Truncation on Combined Word-Based and Class-Based Language Model Using Kullback-Leibler Distance Criterion

Kae-Cherng Yang[1], Tai-Hsuan Ho[2], Juei-Sung Lin[2], Lin-Shan Lee[123]

[1]Department of Electrical Engineering, Nation Taiwan University, Taipei, R.O.C.
[2]Department Computer Science and Information Engineering, Nation Taiwan University, Taipei, R.O.C.
[3]Institute of Information Science , Academia Sinica, Taipei, R.O.C.
E-Mail: bangdoll@speech.ee.ntu.edu.tw Tel: 886-2-369-2535

Abstract

In this paper we present a novel approach to truncate combined word-based and class-based n-gram language model using Kullback-Leibler distance criterion. First, we investigate a reliable backoff scheme for unseen n-gram using class-based language model, which outperforms conventional approaches using (n-1)-gram in perplexity for both training and testing data. As for the language model truncation, our approach uses dynamic thresholds for different words or word contexts determined by the Kullback-Leibler distance criterion, as opposed to the conventional scheme which truncates the language model by a constant threshold. In our experiments, 80% of the parameters are reduced by using the combined word-based and class-based n-gram language model and the Kullback-Leibler distance truncation criterion, while the perplexity only increases 1.6%, as compared with the word bigram language model without any truncation.

1. Introduction

In the large vocabulary continuous speech recognition, the n-gram language model has been widely used as the effective linguistic constraint to determine the final transcription among several text hypotheses. In order to get a reliable language model, we need a lot of text data and therefore the size of a language model will be also very large. However, due to the constraint of memory, a huge language model will make the speech recognition system impractical. Thus reducing the language model size is important.

An intuitive approach to reduce the language model size is to truncate k-gram entries that appear below a given threshold in the training corpus. Another common approach use the class-based n-gram language model (Brown 1992, Jardino 1993, Martin 1993), which is intrinsically more compact and outperforming the word-based model at estimating unseen word sequences. However, given enough training data, the performance of the word-based model usually surpasses that of the class-based model because it is more accurate in capturing sequential relationships between particular words.

To keep the advantage of word-based and class-based language models, combining these two models within the backoff probability estimation phase is a good approach (Niesler 1996). By using the class-based model as the backoff estimation instead of the lower order word-based model, the performance is apparently improved. Furthermore, with this more accurate backoff estimation using class-based model, we can impose a heavier truncation on word-based model, which only slightly degrades the performance. Therefore, a combined model of both word-based and class-based model for backoff estimation under heavy truncation could meet the high accuracy and compact memory storage requirement at the same time, and this is our approach. Another advantage of this combined model we proposed is that its performance is always higher than using class-based language model alone. Even in the worst case, it still performs as well as the class-based model. That is, if all word n-gram entries have been truncated, this combined model will be the same as the class-based model alone. From this viewpoint, heavy truncation can be done since the lowest bound of performance can also keep in the level of class-based models.

In order to get a better truncation for a given amount of parameters, we use the Kullback-Leibler distance criterion (Kneser 1996, Kullback 1958) to determine the thresholds for all k-gram entries where $k < n$. In our truncating procedure, if $N(w_i,w_{i+1},\ldots, w_{i+k}) < Th(w_i,w_{i+1},\ldots,w_{i+k-1})$, then the context entry $(w_i, w_{i+1}, \ldots, w_{i+k})$ will be deleted, where $N(w_i,w_{i+1},\ldots,w_{i+k})$ is the occurrence count of word context $(w_i, w_{i+1}, \ldots, w_{i+k})$ in training corpus, and $Th(w_i,w_{i+1},\ldots,w_{i+k-1})$ is the threshold given context $(w_i,w_{i+1},\ldots,w_{i+k-1})$.

The rest of this paper is organized as follows: Section 2 describes the language model which combining word-based and class-based models; Section 3 describes the truncating criterion named as Kullback-Leibler distance; Section 4 describes the algorithm of truncation using Kullback-Leibler distance criterion; Section 5 presents the experimental results of the perplexity measures. Finally, in Section 6 we will give a brief conclusion.

2. Combined Language Model

This language model combines word-based and class-based models within the backoff framework. The conventional n-gram probability estimated by maximum-likelihood approach has been proven very effective for modeling language. However, word sequence not present in the training corpus will result in zero probability for the test data. Therefore, we need backoff scheme to calculate the probability for unseen events. Briefly, when we compute the likelihood of word contexts, a certain amount of the total probability mass for the conditioning context should be redistributed to the unobserved words. In the conventional model, the redistribution is proportional to the probability from the next lower-order model. However, from past experiences, we know that the class-based language model is more robust for estimating the probabilities for unseen events. Based on this concept, we believe that using class-based language model in backoff phase can make more accurate estimation among unseen word sequences.

In this combined model, the probability estimation formula of a given word context with n words w_{i-n+1}, ..., w_i is as follows:

$$P_n(w_i | h = w_{i-1},...,w_{i-n+1}) = \begin{cases} P_{w,n}(w_i | h) & \text{if } w_i \in W_n(h) \\ \alpha(h) P_{c,n}(w_i | C(h)) & \text{otherwise} \end{cases} \quad (1)$$

where

- $h = w_{i-n+1}$, ..., w_{i-1} means the word history. For example of trigram model, $h = w_{i-2}, w_{i-1}$.
- $W_n(h)$ is the set of words which connect to word context h in training corpus, i.e., if word $w \in W_n(h)$, it means that there is an n-gram entry that stores the word context (h,w) and its count.
- $P_{w,n}(w | h)$ is the word conditional probability given h for which w belongs to $W_n(h)$, i.e., the n-gram entry (h,w) exists in the word-base model. The estimation of $P_{w,n}(w | h)$ will use both word-based and class-based models.
- α(h) is the backoff weight for the given history h. In our combined language model, linear backoff (Placeway 1993) was employed.
- $P_{c,n}(w | C(h))$ is the word conditional probability given C(h) where C(h) is the class sequence of words in the history. The estimation of $P_{c,n}(w | C(h))$ uses class-based

language model only.

The linear backoff approach (Placeway 1993) is a robust and simple method that can be regarded as a HMM grammar structure (Fig. 1). In contrast to conventional backoff scheme (Katz 1987), this approach estimates the probability by a linear combination of direct estimating path and backoff path.

To see the formula of the linear backoff approach, firstly, we define two terms $P_{w|h}(w \mid h)$, the direct estimation probability, and $\alpha(h)$, the backoff probability mass, as follows:

$$P_{w|h}(w|h) = \frac{N(h,w)}{N(h)+R(h)} \tag{2}$$

$$\alpha(h) = \frac{R(h)+T(h)}{N(h)+R(h)} \tag{3}$$

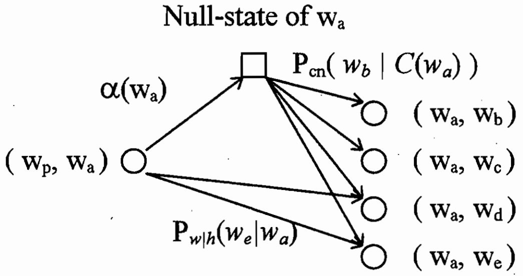

Fig. 1: Example of HMM Bigram Grammar Structure

where N(h,w) is the number of times that word w occurred behind context h in training corpus, R(h) is the number of distinct words that occurred behind context h, and T(h) is the count of total truncated entries, i.e., $T(h) = \Sigma_{w'} N(h,w')$ where w' is the word that the entry (h,w') has been truncated. Note that we should not modify N(h) and R(h) after truncating, i.e., the values of N(h) and R(h) are conditional on whole training corpus and they are independent of truncating process.

In the equations above, $P_{w|h}(w \mid h)$ is the direct estimation probability of word w given the history h and $\alpha(h)$ is the total probability mass through backoff path (null-state) in Fig. 1. For the unseen events, e.g., (w_a, w_b) and (w_a, w_c) in Fig. 1, the probabilities are all estimated by going through the backoff path. As for the observed events, e.g., (w_a, w_d) and (w_a, w_e) in Fig. 1, the probabilities are estimated not only through a direct arc path but also a null-state path. Thus the equation of $P_{w,n}(w \mid h)$ is as follows:

$$P_{w,n}(w|h) = P_{w|h}(w|h) + \alpha(h) P_{c,n}(w|C(h)) \tag{4}$$

where C(h) is {C(w_{i-n+1}), ..., C(w_{i-1})} and C(w) is the class of word w. We assign $P_{c,n}(w \mid C(h))$ as follows:

$$P_{c,n}(w|C(h)) = P(w|C(w)) P_n(C(w)|C(h)) \tag{5}$$

where

$$P(w|C(w)) = \frac{N(w)}{N(C(w))} \qquad (6)$$

$$P_n(C(w)|C(h)) = P_{c(w)|c(h)}(C(w)|C(h)) + \alpha(C(h))P_{n-1}(C(w)|C(h-1)) \qquad (7)$$

The estimation of $P_{C(w)\,|\,C(h)}(C(w) \mid C(h))$ and $\alpha(\,C(h)\,)$ is the same as that for $P_{w\,|\,h}(w \mid h)$ and $\alpha(h)$ except that the word and word history sequence number become class and class sequence history number in training corpus and the backoff weight $\alpha(\,C(h)\,)$ is estimated by lower order class-based language model. Although the above model is complex, we can prove that the summation of probabilities equal to one for all words given the history.

3. Truncation Using Kullback-Leibler Distance Criterion

In order to have a better truncation result, we exploit the Kullback-Leibler distance criterion to measure the quality of truncated language model and determine thresholds for all word k-gram entries where k < n. Let P_I denote the probability distribution of initial language model without any truncation and P_T denote the probability distribution of truncated language model. The Kullback-Leibler distance of these two models is as follows:

$$D(P_T;P_I) = \sum_{h,w} P_I(h,w)\log\frac{P_I(w|h)}{P_T(w|h)} \qquad (8)$$

where $P_I(\,w \mid h\,) = P_n(\,w \mid h\,)$ in the initial model and $P_T(\,w \mid h\,) = P_n(\,w \mid h\,)$ in the truncated model.

We can show that $D(\,P_T; P_I)$ will be greater than or equal to zero and the equality holds if $P_I(\,w \mid h\,) = P_T(\,w \mid h\,)$ only. There is one assumption for using Kullback-Leibler distance criterion: the initial language model without truncation will be the best model comparing to truncated models. Thus if we do any truncation, the resulted model will be worse than initial model. Under this assumption, if the distance of a truncated model is lower, the concerned model with P_T is more near to initial model P_I and therefore the model is considered to be better.

For each time we truncate the k-gram entries, if we use equation (8) to compute the distance of initial model and truncated model, the computation cost is much expensive. To reduce the computational complexity, we can further derive equation (8). Let h_k denote the first k word context in history h. The equation (8) can be rewritten as follows:

$$D(P_T;P_I)=\sum_{(h,w),h_k\neq h_k'} P_I(h,w)\log\frac{P_I(w|h)}{P_T(w|h)}+\sum_{(h,w),h_k=h_k'} P_I(h,w)\log\frac{P_I(w|h)}{P_T(w|h)} \qquad (9)$$

If we change the threshold of k-gram entry with history h_k', we can only calculate the later term in equation (8). Thus we can define the term d(h_k) as follows:

$$\begin{aligned} d(h_k') &= \sum_{(h,w),h_k=h_k'} P_I(h,w)\log\frac{P_I(w|h)}{P_T(w|h)} \\ &= \sum_{(h,w),h_k=h_k'} P_I(h,w)\log P_I(w|h)-\sum_{(h,w),h_k=h_k'} P_I(h,w)\log P_T(w|h) \qquad (10) \\ &= d_0(h_k')-d_T(h_k') \end{aligned}$$

We can calculate the former term in equation (10), d_0 (h_k'), in the initialization procedure and store them. For each time we adjust the threshold, the later term $d_T(h_k')$ is the one that we must calculate .

4. Truncating Algorithm

In the practical system, there is a constraint of the memory that we can use. Therefore the total parameter number of all k-gram entries has an upper bound. Our algorithm is to find the better solution to determine what parameters in word-based language model should be truncated. The parameters in class-based model will not be truncated because the class-based language is much smaller than word-based language model and they are robust to calculate the backoff probabilities.

Before describing our algorithm, firstly we define some terminology as follows:

- Th(h_k'): the threshold of k-gram entries with word context h_k'. If one (k+1)-gram entry with history h_k' is that its count occurring in training corpus is less than Th(h_k'), this entry will be deleted and its count will be added to T(h_k') as describing in backoff phase.

- N_T(h_k' , Th): the total number of m-gram (m = k+1 ~ n) entries with first k symbol history equaling to h_k' and their counts in training corpus are less than Th.

- N_T: the total number of entries that has been deleted, i.e., $N_T = \Sigma_{h_k'} N_T(h_k', Th(h_k'))$.

- N_I: the total number of entries in the initial language model.

- $dn_T(h_k', Th) = d_T(h_k') / (N_T(h_k', Th) - N_T(h_k', Th\text{-}1))$: the normalizing distance of h_k' given threshold Th.

The algorithm is as follows:

```
1) Initialize
   1.1) Set all thresholds, $Th(h_k')$, equal to 2 and total truncated entry number $N_T$=0.
   1.2) Calculate $d_0( h_k' )$, $d_T( h_k' )$, and normalizing distance $dn_T( h_k', Th(h_k') )$ for each $h_k'$.
2) Loop
   2.1) Find the best $h_k'$ that has the smallest distance $dn_T( h_k', Th(h_k') )$ and let $h_B = h_k'$.
   2.2) Calculate $N_T$. If $N_I - N_T <=$ upper bound of parameter number, then break.
   2.3) Set $Th( h_B ) = Th( h_B ) + 1$. Calculate $N_T$ and $dn_T( h_B, Th( h_B ) )$.
3) End
```

Instead of the stop condition for the loop in above algorithm, we can also change the stop condition to control the performance of our combined language model. For this case, we don't need the term N_T, but we must have one term ΔE that is the accumulative distance of truncated model and initial model. If ΔE is larger than a threshold max-ΔE, then we stop the loop.

5. Experimental Results

5.1 Experimental environment

The corpus in our experiments is obtained from newspapers of eight months. Seven out of eight months' data is used for training, and the remaining one is used for testing. The lexicon is provided by CKIP. The vocabulary size is 94188 and the maximum length of word entries is nine. After the word segmentation, there are 10,136,783 words in the training corpus and 1,521,867 words in the testing corpus. The resulted word bigram language model has 2,484,757 bigram entries and 55,380 unigram entries. The perplexity of this model is 307.641. All following experiments use testing corpus to evaluate perplexities.

5.2 Class-Based Bigram Language Model

Our class-based language model is generated by two phases. In the first phase, we use simulated annealing approach [2] to cluster words. However, the result of the first phase is not good enough. In the second phase, we use the clustering result of first phase to be the initial condition and use k-means-style algorithm [3] to improve it.

In both algorithms, we classify 27,829 highest frequency words for three class models with 999, 499, and 249 classes respective, and the words in the residual part are all collected into one class. Thus the total numbers of classes are 1000, 500, and 250.

For the simulating annealing algorithm, the parameters (T_0, T_f, α, i_{max}, r_{max}) are set to (1, 10^{-100}, 0.9, 20000,5000) empirically. It takes at most 48.0 CPU hours on a Pentium 166 machine with 128M ram. For the k-means-style approach, the time complexity is larger. For the case of 1000 classes, we need 5 days for 10 iterations.

Table 1 show perplexity values for simulating annealing approach and k-means-style approach in second phase. The results show that the performance was improved after second phase process.

class number	*250*	*500*	*1000*
Simulating Annealing	538.326	469.951	412.632
k-means-style algorithm	513.094	450.246	392.211

Table 1. Perplexity measures of class-based bigram language model

5.3 Combined Bigram Language Model

The combined bigram language models discussed in section 2 are generated by combining word-based bigram model and class-based bigram models that have 250, 500, 1000 classes respectively. The perplexity values with no truncation are shown in the table 2.

Class number	*250*	*500*	*1000*	*Word-Bigram*
Parameter number	60,030	216,270	620,087	2,540,137
Perplexity value	291.724	292.260	294.523	307.641

Table 2. Perplexity measure of combined bigram and word bigram language model

Note that combined bigram language model with 1000 classed is not the best. Inversely, the combined language model with 250 classes is better than the other two models. Since our language model structure is as HMM that contains a null-state path for not only unseen events but also observed events, there will be an overestimation problem if class number is too large. Therefore it still exists a problem to choose the best class number.

5.4 Truncation on Word Bigram Models with Constant Threshold

We truncate the bigram entries that their counts are smaller than a given threshold and then calculate the perplexity values for word bigram model. The experimental results are shown in table 3.

	Word bigram model	
threshold	Total number of parameters	perplexity value
2	541,014	346.461
3	387,483	363.109
4	300,430	378.260
5	245,734	391.455
6	207,909	403.767
8	158,931	424.919
10	128,077	444.307
20	63,753	515.61

Table 3. Parameter number and perplexity value for word bigram models with constant threshold

5.5 Truncation on Word and Combined Language Models Using Kullback-Leibler Distance

We truncate the bigram entries on both word-based and combined language models by Kullback-Leibler distance criterion. The entry numbers for both models are near the result of constant threshold. The parameter numbers and perplexity values are shown in table 4.

Word bigram model		combined bigram model	
Total number of parameters	perplexity value	Total number of parameters	perplexity value
541,011	347.339	540,992	315.706
387,468	362.261	387,480	326.210
300,425	375.658	300,424	335.350
245,731	387.655	245,733	344.157
207,909	398.966	207,909	352.622
158,931	417.654	158,930	367.882
128,076	434.089	128,077	383.165
63,753	496.737	63,752	485.330

Table 4. Parameter number and perplexity value for word and combined bigram models with dynamic threshold determined by Kssullback-Leibler distance where class number is 250.

Comparing table 4 with table 3, using Kullback-Leibler distance criterion to determine thresholds for all words will be better than constant threshold under the same number of parameters, especially when the total number of truncated entries is large. Besides, the combined language model is better than other two models. The perplexity of combined model is about less than 10% of word model.

6. Conclusion

In this paper, we have present the combined word-based and class-based language model within backoff framework. Our experiments show that this combined language model is better than conventional n-gram language models. Besides, for a practical system, the number of parameters in a language model can not be too much. We develop a truncation algorithm based on Kullback-Leibler distance criterion that show that the resulted model will outperform the model truncated by constant threshold. Finally, in order to get a good trade-off between complexity and performance, we show that the truncation on combined word-based and class-based n-gram language model using Kullback-Leibler distance criterion will have better results.

References

P.F. Brown, V.J. Della Pietra, P.V. deSouza, J.C. Lai, R.L. Mercer: "Class-Based n-gram Models of Natural Language", Computational Linguistics, Vol. 18, No. 4, pp. 467-479, 1992

M. Jardino, G. Adda, "Automatic Word Classification Using Simulated Annealing", Proc. ICASSP II, 1993, Minneapolis, Minnesota, USA, pp. 41-44

S. Martin, J. Liermann, H. Ney, "Algorithms for Bigram and Trigram Word Clustering", EUROSPEECH, 1995, Madrid, September, pp. 1253-1256

T.R. Niesler, P.C. Woodland, "Combination of Word-Based and Category-Based Language Models", ICSLP, 1996, vol I, pp. 220-223

R. Kneser, "Statistical Language Modeling Using a Variable Context Length", Proc. ICSLP, 1996, vol I, pp494-497

S. Kullback, *Information Theory and Statistics*, New York, NY: Wiely, 1958

P. Placeway, R. Schwartz, P. Fung, L. Nguyen, " The Estimation of Powerful Language Models from Small and Large Corpora", Proc. ICASSP, 1993, vol. II, pp. 33-36

S.M. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer", IEEE Trans. Acoustic, Speech, Signal Processing, vol. ASSP-35, no. 3, pp.400-401, Mar. 1987