 50< X <=100         weights: 0.333  0.333  0.333  (12913 items) --> PP=66.46
100< X <=200         weights: 0.333  0.333  0.333  (19280 items) --> PP=80.29
200< X <=400         weights: 0.333  0.333  0.333  (24614 items) --> PP=111.10
400< X               weights: 0.333  0.333  0.333  (53331 items) --> PP=120.57
                        =============>  TOTAL PP = 99.01

by history length X:
  0< X <= 10         weights: 0.441  0.537  0.023  (2940 items) --> PP=62.38
 10< X <= 20         weights: 0.483  0.506  0.011  (2939 items) --> PP=53.22
 20< X <= 50         weights: 0.483  0.475  0.042  (8638 items) --> PP=52.26
 50< X <=100         weights: 0.486  0.456  0.058  (12913 items) --> PP=51.78
100< X <=200         weights: 0.480  0.448  0.072  (19280 items) --> PP=64.19
200< X <=400         weights: 0.476  0.447  0.077  (24614 items) --> PP=89.50
400< X               weights: 0.465  0.450  0.085  (53331 items) --> PP=98.14

                        =============>  TOTAL PP = 78.85 (down 0.2036)

by history length X:
  0< X <= 10         weights: 0.413  0.583  0.004  (2940 items) --> PP=61.18
 10< X <= 20         weights: 0.478  0.517  0.005  (2939 items) --> PP=53.05
 20< X <= 50         weights: 0.498  0.476  0.026  (8638 items) --> PP=52.02
 50< X <=100         weights: 0.516  0.449  0.035  (12913 items) --> PP=51.36
100< X <=200         weights: 0.515  0.441  0.044  (19280 items) --> PP=63.55
200< X <=400         weights: 0.510  0.444  0.046  (24614 items) --> PP=88.45
400< X               weights: 0.498  0.454  0.048  (53331 items) --> PP=96.73
                        =============>  TOTAL PP = 77.93 (down 0.0117)

by history length X:
  0< X <= 10         weights: 0.387  0.612  0.001  (2940 items) --> PP=60.92
 10< X <= 20         weights: 0.473  0.524  0.004  (2939 items) --> PP=53.03
 20< X <= 50         weights: 0.507  0.470  0.023  (8638 items) --> PP=51.99
 50< X <=100         weights: 0.534  0.436  0.030  (12913 items) --> PP=51.28
100< X <=200         weights: 0.537  0.426  0.037  (19280 items) --> PP=63.41
200< X <=400         weights: 0.532  0.431  0.037  (24614 items) --> PP=88.22
400< X               weights: 0.518  0.445  0.037  (53331 items) --> PP=96.44
                        =============>  TOTAL PP = 77.74 (down 0.0024)

by history length X:
```

```
  0< X <= 10         weights: 0.367  0.633  0.000  (2940 items) --> PP=60.80
 10< X <= 20         weights: 0.468  0.528  0.003  (2939 items) --> PP=53.02
 20< X <= 50         weights: 0.513  0.465  0.022  (8638 items) --> PP=51.99
 50< X <=100         weights: 0.548  0.424  0.028  (12913 items) --> PP=51.25
100< X <=200         weights: 0.553  0.412  0.034  (19280 items) --> PP=63.34
200< X <=400         weights: 0.549  0.418  0.033  (24614 items) --> PP=88.13
400< X               weights: 0.533  0.435  0.032  (53331 items) --> PP=96.34
                        =============>  TOTAL PP = 77.66 (down 0.0010)

The weights:

by history length X:
  0< X <= 10     0.3665164465 0.6329447799 0.0005387736
 10< X <= 20     0.4683836866 0.5284524408 0.0031638726
 20< X <= 50     0.5132840973 0.4650724915 0.0216434112
 50< X <=100     0.5481365123 0.4241884394 0.0276750483
100< X <=200     0.5533702047 0.4123587703 0.0342710251
200< X <=400     0.5488876990 0.4179354998 0.0331768012
400< X           0.5332446818 0.4348136401 0.0319416781


NOW TESTING ...

by history length X:
  0< X <= 10         weights: 0.367  0.633  0.000  (4660 items) --> PP=64.18
 10< X <= 20         weights: 0.468  0.528  0.003  (4659 items) --> PP=53.11
 20< X <= 50         weights: 0.513  0.465  0.022  (13364 items) --> PP=51.53
 50< X <=100         weights: 0.548  0.424  0.028  (19913 items) --> PP=55.04
100< X <=200         weights: 0.553  0.412  0.034  (32614 items) --> PP=61.04
200< X <=400         weights: 0.549  0.418  0.033  (44450 items) --> PP=72.71
400< X               weights: 0.533  0.435  0.032  (80340 items) --> PP=88.45

                        =============>  TOTAL TEST-PP = 71.93
```

Appendix C

Best Triggers by the MI-3g Measure

The MI-3g measure was designed to capture the mutual information present in a candidate trigger-pair, while trying to exclude information already provided by the trigram model.

Let $h = w_1^{i-1} \stackrel{\text{def}}{=} \{w_1, w_2, \ldots, w_{i-1}\}$ be the history of a document, and let w by the next word. Define the events W and $W_\circ$ over the joint event space (h, w) as follows:

W : {W=w, i.e. W is the next word. }
$W_\circ$: $\{W \in w_1^{i-3}$, i.e. W occurred somewhere before the last two words of h }

Then given a candidate trigger pair $(A \rightarrow B)$, MI-3g$(A \rightarrow B)$ is defined as:

$$\begin{aligned}
\text{MI-3g}(A \rightarrow B) &\stackrel{\text{def}}{=} I(A_\circ : B) \\
&= P(A_\circ, B) \log \frac{P(B|A_\circ)}{P(B)} + P(A_\circ, \overline{B}) \log \frac{P(\overline{B}|A_\circ)}{P(\overline{B})} \\
&\quad + P(\overline{A_\circ}, B) \log \frac{P(B|\overline{A_\circ})}{P(B)} + P(\overline{A_\circ}, \overline{B}) \log \frac{P(\overline{B}|\overline{A_\circ})}{P(\overline{B})} \qquad \text{(C.1)}
\end{aligned}$$

To derive trigger pairs, I first created an index file for the WSJ training set that contained, for every word, a record of all of its occurrences. I then used this index file to filter the 400 million possible (ordered) trigger pairs of the WSJ's 20,000 word vocabulary. 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-3g$(A_\circ, B)$ 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 sample of the output of the last step is provided below. Each line lists a single triggered-word (B) followed by up to 20 triggers (A) that passed the filtering described above. Within each line, triggers are in decreasing order of MI-3g.

'EM ⇐ 'EM YOU SEASON GAME GAMES LEAGUE TEAM GUYS I BASEBALL COACH TEAM'S FOOTBALL WON HERE ME SEASONS TEAMS MY CHAMPIONSHIP

'N ⇐ 'N PAK BILZERIAN BILZERIAN'S RETAILER KENT STORES PAUL IMPROVEMENT FLORIDA BUYOUT INVESTOR ROCK PAY TAMPA ROLL MUSIC TENDER BEATLES OUTSTANDING

A'S ⇐ A'S OAKLAND DODGERS BASEBALL CATCHER ATHLETICS INNING GAMES GAME DAVE LEAGUE SERIES TEAM SEASON FRANCISCO BAY SAN PARK BALL RUNS

A.'S ⇐ A.'S AGENCY ADMINISTRATION TRANS AGENCY'S AVIATION DRUG SAFETY AIRLINES AERONAUTICS AIR CLOT ICAHN AIRLINE DRUGS CARL SHUTTLE DISSOLVING SHEINBERG FLIGHT

A.S ⇐ A.S TAX RETIREMENT DEDUCTION DEDUCTIONS TAXABLE DEDUCT INDIVIDUAL CONTRIBUTIONS TAXPAYERS ACCOUNTS DEDUCTIBLE TAXES RETURNS INCOME RULES CORPORATIONS TAXED TAXPAYER GRADUATES

AARON ⇐ AARON

ABANDONED ⇐ ABANDONED

ABATEMENT ⇐ ABATEMENT

ABBEY ⇐ ABBEY LLOYDS POUNDS BRITAIN'S LIFE

ABBOTT ⇐ ABBOTT LABORATORIES DRUG BRISTOL PFIZER GENENTECH PHARMACEUTICAL HOSPITAL LILLY MYERS

ABBOUD ⇐ ABBOUD BANCORP BAILOUT HOUSTON CITY TEXAS BANKER CITY'S RESCUE DEPOSIT CHICAGO JENRETTE ROBERT LUFKIN BANKING DONALDSON INSURANCE ASSISTED BANK TROUBLED

ABDUCTED ⇐ LEBANON KIDNAPPERS BEIRUT REBELS KIDNAPPED IRANIAN HOSTAGE

ABDUL ⇐ ABDUL ISLAMIC ARAB ALI MINISTER KUWAIT

ABE ⇐ ABE YASUHIRO TAKESHITA NOBORU NAKASONE NAKASONE'S JAPAN'S LIBERAL FACTION MINISTER PRIME KIICHI MIYAZAWA JAPANESE JAPAN DIET RULING

ABILITY ⇐ ABILITY RATING ABLE RATINGS DEBT

ABITIBI ⇐ ABITIBI NEWSPRINT CANADIAN METRIC TORONTO PRICE PRODUCER CANADA BATHURST PRODUCERS TON MILL FOREST REICHMANN WORLD'S PAPER QUEBEC MILLS INCORPORATED OLYMPIA

ABLE ⇐ ABLE WE HOW WITHOUT WAY TAKE COME CAN'T GOING COMPUTERS PROBLEMS WHETHER USE COMPANIES PROBLEM SYSTEM VERY ABILITY BEING MIGHT

ABNORMAL ⇐ ABNORMAL DISEASE

ABOARD ⇐ ABOARD KILLED PLANE SHIP JET FLIGHT KILLING NAVY JETLINER AIR SOVIET IRAN REBELS MILITARY CRASHED IRANIAN PERSIAN CABIN AIRLINER POLICE

ABORTION ⇐ ABORTION ABORTIONS SUPREME PRO ROE WOMEN WADE INCEST ANTI COURT'S RIGHTS ACTIVISTS PARENTHOOD RAPE COURT REPUBLICAN WEBSTER DEMOCRATIC CHOICE RESTRICTIONS

ABORTIONS ⇐ ABORTION ABORTIONS SUPREME WOMEN PREGNANCY INCEST MEDICAID CLINICS PRO COURT'S ROE LEGISLATURE WADE HEALTH RAPE COURT ANTI FUNDING VETO SENATE

ABOVE ⇐ ABOVE YIELD OH PRICES PRICED POINT TRADERS RATE PAR NONCALLABLE MARKETS POINTS UNDERWRITERS PRICINGS HIGHER DOW JONES ROSE INDEX BONDS

ABRAHAM ⇐ ABRAHAM FEDERATED'S

ABRAMS ⇐ ABRAMS ELLIOTT ASSISTANT CONTRA CONTRAS INTER SECRETARY STATE AID SHULTZ OLLIE IRAN NICARAGUA NORTH'S TESTIMONY OLIVER COLONEL LIEUTENANT NICARAGUAN COSTA

ABRAMSON ⇐ ABRAMSON HUNT BUNKER HERBERT HUNTS UTILITY BANKRUPTCY NELSON

ABROAD ⇐ FOREIGN ABROAD OVERSEAS EXPORTS JAPAN TRADE IMPORTS COUNTRIES FOREIGNERS DOMESTIC ECONOMIC GOODS ECONOMY INTERNATIONAL WORLD DEFICIT CURRENCY JAPANESE GERMANY EXPORT

ABSENCE ⇐ ABSENCE TRADERS LACK NIKKEI FRANKFURT YEN TOKYO PRICES

ABSOLUTE ⇐ ABSOLUTE SUPERCONDUCTORS SUPERCONDUCTIVITY DEMOCRACY SUPERCONDUCTING

ABSOLUTELY ⇐ I YOU WE

ABSORB ⇐ ABSORB BILLION

ABSTRACT ⇐ EXHIBITION ART PAINTINGS ABSTRACT ARTIST ARTISTS MUSEUM RETROSPECTIVE GALLERY PAINTING WORKS PAINTED PAINTER LYRICAL DRAWINGS SCULPTURES SURFACES WORK COLORS IMAGERY

ABU ⇐ ABU NIDAL TERRORIST PALESTINIAN SYRIA LIBYA INTELLIGENCE ARAB AL TERRORISM LEBANON ISRAEL PALESTINIANS TERROR TERRORISTS ISRAELI DAMASCUS ORGANIZATION ATTACKS VIENNA

ABUNDANTLY ⇐ INTERMEDIATE BARREL HEATING CRUDE GASOLINE SEA MERCANTILE DELIVERY

ABUSE ⇐ ABUSE ABUSED DRUG ALCOHOL CHILDREN ABUSING CHILD COCAINE DRUGS CASES TREATMENT CARE SEXUALLY PHYSICAL FAMILIES DEPENDENCY CUSTODY ADDICTS MENTAL PARENTS

ABUSED ⇐ ABUSE ABUSED CHILDREN

ABUSERS ⇐ DRUG DRUGS ABUSERS DISEASE AIDS ABUSE

ABUSES ⇐ ABUSES INVESTIGATION MERC COMMISSION RULES ENFORCEMENT MERCANTILE FRAUD POINT ABUSE DISCIPLINARY CRIMINAL VIOLATIONS ALLEGED MERC'S CASE COMMITTEE TRADES CASES PIT

ABUSIVE ⇐ RIGHTS

ABYSS ⇐ ABYSS

ACADEMIA ⇐ ACADEMIA ACADEMIC

ACADEMIC ⇐ ACADEMIC STUDENTS UNIVERSITY SCHOOL UNIVERSITIES COLLEGE FACULTY SCHOOLS PROFESSORS PROFESSOR EDUCATION STUDENT CAMPUS TEACHING COLLEGES STUDIES UNDERGRADUATE TEACHERS STANFORD ACADEMICS

ACADEMICS ⇐ ACADEMICS UNIVERSITY ACADEMIC UNIVERSITIES

ACADEMY ⇐ ACADEMY SCIENCES ARTS FILMS STOCK FILM POINT TRADING SCHOOL SOVIET CORPORATION DOLLAR COMPANY ACTOR ART STUDIOS SHARES STARS TRAINING OSCAR

ACCELERATE ⇐ INFLATION

ACCELERATED ⇐ PRICES TRADERS

ACCELERATING ⇐ PRICES INFLATION

ACCELERATION ⇐ ACCELERATION AUDI SUDDEN CARS VOLKSWAGEN UNINTENDED HIGHWAY DEFECT TRAFFIC SAFETY AUTO BRAKE CAR TRANSMISSIONS PEDAL MODELS AUTOMATIC COMPLAINTS ENGINE ALLEGED

ACCELERATOR ⇐ ACCELERATOR SUPERCONDUCTING PHYSICS

ACCENT ⇐ MOVIE ACCENT ACTORS HE'S

ACCEPT ⇐ ACCEPT OFFER PROPOSAL NEGOTIATIONS COMPROMISE AGREEMENT ACCEPTING ROSE REAGAN EARNINGS REJECTED TALKS QUARTER INDEX LEADERS POLITICAL FELL MINISTER NET AGREE

ACCEPTABLE ⇐ ACCEPTABLE

ACCEPTANCE ⇐ DEPOSITORY COLLATERAL PREBON FULTON GUIDE REPRESENT ACTUAL PLACED SIXTEENTHS OVERNIGHT TRANSACTIONS BROKERS DISCOUNT DIRECTLY PRIME BASE PAPER INSTITUTIONS KEY AMOUNTS

ACCEPTANCES ⇐ NEGOTIABLE MULTIPLES DEPOSITORY D.S UNSECURED COLLATERAL CERTIFICATES ACCEPTANCE GRADE PREBON GUIDE SECONDARY FULTON DEPOSIT ACTUAL MINIMUM TYPICAL PLACED REPRESENT AMOUNTS

ACCEPTED ⇐ U.S BIDS CITICORP'S AUCTION ACCEPTED TOTALING WEEKLY SUBMITTED WEEK'S PAPER CITICORP SALE COMMERCIAL O. NONCOMPETITIVE ACCEPT RANGED MARKET OFFER SLATED

ACCEPTING ⇐ ACCEPTING FORMER ACCEPT GELLER FEDERICO CORRUPTION EINSTEIN BIAGGI PLEADED AGENCY

ACCESS ⇐ ACCESS COMPANIES TELECOMMUNICATIONS COMMUNICATIONS TELEPHONE SEMICONDUCTOR INFORMATION JAPAN'S ALLOW JAPAN AGREEMENT GOVERNMENT TRADE CHIPS JAPANESE POINT PROVIDE ROSE OPEN COMPUTER

ACCESSORIES ⇐ ACCESSORIES MAKER STORES PRODUCTS APPAREL WHOLESALER INCORPORATED

ACCIDENT ⇐ ACCIDENT SAFETY KILLED ACCIDENTS AVIATION CHERNOBYL NUCLEAR INJURED FLIGHT INVESTIGATORS JET COCKPIT DISASTER PLANE'S TRANSPORTATION CRASHED PLANE TAKEOFF ALOHA REACTOR

ACCIDENTAL ⇐ ACCIDENTAL NUNN MISSILE

ACCIDENTS ⇐ ACCIDENTS SAFETY ACCIDENT INJURIES HIGHWAY DEFECT RECALLING TRAFFIC CARS DEATHS RECALL DRIVERS VEHICLES ENGINE TRANSPORTATION ADMINISTRATION AVIATION INJURED MODEL AUTO

ACCOMMODATIONS ⇐ DISCRIMINATION ACCOMMODATIONS

ACCOMPANIED ⇐ CORPORATION

ACCOMPANYING ⇐ ACCOMPANYING ADJACENT TABLES CHART EXISTED ERLANGER

ACCORD ⇐ ACCORD AGREEMENT PACT AGREED REACHED TALKS MINISTER NEGOTIATORS MINISTERS SIGNED NEGOTIATIONS JAPAN GERMANY REAGAN NATIONS COUNTRIES LOUVRE FOREIGN CURRENCY OFFICIALS

ACCORDING ⇐ ACCORDING INVESTIGATION INVESTIGATORS SECURITIES FUNDS YIELD ATTORNEY FILING MUNICIPAL CRIMINAL INVESTIGATING COMMISSION FEDERAL DOCUMENTS AVERAGED COMMENT FRAUD ALLEGEDLY FAMILIAR OFFICIALS

ACCORDS ⇐ AGREEMENTS ACCORDS AFGHAN ACCORD SOVIET AFGHANISTAN SOVIETS HONDA TROOPS PAKISTAN KABUL PAKISTANI REGIME TALKS GENEVA ZIA AGREEMENT MOSCOW SIGNED HONDA'S

ACCOUNT ⇐ ACCOUNT AD ADVERTISING ACCOUNTS BILLINGS AGENCY SURPLUS SAATCHI INTERPUBLIC AGENCIES THOMPSON CLIENT DEFICIT AYER CLIENTS RUBICAM LINTAS GELLER EINSTEIN TRANSFERS

ACCOUNTABILITY ⇐ ACCOUNTABILITY PUBLIC CONGRESS ACCOUNTABLE HEARINGS RESPONSIBILITY

ACCOUNTABLE ⇐ ACCOUNTABLE ACCOUNTABILITY

ACCOUNTANT ⇐ ACCOUNTANT ACCOUNTANTS ACCOUNTING PROFESSION CERTIFIED AUDIT FINANCIAL PARTNER FRAUD AUDITING BURTON OLD FIRM BUSINESS WHINNEY STATEMENTS FORMER TAX CLARENCE ERNST

ACCOUNTANTS ⇐ ACCOUNTANTS ACCOUNTING CERTIFIED PROFESSION AUDITING ACCOUNTANT AUDIT AUDITS PARTNER INSTITUTE TAX FIRMS STANDARDS AUDITED FIRM DEDUCTIONS CLIENTS AUDITORS ERNST FRAUD

ACCOUNTED ⇐ POINT PERCENT ACCOUNTED SALES EARNINGS ROSE NET SHARE PROFIT QUARTER PRODUCTS BILLION DOLLARS REAGAN TOTAL HIM HOUSE VOLUME HER

ACCOUNTING ⇐ ACCOUNTING ACCOUNTANTS AUDIT AUDITING ANDERSEN AUDITOR NET AUDITORS INCOME ARTHUR EARNINGS ACCOUNTANT FINANCIAL MARWICK AUDITS PROFESSION CERTIFIED WATERHOUSE PEAT PARTNER

ACCOUNTS ⇐ ACCOUNTS DEPOSITS DEPOSIT ACCOUNT FUNDS BANKS BANK ASSETS MONEY REOPEN CUSTOMERS INSURED WITHDRAWALS CHECKING SAVINGS BRANCH BROKER DONOGHUE'S BANKING DOLLARS

ACCREDITATION ⇐ ACCREDITATION HOSPITALS

ACCRUAL ⇐ ACCRUAL LOANS BRAZIL ECUADOR NON BRAZILIAN PLACED LOAN EARNINGS BANK MEDIUM QUARTER STATUS PLACING BRAZIL'S INCOME PAYMENTS BANKS NET DEBT

ACCRUED ⇐ REDEEM REDEMPTION DUE DEBENTURES REDEEMED CONVERTIBLE ACCRUED PREFERRED SUBORDINATED OUTSTANDING DOLLARS PRINCIPAL PERCENT CUMULATIVE FIFTEENTH PRESIDENT WE SERIES BONDS MARKET

ACCUMULATED ⇐ ACCUMULATED

ACCUMULATING ⇐ TAKEOVER STOCK SHARES STAKE RUMORS GILLETTE COMPANY WALL COMPOSITE EXCHANGE INVESTORS TRADING SPECULATION

ACCUMULATION ⇐ ACCUMULATION

ACCURACY ⇐ ACCURACY ACCURATE INACCURATE ERRORS TESTS

ACCURATE ⇐ ACCURATE ACCURACY SHARES INACCURATE

ACCURAY ⇐ ACCURAY COMBUSTION ENGINEERING SYSTEMS OFFER MAKER SHARE INCORPORATED

ACCUSATIONS ⇐ ACCUSATIONS ACCUSED CHARGES

ACCUSED ⇐ ACCUSED COURT CHARGES TRIAL FORMER ALLEGED CHARGED JURY CRIMINAL PROSECUTORS FRAUD ATTORNEY MARKET INDICTMENT INDICTED JUSTICE PERCENT POLICE POINT CONVICTED

ACCUSES ⇐ FILED SUIT COURT LAWSUIT ACCUSES

ACCUSING ⇐ FILED COURT

ACCUTANE ⇐ ACCUTANE ACNE DEFECTS HOFFMANN ROCHE BIRTH DRUG'S DRUG PREGNANT LA WARNINGS FOOD PATIENTS ADMINISTRATION WOMEN SEVERE SWISS DOCTORS USE CASES

ACE ⇐ ACE

ACHIEVE ⇐ ACHIEVE POLICY WE ECONOMIC ACHIEVED GOAL

ACHIEVED ⇐ ACHIEVED

ACHIEVEMENT ⇐ EDUCATION ACHIEVEMENT SCHOOLS STUDENTS POLITICAL

ACID ⇐ ACID RAIN EMISSIONS SULFUR DIOXIDE ENVIRONMENTAL POLLUTION COAL MULRONEY LAKES RESEARCHERS SCIENTISTS POLLUTANTS BURNING FORESTS CLEAN OXIDE PROTEINS DAMAGE CANADIAN

ACKER ⇐ ACKER AM'S PAN AM AIRWAYS EDWARD AIRLINE SHUGRUE SHUTTLE UNIONS BRANIFF CONCESSIONS CHAIRMAN PARENT WORLD AMERICAN TRAFFIC TRAVEL AIRLINES TERRORISM

ACKERMAN ⇐ ACKERMAN DATAPOINT ASHER EDELMAN MARTIN CONSENT

ACKNOWLEDGED ⇐ INTERVIEW ACKNOWLEDGED INVESTIGATION INVESTIGATING OFFICIALS CONTRA ADMINISTRATION INVESTIGATORS TESTIMONY AFFAIR NICARAGUAN TESTIFIED COMMENT CRIMINAL DENIED IRAN COVERT CHAIRMAN REAGAN ATTORNEY

ACKNOWLEDGES ⇐ ADDS I YOU BUSINESS DOESN'T HIM HOW OLD WE ISN'T TOP GOING INDUSTRY PRESIDENT WANT OFTEN LOT OWN TOO WHERE

ACME ⇐ ACME STEEL

ACNE ⇐ ACNE RETIN DRUG ACCUTANE HOFFMANN ROCHE PRESCRIPTION DRUG'S SKIN DEFECTS CREAM LA PHARMACEUTICAL JOHNSON PREGNANT BIRTH

ACQUAINTANCES ⇐ FRIENDS

...

...

...

YATES ⇐ YATES

YEAH ⇐ YOU I ME MY

YEAR'S ⇐ YEAR'S QUARTER NET EARNINGS INCOME EXPECTS PERCENT INCREASE POINT SALES TAX FORECAST FISCAL PROFIT REVENUE COURT ANNUAL COMPARED GROWTH SPENDING

YEARLY ⇐ YEARLY

YEARS' ⇐ SENTENCED YEARS'

YELLOW ⇐ YELLOW DIRECTORIES DIRECTORY BELL NYNEX SAMPLES PAGES PUBLISHERS BELLS DIAL GARDEN FLOWERS ORANGE POWDER DUN BLUE REGIONAL SOUTHWESTERN LISTINGS BIOLOGICAL

YELLOWSTONE ⇐ YELLOWSTONE FIRES PARK FORESTS TREES FOREST ACRES FIRE NATURAL FIREFIGHTERS NATURE BURNED MONTANA NATIONAL WYOMING GROUND ENVIRONMENTAL LAND

YELTSIN ⇐ YELTSIN BORIS GORBACHEV COMMUNIST MIKHAIL SOVIET MOSCOW POLITBURO GLASNOST GORBACHEV'S PARTY PERESTROIKA KREMLIN LEADER LIGACHEV PARTY'S UNION SPEECH REFORM APPARENTLY

YEMEN ⇐ YEMEN ARAB ARABIA SAUDI NORTH EGYPT OIL SEA WEAPONS IRAN EAST COUNTRIES IRAQ

YEN ⇐ YEN TOKYO JAPANESE OH CURRENCY JAPAN JAPAN'S MARKS TRADING DOLLAR'S TRADERS NIKKEI POINT ROSE CURRENCIES UNCONSOLIDATED DOLLAR FELL INTERVENTION GERMAN

YEN'S ⇐ JAPANESE YEN JAPAN'S JAPAN YEN'S TOKYO EXPORTS EXPORT CURRENCY MINISTRY FELL CONSECUTIVE NIKKEI PERCENT POINT DECLINE DOMESTIC ROSE DOLLAR'S CURRENCIES

YES ⇐ YES YOU I ME MY ASKED YOUR HIM KNOW MAN WHY THEN ANSWER EVERY HERE COMPANY TRADING CORPORATION OLD DOES

YESTERDAY'S ⇐ YESTERDAY'S TRADERS TRADING DOW JONES PRICES BOND INDEX RALLY OH POINTS MARKET INDUSTRIAL FUTURES AVERAGE EXCHANGE FELL VOLUME INVESTORS STOCKS

YET ⇐ YET ROSE HASN'T TOO WE OH MAN HOW I HIM MUST QUARTER SEEMS FAR WORLD THOUGH COURSE GREAT FELL HAVEN'T

YETNIKOFF ⇐ YETNIKOFF TISCH SONY LAURENCE RECORDS WALTER S.'S RECORD DIRECTORS DIVISION UNIT OFFICER SOURCES EXECUTIVE JAPAN SALE CHIEF PURCHASE SONY'S MEETING

YEUTTER ⇐ YEUTTER CLAYTON TRADE REPRESENTATIVE IMPORTS TARIFFS REAGAN ADMINISTRATION UNFAIR JAPAN QUOTAS BEEF EXPORTS IMPORT BARRIERS RETALIATION CITRUS JAPAN'S GROWERS OFFICIALS

YIELD ⇐ YIELD BONDS BOND TREASURY'S TREASURY SECONDS NOTES DUE YIELDS INVESTORS PRICED MARKETS EIGHTHS TERM UNDERWRITERS OH MUNICIPAL RATES AMOUNT POINTS

YIELDED ⇐ YIELD TREASURY BONDS BOND YIELDS BENCHMARK TERM TREASURY'S RATES SECONDS BILLS AUCTION TRADERS TREASURYS YIELDED PRICES

YIELDING ⇐ YIELD BONDS YIELDS BOND INVESTORS YIELDING TREASURY RATES GINNIE TERM MUNICIPAL INTEREST MAE PERCENTAGE MARKET BENCHMARK TREASURYS SECURITIES RATE MARKETS

YIELDS ⇐ YIELDS TREASURY YIELD RATES CERTIFICATES D.S DEPOSIT BOND TERM BONDS INTEREST AVERAGE AUCTION BANXQUOTE LIBOR FREDDIE MARKET EURODOLLARS OH QUOTATIONS

YITZHAK ⇐ ISRAELI ISRAEL ISRAEL'S PALESTINIAN PALESTINIANS GAZA ISRAELIS OCCUPIED ARAB PALESTINE LIBERATION YASSER SHIMON

YOGURT ⇐ YOGURT DAIRY FOOD MILK

YONKERS ⇐ YONKERS SAND JUDGE HOUSING FINES LEONARD JAIL CITY DESEGREGATION COUNCIL RESIDENTS COURTS CIVIL COURT IMPOSED RACIAL RIGHTS CASE REFUSED ORDER

YORK ⇐ YORK TRADING IRVING'S GOVERNMENT INCORPORATED IRVING YORK'S TRADERS POLITICAL STOCK SHARES CENTS COMPOSITE EXCHANGE REAGAN MILITARY COMPANY ADMINISTRATION CONGRESS FUTURES

YORK'S ⇐ IRVING'S YORK'S IRVING YORK COMMERCIALE ITALIANA BANCA HOSTILE BANK SUITOR COMMERCIALE'S OLYMPIA FRIENDLY OFFER DEFENSES RICE MILAN MERGER REICHMANN TAKEOVER

YORKER ⇐ YORKER WRITERS MAGAZINE EDITOR WRITER LITERARY CHRYSLER OWNER PUBLISHER EDITORS AD CHRYSLER'S MAGAZINES FEDERICO PAGES

YORKERS ⇐ YORKERS CITY YORK KOCH BROADWAY

YOU ⇐ YOU YOUR I ME MY YOU'RE HOW KNOW THEN I'M WE WANT HER GOOD THAT'S HIM SHE WAY THINK HERE

YOU'D ⇐ YOU I YOU'D YOUR THAT'S YOU'RE

YOU'LL ⇐ YOU YOU'RE YOUR YOU'LL I ME THERE'S OLD MY I'M SEE GO SOMETHING HOW YOURSELF GOT GOING GOOD I'LL THAT'S

YOU'RE ⇐ YOU YOU'RE I YOUR MY ME THAT'S THINK KNOW I'M WANT GOING GOOD SHE LOT GOT HER HOW CAN'T THEN

YOU'VE ⇐ YOU YOU'VE YOUR I YOU'RE GOT ME I'M HOW THAT'S GOOD MY OLD LOOK REALLY I'VE GO NEVER THINK BETTER

YOUNG ⇐ YOUNG OLD AGE MAN HER CHILDREN LIFE HIM I MEN SCHOOL SHE WOMEN COLLEGE POINT TEEN DAE STOCK YOUNG'S KIM

YOUNG'S ⇐ YOUNG YOUNG'S

YOUNGER ⇐ YOUNGER OLDER AGE OLD YOUNG GENERATION WOMEN SON FATHER FAMILY CHILDREN ELDER HER MEN AGING HIM MY RANKS PARENTS AGES

YOUNGEST ⇐ AGE OLD CHILDREN YOUNGEST FAMILY

YOUNGSTERS ⇐ SCHOOL CHILDREN YOUNGSTERS KIDS PARENTS SCHOOLS STUDENTS EDUCATION TEEN YOUNG COLLEGE AGERS LIFE EDUCATIONAL YOU AGE YOUTH DROPOUT SHE ELEMENTARY

YOUNGSTOWN ⇐ DEBARTOLO EDWARD STORES CAMPEAU CAMPEAU'S J. ALLIED

YOUR ⇐ YOUR YOU I MY YOU'RE POINT ME PERCENT TRADING CORPORATION EDITORIAL DOLLARS SHARES HOW YOURSELF BILLION ROSE ANALYSTS KNOW EXCHANGE

YOURSELF ⇐ YOU YOUR YOURSELF YOU'RE I ME MY WANT HIM KNOW WOMAN HOW READ THING THEN LOOK KIND FIND HER

YOUTH ⇐ YOUTH YOUNG OLD TEEN AGE SCHOOL YOUTHS CHILDREN CORPORATION STREETS MAN FATHER SOCIAL LIFE COMPANY HER TRADING COLLEGE YOU SOCIETY

YOUTHS ⇐ YOUTHS YOUTH VIOLENCE POLICE TEEN YOUNG KILLED STUDENTS ISRAELI AGERS BLACK KIDS OCCUPIED

YU ⇐ YU TAIWAN TAIWAN'S CHINA MAINLAND KUOMINTANG CHINESE CHIANG DEMOCRACY BEIJING CHINA'S

YUAN ⇐ YUAN CHINA CHINA'S CHINESE BEIJING SHANGHAI KUOMINTANG CHIANG PEOPLE'S TAIWAN'S TAIWAN REFORM REFORMS ENTERPRISES STATE DENG MAINLAND GOVERNMENT HONG

YUGO ⇐ YUGO CARS CAR IMPORTER AMERICA MOTORS DEALERS MODEL VEHICLE MODELS PRICED

YUGOSLAV ⇐ YUGOSLAVIA YUGOSLAV COMMUNIST YUGOSLAVIA'S BELGRADE SOVIET GORBACHEV MIKHAIL

YUGOSLAVIA ⇐ YUGOSLAVIA YUGOSLAV COMMUNIST YUGOSLAVIA'S BELGRADE SOVIET POLAND MIKHAIL COUNTRY BLOC EAST GORBACHEV HUNGARY ETHNIC PARTY GORBACHEV'S STRIKES COUNTRIES ECONOMIC REFORMS

YUGOSLAVIA'S ⇐ YUGOSLAV YUGOSLAVIA YUGOSLAVIA'S COMMUNIST BELGRADE

YUPPIE ⇐ YUPPIE INSIDER YUPPIES BABY AFFLUENT SCHLOSS ACTORS ADVERTISERS YOUNG COMMERCIALS BOESKY LEVINE BOOM IVAN GARY FEELING HER PERFECT MS. ADS

YUPPIES ⇐ YUPPIES YUPPIE URBAN AFFLUENT YOUNG

YURI ⇐ SOVIET MOSCOW GORBACHEV MIKHAIL GORBACHEV'S YURI GLASNOST BALLET

YVES ⇐ RITZ YVES LAURENT PERFUME FRENCH DIOR VUITTON CARLO DE CERUS FASHION PARIS COSMETICS BENEDETTI LUXURY FINANCIER BRANDS FRANCE'S ITALIAN REVLON

Z. ⇐ Z. WELLCOME AIDS BURROUGHS DEFICIENCY IMMUNE SYNDROME PATIENTS DRUG VIRUS DISEASE WELLCOME'S INFECTED ANEMIA ACQUIRED SYMPTOMS CLINICAL BRODER TREATMENT MARROW

ZACKS ⇐ ZACKS ANALYSTS' EARNINGS ESTIMATES ANALYSTS OMITTED ESTIMATE RESULTS QUARTERLY RECOMMENDED BROKERAGE STOCKS AVERAGE COMPARES PROFIT FORECASTS LYNCH PER DIFFERENCE QUARTER

ZAIRE ⇐ ZAIRE AFRICAN COPPER AFRICA COUNTRIES

ZAMBIA ⇐ AFRICA ZAMBIA AFRICAN COPPER MOZAMBIQUE

ZAPATA ⇐ ZAPATA OFFSHORE DRILLING RESTRUCTURING GAS PAYMENTS

ZAYRE ⇐ ZAYRE ZAYRE'S STORES FRAMINGHAM RETAILER DISCOUNT DEBARTOLO STORE RETAILING AMES SPECIALTY X. MART RETAILERS HAFT DIVISION COMPOSITE EDWARD DEPARTMENT J.

ZAYRE'S ⇐ ZAYRE ZAYRE'S DISCOUNT STORES SPECIALTY STORE X. DEPARTMENT COMPOSITE

ZEALAND ⇐ ZEALAND ZEALAND'S AUSTRALIA AUSTRALIAN LIMITED BRIERLEY EQUITICORP LANGE AUSTRALIA'S SYDNEY FEDERAL PRESIDENT STAKE WELLINGTON PACIFIC HAWKE SIR PEAT WE FLETCHER

ZEALAND'S ⇐ ZEALAND ZEALAND'S

ZELL ⇐ ITEL ZELL HENLEY FE SANTA SAMUEL CHICAGO RAILROAD PACIFIC STAKE ESTATE TRANSACTION ENERGY AGREED REAL ASSETS SAM CORPORATION

ZENITH ⇐ ZENITH ZENITH'S ELECTRONICS GLENVIEW CONSUMER COMPUTER TELEVISIONS ILLINOIS COLOR MAKER PORTABLE LAPTOP SETS COMPUTERS TELEVISION PARTNERS BULL JERRY MACHINES CORPORATION

ZENITH'S ⇐ ZENITH ZENITH'S ELECTRONICS COMPUTER

ZERO ⇐ ZERO INDEX INDUSTRIALS UTILITIES ROSE DOW JONES COMMODITIES TRANSPORTATION PRICES OH FUTURES POINT BONDS SHEARSON VOLUME LEHMAN TREASURY STOCKS FELL

ZHAO ⇐ CHINA'S ZHAO ZIYANG COMMUNIST DENG XIAOPING CHINA DENG'S HU CHINESE PREMIER BEIJING LI PARTY HEIR LEADERSHIP LEADER ACTING DEMONSTRATIONS REFORM

ZIA ⇐ ZIA PAKISTAN PAKISTANI PAKISTAN'S HAQ MOHAMMED AFGHAN AFGHANISTAN ZIA'S PAKISTANIS KHAN SOVIET REFUGEES TROOPS ISLAMABAD KABUL BHUTTO GUERRILLAS SOVIETS REGIME

ZIA'S ⇐ ZIA ZIA'S HAQ PAKISTAN MOHAMMED PAKISTAN'S AFGHAN PAKISTANI AFGHANISTAN PLANE DEATH SOVIET REGIME TROOPS WAR ARMS DIED PRESIDENT KILLED MILITARY

ZICO ⇐ BANCROFT ZICO CONVERTIBLE VIRGIN ISLANDS TENDER FUND HOLDINGS OFFER INVESTMENT BRITISH CONTROLLED MICHAEL SHARES INCORPORATED

ZIEGLER ⇐ ZIEGLER SUTRO GROCERY STORES' STORES LUCKY

ZIMBABWE ⇐ ZIMBABWE AFRICAN AFRICA MOZAMBIQUE AFRICA'S AFRICANS SOUTH WAR INDEPENDENCE COUNTRY GUERRILLAS MARXIST WHITE LIBERATION COLONIAL GUERRILLA NEIGHBORING APARTHEID FORCES

ZIMMER ⇐ ZIMMER SOYBEAN

ZIMMERMAN ⇐ ZIMMERMAN

ZINC ⇐ ZINC COMINCO NICKEL METALS FALCONBRIDGE MINING COPPER NORANDA CANADIAN METAL LIMITED VANCOUVER MINES TONS GOLD LEAD ALUMINUM MINE TORONTO

ZINN ⇐ ZINN ELIAS EDDIE ANTAR CRAZY EDDIE'S ELECTRONICS EDISON ENTERTAINMENT FOUNDER MARKETING CHAIN HOUSTON DISTRIBUTOR STORE CONSUMER J. INCORPORATED STORES RETAILER

ZIP ⇐ ZIP POSTAL MAIL

ZIYANG ⇐ CHINA'S COMMUNIST CHINESE CHINA BEIJING DENG XIAOPING LI PARTY HU REFORM LEADERS LEADER PREMIER INTELLECTUALS

ZOETE ⇐ BARCLAYS WEDD ZOETE LONDON LONDON'S BRITISH POUNDS STOCKBROKERAGE BANKING ANALYST INVESTMENT NATWEST

ZONDERVAN ⇐ ZONDERVAN EVANGELICAL RAPIDS CHRISTIAN PUBLISHER GRAND MICHIGAN INVESTOR CORPORATION GROUP

ZONE ⇐ ZONE ZONES PANAMA MILE PORT STOCK NORIEGA PANAMANIAN TROOPS MILITARY SHIPS VESSELS CANAL BREAKS COMMUNITIES JOE MILES SIDE SOUTH MINISTER

ZONES ⇐ ZONES ZONE COMMUNITIES FOREIGN SENTIMENT UNEMPLOYMENT PORT DEFICITS FACILITIES ENTITLED STATUS GOODS

ZONING ⇐ ZONING DEVELOPER LAND BUILDINGS DEVELOPERS ESTATE RESIDENTS COMMUNITIES HOUSING CITY ORDINANCE BUILDING CITIES PROPERTY ACRES LOCAL NEIGHBORHOODS SUBURBAN COMMUNITY HOMES

ZOO ⇐ ZOO ANIMALS BRONX DIEGO PARK HER ANIMAL SAN BIRDS ACROSS

ZUCKERMAN ⇐ ZUCKERMAN TREASURER CHRYSLER FREDERICK CHRYSLER'S PENSION SALOMON BOND MAGAZINE

ZURICH ⇐ FRANKFURT NIKKEI PENCE LONDON TOKYO DAIMLER WERKE BAYERISCHE MOTOREN BENZ SECTION YEN VOLKSWAGEN DEALERS INDEX MARKS VOLUME BROKERS SESSION STOCKS

ZWEIG ⇐ ZWEIG NEWSLETTER MARTIN EDITOR MUTUAL FUND

ZZZZ ⇐ ZZZZ MINKOW CARPET CLEANING LAUNDERING BARRY BEST'S BEST FRAUD ENTREPRENEUR CARD RESTORATION EXCEEDING BANKRUPTCY RESIGNED CHARGES ANGELES LOS WHINNEY ERNST

Appendix D

The Integrated Language Model (ILM) Interface

The Integrated Language Model (ILM) interface was designed to allow any long-distance language model to be used during the A* stage of SPHINX-II, Carnegie Mellon's speech recognizer. The input to the A* pass is a lattice of word hypothesis, each with multiple begin-times, multiple end-times, and acoustic and linguistic scores. A* proceeds by expanding the most promising nodes, until some number (N) of top-scoring sentence hypotheses are found. When a node is about to be expanded, the language model is consulted. It is given the path ending in that node (h), and a candidate word to be appended to that path (w). The language model must return its estimate of $\log P(w|h)$.

In the following, some implementation details were omitted for clarity.

```
int ilm_init(
   char *ilm_ctl_filename)         /* Language Model control file */


int32 ilm_LOG_prob_of(
   PATH_ID node,           /* index of node to be expanded */
   word    word_id)
/***
  - returns the LOG_probability LOG(P(word|node)).
  - Can be assumed to be called consecutively with many words but same
    node, so caching may be useful.
***/


void ilm_commit_path(
```

```
   PATH_ID node)                    /* ptr to last node in the winning path */
/***
  - used in UNsupervised mode only.
  - called at end of sentence, before starting the next one.
***/


void ilm_commit_external_path(
   PATH_TABLE external_path_table /* special path table for external path */
   PATH_ID node)                  /* ptr to last node in the external path */
/***
  - used in Supervised Adaptation, to commit external sentences.
  - called at end of sentence, before starting the next one.
***/


void ilm_clear_committed_paths()
/***
  - called at end of every context (document, paragraph, ...)
***/
```

Bibliography

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

[Austin+ 91] Steve Austin, Richard Schwartz and Paul Placeway.
The Forward-Backward Search Algorithm.
In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 697–700, Toronto, Canada, 1991.

[Bahl+ 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.

[Bahl+ 89] Lalit Bahl, Peter Brown, Peter deSouza and Robert Mercer.
A Tree-Based Statistical Language Model for natural Language Speech Recognition.
IEEE Transactions on Acoustics, Speech and Signal Processing, **37**, pages 1001–1008, 1989.

[Black+ 92] Ezra Black, Fred Jelinek, John Lafferty, Robert Mercer and Salim Roukos.
Decision Tree Models Applied to the Labeling of text with Parts-of-Speech.
In *Proceedings of the DARPA Workshop on Speech and Natural Language*, published by Morgan Kaufmann, pages 117–121, February 1992.

[Brown+ 90] Peter Brown, John Cocke, Stephen DellaPietra, Vincent DellaPietra, Fred Jelinek, John Lafferty, Robert Mercer and Paul Roosin.
A Statistical Approach to Machine Translation.
Computational Linguistics, Volume 16, pages 79–85, June 1990.

[Brown+ 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+] Peter Brown, Stephen DellaPietra, Vincent DellaPietra, Robert Mercer, Arthur Nadas and Salim Roukos.
Maximum Entropy Methods and Their Applications to Maximum Likelihood Parameter Estimation of Conditional Exponential Models.
A forthcoming IBM technical report.

[Cave+ 80] R. L. Cave and L. P. Neuwirth.
Hidden Markov Models for English.
In *Hidden Markov Models for Speech*, J. D. Ferguson (editor), IDA — CRD, pages 8–15, October 1980.

[Cerf-Danon+ 91] H. Cerf-Danon and M. Elbeze.
Three Different Probabilistic Language Models: Comparison and Combination.
In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 297–300, Toronto, Canada, 1991.

[Chase 93] Lin Chase.
Unpublished work.

[Church 89] Ken Church.
A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text.
In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 695–698, 1989.

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

[Church+ 91] Ken Church and William Gale.
Enhanced Good-Turing and Cat-Cal: Two New Methods for Estimating Probabilities of English Bigrams.
Computer, Speech and Language, Volume 5, pages 19-54, 1991.

[Cover+ 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.

[Csiszar+ 84] I. Csiszar and G. Longo.
Information Geometry and Alternating Minimization Procedures.
Statistics and Decisions, supplement issue **1**, pages 205–237, 1984.

[Csiszar 89] Imre Csiszar.
A Geometric Interpretation of Darroch and Ratcliff's Generalized Iterative Scaling.
The Annals of Statistics, Volume 17, number 3, pages 1409–1413, 1989.

[Darroch+ 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+ 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.

[De Mori+ 91] R. De Mori, R. Kuhn and G. Lazzari. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada, 1991.

[Dempster+ 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+ 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.

[Elbeze+ 90] Marc Elbeze and Anne-Marie Derouault.
A Morphological Model for Large Vocabulary Speech Recognition.
In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 577–580, Albuquerque, NM, April 1990.

[Essen+ 91] Ute Essen and Hermann Ney.
Statistical Language Modelling Using a Cache Memory.
In *Proceedings of the First Quantitative Linguistics Conference*, University of Trier, Germany, September 1991.

[Ferretti+ 89] Marco Ferretti, Giulio Maltese and Stefano Scarci.
Language Model and Acoustic Model Information In Probabilistic Speech Recognition.
In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Pages 707–710, 1989.

[Francis+ 82] W. Francis and H. Kucera. *Frequency Analysis of English Usage.*
Houghton Mifflin Company, Boston, 1982.

[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+ 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+ 93b] Xuedong Huang, Michael Belin, Fileno Alleva and Mei-Yuh Huang.
Unified Stochastic Engine (USE) for Speech Recognition.
In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1993.

[Huang+ 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.

[Isotani+ 94] Ryosuke Isotani and Shoichi Matsunaga.
Speech Recognition Using a Stochastic Language Model Integrating Local and Global Constraints.
In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 87–92, March 1994. To be published by Morgan Kaufmann.

[Iyer 94] Rukmini Iyer, Mari Ostendorf and Robin Rohlicek.
An Improved Language Model Using a Mixture of Markov Components.
In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 82–86, March 1994. To be published by Morgan Kaufmann.

[Jaines 57] E. T. Jaines.
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+ 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+ 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+ 90] Fred Jelinek, Robert Mercer and Salim Roukos.
Classifying Words for Improved Statistical Language Models.
In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Albuquerque, 1990.

[Jelinek+ 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+ 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.

[Kubala+ 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. To be published by 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+ 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+ 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.

[Lafferty+ 92] J. Lafferty, D. Sleator and D. Temperley.
Grammatical Trigrams: A Probabilistic Model of Link Grammar.
Proceedings of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language, Cambridge, MA, 1992.

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

[Lau+ 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+ 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.

[Lee+ 90] Kai-Fu Lee, Hsiao-Wuen Hon and Raj Reddy.
An Overview of the SPHINX Speech Recognition System.
IEEE Transactions on Acoustics, Speech and Signal Processing, pages 35–45, 1990.

[Mercer 92] Robert L. Mercer.
Personal communication.
1992.

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

[Nadas 84] Arthur Nadas.
Estimation of Probabilities in the Language Model of the IBM Speech Recognition System.
IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume ASSP-32, number 4, pages 859–861, August 1984.

[Ney+ 91] Hermann Ney and Ute Essen.
On Smoothing Techniques for Bigram-Based Natural Language Models.
In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 825–828, Toronto, Canada, May 1991.

[Pallet+ 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. To be published by Morgan Kaufmann.

[Paul+ 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.

[Placeway+ 93] Paul Placeway, Richard Schwartz, Pascale Fung and Long Nguyen.
The Estimation of Powerful Language Models from Small and Large Corpora.
In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages II 33–36, Minneapolis, MN, April 1993.

[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+ 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. To be published by Morgan Kaufmann.

[Reddy 76] Raj Reddy.
Speech Recognition by Machine: A Review.
IEEE Proceedings, Volume 64, number 4, pages 502–531, April, 1976.

[Rich 83] Elaine Rich.
Artificial Intelligence.
McGraw-Hill, 1983.

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

[Rosenfeld 94] 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. To be published by Morgan Kaufmann.

[Rosenfeld 94b] Ronald Rosenfeld.
Language Model Adaptation in ARPA's CSR Evaluation.
Oral presentation at *ARPA Spoken Language Systems Workshop*, March 1994.

[Rosenfeld+ 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.

[Schwartz+ 90] Richard Schwartz and Yen-Lu Chow.
Efficient and Exact Procedure for Finding the N Most Likely Sentence Hypotheses.
In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 81–84, Albuquerque, NM, April 1990.

[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.

[Sleator+ 91] D. Sleator and D. Temperley.
Parsing English with a Link Grammar.
Technical Report CMU-CS-91-196, School of Computer Science, Carnegie Mellon University, 1991.

[Soong+ 90] F. Soong and E. Huang.
A Tree-Trellis Based Fast Search for Finding the N-Best Sentence Hypotheses.
In *Proceedings of the DARPA Speech and Natural Language Workshop*, 1990.

[Suhm+ 94] Bernhard Suhm and Alex Waibel.
Towards Better Language Models for Spontaneous Speech.
Subitted to *ICSLP'94*.

[Viterbi 67] A. J. Viterbi.
Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm.
IEEE Transactions on Information Theory, volume IT-13, number 2, pages 260–269, April 1967.

[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.

[Weide 94] Robert Weide.
Personal Communication