ON THE DYNAMIC ADAPTATION OF STOCHASTIC LANGUAGE MODELS

Reinhard Kneser, Volker Steinbiss

Philips GmbH Forschungslaboratorien, Aachen
P. O. Box 1980, 5100 Aachen, Germany

kneser@pfa.philips.de steinbiss@pfa.philips.de

ABSTRACT

This paper introduces a simple and general scheme for the adaptation of stochastic language models to changing text styles. For each word in the running text, the adapted model is a linear combination of specific models, the interpolation parameters being estimated on the preceding text passage. Experiments on an English 1.1-million word corpus show the validity of the approach. The adaptation method improves a bigram language model by 10% in terms of test-set perplexity.

1. INTRODUCTION

The power of stochastic language modelling for large-vocabulary speech recognition has been successfully demonstrated in several systems. While stochastic language models could principally model long-range dependences, they are usually designed to model short-range dependences only, as in the bigram or trigram model cases, due to training and storage problems [1]. Once trained, the language model is usually kept fixed during recognition.

A human listener, in contrast, conditions his expectation of what will be said on a long-ranging history of what has been said. We postulate that there are properties like discourse domain (politics or art), style (newspaper text or database queries), type of vocabulary (general or technical) etc. that vary only slightly within text passages but that can be very different for different types of text. We propose to capture these properties within several specific stochastic n-gram language models. These specific models are linearly interpolated using adaptive interpolation parameters that reflect the type of the preceding part of the text. The adaptive interpolation parameters are chosen to be the maximum-likelihood estimates obtained from the text passage preceding the word under consideration.

Only little has yet been published on language-model adaptation. Kuhn and de Mori adapt a stochastic language model using a cache method which raises the probability of words observed in the preceding text passage [2][3]. The adaptation of an n-gram language model based on a minimum relative entropy criterion is described in [4]. Hattori presented an approach similar to ours for speaker adaptation [5].

2. ADAPTATION SCHEME

We start with K language models, each of them trained beforehand on a different text category such as newspaper text, scientific writing, etc. For simplicity of notation let us assume bigram models (as in our experiments). For the k-th bigram model, $P_k(w_n|w_{n-1})$ is the conditional probability of observing word w_n after word w_{n-1}. Each of the K specific models contributes to the interpolated model via

$$P(w_n|w_{n-1}) = \sum_k \lambda_k P_k(w_n|w_{n-1}) \qquad (1)$$

where λ_k can be interpreted as the probability that it is language model k which produces the current word. Particularly, $0 \leq \lambda_k \leq 1$ and $\sum_k \lambda_k = 1$.

The basic idea now is to dynamically adapt the parameter vector $\underline{\lambda} = (\lambda_1, \ldots, \lambda_K)$, which is used for the prediction of each new word w_n, conditioned on the previously observed L words. The model assumption is that $\underline{\lambda}$ should maximize the likelihood of the preceding words $w_{n-L}, \ldots, w_{n-1}$. The maximum-likelihood estimate is calculated with a few iterations of the forward-backward algorithm [6].

As initial values for $\underline{\lambda}$ at position n, we start at the beginning of the text with equal probabilities and later

0-7803-0946-4/93 $3.00 © 1993 IEEE

Table 1: Test-set perplexities of adaptive models compared to baseline models.

	Method	Without Cache	With Cache
(a)	Baseline model (cf. 3.1)	532.1	410.9
(b)	$\underline{\lambda}$ fixed for each category, optimized on test set	487.6	387.7
(c)	Additionally smoothed specific models on their text categories (cf. 3.3)	505.9	399.0
(d)	Adapted (cf. 3.2)	480.7	384.0
(e)	Adapted smoothed models (cf. 3.3)	482.0	384.2
(f)	Hard Decision (cf. 3.4)	494.2	394.0

on with the estimate for position $n-1$. The iteration step from λ_k to $\widehat{\lambda_k}$ is

$$\widehat{\lambda_k} = \frac{1}{L} \sum_{m=n-L}^{n-1} \frac{\lambda_k P_k(w_m|w_{m-1})}{\sum_i \lambda_i P_i(w_m|w_{m-1})}. \quad (2)$$

3. EXPERIMENTAL RESULTS

3.1 Corpus and Baseline Model

Experiments were performed on an English text corpus comprising 1.1 million words, which consists of 15 different text categories like newspapers, scientific texts, fiction, etc. We divided each of the text categories into a training (3/4) and a test (1/4) portion and then trained 15 specific word bigram language models on the respective training portions. Each of these models is interpolated with unigram and zerogram models, all of them estimated on the respective portions. In the same way an additional general model was trained on all training portions as a whole. With a perplexity of 532.1, the latter serves as a baseline model (left column of *Table 1a*).

We ran an experiment in order to get an idea of how specific the $K = 15 + 1$ language models are. For each text category we chose an optimal parameter vector $\underline{\lambda}^{(k)}$. The columns in *Table 3* show the parameter vectors $\underline{\lambda}^{(k)}$ which differ very strongly among the different text categories. For text category k, the k-th component λ_k has a high value. In almost all cases, the largest value is given to the general model, which has been trained with more data and thus is smoother than each of the specific models. This indicates that the special models are undertrained. We evaluated the interpolated language model (eq. (1)) which uses the constant parameter vector $\underline{\lambda}^{(k)}$ for test portion k (*Table 1b*). As the $\underline{\lambda}^{(k)}$ were estimated on the test portion, the perplexity of 487.6 gives a lower bound on what could be achieved by linearly interpolating the language models, assuming the category of the text under consideration was fixed and known.

3.2 Adaptation Experiment

In the adaptation experiment, the parameter vector $\underline{\lambda}$ was re-estimated for each new word using eq. (2). The optimal window length L, which should clearly be smaller than the typical length of a homogeneous text passage, was determined in a series of experiments (*Table 2*). The results for $L = 400$ (*Table 1d*) show an improvement of 10% over the baseline model. This adaptive model with varying parameter vectors $\underline{\lambda}$ is even better than the one of *Table 1b*, where the parameters were chosen optimally but fixed for each text category. This indicates that also variations within a text category are captured by the adaptation.

3.3 Additional Smoothing of Specific Models Prior to Adaptation

In the experiment above, the specific language models P_j are trained on the respective parts of the whole corpus such that the adaptation process also has to do some smoothing. We ran an experiment where additional smoothing was performed on the specific models prior to adaptation: each $\tilde{P}_k$ is a linear combination

$$\tilde{P}_k(w_n|w_{n-1}) = \sum_j \alpha_{kj} P_j(w_n|w_{n-1}) \quad (3)$$

For each k, we determined the interpolation parameters α_{kj} to be optimal on the training portion of the respective category, using a leaving–one–out method [7]. With these additionally smoothed specific models, we repeated the adaptation experiment of the previous section, obtaining about the same performance (*Table 1e*). The optimal window length (*Table 2*) is shorter.

Fig. 1 shows the interpolation parameters over the varying kinds of text categories 1, ..., 15. There is fluctuation within the $\underline{\lambda}$ but for texts of the same category on the x-axis, certain characteristics can be observed. For each piece of text, all but a few components λ_k are almost 0. One observes that some categories behave similarly, e.g. several kinds of fiction (10, ..., 15), while others, like scientific writings (category 9), are more specific.

Table 2: Test set perplexities for different values of the window length L.
a) Standard specific models (cf. 3.2)
b) Additional smoothing of the specific models (cf. 3.3)

Length	(a)	(b)
25	533.8	492.8
50	504.2	487.1
100	488.6	483.6
200	482.1	482.0
300	481.3	482.2
400	480.7	482.4
500	481.0	482.6
750	481.4	
1000	482.3	
2000	485.5	
5000	489.1	
10000	492.8	
50000	511.2	

3.4 Hard Decision on the Specific Models

The methods described above can be interpreted as making soft decisions about which specific language model produces the next word. As an alternative, one could make a hard decision [8]: take the specific language model that most likely produced the preceding L words. In this case we start with the smoothed specific models from the subsection above. The performance gain is slightly less with this simple method (*Table 1f*) but better than taking the respective specific model for each text category (*Table 1c*).

3.5 Cache Model Added

In order to check whether the improvements were not only due to effects that could be modelled by a cache language model, we ran a control experiment adding an additional cache component to our specific models. The right column of *Table 1* shows that our baseline cache model is still improved by the adaptation method and that the gain is by about one third less in this case.

4. DISCUSSION

The corpus used in our experiments is relatively small (1.1 million words). So, the "specific" language models are not well trained and, due to the large amount of smoothing required, they are not as specific as we would like them to be. However, the results give a clear indication of the validity of the approach.

The adaptation scheme is general and should work for a wider range of language models (e. g. n-gram models, category-based models). It performs well in identifying different text categories but also follows smaller deviations in the running text. Such an adaptive language model could serve as a universal language model covering several applications or topics.

REFERENCES

[1] L. R. Bahl, F. Jelinek, R. L. Mercer: "A Maximum Likelihood Approach to Continuous Speech Recognition", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 5, pp. 179-190, March 1983.

[2] R. Kuhn, R. de Mori: "A Cache-Based Natural Language Model for Speech Recognition", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 12, pp. 570-583, June 1990.

[3] R. Kuhn, R. de Mori: "Corrections to 'A Cache-Based Natural Language Model for Speech Recognition'", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 14, pp. 691-692, June 1992.

[4] S. Della Pietra, V. Della Pietra, R. L. Mercer, S. Roukos: "Adaptive Language Modeling Using Minimum Discriminant Estimation", Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, San Francisco, Ca., pp. I633-I636, March 1992.

[5] H. Hattori: "Speaker Adaptation Based on Markov Modeling of Speakers in Speaker-Independent Speech Recognition", Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Toronto, Canada, pp. 845-848, May 1991.

[6] F. Jelinek, R. L. Mercer, "Interpolation and Estimation of Markov Source Parameters from Sparse Data", *Pattern Recognition in Practice*, E. S. Gelsema, L. N. Kanal (eds.), North-Holland Publishing Comp., 1980.

[7] H. Ney, U. Essen: "On Smoothing Techniques for Bigram-Based Natural Language Modelling", Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Toronto, Canada, pp. 825-828, May 1991.

[8] H. Ney, Personal Communication, Philips Research Laboratories, Aachen, Nov. 1992.

Table 3: Optimal parameter vectors $\underline{\lambda}$ for the 15 different text categories of the text corpus.

Category	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
λ_1	**.29**	.09	.06	.01	.06	.02	.02	.09	.00	.00	.00	.00	.00	.00	.01
λ_2	.01	**.16**	.01	.08	.06	.01	.13	.07	.01	.00	.00	.00	.00	.00	.01
λ_3	.03	.00	**.24**	.00	.00	.06	.02	.00	.07	.00	.00	.01	.00	.00	.02
λ_4	.00	.00	.00	**.17**	.00	.01	.02	.00	.03	.00	.00	.01	.00	.00	.01
λ_5	.03	.03	.05	.00	**.18**	.16	.01	.06	.10	.00	.00	.02	.01	.00	.05
λ_6	.03	.01	.02	.02	.02	**.05**	.02	.01	.01	.02	.00	.06	.02	.01	.05
λ_7	.03	.01	.09	.05	.01	.08	**.09**	.03	.05	.05	.02	.02	.01	.00	.15
λ_8	.03	.06	.00	.00	.08	.03	.05	**.24**	.02	.00	.00	.00	.00	.00	.00
λ_9	.01	.02	.02	.02	.14	.05	.13	.07	**.45**	.00	.00	.00	.00	.00	.00
λ_{10}	.00	.00	.00	.00	.00	.00	.00	.00	.00	**.12**	.07	.06	.11	.08	.04
λ_{11}	.00	.00	.00	.00	.00	.00	.00	.00	.00	.08	**.13**	.04	.10	.10	.02
λ_{12}	.01	.00	.00	.00	.00	.00	.00	.00	.00	.02	.00	**.18**	.05	.01	.02
λ_{13}	.00	.00	.00	.00	.00	.00	.00	.00	.00	.08	.12	.05	**.23**	.17	.03
λ_{14}	.00	.00	.00	.00	.00	.00	.00	.00	.00	.11	.13	.06	.11	**.27**	.01
λ_{15}	.01	.00	.00	.00	.00	.01	.00	.00	.01	.01	.01	.01	.02	.00	**.03**
$\lambda_{general}$	**.51**	**.61**	**.51**	**.64**	**.44**	**.52**	**.51**	**.43**	**.24**	**.50**	**.50**	**.47**	**.35**	**.35**	**.56**

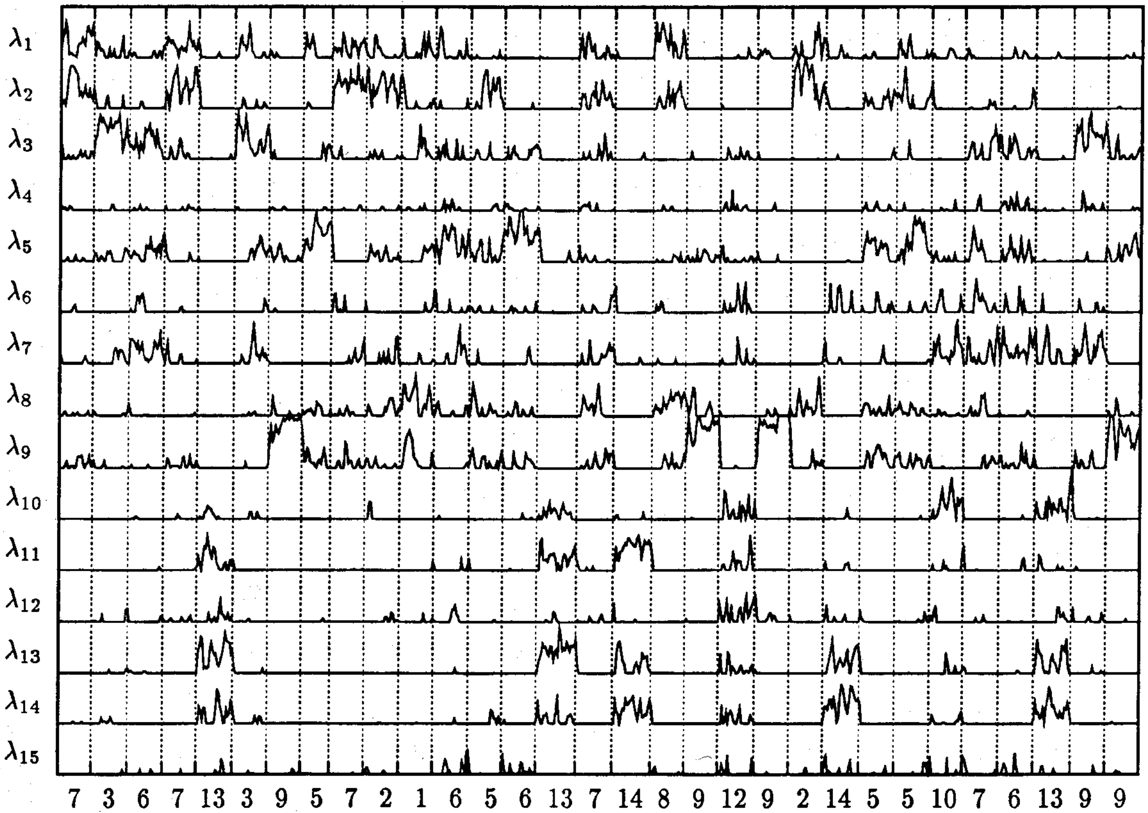

Fig. 1: Interpolation parameters λ_k over running text from various categories (cf. 3.3). The x-axis corresponds to about 70,000 words in portions of about 2000 words from text categories as labelled. On the y-axis, the 15 interpolation parameters λ_k are shown (with values between 0 and 1).