ADAPTIVE LANGUAGE MODELING USING MINIMUM DISCRIMINANT ESTIMATION*

S. Della Pietra, V. Della Pietra, R. L. Mercer, S. Roukos
Continuous Speech Recognition Group,
Thomas J. Watson Research Center
P. O. Box 704, Yorktown Heights, NY 10598

ABSTRACT

We present an algorithm to adapt a *n-gram* language model to a document as it is dictated. The observed partial document is used to estimate a unigram distribution for the words that already occurred. Then, we find the closest *n-gram* distribution to the static *n-gram* distribution (using the discrimination information distance measure) and that satisfies the marginal constraints derived from the document. The resulting minimum discrimination information model results in a perplexity of 208 instead of 290 for the static trigram model on a document of 321 words.

1 INTRODUCTION

Statistical *n-gram* language models are useful for speech recognition and language translation systems because they provide an *a-priori* probability of a word sequence; these language models improve the accuracy of recognition or translation by a significant amount. In the case of trigram ($n = 3$) and bigram ($n = 2$) language models, the probability of the next word conditioned on the previous words is estimated from a large corpus of text. The resulting static language models (SLM) have fixed probabilities that are independent of the document being predicted.

To improve the language model (LM), one can adapt the probabilities of the language model to match the current document more closely. The partially dictated (in the case of speech recognition) document provides significant clues about what words are more likely to be used next. One expects many words to be ***bursty***. For example, if in the early part of a document the word *fax* has been used then the probability that it will be used again in the same document is significantly higher than if it had not occurred yet. In addition, if the words in the early part of a document suggest that the current document is from a particular subdomain, then we may expect other related words to occur at a higher rate than the static model may suggest. For example, the words "***inspection, fire, and insurance***" suggest an insurance report domain, and therefore increased probabilities to words such as "***stairwell and electrical***".

Assume that from a partial document, denoted by h, we have an estimate of the unigram distribution $p_d(w \mid h)$ that a word w may be used in the remaining part of the document. We will denote $p_d(w \mid h)$ by $d(w)$, keeping in mind that this *dynamic* unigram distribution is continuously updated as the document is dictated and is estimated for a subset of R words of the vocabulary. (Typically R is on the order of a document size of about a few hundred words as compared to a vocabulary size of 20,000 words.) In general, the dynamic unigram distribution will be different from the static marginal unigram distribution, denoted by $p_s(w)$. In this paper, we propose a method for adapting the language model so that its marginal unigram distribution matches the desired dynamic unigram distribution $d(w)$.

The proposed approach consists of finding the model that requires the least pertubation of the static model and satisfies the set of constraints that have been derived from the partially observed document. By ***least pertubation*** we mean that the new model is closest to the static model, p_s, using the non-symmetric Kullback-Liebler distortion measure (also known as discrimination information, relative entropy, etc.). The minimum discrimination information (MDI) p^* distribution minimizes:

$$D(p, p_s) = \sum_i p(i) \log \frac{p(i)}{p_s(i)}$$

over all p that satisfy a set of R linear constraints. In this paper, we consider marginal constraints of the

*PAPER SUBMITTED TO THE ICASSP92 PROCEEDINGS.

form

$$\sum_{i \in C_r} p(i) = d_r$$

where we are summing over all events i in the set C_r that correspond to the r-th constraint and d_r is the desired value (for $r = 1, 2, ..., R$). In our case, the events i correspond to bigrams, (w_1, w_2), and the desired value for the r-th constraint, d_r, is the marginal unigram probability, $d(w_r)$, for a word w_r.

The idea of using a window of the previous N words, called a cache, to estimate dynamic frequencies for a word was proposed in [5] for the case of a tri-part-of-speech model and in [6] for a bigram model. In [4] a trigram language was estimated from the cache and interpolated with the static trigram model to yield about 20% lower perplexity and from 5% to 25% lower recognition error rate on documents ranging in length from 100 to 800 words.

2 MINIMUM DISCRIMINATION INFORMATION

The discrimination information can be written as:

$$\begin{aligned} D(p, p_s) &= -\sum p \log p_s + \sum p \log p & (1) \\ &= R_{p_s}(p) - H(p) \geq 0 & (2) \end{aligned}$$

where $R_{p_S}(p)$ is the bit rate in transmitting source p with model p_s and $H(p)$ is the entropy of source p. The MDI distribution p^* satisfies the following *Pythagorean inequality*:

$$D(p, p_s) \geq D(p, p^*) + D(p^*, p_s)$$

for all distributions p in the set P_R of distributions that satisfy the R constraints. So if we have an accurate estimate of the constraints then using the MDI distribution will result in a lower error by at least $D(p^*, p_s)$.

The MDI distribution is the Maximum Entropy (ME) distribution if the static model is the uniform distribution.

Using Lagrange multipliers and differentiating with respect to p_i the probability of the i-th event, we find that the optimum must have the form

$$p_i^* = p_{si} f_{i1} f_{i2} \cdots f_{iR}$$

where the factors f_{ir} are 1 if event i is not in the constraint set C_r or some other value f_r if event i belongs to constraint set C_r. So the MDI distribution is specified by the R factors f_r, $r = 1, 2, ..., R$, that correspond to the R constraints, in addition to the original static model.

3 ALTERNATING MINIMIZATION

Starting with an initial estimate of the factors, the following iterative algorithm is guaranteed to converge to the optimum distribution. At each iteration j, pick a constraint r_j and adjust the corresponding factor so that the constraint is satisfied. In the case of marginal constraints, the update is:

$$f_{r_j}^{new} = f_{r_j}^{old} \frac{d_{r_j}}{p^{j-1}(C_{r_j})}$$

where $p^{j-1}(C_{r_j})$ is the marginal of the previous estimate and d_{r_j} is the desired marginal. This iterative algorithm cycles through the constraints repeatadly until convergence hence the name alternating (thru the constraints) minimization. It was proposed by Darroch and Ratcliff in [3]. A proof of convergence for linear constraints is given in [2] .

4 ME CACHE

We have applied the above approach to adapting a bigram model; we call the resulting model the ME cache. Using a cache window of the previous N words, we estimate the desired unigram probability of all R words that have occurred in the cache by:

$$d(w) = \lambda_c f_c(w)$$

where λ_c is an adjustment factor taken to be the probability that the next word is already in the cache and f_c is the observed frequency of a word in the cache. Since any event (w_1, w_2) participates in 2 constraints one for the left marginal $d(w_1)$ and the other for the right marginal $d(w_2)$ there are $2R + 1$ constraint, a left and right marginal for each word in the cache and the overall normalization, the ME bigram cache model is given by:

$$p_{me}(w_1, w_2) = \alpha_l(w_1)\alpha_r(w_2)p_s(w_1, w_2)$$

We require the left and right marginals to be equal to get a stationary model. (Since all events participate in the normalization that factor is absorbed in the other two.)

The iterations fall into two groups: those in which a left marginal is adjusted and those in which a right marginal is adjusted. In each of these iterations, we adjust two factors simultaneously: one for the desired unigram probability d(w) and the other so that the resulting ME model is a normalized distribution. The update for left marginals is

$$p^j(w_1, w_2) = p^{j-1}(w_1, w_2)a_j s_j$$

where a_j and s_j are adjustments given by:

$$s_j = \frac{1 - d(w_j)}{1 - p^{j-1}(w_j, .)}$$

$$a_j = \frac{d(w_j)}{s_j p^{j-1}(w_j, .)}$$

where $p^{j-1}(w_j, .)$ denotes the left marginal of the $(j-1)$-th estimate of the ME distribution and w_j is the word that corresponds to the selected constraint at the j-th iteration. Similar equations can be derived for the updates for the right marginals. The process is started with $p_0(w_1, w_2) = p_s(w_1, w_2)$.

Note that the marginal $p^j(w, .)$ can be computed by using R additions and multiplications. The algorithm requires order R^2 operation to cycle thru all constraints once. R is typically few hundred compared to the vocabulary size V which is 20,000 in our case. We have found that about 3 to 5 iterations are sufficient to achieve convergence.

5 EXPERIMENTAL RESULTS

Using a cache window size of about 700 words, we estimated a desired unigram distribution and a corresponding ME bigram distribution with an MDI of about 2.2 bits (or 1.1 bits/word). Since the unigram distribution may not be exact, we do not expect to reduce our perplexity on the next sentence by a factor larger than $2.1 = 2^{1.1}$. The actual reduction was a factor of $1.5 = 2^{0.62}$ on the next 93 words of the document. For a smaller cache size the discrepancy between the MDI and actual perplexity reduction is larger.

To evaluate the ME cache model we compared it to the trigram cache model and the static trigram model. In all models we use linear interpolation between the dynamic and static components as:

$$p(w_3|w_1, w_2) = \lambda_c p_c(w_3|w_1, w_2) + (1 - \lambda_c) p_s(w_3|w_1, w_2)$$

where $\lambda_c = 0.2$. The static and cache trigram probabilities use the usual interpolation between unigram, bigram, and trigram frequencies [1]. The cache trigram probability p_c is given by:

$$p_c(w_3|w_1, w_2) = \lambda_1 f_{c1}(w_3) + \lambda_2 f_{c2}(w_3|w_2) + \lambda_3 f_{c3}(w_3|w_1, w_2)$$

where f_{ci} are frequencies estimated from the cache window. The interpolating weights are $\lambda_1 = 0.4, \lambda_2 = 0.5$, and $\lambda_3 = 0.1$. For the ME cache we replace the dynamic unigram frequency $f_{c1}(w_3)$ by the ME conditional bigram probability $p_{me}(w_3|w_2)$ given by:

$$p_{me}(w_3|w_2) = \frac{\alpha_r(w_3) p_s(w_3|w_2)}{\sum_w \alpha_r(w) p_s(w|w_2)}$$

Note that the sum in the denominator is order R since the factors are unity for the words that are not in the cache.

In Table 1, we compare the static, the ME cache, and the trigram cache models on three documents. Both cache models improve on the static. The ME and trigram cache are fairly close as would be expected since they both have the same dynamic unigram distribution. The second experiment illustrates how they are different.

Document	Words	Static	ME Cache	Trigram Cache
T1	321	290	208	218
T3	426	434	291	300
E1	814	294	175	182

Table 1. Perplexity on three documents.

We compared the ME cache and the trigram cache on 2 non-sensical sentences made up from words that have occurred in the first sentence of a document. The 2 sentences are:

- S1: the letter fire to to to
- S2: building building building building

Table 2 shows the perplexity of each sentence at 2 points in the document history: one after the first sentence (of length 33 words) is in the cache and the second after 10 sentences (203 words) are in the cache. We can see that the trigram cache can make some rare bigrams (w_1, w_2) more likely if both w_1 and w_2 have already occurred due to a term of the form $d(w_1)d(w_2)$ whereas the ME cache still has the factor $p_s(w_1, w_2)$ which will tend to keep a rare bigram somewhat less probable. This is particularly pronounced for $S2$, where we expect $d(building)$ to be quite accurate after 10 sentences, the ME cache penalizes the unlikely bigram by a factor of about 13 over the trigram cache.

Sentence	Cache Size	Trigram Cache	ME Cache
S1	33	213	268
S1	203	417	672
S2	33	245	665
S2	203	212	2963

Table 2. Trigram and ME cache perplexity.

6 CONCLUSION

The MDI approach to adapting a language model can result in significant perplexity reduction without a leakage in the bigram probability model. We expect this fact to be important in adapting to a new domain where the unigram distribution $d(w)$ can be estimated from possibly tens of documents. We are currently pursuing such experiments.

REFERENCES

[1] Bahl, L., Jelinek, F., and Mercer, R.,*A Statistical Approach to Continuous Speech Recognition*, IEEE Trans. on PAMI, 1983.

[2] Csiszar, I., and Longo, G., *Information Geometry and Alternating Minimization Procedures*, Statistics and Decisions, Supplement Issue 1:205-237, 1984.

[3] Darroch, J.N., Ratcliff, D. *Generalized Iterative Scaling for Log-Linear Models*, The Annals of Mathematical Statistics, Vol. 43, pp. 1470-1480, 1972.

[4] Jelinek, F., Merialdo, B., Roukos, S., and Strauss, M., *A Dynamic Language Model for Speech Recognition*, Proceedings of Speech and Natural Language DARPA Workshop, pp. 293-295, Feb. 1991.

[5] Kuhn, R., *Speech Recognition and the Frequency of Recently Used Words: a Modified Markov Model for Natural Language*, Proceedings of COLING Budapest, Vol. 1, pp. 348-350, 1988. Vol. 1 July 1988

[6] Kupiec, J., *Probabilistic Models of Short and Long Distance Word Dependencies in Running Text*, Proceedings of Speech and Natural Language DARPA Workshop, pp. 290-295, Feb. 1989.