Language Model Adaptation Using Dynamic Marginals

Reinhard Kneser[1], *Jochen Peters*[2], *and Dietrich Klakow*[2]

[1]Philips GmbH Speech Processing, Kackertstr.10, D-52072 Aachen, Germany
kneser@acn.be.philips.com
[2]Philips GmbH Forschungslaboratorien, Weisshausstr.2, D-52066 Aachen, Germany
{peters | klakow}@pfa.research.philips.com

Abstract

A new method is presented to quickly adapt a given language model to local text characteristics. The basic approach is to choose the adaptive models as close as possible to the background estimates while constraining them to respect the locally estimated unigram probabilities. Several means are investigated to speed up the calculations. We measure both perplexity and word error rate to gauge the quality of our model.

1 Introduction

Common speech recognition systems combine informations from an *acoustical model* and a *language model* to find the most probably spoken sentence [9]. Well–trained language models use *large text corpora* to estimate conditional word probabilities in the context of the preceding $n-1$ words. The resulting n–grams represent the *average* text structures of the training material, but fail to fully describe the fluctuations between *individual* texts.

Many language models thus include a *cache* component to dynamically adapt to the characteristics of the text just recognized [7]. The rather limited cache size restricts the range of local estimates. The *most reliable* information derived is an adaptive unigram. A rough estimate is provided by the relative word frequencies within the cache. More elaborate approaches use trigger effects to increase the probabilities of words that are semantically related to the cached text [8, 5]. Additionally, the probabilities of the few cached bi– or trigrams may be increased.

A crucial question is how to combine the local estimates and the prior knowledge of an n–gram, which was trained on a large background corpus. A sophisticated model uses Maximum Entropy to integrate trigger effects into the background model. This approach, however, requires *enormous training times* [8]. The most widespread type of cache models just linearly interpolates between a *static* n–gram and the locally estimated unigram. Unfortunately, this leads to a *major disadvantage*: The probabilities for cached words are uniformly increased without respect to the history. This effect partly destroys the discriminating power of a well–trained conditional model.

In this paper we describe and evaluate a new method to quickly modify a given static n-gram such that the local unigram properties are correctly modelled without destroying the full history dependence of the original n-gram.

2 Using Dynamic Marginals

To overcome the above mentioned disadvantages we propose to use the cache information in a different way. Instead of linearly interpolating between the background n–gram $p_{\text{back}}(w \,|\, h)$ and the locally estimated (smoothed) unigram $p_{\text{adap}}(w)$ we treat the latter as a *dynamic marginal* restricting the allowed adaptive n–grams:

$$\sum_h p_{\text{adap}}(h) \cdot p_{\text{adap}}(w \,|\, h) \overset{!}{=} p_{\text{adap}}(w) \tag{1}$$

for all w in the vocabulary. Here, h denotes any history of length $(n-1)$. For the moment, $p_{\text{adap}}(h)$ and $p_{\text{adap}}(w)$ are assumed to be already estimated. Following the idea of *Minimum Discriminant Estimation* [4], we are now looking for that n–gram $p_{\text{adap}}(w \,|\, h)$ which, while satisfying (1), is closest to the well–trained model $p_{\text{back}}(w \,|\, h)$ in terms of relative entropy D (also called Kullback–Leibler distance, discriminant information, etc. [2]). More precisely, we define the adaptive n–gram model as follows:

$$p_{\text{adap}}(* \,|\, *) = \underset{p(*\,|\,*)}{\operatorname{argmin}} \sum_h p_{\text{adap}}(h) \cdot D\big(\, p(* \,|\, h) \,\|\, p_{\text{back}}(* \,|\, h) \,\big) \tag{2}$$

Here, the minimum is taken over all normalized distributions $p(* \,|\, *)$ satisfying the constraints (1). The *weighted average* of relative entropies to be minimized reflects the fact that the importance of $p(* \,|\, h)$ increases with the likelihood of h during recognition.

Usual calculus techniques for restricted optimization reveal a simple structure of the adaptive n–grams [2, Sec.11.1]:

$$p_{\text{adap}}(w \,|\, h) = \frac{\alpha(w) \cdot p_{\text{back}}(w \,|\, h)}{\sum_v \alpha(v) \cdot p_{\text{back}}(v \,|\, h)} \tag{3}$$

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 701 C9. The responsibility for the contents of this study lies with the authors.

Here, the parameters $\alpha(w)$ must be adjusted such that the given constraints are respected.

We want to stress that the derived model structure retains most of the differentiating structure of the background n–gram, since the well–trained history dependence remains an integral part of the adaptive model.

It might interest the reader that we could have started with the model structure given in (3). Then, minimizing the perplexity of some adaptation material would have yielded the set of equations (1) as extremum conditions. In that case, $p_{\text{adap}}(h)$ and $p_{\text{adap}}(w)$ would be the history and the unigram distributions within the adaptation material.

3 Calculating the Parameters

3.1 Iterative Scaling

Given the model structure (3) and the constraints (1) we now proceed to determine the adaptive parameters $\alpha(w)$. Since there is no closed solution to the resulting nonlinear equations we will make use of the well–established *Generalized Iterative Scaling*–algorithm (GIS) of Darroch and Ratcliff [3]. In order to use their approach we first reformulate the constraints (1):

$$\sum_{h,v} p_{\text{adap}}(h) \cdot p_{\text{adap}}(v \,|\, h) \cdot f_v(h,w) = p_{\text{adap}}(w) \quad (4)$$

using the following unigram feature functions:

$$f_v(h,w) = \delta_{v,w} \quad (5)$$

After a uniform scaling of all the features by a factor β $(0 < \beta \leq 1)$ we obtain the following iterative procedure to calculate the wanted parameters α:

$$\begin{aligned} \alpha^{(0)}(w) &= 1 & (6)\\ \alpha^{(k+1)}(w) &= \left(\frac{p_{\text{adap}}(w)}{\sum_h p_{\text{adap}}(h) \cdot p^{(k)}(w \,|\, h)} \right)^{\beta} \cdot \alpha^{(k)}(w) \end{aligned}$$

Here, the conditional probabilities $p^{(k)}(w \,|\, h)$ are calculated as in (3) but based on the intermediate parameters $\alpha^{(k)}$. As $k \to \infty$ the $\alpha^{(k)}$ converge to the solution of (1).

Please note, that a scaling of the features is absent in most applications of the GIS to language models known from the literature. The "standard" version of GIS corresponds to $\beta = 1$.

3.2 Fast Approximations

As we are interested in a fast algorithm which allows for frequent online adaptations we proceed to further simplify the above outlined method. As a first step to minimize the computational efforts we will stop after the first GIS iteration, whence we have:

$$\alpha(w) \approx \alpha^{(1)}(w) = \left(\frac{p_{\text{adap}}(w)}{\sum_h p_{\text{adap}}(h) \cdot p_{\text{back}}(w \,|\, h)} \right)^{\beta} \quad (7)$$

Up to now we have not discussed how the adaptive history distribution $p_{\text{adap}}(h)$ is determined. Since a sound estimation would require much more data than is usually available for adaptation we decided to take the background distribution $p_{\text{back}}(h)$ as an approximate substitute. As an additional benefit we avoid the expensive calculation of the denominator in (7):

$$\alpha(w) \approx \left(\frac{p_{\text{adap}}(w)}{p_{\text{back}}(w)} \right)^{\beta} \quad (8)$$

Using this approximation the complete model now reads as follows:

$$p_{\text{adap}}(w \,|\, h) = \frac{\alpha(w)}{z(h)} \cdot p_{\text{back}}(w \,|\, h) \quad (9)$$

with

$$z(h) = \sum_w \alpha(w) \cdot p_{\text{back}}(w \,|\, h) \quad (10)$$

3.3 Accelerated Normalization

Let us suppose that the background model has a backing–off structure. Here, different probability estimates $p_{\text{back}}(w \,|\, h)$ are used for the set $\mathcal{T}$ of all n–grams (h,w) that have been observed in the background data and for the set of unseen n–grams. The second case is generalized to the less specific transition $\hat{h} \to w$ with a shortened history $\hat{h}$.

In this case a further reduction of computational complexity can be achieved by introducing an additional constraint for each history h. Instead of only requiring the distribution $p_{\text{adap}}(* \,|\, h)$ to be normalized we further constrain it to leave the total probability of observed transitions unchanged:

$$\sum_{w:(h,w)\in\mathcal{T}} p_{\text{adap}}(w \,|\, h) \overset{!}{=} \sum_{w:(h,w)\in\mathcal{T}} p_{\text{back}}(w \,|\, h) \quad (11)$$

These additional constraints motivate the following structure for an adaptive model:

$$p_{\text{adap}}(w \,|\, h) = \begin{cases} \dfrac{\alpha(w)}{z_0(h)} \cdot p_{\text{back}}(w \,|\, h) & \text{if} \quad (h,w) \in \mathcal{T} \\[2ex] \dfrac{1}{z_1(h)} \cdot p_{\text{adap}}(w \,|\, \hat{h}) & \text{else} \end{cases} \quad (12)$$

with the following partial normalization factors $z_0(h)$ and $z_1(h)$:

$$z_0(h) = \frac{\sum_{w:(h,w)\in\mathcal{T}} \alpha(w) \cdot p_{\text{back}}(w \,|\, h)}{\sum_{w:(h,w)\in\mathcal{T}} p_{\text{back}}(w \,|\, h)} \quad (13)$$

and

$$z_1(h) = \frac{1 - \sum_{w:(h,w)\in\mathcal{T}} p_{\text{adap}}(w \,|\, \hat{h})}{1 - \sum_{w:(h,w)\in\mathcal{T}} p_{\text{back}}(w \,|\, h)} \quad (14)$$

This results in a considerable speed–up as compared to (10) since the sums are no longer taken over the full vocabulary but are now restricted to the comparatively few words following the respective history h in the background corpus.

4 Experiments

Various tests have been performed on the Spoke 2 adaptation task of the 1994 ARPA evaluation [6] which provided two sets of domain specific newspaper articles, one about Jackie Kennedy and one about Korea, henceforth denoted *Kennedy* and *Korea*. Both data sets are devided into an adaptation and a test set, each containing approximately 12 000 words. The background models $p_{\text{back}}(w)$ and $p_{\text{back}}(w\,|\,h)$ were trained on the ARPA North American Business News (NAB) corpus comprising 240 million words of newspaper texts. All tests used a 64k vocabulary. To evaluate the adaptive models we calculated the test set perplexities (PP) and determined word error rates (WER) using acoustic material which was available for about 2 000 words from each of the test sets.

4.1 Domain Adaptation

We now investigate the power of the new adaptation technique and compare it to other known methods. Here, the task is to derive one fixed adapted model $p_{\text{adap}}(w\,|\,h)$, given the well–trained n–gram $p_{\text{back}}(w\,|\,h)$ and an adaptation set. Our marginal distributions $p_{\text{adap}}(w)$ were estimated on the respective adaptation material using a standard backing–off scheme for unseen events.

First of all, various values of β were tested to reduce both perplexity (Fig. 1) and word error rate (Fig. 2). The perplexity has a clear minimum around $\beta = 0.5$ whereas the WER is not as simple to interprete, since the error counts show irregular fluctuations as β is varied. Considering both test cases, a value of $\beta \approx 0.5$ appears to be a suitable choice. The presented figures were calculated for bigram models, but trigrams lead to similar conclusions (see also Table 1 and 2).

Next, we investigated the influence of the type of normalization used in (9) and (12). The perplexity and WER reductions shown in Table 1 reveal that both methods perform almost identically. All other tests in this paper were thus done using the much faster version (12). Fill–up is a different method for domain adaptation [1]. On *Kennedy*, fill–up is not as good as our method with the recommended $\beta = 0.5$, but on *Korea* there are no significant differences. Finally, a bigram trained on the rather limited adaptation data is obviously not a competitive adaptation technique, since it is even worse than the background model. For the trigram, we obtain a significant WER reduction on *Kennedy* (10% relative) and a degradation (5% relative) on *Korea*, see Table 2. (Perplexities are reduced for both domains.) The overall performance for $\beta = 0.5$ is very encouraging.

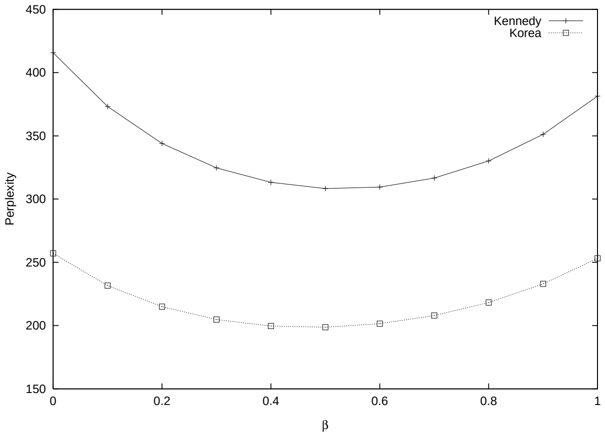

Figure 1: Perplexity versus β. Note that $\beta = 0$ corresponds to the non adapted background model. $\beta = 1$ represents the GIS "standard" version.

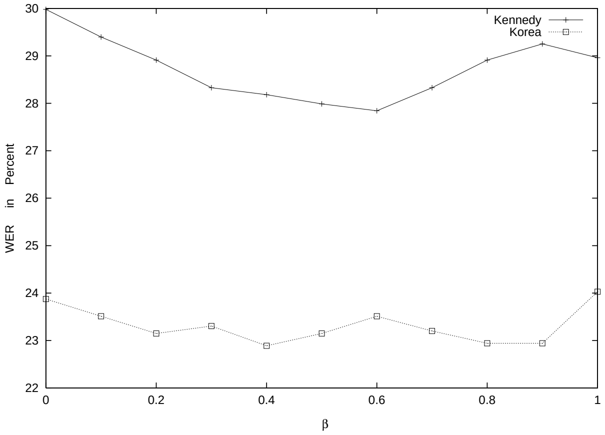

Figure 2: Word error rate versus β.

Bigrams		*Kennedy*		*Korea*	
		PP	WER	PP	WER
Background model		416	30.0%	257	23.9%
(9)	$\beta = 0.5$	305	27.8%	199	23.5%
	$\beta = 1.0$	389	29.1%	259	24.2%
(12)	$\beta = 0.5$	308	28.0%	199	23.1%
	$\beta = 1.0$	381	29.0%	253	24.0%
Fill–up		373	29.3%	229	23.4%
Bigram on adap.		894	38.3%	623	31.8%

Table 1: Domain adaptation of bigram models.

Trigrams		*Kennedy*		*Korea*	
		PP	WER	PP	WER
Background model		266	28.6%	164	19.8%
(12)	$\beta = 0.5$	214	25.8%	134	20.7%
	$\beta = 1.0$	281	26.6%	176	21.0%

Table 2: Domain adaptation of trigram models.

4.2 Online Adaptation

In a second experimental setup we ignored the adaptation sets. Instead, we tested the use of our method for online adaptation to a previously unknown domain. Here, the marginals $p_{\mathrm{adap}}(w)$ were estimated as a linear combination of the dynamically updated word frequencies within the local cache, $p_{\mathrm{cache}}(w)$, and the background unigram, $p_{\mathrm{back}}(w)$:

$$p_{\mathrm{adap}}(w) = \lambda \cdot p_{\mathrm{cache}}(w) + (1 - \lambda) \cdot p_{\mathrm{back}}(w) \qquad (15)$$

The cache itself was updated after every sentence. (In supervised adaptation the recognized sentences are corrected before the updates. Unsupervised adaptation uses the "raw" recognition results.) The interpolation parameter λ can be adjusted to minimize the $p_{\mathrm{adap}}(w)$ unigram perplexity of any sample corpus. If we do not know anything about the domain to be recognized we may estimate λ using the broad mixture of NAB texts. This yields $\lambda = 0.2$. If instead we take the NAB–untypical adaptation texts for *Kennedy* and *Korea* as prior knowledge the estimate shifts to $\lambda = 0.5$.

For comparison we also evaluated a standard cache model:

$$p_{\mathrm{adap}}(w \mid h) = \lambda \cdot p_{\mathrm{cache}}(w) + (1 - \lambda) \cdot p_{\mathrm{back}}(w \mid h) \qquad (16)$$

Our results are summarized in Table 3 and 4. We always observe reductions in perplexity, but only small changes in the WER for supervised adaptation and even an increase in WER for unsupervised adaptation. Again, *Kennedy* seems to be more sensitive to adaptation than *Korea*.

Kennedy		PP	WER	
			sup.	unsup.
Background model		416	30.0%	
Standard cache model		336	30.5%	31.3%
$\lambda = 0.2$	$\beta = 0.5$	334	28.8%	29.6%
	$\beta = 1.0$	298	29.0%	29.9%
$\lambda = 0.5$	$\beta = 0.5$	314	28.6%	30.0%
	$\beta = 1.0$	296	29.4%	31.5%

Table 3: Bigram perplexities and WER for supervised and unsupervised online adaptation.

Korea		PP	WER	
			sup.	unsup.
Background model		257	23.9%	
Standard cache model		219	23.6%	23.9%
$\lambda = 0.2$	$\beta = 0.5$	211	23.8%	24.4%
	$\beta = 1.0$	188	23.9%	24.5%
$\lambda = 0.5$	$\beta = 0.5$	197	23.9%	24.6%
	$\beta = 1.0$	188	23.7%	24.5%

Table 4: Like Table 3 but for the domain *Korea*

5 Conclusion

In this paper we proposed a new method to *quickly modify* a given language model in order to adapt to a special topic or speaker. Contrary to a linear interpolation between the static conditional model and the dynamically estimated unigram we minimally distort the former such that it respects the marginal distribution given by the local unigram. For domain adaptation we can achieve improvements both in perplexity and in word error rate. For online adaptation the changes are not very significant. Future experiments will include semantic word classes to improve the quick and robust estimation of the adaptive unigrams.

References

[1] S. Besling and H.-G. Meier. Language model speaker adaptation. In *Proc. EUROSPEECH*, pages 1755–1758, Madrid, Spain, Sep. 1995.

[2] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, 1991.

[3] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log linear models. *Annals Math. Stat.*, 43(5):1470–1480, 1972.

[4] S. A. Della Pietra, V. J. Della Pietra, R. L. Mercer, and S. Roukos. Adaptive language modeling using minimum discriminant estimation. *Proceedings of the ICASSP*, pages I633–I636, 1992.

[5] R. Kneser and J. Peters. Semantic clustering for adaptive language modeling. *Proceedings of the ICASSP*, 2:779–782, April 1997.

[6] F. Kubala. Design of the 1994 CSR Benchmark Tests. *Proc. Spoken Language Systems Technology Workshop*, pages 41–46, Jan. 1995.

[7] R. Kuhn and R. de Mori. A cache-based natural language model for speech recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12:570–583, June 1990.

[8] R. Rosenfeld. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach.* PhD thesis, School of Computer Science, CMU, 1994.

[9] V. Steinbiss, H. Ney, U. Essen, B.-H. Tran, X. Aubert, C. Dugast, H.-G. Meier, M. Oerder, R. Haeb-Umbach, D. Geller, W. Höllerbauer, and H. Bartosik. Continuous speech dictation — from theory to practice. In *Speech Communication*, volume 17(1-2), pages 19–38, Amsterdam, North–Holland, Aug. 1995.