STATISTICAL LANGUAGE MODELING USING A VARIABLE CONTEXT LENGTH

Reinhard Kneser

Philips GmbH Forschungslaboratorien Aachen, Weißhausstr. 2, D-52066 Aachen, Germany
E-mail: kneser@pfa.research.philips.com

ABSTRACT

In this paper we investigate statistical language models with a variable context length. For such models the number of relevant words in a context is not fixed as in conventional M-gram models but depends on the context itself. We develop a measure for the quality of variable-length models and present a pruning algorithm for the creation of such models, based on this measure. Further we address the question how the use of a special backing-off distribution can improve the language models.

Experiments were performed on two data bases, the ARPA-NAB corpus and the German Verbmobil corpus, respectively. The results show that variable-length models outperform conventional models of the same size. Furthermore it can be seen that if a moderate loss in performance is acceptable, the size of a language model can be reduced drastically by using the presented pruning algorithm.

1. Introduction

The usual approach in statistical language modeling for automatic speech recognition to obtain an estimate for the probabilities of a given word sequence $w_1 \dots w_n$ is to decompose the probability into a product of conditional probabilities. In order to be able to estimate the huge number of possible conditional probabilities, additional model assumptions must be made, the most common being the M-gram model, where the dependence is assumed to be only on the last $M-1$ words:

$$p(w|h) = p(w|h_{M-1}). \tag{1}$$

Here we use the short-hand notations h for the history consisting of all words in the sequence preceding the current word, and h_k for the last k words in h.

Theoretically, an improvement of the model should always be achieved by making the length M of the considered context larger. But also the number of parameters grows rapidly with M, such that estimation and storage problems arise. Instead of enlarging the length globally for all contexts it is therefore advisable to do this only selectively, depending on the context itself.

Several different approaches to the problem of determining this context length have been undertaken. One way is to enlarge the considered context successively, as long as this leads to a significant improvement of the model as measured by some criterion [8][6]. In [2] the modeling of long-range dependences was achieved by allowing for multi-word sequences as units while keeping the context length M fixed. Finally, also the very simple cut-off method where M-grams are selected according to their frequency and which was used for the generation of the official compact ARPA-NAB models can be regarded as a variable-length model.

We propose a pruning strategy in order to obtain a variable-length model. Starting with a full M-gram model with some large number M, we successively discard those M-grams which have the smallest impact on the model.

2. Specifying Variable-Length Models

Variable-length models are determined to a great part by the set $\mathcal{S}$ of the word sequences for which a distinct probability is modeled. For each word sequence $(h_k, w) \in \mathcal{S}$ let $\alpha(w|h_k)$ be an estimate of the conditional probability $p(w|h_k)$ which is usually calculated from frequencies in the training data by some discounting method [5]. We now apply the backing-off technique [3] and obtain estimates for word sequences not contained in $\mathcal{S}$ by considering only a shorter context length:

$$P_S(w|h_k) := \begin{cases} \alpha(w|h_k) & \text{if } (h_k, w) \in \mathcal{S} \\ \gamma(h_k)\, P_S(w|h_{k-1}) & \text{otherwise} \end{cases} . \tag{2}$$

This defines the model completely since the resulting model is an M-gram model with M being the length of the longest sequence in $\mathcal{S}$. The normalization factor γ (or γ_S if we want to stress the dependence on $\mathcal{S}$) is necessary in order to get distributions that sum to unity and is completely determined by α and $\mathcal{S}$. Hence α is the only free parameter and the number of parameters is basically determined by the number of elements in $\mathcal{S}$.

Note that if $\mathcal{T}$ is the set of all word sequences in the training data with a maximal length of M, $P_{\mathcal{T}}$ represents the

This work was partially funded by the German Federal Ministry for Education, Science, Research and Technology (BMBF) in the framework of the Verbmobil Project under Grant 01 IV 102 B. The responsibility for the contents of this study lies with the author.

standard fixed-length M-gram model. Starting from this we may obtain a variable length model by adding and removing selected word sequences, which makes the considered context length larger and shorter, respectively. Variable-length models can for example be defined by removing all word sequences from $\mathcal{T}$ for which the frequency in the training data is less or equal to some cut-off value, where different values may be used for sequences of different length. This cut-off method was for example used to generate the so-called compact language models for the ARPA-NAB evaluations.

3. Measuring the Quality

We want to create variable-length models by finding a suitable subset $\mathcal{S}$ of the set $\mathcal{T}$ of all M-gram sequences in the training data with a given maximal context length M. Therefore we must be able to measure the quality of a model. The perplexity of the model $P_{\mathcal{S}}$ on a corpus generated according to the full M-gram model $P_{\mathcal{T}}$ might serve as criterion, assuming that this full model is a good reference. This leads in a natural way to the notation of the conditional relative entropy

$$D_1(P_{\mathcal{T}}\|P_{\mathcal{S}}) := \sum_{h_{M-1},w} P_{\mathcal{T}}(h_{M-1},w)\log\frac{P_{\mathcal{T}}(w|h_{M-1})}{P_{\mathcal{S}}(w|h_{M-1})} \quad (3)$$

which can also be regarded as average Kullback Leibler distance. If this distance is low, the concerned model $P_{\mathcal{S}}$ is near to $P_{\mathcal{T}}$ and thus the model is considered to be good. Unfortunately this distance is expensive to calculate. Especially it is not possible to reuse parts of the calculation for other sets since almost all terms in the sum really depend on $\mathcal{S}$. Therefore it is desirable to simplify this distance.

We assume $\mathcal{S} \subset \mathcal{T}$ and take $\alpha = P_{\mathcal{T}}$ in eq. (2). In a first approximation we ignore the dependence of γ on $\mathcal{S}$. With this approximation we observe that the estimates $P_{\mathcal{S}}$ and $P_{\mathcal{T}}$ are the same for word sequences (h_k, w) which are contained in $\mathcal{S}$. For $(h_k, w) \notin \mathcal{S}$ we get

$$\frac{P_{\mathcal{T}}(w|h_k)}{P_{\mathcal{S}}(w|h_k)} = \frac{P_{\mathcal{T}}(w|h_k)}{\gamma(h_k)P_{\mathcal{T}}(w|h_{k-1})}\frac{P_{\mathcal{T}}(w|h_{k-1})}{P_{\mathcal{S}}(w|h_{k-1})} \quad (4)$$

and for $(h_k, w) \notin \mathcal{T}$ this simplifies to

$$\frac{P_{\mathcal{T}}(w|h_k)}{P_{\mathcal{S}}(w|h_k)} = \frac{P_{\mathcal{T}}(w|h_{k-1})}{P_{\mathcal{S}}(w|h_{k-1})}. \quad (5)$$

Finally we assume that if $(h_k, w) \notin \mathcal{S}$ there is no longer history $h_{k'}$ ending in h_k for which $(h_{k'}, w) \in \mathcal{S}$. Applying these approximations to eq. (3) motivates the distance measure

$$D_2(P_{\mathcal{T}}\|P_{\mathcal{S}}) := \sum_{k=0}^{M-1} \sum_{(h_k,w)\in\mathcal{T}\setminus\mathcal{S}} d_1(h_k, w) \quad (6)$$

where the terms of the sum are defined by

$$d_1(h_k, w) := P_{\mathcal{T}}(h_k, w)\log\frac{P_{\mathcal{T}}(w|h_k)}{\gamma_{\mathcal{T}}(h_k)P_{\mathcal{T}}(w|h_{k-1})}. \quad (7)$$

Since the terms $d_1(h_k, w)$ do no longer depend on $\mathcal{S}$ they can be calculated beforehand for all elements of $\mathcal{T}$ and the search for some optimal $\mathcal{S}$ can thus be performed very efficiently.

4. Pruning Algorithm

We have seen that a subset $\mathcal{S} \subset \mathcal{T}$ corresponds to a variable length model and that eq. (6) can be used to measure the quality of this model. The creation of a variable-length model can thus be regarded as the search problem of finding the set $\mathcal{S}$ of a given size which minimizes eq. (6). In the general case this is easily done by simply removing those elements from $\mathcal{T}$ which have the lowest values $d_1(h_k, w)$.

In our implementation we save access time and memory by storing the word sequences of $\mathcal{S}$ in a tree structure. Each node of this tree corresponds to a word sequence and each arc is labeled with a word identity, such that the parent node of a word sequence $w_1 \dots w_k$ corresponds to the word sequence $w_1 \dots w_{k-1}$ with the last word removed and such that the connecting arc is labeled with the identity of the removed word w_k. Since in this implementation memory gets allocated anyway for each node, we consider only sets $\mathcal{S}$ where each node W [1] in the corresponding tree is also contained in $\mathcal{S}$. This implies that if we remove W from $\mathcal{S}$, and thus from the tree, we also remove all successor nodes $Succ(W)$, i. e. the set of all longer word sequences starting with the same words as W.

Also in this special case where only subsets $\mathcal{S}$ are admitted which are representable as trees we follow the same basic idea as in the general case. Starting with the full subset $\mathcal{S} = \mathcal{T}$, we successively remove those sequences that contribute least to the sum in eq. (6) until $\mathcal{S}$ reaches some size K specified in advance. For each node $W = (h_k, w) \in \mathcal{S}$ we calculate the average contribution to the sum in eq. (6) for the case that this node including all its successor nodes gets removed:

$$d_2(W) = \frac{d_1(W) + \sum_{V\in Succ(W)} d_1(V)}{1 + |Succ(W)|}. \quad (8)$$

If we always prune the nodes with the lowest average contribution, the resulting tree is the best among all trees of the same size. Hence we have the following pruning algorithm:

```
Start with S = T.
while (|S| > K)
  For all nodes in S calculate d2.
  Remove node with lowest d2.
```

For each node we might save its average contribution d_2 at the time when it gets removed. Once we have these values we can reproduce the result by just removing all nodes with a value below the average contribution of the last node that was pruned. The fact that removing a node from $\mathcal{S}$ only effects the average contribution of predecessor nodes makes it possible to efficiently calculate these values in a bottom up fashion in advance, such that the cost for the search is negligible compared to the cost needed for the calculation of d_1.

[1] We do not differentiate between the node W, the corresponding word sequence $w_1 \dots w_k$ and the word w_k in the context of the history $h_{k-1} = w_1 \dots w_{k-1}$

5. Improved Backing-Off Distribution

In the backing-off case of eq. (2) we assume the probability to be proportional to the probability which considers only a smaller context. This does not account for the additional information that, since the backing-off branch was selected, this event is not covered by some longer context. In [4] a backing-off model was improved by taking a modified distribution $\beta(w|h_k)$ in the backing-off branch of eq. (2), instead of $P_S(w|h_k)$. This β was selected according to some constraints on the marginal distributions of $P_S(h_k, w)$. Applying the same method to the case of variable-length models gives the solution

$$\beta(w|h_k) \approx \frac{N(h_k, w) - \sum_{v:(v,h_k,w)\in S} N(v, h_k, w) - d}{\sum_{w'}[N(h_k, w') - \sum_{v:(v,h_k,w')\in S} N(v, h_k, w') - d]}, \quad (9)$$

where we write N for the frequencies in the training data and absolute discounting with the discounting parameter d is assumed. This equation holds only approximately since the values must be smoothed in order to avoid values of $\beta = 0$. We will refer to this as the *optimized* backing-off distribution, in contrast to the *standard* distribution.

6. Experiments

Experiments have been performed on two different data sets, the ARPA North-American-Business-News (NAB) corpus [7] and the German Verbmobil appointment-scheduling corpus [9]. The NAB corpus comprises about 240 million words of newspaper data. A vocabulary of 64k words with a coverage of about 99.5% was used in the experiments. The development and evaluation data of the 1994 ARPA evaluations served as test set. The Verbmobil corpus consists of more than 600 spontaneous dialogs between two partners trying to fix a date for an appointment and comprises a total of 300,000 words with a vocabulary of about 5000 words.

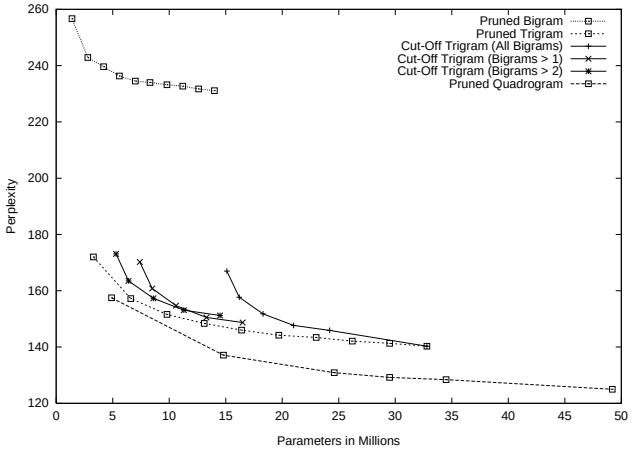

Figure 1: Perplexities of variable-length models of different size (NAB)

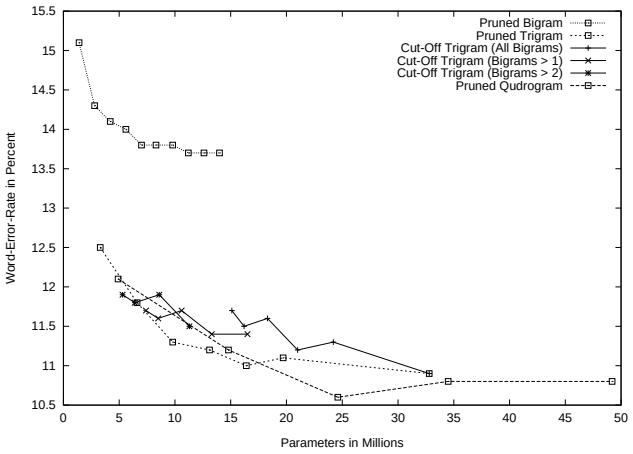

Figure 2: Word error rates for variable-length models of different size (NAB)

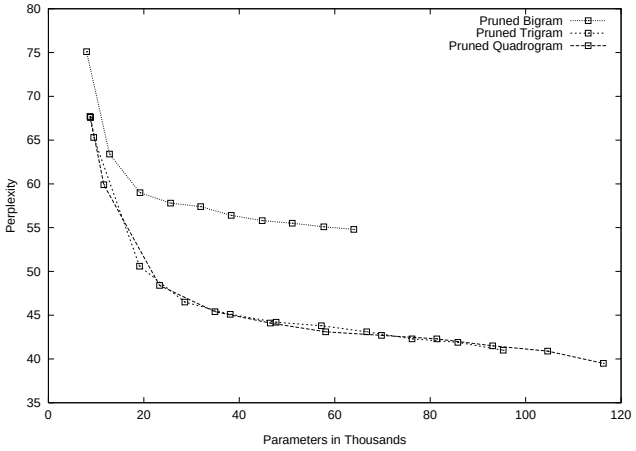

Figure 3: Perplexities of variable-length models of different size (Verbmobil)

We have built variable-length models for a maximal context length M of up to 5 words by starting with a full M-gram model and by successively pruning this as described in section 4. In order to avoid memory problems these start models were already slightly pruned by ignoring all 3-, 4- and 5-grams which occurred only once. Backing-off models were trained using the optimized backing-off distribution and the method of non-linear interpolation [5], and test-set perplexities were calculated for each model. The results are presented in Figures 1 and 3. Variable-length models with the same maximal context length are connected with dotted lines and the full model, having the largest number of parameters, is represented by the rightmost point. Although 5-gram models have been calculated for both tasks, they were not included in the Figures, since they are almost identical to the 4-gram results. In addition to the calculation of perplexities we have run recognition experiments for the NAB task. This was done by applying the various language models in the last

step of our lattice rescoring algorithm [1].

We observe that longer contexts generally perform better even when comparing models with the same number of parameters. This difference is largest between bigram and trigram models and amounts to a factor of about 1.5 in terms of perplexity. When going to longer context lengths this difference becomes smaller. 3-gram and 4-gram models of same size perform similar on the Verbmobil corpus, while on the much larger NAB corpus this limit lies between 4-gram and 5-grams. Nevertheless these models can still be slightly improved by allowing for a larger number of parameters. The performance of models having the same maximal context length depends directly on the number of parameters but the difference is small for a long range, such that if a small loss in performance is acceptable the size of the language model can be reduced drastically. Reducing for example the size of the standard NAB trigram model by a factor of 3 gives only a loss of 7% in terms of perplexity and 3% in terms of word error rate, respectively.

In an additional experiment the cut-off method of section 2. was used to create trigram models for the NAB corpus. Different cut-off values for bi- and trigrams were used. The results are plotted in Figures 1 and 2 by connecting models having the same bigram cut-off values with solid lines. The performance of all of these models is not as good as the corresponding variable-length model of same size, generated by our pruning algorithm. The difference depends much on the selection of the cut-off values and seems to be larger for high trigram cut-offs. Note also that the problem of the right choice of cut-off values gets even more complicated if an additional cut-off value for 4-grams is necessary.

There is a good correspondence between the results of the recognition experiments and the perplexities, although the curves are not as smooth. A discrepancy can be observed in the case of the 4-gram which might be explained by search errors, since in contrast to the trigram models, only an approximate search is used for the 4-gram models.

In Table 1, results for the standard and optimized backing-off distributions are listed for selected models. For all models the optimized distribution performs better in terms of perplexity as well as in terms of error rate. The effect is larger for longer contexts.

Table 1: Perplexities (PP) and word error rates (WER) for different backing-off distributions (NAB)

	Standard BO		Optimized BO	
M	PP	WER	PP	WER
14 Million Parameters				
2	235.7	14.0%	231.1	13.7%
3	164.0	12.1%	148.4	11.2%
4	160.2	11.8%	137.1	11.2%
5	161.5	–	134.4	–
33 Million Parameters				
3	153.3	11.6%	140.3	10.9%
4	150.3	11.4%	128.4	10.8%
5	154.3	–	127.9	–

7. Conclusions

We presented an algorithm for the calculation of a statistical language model with a variable context length. In experiments on two different corpora we were able to build variable-length language models having the same number of parameters but performing better, compared to conventional models. The improvement depends on the length of the fixed context and on the amount of available training data and was e. g. 10% in terms of perplexity in the case of the NAB corpus, compared to our best trigram model. The same algorithm can be used to build language models of scalable size. When accepting a moderate loss in performance, the memory requirements of a model can be reduced drastically. The application of this algorithm showed to be superior to the simpler cut-off method. Finally we showed that the use of a special backing-off distribution which was modified according to the needs of the variable-context-length models gave consistently better results compared to the standard backing-off distribution.

8. REFERENCES

1. X. Aubert, C. Dugast, H. Ney, and V. Steinbiss. Large vocabulary continuous speech recognition of Wall Street Journal data. In *Proc. ICASSP*, volume 2, pages 129–132, Adelaide, Australia, Apr. 1994.
2. S. Deligne and F. Bimbot. Language modeling by variable length sequences: Theoretical formulation and evaluation of multigrams. In *Proc. ICASSP*, volume 1, pages 169–172, Detroit, MI, May 1995.
3. S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. on Acoustics, Speech and Signal Proc.*, ASSP-35:400–401, March 1987.
4. R. Kneser and H. Ney. Improved smoothing for M-gram language modeling. In *Proc. ICASSP*, volume 1, pages 181–184, Detroit, MI, May 1995.
5. H. Ney, U. Essen, and R. Kneser. On structuring probabilistic dependencies in stochastic language modelling. *Computer Speech and Language*, 8:1–38, 1994.
6. T. R. Niesler and P. C. Woodland. A variable-length category-based n-gram language model. In *Proc. ICASSP*, Atlanta, GA, May 1996.
7. D. B. Paul and J. M. Baker. The design for the Wall Street Journal-based CSR corpus. In *Proc. DARPA Speech and NL Workshop*, pages 357–362, Harriman, NY, Feb. 1992.
8. D. Ron, Y. Singer, and N. Tishby. The power of amnesia. In J. Cowan et al., editor, *Advances in Neural Information Processing Systems*, volume 6, pages 176–183. Morgan Kauffmann, 1994.
9. W. Wahlster. VERBMOBIL: Translation of face-to-face dialogs. In *Proc. Eurospeech*, volume Opening and Plenary Sessions, pages 29–38, Berlin, Germany, Sep. 1993.