Computer Speech and Language (1992) **6**, 331–344

A language model for very large-vocabulary speech recognition

V. Gupta, M. Lennig and P. Mermelstein

*INRS-Télécommunications, 16 Place du Commerce, Montreal, Quebec Canada H3E 1H6**

Abstract

We apply a trigram language model to an 86 000-word vocabulary speech recognition task. The recognition task consists of paragraphs chosen arbitrarily from a variety of sources, including newspapers, books, magazines, etc.

The trigram language model parameters correspond to probabilities of words conditioned on the previous two words. The number of parameters to be estimated is enormous: $86\,000^3$ parameters in our case. Even a training set consisting of 60 million words is too small to estimate these parameters reliably. Parameter estimates using relative frequencies would assign a value of zero to a large fraction of the parameters. Many algorithms have been proposed to estimate probabilities of events not observed in the training text. We propose here a simple algorithm for estimating the probabilities of such events using Turing's formula.

The resulting trigram language model reduces the acoustic recognition errors by 60%. We also show that the effectiveness of the trigram language model for correcting an acoustic word recognition error depends on whether or not the neighbouring word contexts occur in the training text corpus for the language model.

1. Introduction

The task of the 86 000-word recognizer designed by INRS-Télécommunications is to transcribe speech into text automatically. The recognition task is divided into two parts: acoustic recognition and language-model-based sentence decoding. For each word of spoken input, the acoustic recognizer generates a list of word hypotheses and their associated acoustic likelihoods. The language component takes this probabilistic word lattice as input and uses a statistical model of the syntactic, semantic and pragmatic properties of English to generate the *a posteriori* most likely word string. The focus of this paper is statistical language modelling.

A number of language models have been used previously for speech recognition. The trigram language model (Jelinek, 1985) has been used successfully for both a 5000-word and a 20 000-word office correspondence task (Averbuch *et al.*, 1987). Derouault and Merialdo (1986) use a tri-POS (parts-of-speech) model for a 250 000-word French recognizer. Application of a trigram language model to a recognition task with such a

* The authors are also with Bell-Northern Research, Montreal, Canada.

0885–2308/92/040331 + 14 $08.00/0

large vocabulary was considered infeasible by Derouault and Merialdo (1986). They also apply global syntactic constraints using a sentence parser. However, application of global syntactic constraints results in only a marginal improvement in the recognition accuracy of their system. The advantage of a tri-POS language model is that it requires significantly less memory for storage than a trigram language model and can be trained from a small training text corpus. However, the tri-POS language model estimates the probability of a word conditioned on the parts-of-speech of the previous two words, resulting in much weaker linguistic constraints than the trigram language model. Lee (1988) and Rohlicek, Chow and Roucos (1988) have used a simpler bigram or word-pair language model in a 997-word resource management task. To make best use of the available syntactic and semantic constraints, we have used a trigram language model in our 86 000-word vocabulary recognition system.

A number of algorithms exist for estimating parameters for the trigram language model from sparse data. These algorithms include the deleted interpolation method (Bahl, Jelinek & Mercer, 1983) and the backoff method (Katz, 1987). In the deleted interpolation method, the probability of word w_3 conditioned on the previous two words w_1 and w_2, $P(w_3|w_1w_2)$, is computed as a weighted average of the relative frequencies[1] $f(w_3|w_1w_2)$, $f(w_3|w_2)$ and $f(w_3)$ in the training text corpus. The deleted interpolation method requires large amounts of storage for the parameters since both the weights and the relative frequencies are stored. The weights are estimated using the forward–backward algorithm, which requires significant computing. The backoff method (Katz, 1987) is storage efficient as it does not require storage of weights. It uses Turing's estimate (Good, 1953) to compute the probability mass of all the trigrams which do not occur in the training text corpus. This probability mass is then distributed among the unseen trigrams using the bigram and monogram counts. In our implementation of the trigram language model, we estimate the probabilities $P(w_3|w_1w_2)$ for the trigrams $w_1w_2w_3$ which do not occur in the training text corpus by direct application of Turing's estimate, without resorting to bigram and monogram counts to partition the probability mass.

To obtain reliable trigram statistics for our 86 000-word recognizer, we analysed a language model training text corpus containing 60 million words. Application of the trigram language model results in a significant reduction in speech recognition errors, verifying the feasibility of using a trigram language model for a very large vocabulary recognition task.

2. Overview of the word recognizer

A block diagram of the recognizer is shown in Fig. 1. Words are recognized in two steps. The first step is the acoustic recognition of the spoken words. The acoustic recognizer has been described in detail by Gupta, Lennig and Mermelstein (1988). We will only discuss here the details pertinent to language modelling. The input to the acoustic recognizer consists of words separated by pauses of at least 150 ms. The spoken text consists of sentences read (without punctuation) from text which was selected randomly from magazines, books and newspaper articles. An end-point detector segments the acoustic data A for the spoken word string $W = w_1^n = w_1, w_2, \ldots, w_n$ into subsegments $A_1^n = A_1, \ldots, A_n$ corresponding to the words $w_1, w_2, \ldots, w_n$ in the word string. For each

[1] For example, the relative frequency of w_3 conditioned on the context w_1w_2 is $f(w_3|w_1w_2) = \frac{C[w_1w_2w_3]}{C[w_1w_2]}$, where the function C counts the number of occurrences of its argument in the text.

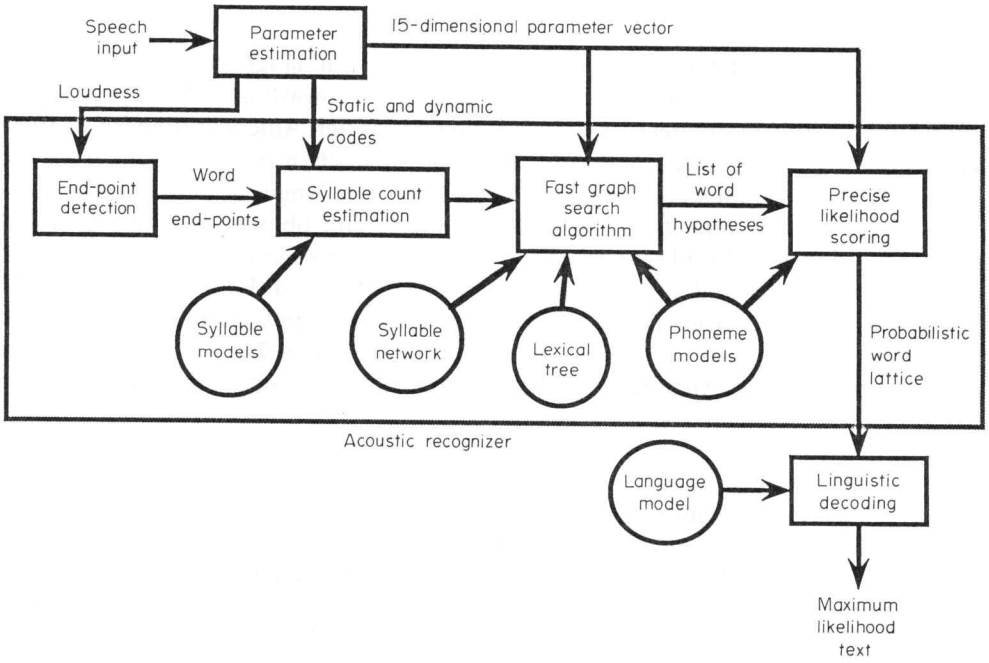

Figure 1. Block diagram of the large vocabulary speech recognition system.

segment A_i, the acoustic recognizer generates a list $\omega_{i1},\ldots,\omega_{iNi}$ of N_i most likely word choices, together with their likelihoods $(P(A_i|\omega_{ij}), j=1,\ldots N_i)$. These likelihoods are smoothed by taking their seventh root before being passed to the second step of recognition. Such a normalization achieves a balance between likelihoods derived from the language model and the acoustic recognizer (Bahl *et al.*, 1980). (The value of seven for the normalizing constant was derived by experimenting on a test set not used in this paper.) The probabilities:

$$P(A|W_k)=P(A_1|\omega_{1k_1})P(A_2|\omega_{2k_2})\ \ldots\ P(A_n|\omega_{nk_n}),$$

where ω_{ik_i} corresponds to one of the hypothesized words for the acoustic segment A_i, are used in the second step of recognition.

The second step of word recognition applies the trigram language model to find the most likely word string $\hat{W}$ using:

$$\hat{W}=\underset{W_k}{\text{argmax}}\ P(W_k)P(A|W_k).$$

The acoustic recognizer generates the probabilities $P(A|W_k)$, while the language model provides the probabilities $P(W_k)$. The search algorithm we have used to find $\hat{W}$ is called the *stack decoding* algorithm (Jelinek, 1976), also known as the A^* algorithm (Nilsson, 1980). The focus of this article is the language model which estimates the word-string probabilities $P(W_k)$.

3. Text databases

We have used four different text databases to train our language model. The first database is the *Brown Corpus* (Francis & Kucera, 1979) consisting of 1·1 million words of text collected from different sources to represent written American English in 1960. The second database, called *hansard*, consists of 14 million words of text from the proceedings of the Canadian Parliament in Ottawa. The language in hansard is more formal and contains repetitious use of many words and phrases, resulting in a low perplexity of around 70. (The perplexity for the text from the *Brown Corpus* and the newspapers is higher and varies from 500 to 1000.) The third database consists of 22 million words of text from *The Globe and Mail* newspaper published in Toronto. The fourth database consists of 23 million words of text from *The Gazette* newspaper published in Montreal. All together, we have trained our language model with 60 million words of text.

4. The lexicon

All recognition experiments reported here are based on a vocabulary consisting of 86 000 orthographically distinct words. For example, *book* and *books* are considered two different words in the lexicon. The philosophy behind the design of this dictionary has been described in detail elsewhere (Seitz *et al.*, 1988, 1989). In short, the 86 000 words in our lexicon include the 60 000 words in the *Merriam Webster's Seventh Collegiate Dictionary* (1965). The remaining 26 000 words are taken from the most frequent words in an 11-million word text corpus (consisting of the *Brown Corpus* and a 10-million word subset of *The Globe and Mail*). These additional 26 000 lexical items include many names, acronyms[2] and inflected forms of words.

5. Estimation of parameters for the trigram language model

The trigram language model estimates the $P(W)$ for all possible word strings $W = w_1^n$ in the language as:

$$P(W) = P(w_1)P(w_2|w_1)\prod_{i=3}^{n} P(w_i|w_{i-2}^{i-1}).$$

In the trigram language model, the only parameters to be estimated are the conditional probabilities $P(w_3|w_1^2)$ for all possible words w_1, w_2, w_3 in the lexicon.

In our case, the total number of parameters to be estimated for the trigram language model is $86\,000^3$. Estimating these conditional probabilities using relative frequencies requires a very large training text corpus to obtain reliable statistics. Such corpora are not available today. Additionally, training using such large corpora could require excessive amounts of computation and storage. We therefore resort to an alternative technique, one using Turing's estimates (Good, 1953) to compute statistics from insufficient data.

Let us first outline Turing's estimates of word probabilities. To estimate word probabilities from a text corpus of size N, let n_r denote the number of words which occur exactly r times in this text. Then the probability of the word w which occurs exactly r times in the text is given by:

[2] Acronyms are treated as single words. For example, GNP is represented as /dʒiɛnpi/.

$$P(w)=\frac{(r+1)n_{r+1}}{n_r N}. \tag{1}$$

The quantity $\frac{(r+1)n_{r+1}}{N}$ is the probability mass of all the words which occur exactly r times in the text. Turing's Formula (1) is an empirical Bayes' estimator of a multinomial probability (Nádas, 1985).

In this application we have used Turing's Formula (1) primarily to estimate the conditional probabilities $P(w_3|w_1^2)$ when the trigram w_1^3 is not observed in the training text corpus. (Note that, using Turing's Formula, the probability of a word (or n-gram) not observed in the text is $\frac{n_1}{n_0 N}$). This probability mass has to come from probability mass of trigrams with counts greater than zero. We have accounted for this probability mass by renormalizing the probability masses for trigrams with small counts in the training text.

Let us first estimate conditional probabilities $P(w_3|w_1^2)$ for trigrams w_1^3 not observed in the training text corpus. The estimation of these conditional probabilities can be divided into three cases: when the bigram w_2^3 is observed in the training text, when the bigram w_2^3 is not observed but the word w_3 is observed and when the word w_3 does not occur in the training text.

If the bigram w_2^3 is observed in the training text, we estimate the conditional probability $P(w_3|w_1^2)$ as an average over all the words w_1 for which the trigram $w_1 w_2 w_3$ does not occur in the training text:

$$P(w_3|w_1 w_2)=\frac{\Sigma_{w_a} P(w_a w_2 w_3)}{\Sigma_{w_a} P(w_a w_2)},\ w_a : C[w_a w_2 w_3]=0, \tag{2}$$

where the function C counts the number of occurrences of its argument in the training text corpus. To estimate the numerator, we group together all trigrams with the same two last words w_2^3. Denote by $C[w_a : C[w_a w_2 w_3]=1]$, the total count of all possible trigrams of the form $w_a w_2 w_3$ which occur only once in this group. Then the estimate of the total probability mass for trigrams $w_a w_2 w_3$ in this group with a count of zero (using Turing's Formula) is $(C[w_a : C[w_a w_2 w_3]=1]/N_1)$, where N_1 is the total number of trigrams in this group. Over the entire training set, the total probability mass is $(C[w_a : C[w_a w_2 w_3]=1]/N_1)(N_1/N)$, where N is the total number of trigrams in the training text.

The denominator in Equation (2) is the total probability mass of all bigrams of the form $w_a w_2$, where w_a corresponds to all possible words with $C[w_a w_2 w_3]=0$ in the training text corpus. In general many such bigrams $(w_a w_2)$ occur in the training text corpus, and we do not have to rely on Turing's Formula to estimate their probability mass. We compute the denominator in Equation (2) as:

$$\sum_{w_a : C[w_a w_2 w_3]=0} P(w_a w_2)=\sum_{w_a} P(w_a w_2)-\sum_{w_a : C[w_a w_2 w_3]>0} P(w_a w_2)$$

$$=C[w_2]/N-\sum_{w_a : C[w_a w_2 w_3]>0} C[w_a w_2]/N$$

$$\approx C[w_2]/N.$$

The last approximation is reasonable in most cases. When $C[w_2]$ is reasonably large, the second sum is significantly smaller than the first sum. The only time the approximation is not accurate is when the count $C[w_2]$ is small. In such cases, the conditional probability estimates are not going to be accurate in any case, and such approximations will not affect recognition accuracy. We have experimented with language models with and without this approximation. The approximation only affects a few words in the 7000-word test set, and the overall recognition accuracy is not affected.

On dividing the numerator by the denominator in (2), we get:

$$P(w_3|w_1w_2) = C[w_a : C[w_aw_2w_3] = 1]/C[w_2]. \tag{3}$$

This is a nice result, since the conditional probability is based on reasonably large counts for the numerator and the denominator. Note that, in Equation (3), the conditional probabilities are no longer proportional to the bigram counts. In a few cases, it is possible that even though the bigram w_2w_3 occurs, the count $C[w_a : C[w_aw_2w_3] = 1]$ is zero. One example is when the sequence $w_bw_2w_3$ occurs more than once, and w_2 does not occur anywhere else in the entire text. In such a case, we feel that using the count $C[w_bw_2w_3]$ would result in a rather high estimate of the probability $P(w_3|w_1w_2)$ for trigrams with $C[w_1w_2w_3] = 0$. Therefore, when $C[w_a : C[w_aw_2w_3] = 1]$ is zero, we fall back to case two or three as appropriate. In other words, case one applies when $C[w_1^3] = 0$ and $C[w_a : C[w_aw_2w_3] = 1] > 0$.

In the second case we estimate the conditional probability $P(w_3|w_1^2)$ when $C[w_2^3] = 0$, but $C[w_3] > 0$. In this case we estimate the $P(w_3|w_1w_2)$ as an average over all bigrams $w_1w_2 : C[w_2w_3] = 0$. We have:

$$P(w_3|w_1^2) = \frac{\Sigma_{w_aw_b} P(w_aw_bw_3)}{\Sigma_{w_aw_b} P(w_aw_b)}, \quad w_b : C[w_bw_3] = 0. \tag{4}$$

In Equation (4) the numerator is estimated by grouping together all bigrams with the same last word w_3. The total probability mass in this group for all bigrams with $C[w_2w_3] = 0$ is given by $C[w_b : C[w_bw_3] = 1]/C[w_3]$. Over the entire text, the probability mass is $C[w_b : C[w_bw_3] = 1]/C[w_3](C[w_3]/N)$. The denominator can be evaluated as:

$$\sum_{w_aw_b : C[w_bw_3] = 0} P(w_aw_b) = \sum_{w_aw_b} P(w_aw_b) - \sum_{w_aw_b : C[w_bw_3] > 0} P(w_aw_b)$$

$$= 1 - \sum_{w_aw_b : C[w_bw_3] > 0} P(w_aw_b)$$

$$\approx 1.$$

Note that the approximation is quite reasonable, since there are very few words w_b for which $C[w_bw_3] > 0$, and the second term is significantly smaller than one. (We have tried our language model with and without this approximation, and we find that the approximation does not affect the recognition results for the 7000 words on which the language model was tested.) In other words, Equation (4) can be written as:

$$P(w_3|w_1w_2) = C[w_b : C[w_bw_3] = 1]/N. \tag{5}$$

In a few cases it is possible that even though the word w_3 occurs, the count $C[w_b : C[w_b w_3] = 1]$ is zero. One example is when the sequence $w_a w_3$ occurs more than once, and w_3 does not occur anywhere else in the entire text. In such cases we feel that the sequence $w_b w_3$ should not be used to estimate the probability $P(w_3|w_1 w_2)$ as it may lead to probability estimates which are too high. In such cases we fall back to the third case. In other words, case two applies when $C[w_2^3] = 0$ and $C[w_b : C[w_b w_3] = 1] > 0$.

In the third case we evaluate the conditional probabilities $P(w_3|w_1^2)$ when $C[w_3] = 0$. $P(w_3|w_1^2)$ is evaluated as the average probability over all words w_3 in the vocabulary such that $C[w_3] = 0$. We have:

$$P(w_3|w_1 w_2) = \frac{(1/N_1)\Sigma w_a w_b w_c P(w_a w_b w_c)}{\Sigma w_a w_b P(w_a w_b)}, \quad w_c : C[w_c] = 0, \tag{6}$$

where N_1 is the total number of words in the vocabulary which do not occur in the training text corpus. Using Turing's Formula, the numerator in Equation (6) is evaluated as $(1/N_1)C[w_c : C[w_c] = 1]/N$, and (6) is reduced to:

$$P(w_3|w_1 w_2) = C[w_c : C[w_c] = 1]/NN_1. \tag{7}$$

Equations (3), (5) and (7) are used to compute the conditional probabilities $P(w_3|w_1^2)$, when $C[w_1^3] = 0$. The complete set of equations for computing the conditional probabilities for all possible trigrams w_1^3 is:

$$P(w_3|w_1^2) = C[w_1^3]/C[w_1^2], \qquad \text{if } C[w_1^3] > 2 \tag{8a}$$

$$= d \times C[w_1^3]/C[w_1^2], \qquad \text{if } C[w_1^3] > 0 \tag{8b}$$

$$= d \times C[w_a : C[w_a w_2 w_3] = 1]/C[w_2], \qquad \text{if } C[w_a : C[w_a w_2 w_3] = 1] > 0 \tag{8c}$$

$$= \mathrm{d} \times C[w_b : C[w_b w_3] = 1]/N, \qquad \text{if } C[w_b : C[w_b w_3] = 1] > 0 \tag{8d}$$

$$= d \times C[w_c : C[w_c] = 1]/NC[w_a : C[w_a] = 0], \qquad \text{otherwise} \tag{8e}$$

where d is a renormalization factor so that $\Sigma_{w3} P(w_3|w_1^2)$ adds up to one. When the total mass of the trigrams in Equation (8b) is large, then the factor d is close to 1·0. The factor d is much smaller when the total mass of the trigrams in Equation (8b) is small. Because of the small mass, we have less confidence in the relative frequencies, and a significant portion of this mass is distributed into unseen events through Equations (8c), (8d) and (8e). There could be other ways of choosing the renormalizing factor d, and d does not have to be the same for Equations (8b)–(8e).

Equations (8a)–(8e) are quite simple to implement for very large training text corpora. In our implementation we have assumed that the conditional probabilities $P(w_3|w_1^2)$ can be correctly estimated from the relative frequencies $f(w_3|w_1 w_2)$ when $C[w_1^3] > 2$, while the conditional probabilities involving trigrams with counts of two or less are all renormalized by a factor d so that $\Sigma w_3 P(w_3|w_1^2)$ adds up to one.

The above equations were applied to train a language model from approximately 60 million words of text: one million from the *Brown Corpus*, 14 million from the debates of the Canadian Parliament, 22 million from *The Globe and Mail* newspaper and 23 million from *The Gazette* newspaper stories. In training the language model we ignore all

punctuation. For example, in the word sequence *Mary went to Canada. In Canada*, the sequences *to Canada in* and *Canada in Canada* are considered trigrams. This is consistent with the way the text is read: without pronouncing punctuation marks or explicitly indicating which words are capitalized.

6. Comparison with the backoff language model of Katz

Equations (8a)–(8e) are quite different from the recursive procedure outlined by Katz (1987) for his backoff algorithm. It is interesting to compare Katz's backoff algorithm with the algorithm outlined here. The major difference is in the way we estimate probabilities $P(w_3|w_1w_2)$ when $C[w_1^3]=0$. In the case of Katz's algorithm, a conditional probability mass $\tilde{\beta}(w_1^2)$ (Katz, 1987) is distributed among all w_3 such that $C[w_1^3]=0$, and the estimate $P(w_3|w_1^2)$ is proportional to $P(w_3|w_2)$. The conditional probability mass $\tilde{\beta}(w_1^2)$, computed by discounting the relative frequencies $\dfrac{C[w_1w_2w_3]}{C[w_1w_2]}$ using a discounting coefficient d_c, is distributed among all w_3 for which $C[w_1^3]=0$. The distribution among various w_3 is done according to the conditional estimates $P(w_3|w_2)$. When $C[w_2w_3]=0$, a similar argument is used to compute and distribute the probability mass according to the estimates $P(w_3)$.

For the estimates given by Equations (8a)–(8e) we estimate the probability mass as $C[w_1 : C[w_1^3]=1]/N$ (as opposed to the use of discounting by Katz) and distribute this mass over all w_1 such that $C[w_1^3]=0$. We feel that, in order realistically to estimate $P(w_3|w_1^2)$, the probability mass of w_1^3 should depend on occurrences of w_1^3, or w_2^3 or w_3, wherever possible. This change in the way we estimate the probability mass leads to some differences in the probability estimates. In many cases, Katz's estimates are close to those obtained by us. In cases where very few samples are available, we see major differences in the estimates for conditional probabilities. Let us examine some of these differences in detail.

Let us compare the case when $C[w_1^3]=0$ but $C[w_2^3]\neq 0$. Consider two possible cases under such circumstances. First, take a case where the sequence $w_aw_bw_c$ occurs five times, and the words w_a and w_b only occur in these five trigrams. According to Katz's estimate, $P(w_d|w_aw_b)$ $(w_d\neq w_c)$ is proportional to $P(w_d)$. We split this situation into two possibilities (Equation 8d or 8e). If $C[w_1 : C[w_1w_d]=1]>0$, then Equation (8d) applies. If $C[w_d]$ is large, then $C[w_1 : C[w_1w_d]=1]$ will be proportional to $C[w_d]$ (approximately 60% of the bigrams have a count of one), and therefore $P(w_d|w_aw_b)$ will be proportional to $P(w_d)$. In such a case, Katz's estimates are similar to our estimates. In the second case, when $C[w_1 : C[w_1w_d]=1]=0$, our estimates are given by Equation (8e). This can happen, for example, when w_d occurs only in one bigram w_ew_d many times. In such cases, our estimate of $P(w_d|w_aw_b)$ will be significantly lower than the estimate due to Katz. We could argue that our estimate is appropriate since the bigram w_ew_d is special and it should not contribute anything towards the estimate of the $P(w_d|w_aw_b)$. Obviously, we can also argue that the estimates differ because of the small amount of data available to estimate these conditional probabilities, and how we interpret this sparse data.

Let us take another case where $C[w_aw_bw_d]=0$, but $C[w_1 : C[w_1w_bw_d]=1]>0$. In such a case, Katz's estimate of $P(w_d|w_aw_b)$ is proportional to $P(w_d|w_b)$. In our algorithm, the probability is determined by Equation (8c). When the count $C[w_bw_d]$ is reasonably large, the count $C[w_bw_d]$ is proportional to the count $C[w_a : C[w_aw_bw_d]=1]$ (approximately 75% of the trigrams have a count of one). In such cases, the probability $P(w_d|w_aw_b)$ is

proportional to $P(w_d|w_b)$. However, when the bigram w_bw_d occurs in very few distinct trigrams, then our probability estimates can differ significantly from those obtained by Katz. For example, let us assume that the bigram w_bw_d occurs five times in the text corpus, and w_b and w_d do not occur anywhere else in the text. In this case, $P(w_d|w_aw_b)$ is proportional to $P(w_d|w_b)$ in Katz's case. In our case, the probability estimates vary depending on the count $C[w_1 : C[w_1w_bw_d] = 1]$, which can be as high as five and as low as zero. It can be zero when w_d occurs five times in the sequence $w_1w_bw_d$ and nowhere else. In such a case, the probability estimate $P(w_d|w_aw_b)$ is based on Equation (8e). Our probability estimate is much smaller than that due to Katz. We can argue that our probability estimate is much more realistic than that due to Katz. In reality, both estimates are probably way off due to a small sample size. We can similarly pick many cases where our probability estimates differ significantly from those of Katz, but the primary reason is the very small number of samples involved in these cases and the interpretation of these observed samples.

7. Recognition results

We have applied the language model on a total of 7240 words spoken by five male and five female speakers. The perplexity (Jelinek, 1985) of this text is 670. The words spoken were taken from various newspaper articles, books and magazines. These articles represent discourse domains similar to the training text corpus. Some samples of the test text are given in the Appendix. The speakers did not pronounce the punctuation marks (*comma*, *period*, etc.) in the text. Since we do not attempt to hypothesize phrase or sentence markers in the recognized word sequences, we only quote a word recognition rate (and not a sentence recognition rate).

Let us first look at details of the acoustic recognizer pertinent to the discussion here. For each word, the acoustic recognizer uses a fast graph search algorithm (Gupta, Lennig & Mermelstein, 1988) to restrict the possible word hypotheses to a maximum of 300 word choices. These hypotheses are then reordered using exact likelihood scores. Since the fast graph search algorithm uses rough likelihoods to rank word hypotheses, it is possible that the correct word would have ranked near the top, even though it was missed by the search algorithm. In fact, if we put the correct word back in the hypothesis list, then approximately 20% of the correct words (missed originally) would be classified as the top choice. Also, 40% of these words would turn out to be top choices after the language model. Hence, when the correct word is missed by the fast graph search algorithm, we call it a search error. In all the experiments reported here, we have not put back the words which were missed by the search algorithm.

It is interesting to examine where the language model is most effective in correcting acoustic recognition errors. We analysed the language model's ability to correct acoustic recognition errors as a function of the coverage of the word and its context by the training set of the language model. Each word in the test set is classified into one of six possible context coverage categories based on the neighbouring words found in the training text corpus. To clarify these coverage categories, let us consider the word *sank* in the partial test sentence *every chair sank several inches*. The word *sank* will be classified into one of the six categories as follows.

- Five-word context: if $C[\textit{every chair sank}] > 0$ and $C[\textit{sank several inches}] > 0$ in the training text, then we consider word *sank* to have a five-word context. Note that we

are not implying that the sequence *every chair sank several inches* occurs in the training text nor that the sequence *chair sank several* occurs in the training text. We are only saying that both the sequences *every chair sank* and *sank several inches* are observed in the training text.

- Four-word context: if $C[\textit{every chair sank}]>0$ and $C[\textit{sank several}]>0$, or $C[\textit{chair sank}]>0$ and $C[\textit{sank several inches}]>0$ in the training text, then the word *sank* has a four-word context.
- Three-word context: if $C[\textit{every chair sank}]>0$ and $C[\textit{sank several}]=0$, or $C[\textit{chair sank}]=0$ and $C[\textit{sank several inches}]>0$, or $C[\textit{chair sank}]>0$ and $C[\textit{sank several}]>0$ in the training text, then *sank* has a three-word context.
- Two-word context: if $C[\textit{chair sank}]>0$ or $C[\textit{sank several}]>0$ (but not both) in the training text, then *sank* has a two-word context.
- One-word context: if $C[\textit{chair sank}]=0$ and $C[\textit{sank several}]=0$, but $C[\textit{sank}]>0$ in the training text, then *sank* has a one-word context.
- Zero-word context: if $C[\textit{sank}]=0$ in the training text, then *sank* has a zero-word context.

In the following text, when we refer to a test word having three-word context, we mean that for this word the conditions outlined are satisfied for the three-word context above, but not for any higher contexts (four-word or five-word contexts). In Table I we have tabulated how effective the language model is in correcting acoustic recognition errors depending on these contexts. In compiling this table we only consider words where the word hypothesis list includes the correct word, since only these words can be corrected by the language model. The search algorithm commits 3·4% search errors; therefore, the acoustic and language model recognition accuracies in Table I have been compiled from 96·6% of the test words.[3] The effectiveness of the language model is found to be directly related to the context coverage observed for that word. For example, over 93% of the acoustic recognition errors are corrected for words having five-word context. Most of the remaining errors for these words are due to very low acoustic recognition likelihoods.

The language model is able to correct acoustic recognition errors even for words for which only two-word context is found. That is, the language model is effective even when only one of the bigrams (with the word on the left or right of it) has a non-zero count in the training text. One example of a word from our test set having two-word context is *this* in the sequence *have weighed this thought with*. The 60-million word training text has $C[\textit{weighed this}]=0$, $C[\textit{this thought with}]=0$, and $C[\textit{this thought}]>0$. Another example is the word *wives* in the sequence *men without wives whereas the*. In this case, the training set has $C[\textit{without wives}]=0$, $C[\textit{wives whereas the}]=0$, but $C[\textit{wives whereas}]>0$.

The language model increases the recognition errors for words with neither the left nor the right context in the training text. In such cases, the language model does not have enough contextual information. The conditional probability is estimated as the average over all possible preceding words, as is evident from Equations (8d) and (8e). Luckily, only 5% of the words fall in this category. Some examples of such words are *energy* and *wasters* in the sequence *most profligate energy wasters on*. Neither the bigram *profligate*

[3] Search errors, however, are included in compiling statistics in Table II. Therefore, the recognition accuracies in Table II are lower than those in Table I. The recognition rate of the acoustic recognizer is given by the per cent of words for which the top word choice is correct. Homophone confusions are not counted as errors. For recognition rates using the language model, orthographic differences are counted as errors.

TABLE I. Effectiveness of the trigram language model to correct words in the test set depending on contexts occurring in the training text corpus

Context	Number of words	Acoustic recognition (%)	Recognition with language model (%)	Percentage of acoustic errors corrected
Five-word	2000	88·3	99·3	93·6
Four-word	2156	85·3	97·9	85·8
Three-word	1646	82·0	93·1	61·6
Two-word	840	81·2	85·1	20·9
One-word	306	80·4	71·6	−45·0
Zero-word	47	87·2	59·6	−216·0
Search errors	245			
Average		84·7	94·2	62·3

energy nor *energy wasters* occurs in the training text. The word *energy* does occur in the training text. The word *wasters* does not occur even once in the 60-million word training text. Some other examples of words in the test set which do not occur in the training text are *barnburner*, *gondolier*, *heuristics*, *amoebas*, *sashaying*, *luminol*, *Marley*, *doorless*, etc.

The information in Table I cannot be exploited during the recognition process since the table is generated using the identity of the spoken words. Does the language model in fact increase the error rate for words for which no more than one-word context is found based on their recognized identities? To answer this, Table II reformulates the information in Table I based on the context found for words as recognized instead of as spoken. As can be seen from Table II, the error rate does not increase for words with only one-word context or less. In fact, the error rate goes down marginally for such words.

Some interesting results are evident in Table II. Recognized words with more restricted contexts in the training text corpus are less likely to be correctly recognized by

TABLE II. Acoustic and language model recognition rate depending on the contexts of the recognized words as observed at the output of the language model processor

Context	Number of words	Acoustic recognition (%)	Recognition with language model (%)	Search errors (%)
Five-word	2215	87·7	97·5	0·5
Four-word	2366	83·4	94·5	1·3
Three-word	1581	78·0	87·7	3·7
Two-word	753	75·3	79·2	8·8
One-word	278	65·8	66·9	21·6
Zero-word	47	51·0	51·0	38·3
Overall	7240	81·8	91·0	3·4

the acoustic recognition module. In fact, words having zero-word context have only a 51% recognition accuracy. The reason is that words like *the* which occur frequently in the training set (for acoustic recognition) are recognized with much higher recognition accuracy than words like *wasters* which do not occur in the training set. Only 30% of the words in the test set also occur at least once in the training set. A higher percentage of these words correspond to words with five-, four- or three-word contexts than with zero or one-word contexts. For the same reason, the probability of search error during acoustic recognition increases with decreasing context. There is a 38% probability of search error for words with zero-word context (words which do not occur in the language model training text corpus). Rare words are less likely to be correctly recognized by the acoustic recognizer, and the language model is also less likely to correct these errors because of the poor context coverage.

Since the training text corpus does not have enough contextual information for words having less than three-word context, some alternate strategies could be explored for improving the recognition accuracy for these words. One strategy could be to use parts-of-speech contexts instead of word contexts. Another possible strategy is to use a parser to identify words which result in parsing errors (O'Shaughnessy, 1989). These erroneous words could then be replaced by alternate hypotheses using the language model.

8. Conclusions

In conclusion, we have shown that a trigram language model can be effectively applied to reduce acoustic recognition errors in an 86 000-word recognition task. The average reduction in the error rate is 62%. We have derived a simple algorithm (using Turing's formula) for training the conditional probabilities which involve trigrams with zero counts in the training text. This algorithm allows easy estimation of the language model parameters from a very large text corpus.

We have also shown that the language model is effective in reducing acoustic recognition errors for words in the test set which have at least the left or the right word context occurring in the training text corpus. The language model provides dramatic improvement for words having more than two-word contexts. To improve the recognition accuracy of the language model for words with two-word or fewer contexts, a language model based on parts of speech is suggested. Another possibility is the use of a parser to identify incorrect word sequences (O'Shaughnessy, 1989). This parser can identify 30% of the language errors in the form of incorrect noun phrases, unmatched verbs, loose prepositional phrases, etc. By disallowing such erroneous words or sequences, we may be able to improve the recognition accuracy of the language model.

Our speech recognition task is from a general discourse domain, consisting of articles chosen from a variety of sources. Improved performance may be achievable with a language model trained from a much larger training text corpus collected from many different sources.

We would like to thank InfoGlobe for providing the *Globe and Mail* database, and InfoMart for providing the *Gazette* database. This work was supported by the Natural Sciences and Engineering Research Council of Canada.

References

Averbuch, A., Bahl, L., Bakis, R., Brown, P., Daggett, G., Das, S., Davies, K., Gennaro, S. De, Souza, P. de, Epstein, E., Fraleigh, D., Jelinek, F., Lewis, B., Mercer, R., Moorhead, J., Nádas, A., Nahamoo, D., Picheny, M., Shichman, G., Spinelli, P., Compernolle, D. Van & Wilkens, H. (1987). Experiments with the Tangora 20,000 word speech recognizer. In *Proceedings of the 1987 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Dallas, pp. 701–704.

Bahl, L. R., Bakis, R., Jelinek, F. & Mercer, R. L. (1980). Language-model/acoustic-channel-model balance mechanism. *IBM Technical Disclosure Bulletin*, **23**, 3464–3465.

Bahl, L. R., Jelinek, F. & Mercer, R. L. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI 5**, 179–190.

Derouault, A.-M. & Merialdo, B. (1986). Natural language modeling for phoneme-to-text transcription. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI 8**, 742–749.

Francis, W. N. & Kucera, H. (1979). Manual of Information to Accompany a Standard Sample of Present-day Edited American English, For Use With Digital Computers. Department of Linguistics, Brown University.

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, **40**, 237–264.

Gupta, V., Lennig, M. & Mermelstein, P. (1988). Fast search strategy in a large vocabulary word recognizer. *Journal of the Acoustical Society of America*, **84**, 2007–2017.

Rohlicek, J. R., Chow, Y.-L. & Roucos, S. (1988). Statistical language modeling using a small corpus from an application domain. In *Proceedings of the 1988 IEEE International Conference on Acoustics, Speech and Signal Processing*. New York, pp. 267–270.

Jelinek, F. (1976). Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, **64**, 532–556.

Jelinek, F. (1985). The development of an experimental discrete dictation recognizer. *Proceedings of the IEEE*, **73**, 1616–1624.

Katz, S. M. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions Acoustics, Speech, and Signal Processing*, **ASSP 35**, 400–401.

Lee, K.-F. (1988). Large-vocabulary speaker-independent continuous speech recognition: The SPHINX System. Ph.D. Thesis. Computer Science Department, Carnegie Mellon University.

G. & C. Merriam Co. (1965). *Webster's Seventh New Collegiate Dictionary*.

Nádas, A. (1985). On Turing's formula for word probabilities. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **ASSP 33**, 1414–1416.

Nilsson, N. J. (1980). *Principles of Artificial Intelligence*. Tioga Publishing Co., Palo Alto, California.

O'Shaughnessy, D. (1989). Using syntactic information to improve large-vocabulary word recognition. In *Proceedings of the 1989 IEEE International Conference on Acoustics, Speech and Signal Processing*. Glasgow, pp. 715–718.

Seitz, F., Gupta, V., Lennig, M. & Mermelstein, P. (1988). Designing a dictionary for a very large vocabulary word recognition system. In *Proceedings of NWAVE-XVII*. Montreal, pp. 48–49.

Seitz, F., Gupta, V., Lennig, M., Kenny, P., Deng, L. & Mermelstein, P. (1990). A dictionary for a very large vocabulary word recognition system. *Computer Speech and Language*, **4**, 193–202.

Appendix

Some examples of texts used as test sentences are as follows. They are shown here as they were spoken.

Text 1

the home was as sensually comfortable as the human womb supposedly is every chair sank several inches at the lightest touch foam and down surrendering abjectly to any pressure the tufts of the acrylic nylon carpets tickled the ankles of anyone kind enough to walk on them beside the bar what looked like a radio dial would upon being turned make the lighting throughout the house as mellow or as bright as the mood demanded located throughout the house within easy walking distance of one another were contour chairs a massage table and a motorized exercising board whose many sections prodded the body with a motion that was at once gentle yet suggestive

Text 2

his future assignments will involve the research of new algorithms and heuristics to model and solve new networking problems introduced by integrated network technologies his strong background in mathematics is sure to be an asset to the department

Text 3

doctors say the most disturbing trend they have noticed in the past few months is the increase in the number of heart attack victims and other seriously ill patients who turn up in taxis or on foot instead of in ambulances where they belong that's because until last month hospitals were turning away ambulances when their wards became too full except in critical cases so patients were unable to choose their hospitals

Text 4

the institute noted that conventional wisdom which ignores the potential for energy conservation holds that global energy use will triple in forty years in north America where forests and lakes are already threatened by acid rain coal burning may quadruple by twenty twenty five this would vastly increase sulphur dioxide emissions the major source of acid rain unless energy conservation measures are implemented in other words the world will pay an enormous economic and environmental price if it doesn't wake up the problem is not that energy efficiency technologies do not exist rather they receive virtually no governmental support enormous potential exists for energy conservation particularly in Canada where we have the dubious distinction of being the world's most profligate energy wasters on a per capita basis