MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Comparison of Width-wise and Length-wise Language Model Compression

Edward Whittaker, Bhiksha Raj

TR-2001-42 December 2001

Abstract

In this paper we investigate the extent to which Katz back-off language models can be compressed through a combination of parameter quantization (width-wise compression) and parameter pruning (length-wise compression) methods while preserving performance. We compare the compression and performance that is achieved using entropy-based pruning against that achieved using only parameter quantization. We then compare combinations of both methods. It is shown that a broadcast news language model can be compressed by up to 83% to only 12.6Mb with no loss in performance on a broadcast news task. Compressing the language model further by quantization to 10.3Mb resulted in only a 0.4% degradation in word error rate which is better than can be achieved through entropy-based pruning alone.

In Eurospeech'2001

Publication History:–

1. First printing, TR-2001-42, December 2001

Comparison of Width-wise and Length-wise Language Model Compression

E. W. D. Whittaker[1] *and B. Raj*[2]

1. Compaq Cambridge Research Laboratory
Cambridge, MA 02142 USA
2. Mitsubishi Electric Research Laboartories
Cambridge, MA 02139 USA

Abstract

In this paper we investigate the extent to which Katz back-off language models can be compressed through a combination of parameter quantization (width-wise compression) and parameter pruning (length-wise compression) methods while preserving performance. We compare the compression and performance that is achieved using entropy-based pruning against that achieved using only parameter quantization. We then compare combinations of both methods. It is shown that a broadcast news language model can be compressed by up to 83% to only 12.6Mb with no loss in performance on a broadcast news task. Compressing the language model further by quantization to 10.3Mb resulted in only a 0.4% degradation in word error rate which is better than can be achieved through entropy-based pruning alone.

1. Introduction

In this paper we examine several techniques for compressing language models including quantizing the values of parameters in the language model and pruning parameters from the language model. In particular we examine the interaction between both quantizing and pruning parameters. The main aim of this investigation is to determine how best to compress the size of language models in memory while minimising the degradation in language model performance. There are several compelling reasons for addressing this issue. The main reason is that the language model is in general by far the largest component of a speech recognition system. From desktop dictation applications to incorporating speech on hand-held PCs, memory limits the size of the language model that can be used and severely restricts the performance of the speech recognition system.

Conventionally, language models are compressed by reducing the length of the lists of explicitly stored N-gram probability events in the language model. The most efficient manner of reducing the size of these lists is by eliminating all those elements in the list whose contribution to the language model entropy lies below some threshold [1]. We refer to this method of language model compression as length-wise compression.

In [2] we describe two language model compression methods that achieve reduction in language model size by both width-wise and length-wise compression of the lists of language model probabilities and back-off weights. It is shown that broadcast news language models can be compressed by up to 60% of their original size with no significant increase in word error rate on a broadcast news task. This is achieved through a combination of quantization and parameter pruning. Both methods are shown to provide an effective means of compressing language model parameters while minimising the degradation in recognition performance.

In this paper we evaluate the performance of the entropy-based parameter pruning technique [1] and compare the compression and recognition performance achieved using this method against that achieved with the parameter quantization methods. In addition, we investigate combining both methods, first to reduce the initial size of the language model using the entropy-based technique and then to quantize the remaining parameters. The performance of these models is compared against ones of equal size pruned using only the entropy-based technique.

In Section 2 we briefly outline the storage requirements for the different elements of a language model. In Section 3 we describe width-wise compression techniques. Length-wise compression techniques are described in Section 4. In Section 5 we present the word error rate results of recognition experiments on a broadcast news task and evaluate the quantization and pruning techniques both individually and combined. These results permit the optimal combination of pruning and quantization to be determined for a desired language model compression and performance.

2. Language model memory requirements

Conventional methods of storing Katz back-off trigram language models require 2 bytes for every explicitly stored probability and back-off weight. In the CMU SPHINX-III speech recognition system [3] used in the experiments in this paper this is achieved by truncating each parameter to 4 decimal places. In general this ensures that there are no more than 65536 unique values in each list of N-grams or back-off weights. Additional storage is required for the tree structure which is the common method for compactly storing these parameters. This overhead equates to an additional 1 byte for every unigram and bigram parameter in the model and another 2 bytes for every bigram and trigram which is used for word identifiers in the tree structure. Overall the memory requirement in bytes is: 5*$N(unigrams)$ + 7*$N(bigrams)$ + 4*$N(trigrams)$, where $N(\cdot)$ is the number of the types of events in parentheses. Language model storage requirements are explained in more detail in [2].

3. Compression by quantization

In [2] we describe two methods for width-wise compression of language model parameters using quantization which were called *absolute parameter compression* and *difference parameter compression*. In the original methods, the Lloyd-Max algorithm [4] was used to perform the quantization. This method iteratively minimises the average squared error introduced in the parameter through quantization. However other quantization methods, such as linear quantization which simply partitions the range of the parameter into equally sized segments,

may also be used.

3.1. Absolute parameter compression

All unigram, bigram and trigram log probabilities and unigram and bigram log backoff weights are quantized to a small number of quantization levels. Quantization is performed separately on each of the N-gram probability and back-off weight lists and separate quantization level look-up tables generated for each of these sets of parameters. If $Q_i^{\{P,\alpha\}}[\cdot]$ is a function that maps either a probability (P) or back-off weight (α) in the i-gram table to its quantized value, $P(\cdot)$ is the original probability of an event and $\alpha(\cdot)$ is the back-off weight of some context, then each explicit probability in the language model ($N > 0$) is mapped to a quantized probability

$$Q_N^P[P(w_i \mid w_{i-N+1}^{i-1})], \tag{1}$$

and each back-off weight is mapped to a quantized back-off weight for $N > 1$

$$Q_{N-1}^{\alpha}[\alpha(w_{i-N+1}^{i-1})]. \tag{2}$$

Compression results from the reduced number of bits needed to store the indices into the look-up tables.

3.2. Difference parameter compression

Here, for N-gram events where $N > 1$, we quantize the difference between N-gram log probabilities and their quantized log back-off estimates. Using the above definitions each quantized difference probability is determined as follows:

$$Q_N^P[P(w_i \mid w_{i-N+1}^{i-1})] - Q_{N-1}^{\alpha}[\alpha(w_{i-N+1}^{i-1})] \cdot Q_{N-1}^P[P(w_i \mid w_{i-N+2}^{i-1})]. \tag{3}$$

The stored values now represent indices to the quantized probability differences. During recognition the true probability must be composed by adding the backed-off estimate to the quantized differences. Unigram probabilities and all back-off weights are quantized as for absolute parameter compression.

Procedurally, first the unigram probabilities and back-off weights are quantized. Bigram back-off weights and the differences between the true bigram probabilities and their quantized backed-off estimates are then quantized. Finally the differences between the true trigram probabilities and their quantized backed-off estimates are quantized.

4. Compression by pruning

4.1. Pruning by quantization

Both the absolute and difference parameter compression methods described above incorporate a stage of parameter pruning. The criterion for pruning a parameter is how similar the quantized backed-off probability is to the quantized original probability. For absolute parameter compression if the quantized backed-off probability falls in the same quantization bin as the quantized original probability then the original parameter is discarded i.e. if

$$Q_3^P[P(w_i \mid w_{i-2}, w_{i-1})] = Q_3^P[Q_2^{\alpha}[\alpha(w_{i-2}, w_{i-1})] \cdot Q_2^P[P(w_i \mid w_{i-1})]], \tag{4}$$

the parameter is removed. For difference parameter compression, a zero-valued quantization level is introduced during quantization. Any parameter that is quantized into this bin is discarded.

4.2. Entropy-based parameter pruning

The operation of the entropy-based pruning of language models is described in [1]. Explicit probability estimates are removed from the language model if it is shown that doing so results in an improvement of the language model perplexity or a degradation that is deemed acceptable. For each context h, every N-gram event stored in an $(N-1)$-gram context has its explicit probability estimate tentatively replaced by the implicit (backed-off) $(N-1)$-gram estimate,

$$P'(w \mid h) = \alpha'(h) P(w \mid h'), \tag{5}$$

where h' is the last $(N-2)$ words in h.

The pruning algorithm aims to minimise the divergence between the original distribution $P(\cdot \mid \cdot)$ and the pruned distribution $P'(\cdot \mid \cdot)$. Assuming that each N-gram has an independent effect on the divergence, the relative entropy can be used to quantify this change

$$D(P \mid\mid P') = -\sum_{w_i} P(w_i, h)[\log P'(w_i \mid h) - \log P(w_i \mid h)]. \tag{6}$$

The removal of an explicit N-gram event (h, w) changes the back-off weight for that context and therefore affects the contribution from all backed-off estimates,

$$D(P \mid\mid P') = -P(h)\{P(w \mid h)[\log P'(w \mid h) - \log P(w \mid h)] + \sum_{\forall w_i : N(h,w_i)=0} P(w_i \mid h)[\log P'(w_i \mid h) - \log P(w_i \mid h)]\}. \tag{7}$$

Insertion of the backed-off estimates into the above equation removes the need for a summation over all vocabulary words and allows the relative entropy to be computed efficiently,

$$D(P \mid\mid P') = -P(h)\{P(w \mid h)[\log P(w \mid h') + \log \alpha'(h) - \log P(w \mid h)] + [\log \alpha(h') - \log \alpha(h)] \sum_{\forall w_i : N(h,w_i)=0} P(w_i \mid h)\}. \tag{8}$$

The summation in the above equation is simply the probability mass of unseen events used in computing the back-off weights. The marginal history probabilities $P(h)$ can be obtained by multiplying together the appropriate conditional probabilities $P(w_{i-N+1}) \cdot P(w_{i-N+2} \mid w_{i-N+1}) \cdots$ and the updated back-off weight $\alpha'(h)$ is obtained by omitting the contribution from the pruned N-gram when computing the back-off weight.

Since relative entropy is directly related to the intrinsic perplexity of the language model $PP = e^{-\sum_{h,w} P(h,w) \log P(w|h)}$ the change in perplexity between the original and the pruned model is given by $e^{D(P||P')} - 1$. Consequently a selection criterion can be defined so that explicit N-gram estimates are retained which, if they were to be removed, would increase the perplexity by more than some threshold value.

5. Experiments

In this section we investigate the effect of the different width-wise and length-wise language model compression schemes presented above on the 1998 DARPA HUB4 broadcast news task [5]. A trigram language model using Katz back-off together with Good-Turing discounting was built using a 65k word vocabulary and approximately 100 million words of broadcast news transcriptions and newspaper texts. In addition, all singleton bigrams and trigrams were discarded. The baseline language model required 71.9Mb in memory and gave a word error rate of 22.1%.

5.1. Compression by quantization

We investigated linear and Lloyd-Max quantization for both absolute and difference parameter compression. The results for compression using absolute and difference parameter compression are given for 2, 4 and 8-bit linear quantization in Table 1. In Table 2 we give the results using 2 and 4-bit Lloyd-Max quantization. No parameters were pruned from any model. In the tables, Q_u, Q_b, Q_t indicates that 2^{Q_u}, 2^{Q_b} and 2^{Q_t} quantization levels were used for unigram, bigram and trigram elements, respectively.

Method	Q_u, Q_b, Q_t	size (Mb)	WER%
abs	8,8,8	53.1	22.1
abs	4,4,4	43.5	22.4
abs	4,4,2	40.6	25.5
dif	8,8,8	53.1	22.1
dif	4,4,4	43.5	23.9
dif	4,4,2	40.6	40.6

Table 1: Recognition performance of language model quantized using absolute (abs) and difference (dif) parameter compression and linear quantization.

Method	Q_u, Q_b, Q_t	size (Mb)	WER%
abs	4,4,4	43.5	22.2
abs	4,4,2	40.6	23.1
dif	4,4,4	43.5	21.9
dif	4,4,2	40.6	22.8

Table 2: Recognition performance of language model quantized using absolute (abs) and difference (dif) parameter compression and Lloyd-Max quantization.

5.2. Compression by pruning

Two methods of pruning language model parameters were investigated: pruning as a result of the quantization process and explicit parameter pruning using the entropy-based method.

5.2.1. Pruning by quantization

Only trigrams were considered for removal from the language model after quantization had been performed. The word error rate, the number of trigrams discarded and the size of the resulting language model are shown in Table 3. Both the absolute and difference compression methods were used with Lloyd-Max quantization at 2 and 4 bits.

Method	Q_u, Q_b, Q_t	3-gram del.	size(Mb)	WER%
abs	4,4,4	1686294	39.3	22.2
abs	4,4,2	5260335	36.9	23.3
dif	4,4,4	1119492	40.7	22.1
dif	4,4,2	3526503	32.8	22.5

Table 3: Recognition performance of language models compressed using absolute (abs) and difference (dif) parameter compression with the Lloyd-Max algorithm, showing the number of trigrams pruned.

5.2.2. Entropy-based pruning

Entropy-based pruning was applied to the baseline language model (from which all singleton bigrams and trigrams had initially been removed) using a threshold parameter ranging from 10^{-9} to 10^{-6}. The word error rate and the size in memory of each pruned language model is shown in Table 4.

Threshold	size (Mb)	WER%
1×10^{-9}	61.6	22.1
5×10^{-9}	50.4	22.1
1×10^{-8}	42.0	21.9
5×10^{-8}	17.3	22.1
1×10^{-7}	10.5	23.0
5×10^{-7}	3.29	25.0
1×10^{-6}	2.00	25.9

Table 4: Recognition performance of language model compressed using entropy-based pruning with 16-bit parameters.

5.3. Compression by quantization and pruning

The parameters in each of the language models obtained using entropy-based pruning with different threshold values were quantized using different numbers of quantization levels. The word error rate obtained for each language model is plotted in Figure 1 against the memory requirement of the model. The memory requirement takes into account the number of remaining parameters after pruning and the degree of quantization used.

6. Discussion

From the results in Section 5.1 it is clear that the choice of quantization method can have a large effect on the performance of the language model. Despite the simplicity of linear quantization no loss in incurred when parameters are quantized to 8 bits using this method. However, as the number of quantization levels is reduced the word error rate increases rapidly, significantly so for difference parameter compression. A comparison with the performance of models quantized using the Lloyd-Max algorithm shows the latter to be superior to linear quantization especially when fewer quantization levels are used. Nonetheless, certain problems were discovered with the Lloyd-Max algorithm. For example, the greedy nature of the algorithm renders it susceptible to getting stuck in local minima which are dependent on the initialisation. A crucial observation however was that there is a distinct correlation between average quantization error and word error rate. This facilitates improving the quantization without running a recognition experiment to determine the performance.

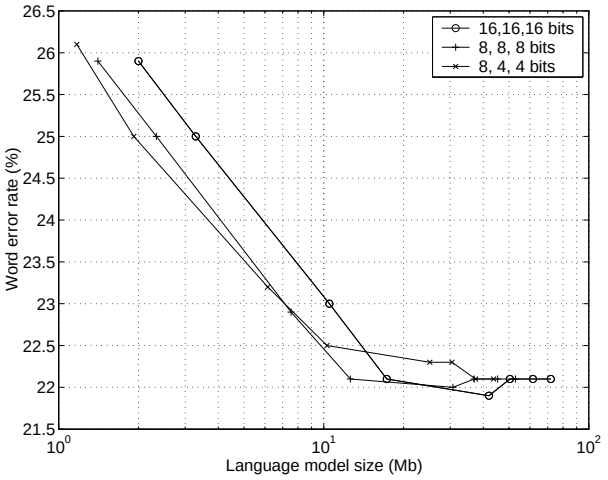

Figure 1: Word error rate against language model size.

The results in Section 5.2 show that it is preferable to prune the language model using entropy-based pruning before quantizing the remaining parameters, than to prune parameters based on quantizing the parameters in the original model. Moreover, the entropy-based pruning method can achieve a greater amount of compression in language model memory requirements over using only quantization while incurring no increase in word error rate. It is seen however that pruning the language model to a size much below 18Mb results in a rapid degradation in performance for this particular language model. When quantization is applied to the pruned language models the fewer the quantization levels used, the greater the compression but also the greater the corresponding degradation in performance. Nonetheless, the combination of pruning a language model followed by quantization of the parameters results in greater compression for a fixed performance than either method can achieve by itself. Furthermore, for a given number of parameters additional compression can be achieved by quantization with a minimal loss in performance. For example, our language model can be compressed by an additional 20% from 12.6Mb to 10.3Mb for only a 0.4% degradation in word error rate.

An interesting observation is that greater resolution in terms of the number of quantization levels is required by the lower order parameters (bigrams and bigram back-off weights) as the degree of pruning is increased. The explanation for this lies in the fact that as a model is pruned more severely there are fewer bigrams and trigrams left and consequently the model will need to back-off more frequently. Backed-off estimates compound the quantization errors in the back-off weights and lower-order parameters resulting in greater overall error in the probability estimates. As a result it becomes important to decrease the quantization error in lower-order parameters. Thus the results obtained with 4-bit quantization of unigram parameters were observed to be worse than when 8-bit quantization was used. At extreme levels of pruning, however, the total number of parameters becomes smaller so the average quantization error actually decreases resulting in lower degradation due to quantization.

We observe also that entropy-based pruning reduces the size of the language model by up to 76% with no loss in recognition accuracy. This is because the criterion for pruning is such that removing parameters has a minimal effect on the entropy of the language model. In contrast, the Lloyd-Max algorithm, which yielded the least degradation in performance, is based on minimising the squared error between the quantized and unquantized language models. Since there exists a correlation between the average quantization error and word error rate, we believe that the degradation in recognition performance can be minimised further through the application of better quantization methods. Indeed we hypothesise that a quantization method based on minimising the effect of quantization on the entropy of the language model would significantly reduce any performance degradation.

7. Conclusion

In this paper we have compared the performance of a language model against the extent of the compression applied. Compression was achieved through quantizing parameters, pruning parameters and combinations of quantization and pruning. It was shown that pruning a language model first and then quantizing the parameters gave superior compression for a given word error rate than employing either quantization or pruning alone. Through this combination a broadcast news language model was compressed by up to 83% to only 12.6Mb for no loss in performance on a broadcast news task. Compressing the language model further by quantization to 10.3Mb resulted in only a 0.4% degradation in word error rate which is better than can be achieved through entropy-based pruning alone.

8. References

[1] A. Stolcke, "Entropy-based Pruning of Backoff Language Models," in *Proceedings of the 1998 DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[2] E. W. D. Whittaker and B. Raj, "Quantization-based Language Model Compression," Submitted to *the European Conference on Speech Communication and Technology*, Aalborg, Denmark, 2001.

[3] P. Placeway et al., "The 1996 HUB-4 SPHINX-3 System," in *Proceedings 1997 DARPA Speech Recognition Workshop*, Chantilly, Virginia, Feb 2-5 1997.

[4] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, 1982.

[5] D. S. Pallet, J. G. Fiscus, J. S. Garofolo, A. Martin, and M. A. Przybocki, "1998 Broadcast News benchmark Test Results," in *Proceedings of the DARPA broadcast news workshop*, Feb 28 - Mar 3 1999.