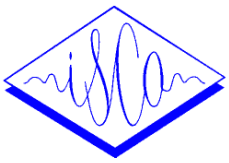

4th European Conference on
Speech Communication and Technology
EUROSPEECH '95
Madrid, Spain, September 18-21, 1995

EXTENSIONS OF ABSOLUTE DISCOUNTING FOR LANGUAGE MODELING

M. Generet, H. Ney, F. Wessel

Lehrstuhl für Informatik VI, RWTH Aachen – University of Technology,
D-52056 Aachen, Germany

ABSTRACT

In this paper, we extend the absolute discounting technique along various directions. To estimate the backing-off distribution, we use m-gram singletons, i.e. m-grams that were seen exactly once in the training data. This method is applied in addition to the usual estimation of discounting parameters. The improvement in perplexity is typically between 8% and 12%. We also investigate a cache model. In experimental tests on a large text corpus, the cache model improved the perplexity by up to 28%. The experimental evaluations were carried out on a set of 38 million words from the Wall Street Journal task. We compare our results with the results reported by CMU.

1. INTRODUCTION

Recently, the use of the absolute discounting method [5] has produced good results on the 1-million LOB corpus, which is a small corpus by today's standards. In this paper, we extend this technique along the following directions:

- use of leaving-one-out training to estimate the backing-off distribution (singleton backing-off) in addition to the parameters of absolute discounting;
- combination with a cache model;
- experimental tests on a large text corpus;
- comparison with results reported by Rosenfeld of CMU [7].

The experimental evaluations were carried out on a set of 38 million words from the Wall Street Journal task (WSJ).

2. BASIC METHODS

A stochastic language model for a word sequence $w_1^N = w_1 \ldots w_N$ can be written in the form:

$$Pr(w_1^N) = \prod_{n=1}^{N} Pr(w_n|w_1^{n-1}) \quad .$$

The conditional probability $Pr(w_n|w_1^{n-1})$ is calculated by an m-gram language model. To present the general framework, we denote the history under consideration by the symbol h and its generalization by $\overline{h}$. E.g., for a trigram model, a history h is defined by the two predecessor words (u, v), and its generalized history $\overline{h}$ is given by all word pairs ending in v. Even in a large training corpus, many of the possible histories never occur, and it is necessary to smooth the history-specific model with a more general model.

2.1. Absolute Discounting

Like most other smoothing techniques, absolute discounting is formulated in terms of the counts $N(h, w) > 0$ of the joint event (h, w). The symbol h denotes the given history, i.e. typically one or two predecessor words depending on the use of a bigram or trigram model. The basic idea is to leave the high counts virtually unchanged. In absolute discounting the counts are reduced by a constant b_h that may depend on the history h:

$$p(w|h) = \begin{cases} \dfrac{N(h,w) - b_h}{N(h,\cdot)} & \text{if } N(h,w) > 0 \\ b_h \cdot \dfrac{W - N_0(h,\cdot)}{N(h,\cdot)} \cdot \dfrac{\beta(w|\overline{h})}{\sum\limits_{w'} \beta(w'|\overline{h})} & \text{if } N(h,w) = 0 \end{cases}$$

with $\beta(w|\overline{h})$ representing the distribution for the generalized history $\overline{h}$ and W the vocabulary size.
Here and in the following, we use the notation:

$N(h, w)$: count of joint event (h, w);
$N(h, \cdot) = \sum_w N(h, w)$;
$N_r(h, w)$: count of event pairs (h, w) that occur exactly r times in the training data;
$N_r(h, \cdot) = \sum_w N_r(h, w)$.

The backing-off method as described above is convenient for analytic calculations. In the implementation, it is sometimes useful to apply a slight modification of the above model which is referred to as an interpolation with absolute discounting [5]:

$$p(w|h) = \frac{\max[N(h,w) - b_h, 0]}{N(h,\cdot)} + b_h \cdot \frac{W - N_0(h,\cdot)}{N(h,\cdot)} \cdot \beta(w|\overline{h})$$

with $\sum_w \beta(w|\overline{h}) = 1$.

Typically in the experiments, the difference between backing-off and interpolation is negligible.

2.2. Parameter Estimation

We have to estimate the discounting parameters b_h and the backing-off distribution $\beta(w|\overline{h})$. The discounting parameters b_h can be approximated by

$$b_h \cong \frac{N_1(h,\cdot)}{N_1(h,\cdot) + 2 \cdot N_2(h,\cdot)} \quad .$$

To reduce the number of the discounting parameters, we typically pool these parameters over all histories h.

Apart from smoothing, the backing-off distribution $\beta(w|\overline{h})$ was basically estimated by the relative frequencies [5]:

$$\beta(w|\overline{h}) = \frac{N(\overline{h},w)}{N(\overline{h},\cdot)} \quad .$$

However, strictly speaking, the backing-off distribution should be estimated by leaving-one-out. In a first approximation, it can be shown [2] that the resulting estimate is:

$$\beta(w|\overline{h}) \cong \frac{N_1(\overline{h},w)}{N_1(\overline{h},\cdot)} \quad ,$$

where $N_1(\overline{h},w)$ is computed over the joint events (h,w) that were seen exactly once in the training data. This type of distribution will be referred to as singleton distribution. The interesting property is that strongly correlated word pairs are omitted from the estimation of the backing-off distribution.

2.3. The Cache Model

In a cache model [3], the probability of the most recent M words is increased. A typical value of M is between 100 and 1000. The most simple case of a cache is the unigram cache model. The unigram cache probability for a word w_n at the position n is calculated as:

$$p_c(w_n|w_{n-1}^{n-m}) = \frac{1}{M}\sum_{m=1}^{M}\delta(w_n,w_{n-m}) \quad ,$$

where the Kronecker function $\delta(x,y)$ is 1 if $x=y$ and 0 otherwise.

In the Wall Street Journal task, the text consists of newspaper articles, the boundaries of which are known. Therefore the cache is initialized to zero at each article boundary, which requires a modification of the formula. In general, the cache is interpolated with an m-gram model.

3. FULL TRIGRAM MODEL

The backing-off distribution $\beta(w|\overline{h})$ described in Section 2.2 is not sufficient, because for the generalized history $\overline{h}$ some events do not occur. Therefore, we have to construct a hierarchical language model.

3.1. The m-Gram Hierarchy

We formulate the full trigram model for the case of backing-off distribution using trigram singletons. For this purpose, we define the following types of counts:

$N(u,v,w)$: count of trigram (u,v,w);

$N(u,v,\cdot)$: count of bigram (u,v);

$N_1(\cdot,v,w)$: count of trigram singletons ending with (v,w);

$N_1(\cdot,v,\cdot)$: count of trigram singletons having v in the middle;

$N_1(\cdot,\cdot,\cdot)$: count of all trigram singletons;

$N_1^0(\cdot,v,\cdot)$: count of unseen events $(v,\cdot)$ in the set of trigram singletons;

$N_1(\cdot,w)$: count of bigram singletons ending with w in the set of trigram singletons;

$N_1(\cdot,\cdot)$: count of all bigram singletons in the set of trigram singletons;

$N_1^0(\cdot,\cdot)$: count of unseen events $(\cdot)$ in the set of the bigrams belonging to the trigram singletons.

The count $N_1(\cdot,v,w)$ is equivalent to the count of bigrams beginning with v in the set of the trigram singletons. Using this notation, we have the following trigram model:

$$\begin{aligned}
p_{tri}(w|uv) &= \frac{\max[N(u,v,w) - b_{tri}, 0]}{N(u,v,\cdot)} \\
&+ b_{tri} \cdot \frac{W - N(u,v,\cdot)}{N(u,v,\cdot)} \cdot p_{bi}(w|v) \\
p_{bi}(w|v) &= \frac{\max[N_1(\cdot,v,w) - b_{bi}, 0]}{N_1(\cdot,v,\cdot)} \\
&+ b_{bi} \cdot \frac{W - N_1^0(\cdot,v,\cdot)}{N_1(\cdot,v,\cdot)} \cdot p_{uni}(w) \\
p_{uni}(w) &= \frac{\max[N_1(\cdot,w) - b_{uni}, 0]}{N_1(\cdot,\cdot)} \\
&+ b_{uni} \cdot \frac{W - N_1^0(\cdot,\cdot)}{N_1(\cdot,\cdot)} \cdot \frac{1}{W} \quad .
\end{aligned}$$

3.2. The Cache Component

In the cache model the trigram model is linearly interpolated with the cache component. The formula is:

$$\begin{aligned}
p(w_n|w_{n-M}^{n-1}) &= (1-\lambda_c) \cdot p_{tri}(w_n|w_{n-2},w_{n-1}) \\
&+ \lambda_c \cdot p_c(w_n|w_{n-M}^{n-1}) \quad .
\end{aligned}$$

As in [1] and [7], we consider two types of cache models, namely unigram cache and bigram cache. For the unigram cache, we found that the combination with a floor value improves the perplexity. This variant of the unigram cache is defined as follows:

$$p_c^{uni}(w_n|w_{n-M}^{n-1}) = \begin{cases} \dfrac{N_c(w_n) - d_c^{uni}}{M} & \text{if } N_c(w_n) > 0 \\ d_c^{uni} \cdot \dfrac{(W - N_{c_0}(\cdot))}{M} \cdot \dfrac{1}{N_{c_0}(\cdot)} & \text{if } N_c(w_n) = 0 \end{cases}$$

with: d_c^{uni} the discounting parameter,

$$N_c(w_n) = \sum_{m=1}^{M}\delta(w_n,w_{n-m})$$

$N_{c_0}(\cdot)$ the number of words not seen in the cache.

The bigram cache has to be combined with the unigram cache. We use two methods A and B.
Method A is based on absolute discounting:

$$p_c(w_n|w_{n-M}^{n-1}) = \frac{\max[N_c(w_{n-1},w_n) - d_c^{bi}, 0]}{N_c(w_n)} + d_c^{bi} \cdot \frac{(W - N_{c_0}(w_{n-1},\cdot))}{N_c(w_n)} \cdot p_c^{uni}(w_n|w_{n-M}^{n-1}) .$$

Here, $N_c(w_{n-1},w_n)$ is the bigram count for the cache history and $N_{c_0}(w_{n-1},\cdot)$ is the count of unseen events $(w_{n-1},\cdot)$ for the M most recent words.
Method B is a linear interpolation of the bigram cache and the unigram cache with an additional interpolation parameter.

4. RESULTS

4.1. Corpus

The above described methods were tested on the WSJ task [6]. The vocabulary was ARPA's official '20o.nvp' (20 000 most common WSJ words including the unknown word, non-verbalized punctuation). In our tests, we used exactly the same conditions as reported by Rosenfeld [7] so that a direct comparison of the perplexities is possible. There were three trainings sets of different sizes: 1 million, 5 million and 38 million words. Table 1 gives a detailed overview of the database. For the development data, we used the development text files for WSJ0 from 1989. The development data was used to adjust the discounting and interpolation parameters. Each of the different language models was tested on the same test set of 325 000 words. The test data was never used for the language model training.

4.2. Experimental Results

The performance of the difference methods was evaluated in several experiments. We compare the results with an interpolated absolute discounting method without singletons. In contrast to Rosenfeld [7], we did not omit the trigram singletons from the training data.

For each type of the language model, the following parameters have to be estimated:

- the discounting parameters for the m-gram model b_{tri}, b_{bi} and b_{uni};
- the discounting parameters for the cache model d_c^{uni} and d_c^{bi};
- the interpolation parameter λ_c for the combination of the trigram model and the cache model.

Table 1: Database for training, development and test (WSJ task).

	words	sentences	articles
1 MW	972 868	41 156	2 179
5 MW	4 513 716	189 678	10 018
38 MW	38 532 517	1 611 572	79 452
development	76 646	3 204	165
test	324 655	13 542	757

Table 2: Counts of the different m-grams.

	corpus	diff. events	singletons
unigram	1 MW	17 456	2 296
	5 MW	19 730	231
	38 MW	19 983	0
bigram	1 MW	303 858	211 105
	5 MW	881 263	566 093
	38 MW	3 500 636	2 046 462
trigram	1 MW	648 482	556 185
	5 MW	2 420 168	1 990 507
	38 MW	14 096 109	10 907 373

The different parameters were estimated by leaving-one-out on the training data or cross validation on the development data. The choice of the discounting parameter b_{uni} had no effect on the perplexity. We carried out systematic tests for each discounting parameter b_{tri}, b_{bi}, d_c^{uni} and d_c^{bi} on the development data. These tests showed that for each of these parameters there was an extremely flat minimum of the perplexity. As an example, Fig. 1 shows the perplexity as a function of the discounting parameter b_{tri} for the 5 million words of training data. The perplexity was measured as a function of b_{tri} for three different methods: with and without leaving-one-out on the 5 MW training data and in a cross validation fashion on the development data. We compared history-specific discounting parameters with history independent ones. For the history independent case, we also compared leaving-one-out estimates and cross-validation estimates. As the experiments shown in Table 3 indicate, there is no advantage in having a history-specific parameter b_{tri}.

Table 4 summarizes the perplexities for the different types of language models studied in this paper. For the bi-/unigram cache, we used method A to combine the bigram and the unigram cache. In addition to the perplexity measurements, Table 4 shows that there is

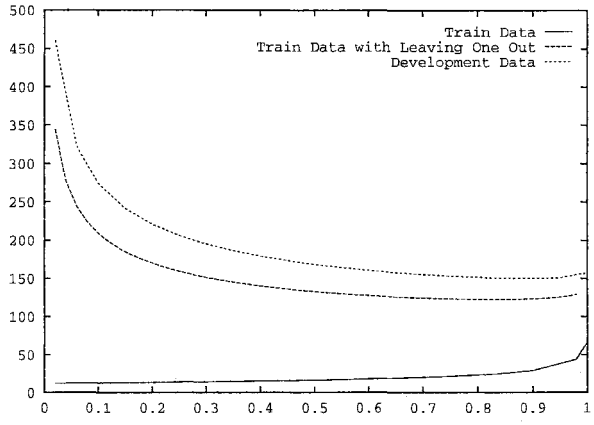

Figure 1: Perplexity as a function of b_{tri} for the 5-MW train data and the development data.

Table 3: Perplexities of the trigram model on the test data for different estimates of discounting parameter b_{tri}.

	leaving-one-out		dev. data
	hist. dep.	hist. indep.	hist. indep.
1 MW	231.2	224.3	221.6
5 MW	156.2	151.7	150.1
38 MW	98.7	96.7	96.5

a strong effect of the training set size. The difference between the perplexities of the trigram and the bigram language model increases with the size of training corpus (1 MW: 13%, 5 MW: 23% and 38 MW: 38%). The best results were obtained for the combination of the trigram model with singleton backing-off and the bi-/unigram cache (method A). The singleton model reduced the perplexity by 8% to 12% depending on the size of the training corpus. The bi-/unigram cache model improved the perplexity of the baseline trigram model by up to 28%.

In addition to our perplexity results, Table 4 shows the results obtained by Rosenfeld at CMU [7]. Comparing his results with our results, we see:

- As to the baseline methods, absolute discounting is slightly better than Katz's backing-off, in particular when the training corpus is small.
- When combining all methods, the CMU results are better than ours. We attribute this difference to the use of maximum entropy in connection with long-distance bigrams and trigrams [7].

Table 5 gives the results for the four types of cache models tested. In each case, the cache model was combined with the trigram/singleton language model. The experimental results showed only a small difference (2%) between the two combination methods, namely linear interpolation and absolute discounting, of the cache models.

Table 4: Perplexities for different language models on the test data.

size of training corpus	1 Mio	5 Mio	38 Mio
absolute discounting and interpolation			
bigram (no singletons)	288	217	168
trigram	250	167	104
+ singleton	222	150	96
+ bi-/unigram cache	178	127	86
+ singleton + bi-/unigram cache	170	122	84
CMU results			
Katz's trigram model	269	173	105
+ bi-/unigram cache	193	133	88
+ bi-/unigram cache + maximum entropy	163	108	71

Table 5: Comparison of perplexities (PP) for different cache models (5 MW train data).

model	PP
unigram cache without floor	128.8
unigram cache	126.4
bi-/unigram cache:	
method A: absolute discounting	121.7
method B: linear interpolation	123.6

5. SUMMARY

In this paper we have examined the absolute discounting methods. The conventional trigram language model was improved by incorporating a so called singleton backing-off distribution. The perplexity improvements obtained by the singleton method varied between 8% and 12%, depending on the size of the training corpus. Furthermore, we tested the cache model at the level of both unigrams and bigrams. The combination of the cache model with a trigram model resulted in an improvement of up to 28%.

REFERENCES

1. F. Jelinek, B. Marialdo, S. Roukos, M. Strauss: "A Dynamic Language Model for Speech Recognition", DARPA Workshop on Speech and Natural Language, pp. 293-295, Pacific Groves, CA, February 1991.
2. R. Kneser, H. Ney: "Improved Backing-off for m-gram Language Modeling", IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 181-184, Detroit, MI, May 1995.
3. R. Kuhn, R. De Mori: "A Cache-Based Natural Language Model for Speech Recognition", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. PAMI-14, no. 6, pp. 691-692, June 1992.
4. H. Ney, U. Essen: "On Smoothing Techniques for Bigram-Based Natural Language Models", International Conference on Acoustic, Speech and Signal Processing, pp. 825-828, Toronto, May 1991.
5. H. Ney, U. Essen, R. Kneser: "On Structuring Probabilistic Dependences in Stochastic Language Modelling", Computer Speech and Language, Vol. 8, pp. 1-38, 1994.
6. D. Paul, J.M. Baker: "The Design for the Wall Street Journal-based CSR Corpus", DARPA Spoken Language Systems Workshop, February 1992.
7. R. Rosenfeld: "Adaptive Statistical Language Modeling: A Maximum Entropy Approach", School of Computer Science, Carnegie Mellon University, Ph. D. Thesis, Pittsburgh, PA, CMU-CS-94-138, 1994.