A Dynamic Language Model for Speech Recognition

F. Jelinek, B. Merialdo, S. Roukos, and M. Strauss[1]

IBM Research Division, Thomas J. Watson Research Center,
Yorktown Heights, NY 10598

ABSTRACT

In the case of a trigram language model, the probability of the next word conditioned on the previous two words is estimated from a large corpus of text. The resulting static trigram language model (STLM) has fixed probabilities that are independent of the document being dictated. To improve the language model (LM), one can adapt the probabilities of the trigram language model to match the current document more closely. The partially dictated document provides significant clues about what words are more likely to be used next. Of many methods that can be used to adapt the LM, we describe in this paper a simple model based on the trigram frequencies estimated from the partially dictated document. We call this model a cache trigram language model (CTLM) since we are caching the recent history of words. We have found that the CTLM reduces the perplexity of a dictated document by 23%. The error rate of a 20,000-word isolated word recognizer decreases by about 5% at the beginning of a document and by about 24% after a few hundred words.

INTRODUCTION

A language model is used in speech recognition systems and automatic translation systems to improve the performance of such systems. A trigram language model [1], whose parameters are estimated from a large corpus of text (greater than a few million words), has been used successfully in both applications. The trigram language model has a probability distribution for the next word conditioned on the previous two words. This static distribution is obtained as an average over many documents. Yet we know that several words are bursty by nature, i.e., one expects the word "language" to occur in this paper at a significantly higher rate than the average frequency estimated from a large collection of text. To capture the "dynamic" nature of the trigram probabilities in a particular document, we present a "cache" trigram language model (CTLM) that uses a window of the n most recent words to determine the probability distribution of the next word.

The idea of using a window of the recent history to adjust the LM probabilities was proposed in [2, 3]. In [2] the dynamic component adjusted the conditional probability, $p_n(w_{n+1} \mid g_{n+1})$, of the next word, w_{n+1}, given a predicted part-of-speech (POS), g_{n+1}, in a tripart-of-speech language model. Each POS had a separate cache where the frequencies of all the words that occurred with a POS determine the dynamic component of the language model. As a word is observed it is tagged and the appropriate POS cache is updated. At least 5 occurrences per cache are required before activating it. Preliminary results for a couple POS caches indicate that with appropriate smoothing the perplexity, for example, on NN-words is decreased by a factor of 2.5 with a cache-based conditional word probability given the POS category instead of a static probability model.

In [3], the dynamic model uses two bigram language models, $p_n(w_{n+1} \mid w_n, D)$, where D=1 for words w_{n+1} that have occurred in the cache window and D=0 for words that have occurred in the cache window. For cache sizes from 128 to 4096, the reported results indicate an improvement in the average rank of the correct word predicted by the model by 7% to

[1]Now at Rutgers University, NJ, work performed while visiting IBM

Test Set	static Perplexity	dynamic Perplexity	λ_c
A	91	75	0.07
B	53	49	0.12
C	262	202	0.07

Table 1: Perplexity of static and dynamic language models.

Cache Size	0	200	1000
Perplexity	262	217	202

Table 2: Perplexity as a function of cache size on test set C.

Static	Unigram Cache	Trigram Cache
262	230	202

Table 3: Perplexity of unigram and trigram caches.

17% over the static model assuming one knows if the next word is in the cache or not.

In this paper, we will present a new cache language model and compare its performance to a trigram language model. In Section 2, we present our proposed dynamic component and some results comparing static and dynamic trigram language models using perplexity. In Section 3, we present our method for incorporating the dynamic language model in an isolated 20,000 word speech recognizer and its effect on recognition performance.

CACHE LANGUAGE MODEL

Using a window of the n most recent words, we can estimate a unigram frequency distribution $f_n(w_{n+1})$, a bigram frequency distribution, $f_n(w_{n+1} \mid w_n)$, and a trigram frequency distribution, $f_n(w_{n+1} \mid w_n, w_{n-1})$. The resulting 3 dynamic estimators are linearly smoothed together to obtain a dynamic trigram model denoted by $p_{cn}(w_{n+1} \mid w_n, w_{n-1})$. The dynamic trigram model assigns a non-zero probability for the words that have occurred in the window of the previous n words. Since the next word may not be in the cache and since the cache contains very few trigrams, we interpolate linearly the dynamic model with the the static trigram language model:

$$p_n(w_{n+1} \mid w_n, w_{n-1}) = \quad (1)$$
$$\lambda_c p_{cn}(w_{n+1} \mid w_n, w_{n-1}) +$$
$$(1 - \lambda_c) p_s(w_{n+1} \mid w_n, w_{n-1})$$

where $p_s(\cdots)$ is the usual static trigram language model. We use the forward-backward algorithm to estimate the interpolation parameter λ_c [1]. This parameter varies between 0.07 and 0.28 depending on the particular static trigram language model (we used trigram language models estimated from different size corpora) and the cache size (varying from 200 to 1000 words.)

We have evaluated this cache language model by computing the perplexity on three test sets:

- Test sets A and B are each about 100k words of text that were excised from a corpus of documents from an insurance company that was used for building the static trigram language model for a 20,000-word vocabulary.
- Test set C which consists of 7 documents (about 4000 words) that were dictated in a field trial in the same insurance company on TANGORA (the 20,000-word isolated word recognizer developed at IBM.)

Table 1 shows the perplexity of the static and dynamic language models for the three test sets. The cache size was 1000 words and was updated word synchronously. The static language model was estimated from about 1.2 million words of insurance documents. The dynamic language model yields from 8% to 23% reduction in perplexity, with the larger reduction occurring with the test sets with larger perplexity. The interpolation weight λ_c was estimated using set B when testing on sets A and C and set A when testing on set B. Table 2 shows the effect of cache size on perplexity where it appears that a larger cache is more useful. These results were on test set C. On test set C, the rate that the next word is in the cache ranges from 75% for a cache window of 200 words to 83% for a window of 1000. Table 3 compares a cache with unigrams only with a full trigram cache (for the trigram cache, the weights for the unigram, bigram, and trigram frequencies were 0.25, 0.25, 0.5 respectively and were selected by hand.) A second set of weights (0.25,0.5,0.25) produced a perplexity of 190 for the trigram cache. In all the above experiments, the cache was not flushed between documents. In the next section, we compare the different models in an isolated speech recognition experiment.

We have tried using a fancier interpolation scheme where the reliance on the cache depends on the cur-

Text Length	0-100	100-200	200-300	300-400	400-500	500-800
% Reduction in Error Rate	6.1%	5.3%	4.7%	10.5%	16.3%	23.8%

Table 4: Percentage reduction in error rate with trigram cache.

rent word w_n with the expectation that some words will tend to be followed by bursty words whereas other words will tend to be followed by non-bursty words. We typically used about 50 buckets (or weighting parameters). However, we have found that the perplexity on independent data to be no better than the single parameter interpolation.

ISOLATED SPEECH RECOGNITION

We incorporated the cache language model into the TANGORA isolated speech recognition system. We evaluated two cache update strategies. In the first one, the cache is updated at the end of every utterance, i.e., when the speaker turns off the microphone. An utterance may be a partial sentence or a complete sentence or several sentences depending on how the speaker dictated the document. In the second strategy, the cache is updated as soon as the recognizer makes a decision about what was spoken. This typically corresponds to a delay of about 3 words. The cache is updated with the correct text which requires that the speaker correct any errors that may occur. This may be unduly difficult with the second update strategy. But in the context of our experiments, we have found that using the simpler (and more realistic) update strategy, i.e., after an utterance is completed, to be as effective as the more elaborate update strategy.

The TANGORA system uses a 20,000-word office correspondence vocabulary with a trigram language model estimated from a few hundred million words from several sources. The cache language model was tested on a set of 14 documents dictated by 5 speakers with an internal telephone system (private branch exchange.) The speakers were form the speech group typically dictating electronic mail messages or internal memoranda. The size of a document ranged from about 120 words to 800 words. The total test corpus was about 5000 words. The maximum cache size (4000 words) was larger than any of the documents. In these tests, the cache is flushed at the beginning of each document.

In these experiments, the weights for interpolating the dynamic unigram, bigram, and trigram frequencies were 0.4, 0.5, and 0.1, respectively. The weight of the cache probability, λ_c, relative to the static trigram probability was 0.2. Small changes in this weight does not seem to affect recognition performance. The potential benefit of a cache depends on the amount of text that has been observed. Table 4 shows the percentage reduction in error rate as a function of the length of the observed text. We divided the documents into 100-word bins and computed the error rate in each bin. For the static language model, the error rate should be constant except for statistical fluctuations, whereas one expects that the error rate of the cache to decrease with longer documents. As can be seen from Table 4, the cache reduces the error rate by about 5% for shorter documents and up to 24% for longer documents. The trigram cache results in an average reduction in error rate of 10% for these documents whose average size is about 360 words. The trigram cache is very slightly better than a unigram cache eventhough the earlier results using perplexity as a measure of performance indicated a bigger difference between the two caches.

REFERENCES

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

[2] 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

[3] 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.