Development of an N-Gram Based Language Model for Continuous Speech Recognition

S. P. Given

The Language Modeling Group
Department of Electrical and Computer Engineering
Mississippi State University
Mississippi State, Mississippi 39762
given@ee.msstate.edu

ABSTRACT

An essential element of any speech recognition system is the language model. A language model attempts to identify and make use of the regularities in natural language to better define language syntax for easier recognition. One major obstacle in speech recognition is variability and uncertainty of message content. This, coupled with inherent noise, distortion and losses that occur in speech, emphasize the need for a good language model[1].

Several different types of language modeling techniques exist. This project will concern itself mainly with statistical language modeling. Statistical language modeling uses large amounts of text to automatically compute the model's parameters. This is called training. Language models can be compared using standard measures such as perplexity and recognition or word error rate. This project will use perplexity as a benchmark.

A good language model will provide *a priori* probabilities for all possible queries that the search algorithm may request pertaining to the learned vocabulary. Hence, the complexity of the model is directly related to the size of the corpus upon which it is trained.

46. INTRODUCTION

Building a language model for a continuous speech recognition system is a formidable task. The type of language model to be used is one of the first things that must be considered and this choice has a marked effect on the speech recognition system's performance. Statistical language models have been heavily favored for some time in much of the speech research community because of their savings in complexity. Statistical language models derive their parameters directly from the training corpus[11]. A language model must properly model the training corpus and it must also utilize a method for handling outliers(words or phrases that occur infrequently, possibly never, in the training data). Methods that adjust the model parameters to account for outliers by shaping distributions are called smoothing.

The speech recognition problem is of major importance in the development of more "human" like computers. It is desirable for the computer to understand and act upon vocal commands. This desire is derived from the fact that oral communication is the most common type of communication among humans and therefore most are highly skilled at communicating ideas through speech[5]. Since the majority of end users of computers are not fluent in any computer language the drive is to make the computer fluent in human language. If this push toward more "user-friendly" computing resources is successful, the applications are only a function of ones ability to imagine. Speech recognition has many areas of commercial applications such as: dictation, personal computers, automated telephone services, and special purpose industry applications[5].

Current dictation systems are designed to operate in office settings with head-mounted noise cancellation microphones. They are also speaker dependent as it is expected that one speaker will be using the system for an extended length of time. The two types of dictation systems are “unrestricted”, which are used for letter writing or newspaper articles, and structured systems which are used for things such as report generation where the vocabulary is restricted to a certain type of report(medical, insurance, etc.). Current vocabulary sizes for dictation systems are about 40,000 words[5].

More people are exposed to computers now than ever and the trend toward silicon is ever increasing. To this end the goals of speech recognition are not only to make computers more easy for new users but also to make them more efficient for old pros. The ideas of changing font in mid-keystroke or opening a file by spoken command appeal to people who are on computers for the majority of every day[5]. Some people even suggest that with the trend toward smaller packages, the keyboard may be the limiting factor in size. This school of thought has visions of a totally speech driven interface.

Telephone-based recognition has potential in the areas of banking, credit card application and validation, shopping by catalog, and various customer service avenues. The main problem with this type of system is the uncertainty of conditions of use such as handset and microphone differences, channel noise, and low signal bandwidth[5].

Some other applications for speech recognition are industrial automation and providing necessary interfaces for people with disabilities. The main reason for success in industry is the marked increase in productivity in applications in which recognition systems help or replace human workers[5].

Table 1 shows the progression of speech recognition over approximately two decades[5]. Some helpful acronyms are: SI-speaker independent, SD-speaker dependent, CSR-continuous speech recognition, and IWR-independent word recognition.

Task	Late 70’s	Mid 80‘s	Early 90‘s
SI IWR Alphabet	30%	10%	4%
SI CSR Digits	10%	6%	0.4%
SD CSR Query, 1,000 word (perplexity 6)	2%	0.1%	----
SI CSR Query, 1,000 word (perplexity 60)	----	60%	3%
SD IWR Dictation, 5,000 word	----	10%	2%
SI CSR Dictation, 5,000 word	----	----	5%
SI CSR Dictation, 20,000 word	----	----	13%

Table 1: Progress in speech recognition, as expressed by word error rate.

47. The Statistical Language Model

Natural language can be viewed as a stochastic process with each unit of speech(in our case words) being a random variable with some probability density distribution[1]. Given some speech signal, S, we would like to form some hypothesis as to what gave rise to that particular signal. The front end operates directly on the signal and maps it to some acoustic vector, Y. Providing a measure of the probability of the acoustic vector given some word sequence is the job of the acoustic model.

So, what we would like to know is, given Y, what word or word sequence, W, corresponds to that particular signal. It is the job of the language model to provide a set of probabilities that effectively rank the hypothesis that it is given(figure 1). That is, the language model will give the probability of some future word given a history of previous words that were spoken[3].

This whole idea of ranking a hypothesis can be viewed from an information theory standpoint as well. If we think of an information source emitting messages, W, from a distribution, p(W), into a noisy channel, we can view this channel as a transformation of W into observables, Y, governed by a conditional distribution, p(Y|W)[2].

47.1. N-Gram

The n-gram model uses the previous (n-1) words as the only information source to generate the model parameters. N-grams are easy to implement, easy to interface with, and good predictors of short term dependencies, and thus have become the model of choice among statistical language models[1].

We can view n-grams from the approach that given any state (w_k,w_{k+1}), we will proceed to state (w_{k+1},w_{k+2}) with probability P(w_{k+2}|$w_{k+1}w_k...w_{k-(n-3)}$)[2]. Let's view this approach mathematically. The recognizer would like to find some word sequence

$$W = w_1, w_2, \ldots, w_N$$

that satisfies the argument,

$$p(W|Y) = \overset{max}{\hat{W}} p(W|Y)$$

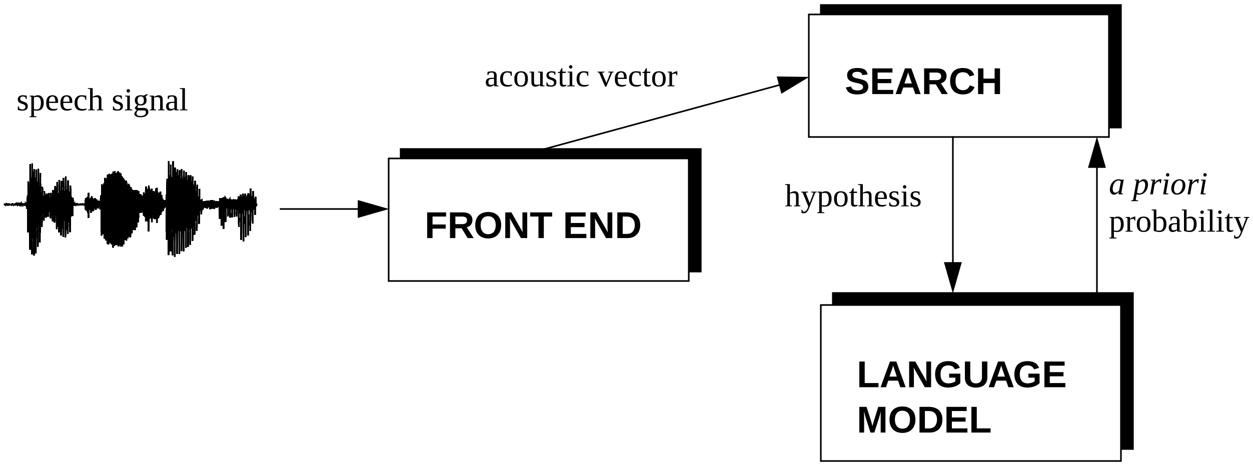

Figure 1: Basic speech recognition system overview with component responsibilities.

or similarly from Baye‘s rule,

$$W = \underset{W}{\arg max}\, p(W)p(Y|W)$$

where W is any word string and Y is the string of acoustical observations[17]. The acoustic model provides the probability p(Y|W). It is the job of the language model to provide the *a priori* information of the training corpus, p(W), which is given by

$$p(W) = \prod_{i=1}^{N} p(w_i|(w_1, w_2, \ldots, w_{N-1}))$$

47.2. Coverage

There is always a trade-off between reliability and detail. This is completely dependent on the size of the training data. The larger the training data, the more likely a large N will yield acceptable results. A smaller training corpus will necessitate a smaller choice in N in order to be reliably sure that an n-gram will exist. In cases where reliability becomes a problem a backoff approach can be used by the search algorithm.

Assume the search engine queries the language model for a certain probability of an n-gram occurring. If the score that is returned by the language model is not “good” according to the search engines criteria, or if the n-gram does not exist, the search engine can then “back off’ from looking for match of length n, to looking for a suitable match of length n-1. This method can be applied until an acceptable score is returned or until unigrams are reached. This is an ARPA standard language model which utilizes the format introduced by Doug Paul. The algorithm for using this model can be seen in figure 2.

The search starts looking for a suitable trigram. If none exists it searches for the corresponding bigram. If the bigram exists then the returned probability will be a product of the bigram backoff weight and the conditional probability associated with words three and two, and if the bigram does not exist, the aforementioned conditional probability is returned. It is easy to see that backoff models provide an efficient method for increasing coverage and hence increasing overall performance of the system.

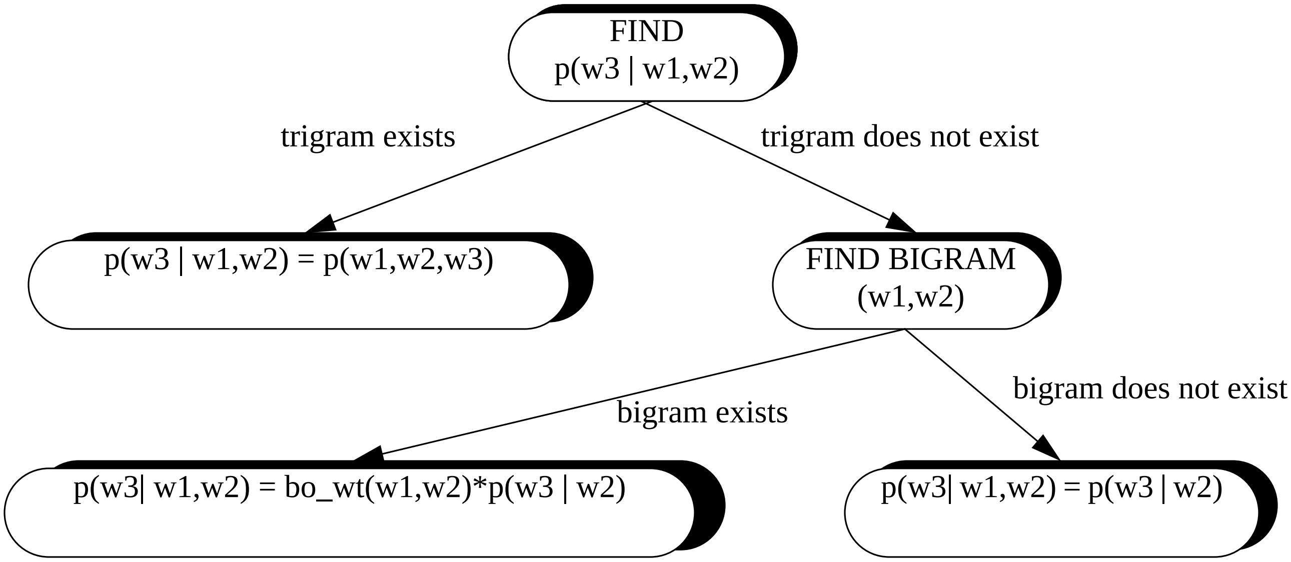

Figure 2: Trigram example of backoff approach.

47.3. Smoothing

Obtaining reliable estimates of the parameters of probabilistic language models is always an issue.Regardless of the size of the training corpus, it is impossible to cover all possible words and word sequences that the model will be queried for ranking. It is, however, desirable that the outliers that do not occur in the training data be accounted for in some manner. There are several methods for dealing with this apparent contradiction and avoiding the possibility of assigning zero probability to any words or word sequences. Setting some type of floor for ranking is one solution. In this case, some probability is assigned to the case where an unknown query is made. The rest of the model's distributions are adjusted accordingly. Another method for resolving this problem is deleted interpolation.

Deleted interpolation is a method for supplementing training data that is insufficient for the speech recognition application for which it is being used[4]. This method uses both tied and untied models properly weighted to lower the number of parameters and thus the variance of the model's distributions. It has the effect of insuring that no zero probabilities are assigned.

For a more detailed insight into deleted interpolation, let's assume that we have a set of training sequences, τ. If we divide τ into two subsets, τ' and τ'', we can train each model, tied(M_t) and untied(M_u), on the two different subsets of training data. The next step is to run recognition experiments on each of the sequences in τ'', using both M_t and M_u. One of the models will produce a better likelihood score for each case. Let ε_t be the fraction of strings in τ'' for which the tied HMM performed better. If A_t and B_t are the state transition and observation matrices for M_t, and M_u has similar matrices associated with it, the matrices for the “hybrid” model that is given by deleted interpolation can be seen below[4].

$$A = \varepsilon_t A_t + (1 - \varepsilon_t) B_u$$

$$B = \varepsilon_t B_t + (1 - \varepsilon_t) B_u$$

As is mentioned in [4], deleted interpolation is not generally as straightforward as stated above. In practice, the training space is usually partitioned iteratively in an effort to obtain even better coverage.

Aside from the intuitive notion that assigning zero probabilities will result in inferior speech recognition performance, there are some mathematical consequences as well. These implications will be discussed in the evaluation section.

48. BUILDING THE MODEL

The Carnegie Mellon University Statistical Language Modeling(CMU SLM)Toolkit[21] was used to create the language model. Some of the issues that must be considered when building a language model are the type of corpus that is available, what size n-gram to use and what type of standard will the model adhere to(i.e. ARPA, LDC, . . .).

48.1. Formatted Data

There are several types of data formats that are common in speech research: verbalized punctuation(VP), some verbalized punctuation(SVP), SGML(ARPA), and text are the main ones that are supported by the CMU SLM.

VP consist of transforming symbols to words: @ -----> at, + ------> plus, etc. SGML is the ARPA standard for CSR use. It consists of text words separated by whitespace and SGML markers enclosed within angled brackets that indicate beginning of article, beginning/end of sentence and various other occurrences that can be exploited to gain some contextual edge. Finally, text is simply words separated by whitespace. The text format also supports beginning/end of sentence markers as well as article markers. For the language model that was created, the text format was used.

48.2. Text-to-Model

The method that is used to build the model can be seen in Figure 3. The text corpus is taken in and compiled into basic trigram counts. From there a vocabulary is generated. Once the vocabulary is generated, the words in the vocabulary are mapped to word ID's numbered 1 to V where V is the number of words in the vocabulary. At this point, the final step is taken to create the language model of choice(in this case the ARPA Backoff)[21].

All of the steps in model generation are performed using scripts that are written for each specific purpose. The CMU SLM Toolkit is a versatile tool that allows the user to build model or a piece of a model and then customize it for some specific application.

49. EVALUATION

Evaluation is arguably the most important part of any research project. Without proper methods and some widely accepted measures, it is difficult to benchmark one's progress. Evaluation plays an important role for system developers(to tell if their system is improving), for consumers(to identify which system best meets their needs) and evaluation also has a way of focusing research[22]. Proper evaluation has led to many advances in the field of speech research such as: development of test corpora, creation of at least four performance workshops, and has resulted in the word error rate decreasing by a factor of two every two years for six years in a row. One of these afore-mentioned performance workshops is the ARPA workshop which features an annual competition evaluation of systems on a common test corpus. This workshop has led to rapid algorithm development and improvement in speech research[5].

In speech recognition, our criterion is recognition accuracy. One direct measurement

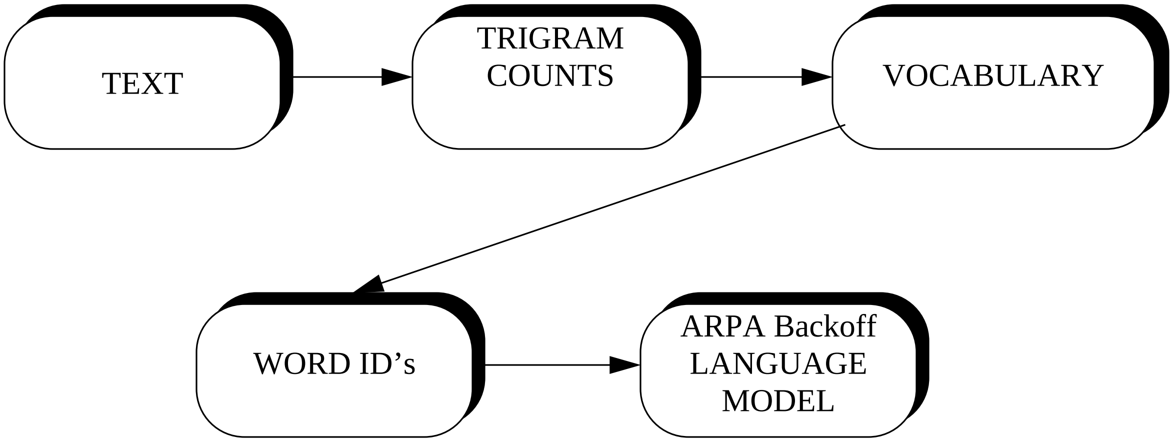

Figure 3: Creation of ARPA Backoff Language Model.

of this is word error rate. Another measure of accuracy, which is not quite as direct, is obtained by measuring the fitness of the language model via some accepted norm such as perplexity.

49.1. Perplexity

Perplexity is a measure of the performance of a language model. Care must be taken when discussing perplexity as it depends on the model and the training data. One can only get valid comparisons when all of these factors are taken into account. In particular, the training data must be the same for the perplexity to have any real meaning at all.

Perplexity can be viewed as the geometric mean of the branch-out factor of the language model[1]. When viewed from this definition the importance of comparing perplexities based on the same training corpus becomes obvious. A small training corpus would result in fewer branches at each node and hence lower perplexity than that of a larger corpus. These corpus size issues illustrate the fact that perplexity measures only give information for specific environments. Perplexity taken out of context has no meaning. The perplexity can be defined as[4]:

$$Q(\underline{w}) = 2^{\hat{H}(\underline{w})} \approx \frac{1}{\sqrt[N]{\hat{P}\left(w_1^N\right)}}$$

where $\hat{H}(\underline{w})$ is the entropy of the model[16].

$$\hat{H}(\underline{w}) = -\lim_{N \to \infty} \log \hat{P}\left(\underline{w}_1^N = \underline{w}_1^N\right)$$

Since entropy is the measure of uncertainty[14,15], it should be obvious that the highest entropic state occurs when all paths leaving the node are equi-probable. It is duly noted that entropy gives a sound measure of difficulty, but speech researchers have chosen to use perplexity instead[19].

It can be seen that if a zero probability were ever assigned in the model, an infinite perplexity would arise. This illustrates the need for some type of smoothing in order to be certain that outliers will be properly modeled. A low perplexity is not a sufficient condition to guarantee a low word error rate[3]. Perplexity, however, is often used as it is uncommon for a low perplexity to lead to a high word error rate.

49.2. Model Specifications

This model uses a vocabulary of 4000 words. It is an open vocabulary, which means that there are no assumptions made about the type of words that it will be queried for(i.e. it expects to be tested using words that do not occur in the training corpus). This model was constructed using 18411 trigrams and their counts.

The unigram part is derived from 17354 words which results in 80 distinct counts and a maximum count of 1049. The bigram component is based on 11727 distinct bigrams with a maximum count of 1049. Only 248 bigrams are kept as 11479 of them occur 5 or fewer times and are consequently discarded. The final component(trigrams) is taken from 16219 distinct trigram counts with the maximum count being 107. Of the 16219 trigrams, only 29 were allowed in the model as 16190 of them resulted in counts less than 10 and were excluded.

It can be reasoned from the above data that this model will serve well as a test model only. While sparse data is one of the foremost problems in language modeling today[20], this model suffers even more severely than most and is intended only as a testbed for small scale recognition.

50. Future Improvements

A larger training corpus is always a good method for improving results. Other improvements that can be made to the basic n-gram language models are usually aimed at making use of any long term dependencies in an attempt to add some contextual element to the model to improve recognition.

One such example is presented in [6]. The approach in this instance is to use grammatical trigrams which utilize a highly lexical grammar, with standard n-grams as a sub-class, to reduce the relative entropy of natural language. In another research effort[7], long-range trigrams are used to allow prediction from not only the two immediate preceeding words, but also from any two preceeding pairs of adjacent words in the same sentence. Some other areas of improvement are cluster models and trigger pairs.

In [2], cluster models are based on a topic-dependent corpus. The attempt is to improve recognition by placing unigram constraints on words having the most mutual information with the topic. A maximum entropy approach using trigger pairs are used in [3] to allow topical adaptation of the model. One study by Ronald Rosenfeld[23] sought improvement by optimizing the size of the vocabulary. The attempt here is to find the best vocabulary size as a trade-off between reducing OOV rate and increasing the model's entropy.

It is evident that an abundance of research is geared toward the advancement of speech technology. The number of ways that models can be "improved" is essentially limitless as there are so many aspects to be optimized and even marginal improvements are considered worthwhile.

51. REFERENCES

159.. Rosenfeld, R. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*, Ph.D. thesis, Carnegie Mellon University, April 1994.

160. Lafferty, S. and Suhm, B., *Cluster Expansions and Iterative Scaling of Maximum Entropy Language Models,* in Fifteenth International Workshop on Maximum Entropy and Bayesian Methods, Kluwer Academic Publishers, 1995.

161. Koppleman, J., *A Statistical Approach to Language Modeling in the ATIS Domain*, MIT Department of Electrical Engineering and Computer Science, January, 1995.

162. Deller, J. R., Proakis, J. G., and Hansen, J. H.L., *Discrete-Time Processing of Speech Signals*. Macmillan Publishing Co., New York, 1993.

163. Rudnicky, A. T. and Hauptmann, A. G. *Survey of Current Speech Technology.* School of Computer Science Carnegie Mellon University, Pittsburg PA, 1993.

164. Lafferty, J., Sleator, D., and Temperley, D. *Grammatical Trigrams: A Probablistic Model of Link Grammar.* presented to 1992 AAAI Fall Symposium on Probablistic Approaches to Natural Language.

165. Gillett, J., Lafferty, J., Pietra, S. D., Pietra, V. D., Printz, H., and Ures, L. *Inference and Estimation of a Long-Range Trigram Model.* IBM, T.J. Watson Research Center, Yorktown Heights, NY.

166. Carter, D. *Improving Language Models by Clustering Training Sentences.* SRI International, 1994.

167. Segal, J., Stolcke, A. *Precise n-gram Probabilities from Stochastic Context-free Grammars.* International Computer Science Institute, Berkeley, California,

appeared in ACL-94.

168. Omohundro, S. M., Andreas, S. *Best-first Merging for Hidden Markov Model Induction.* International Computer Science Institute, Berkeley, California, 1994.

169. Kita, K. *A Study on Language Modeling for Speech Recognition.* KITA Laboratories, 1992.

170. Jelinek, F., Mercer, R. C., and Roukos, Salim. *Principles of Language Modeling for Speech Recognition.* Advances in Speech Signal Processing, pp.651-699, Marcel Dekker,1992.

171. Chollet, G., Capman, F., and Daoud, J. F. A. *On the Evaluation of Recognizers-Statistical Validity of the Tests*. Technical Report, SAM-ENST-02, SAM, 1991.

172. Berger, A. L., Dellapietra, S. A., and Dellapietra, V. J. *A Maximum Entropy Approach to Natural Language Processing.* IBM Technical Disclosure Bulletin, August 1994.

173. Kapur, J. N., and Kesavan, H. K. *Entropy Optimization Principles with Applications.* Academic Press Incorporated, 1992.

174. Blahut, R. E. *Principles and Practice of Information Theory.* Addison-Wesley Publishing Company, 1987.

175. Picone, J., and Deshmukh, N. *Methodologies for Language Modeling and Search in Continuous Speech Recognition.* Institute for Signal and Information Processing, Mississippi State University, ECE Dept.

176. Kupiec, J. *Probabilistic Models of Short and Long Distance Word Dependencies in Running Text.* Xerox Palo Alto Research Center, Palo Alto, CA.

177. Bahl, L. R., Jelinek, F., and Mercer, R. L. *A Maximum Likelihood Approach to Continuous Speech Recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 5, pp. 179-190, Mar. 1983.

178. Katz, S. M. *Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recogniser.* IEEE Trans. ASSP, Vol. 35, No. 3, pp.400-401, 1987.

179. Rosenfeld, Roni. CMU SLM Toolkit Documentation. August 1994.

180. Hirshman, L. and Thompson, H. S. *Overview of Evaluation in Speech and Natural Language Processing.* MITRE Corporation, Bedford, MA.

181. Rosenfeld, R. Optimizing Lexical and N-gram Coverage Via Judicious Use of Linguistic Data. School of Computer Science, Carnegie Mellon University.

182. http://sls-www.les.mit.edu/SLSpubs.html

183. http://www.cs.cmu.edu/

184. http://www.ri.cmu.edu/

185. http://www.mambo.ucsc.edu

186. http://www.speech.cs.cmu.edu/

187. http://www.ll.mit.edu