

# Statistical Language Modeling and Word Triggers

Christoph Tillmann      Hermann Ney

Lehrstuhl für Informatik VI, RWTH Aachen  
D-52056 Aachen, Germany

{tillmann, ney}@informatik.rwth-aachen.de

## ABSTRACT

This paper describes the use of word triggers in the context of statistical language modeling for speech recognition. It consists of two parts: First we describe the use of trigram models and smoothing in language modeling; smoothing techniques are necessary due to unseen events in training data. In the second part we consider the use of word triggers in language modeling to capture long-distance dependencies. The experimental results presented are based on the Wall Street Journal task.

## 1. INTRODUCTION

The need for a stochastic language model in speech recognition arises from Bayes' decision rule for minimum error rate [1]. The word sequence  $w_1 \dots w_N$  to be recognized from the sequence of acoustic observations  $x_1 \dots x_T$  is determined as that word sequence  $w_1 \dots w_N$  for which the posterior probability  $Pr(w_1 \dots w_N | x_1 \dots x_T)$  attains its maximum. This rule can be rewritten in the form:

$$\arg \max_{w_1 \dots w_N} \{Pr(w_1 \dots w_N) \cdot Pr(x_1 \dots x_T | w_1 \dots w_N)\} \quad , \quad (1)$$

where  $Pr(x_1 \dots x_T | w_1 \dots w_N)$  is the conditional probability of, given the word sequence  $w_1 \dots w_N$ , observing the sequence of acoustic measurements  $x_1 \dots x_T$  and where  $Pr(w_1 \dots w_N)$  is the prior probability of producing the word sequence  $w_1 \dots w_N$ .

The task of the stochastic language model is to provide estimates of these prior probabilities  $Pr(w_1 \dots w_N)$ . Using the definition of conditional probabilities, we obtain the decomposition:

$$Pr(w_1 \dots w_N) = \prod_{n=1}^N Pr(w_n | w_1 \dots w_{n-1}) \quad . \quad (2)$$

The organization of this paper is as follows. In Section 2, we review our baseline method for trigram modeling and smoothing; the smoothing is based on the leaving-one-out method. In Section 3, we study the word triggers to model long-distance dependencies. As we will see, we again apply some

sort of leaving-one-out concept to select good trigger pairs. In both sections, we present experimental results that were obtained on the Wall Street Journal corpus.

## 2. M-GRAM MODELING AND SMOOTHING

### 2.1. Bigram and Trigram Modeling

For large vocabulary speech recognition, these conditional probabilities are typically used in the following way [1]. The dependence of the conditional probability of observing a word  $w_n$  at a position  $n$  is assumed to be restricted to its immediate  $m$  predecessor words  $w_{n-m} \dots w_{n-1}$ . The resulting model is that of a Markov chain and is referred to as  $(m+1)$ -gram model. For  $m = 1$  and  $m = 2$ , we obtain the widely used bigram and trigram models, respectively. These bigram and trigram models are estimated from a text corpus during a training phase. But even for these restricted models, most of the possible events, i.e. word pairs and word triples, are never seen in training because there are so many of them. Therefore in order to allow for events not seen in training, the probability distributions obtained in these  $m$ -gram approaches are smoothed with more general distributions.

Strictly speaking, to evaluate the quality of a stochastic language model, we would have to run a whole recognition experiment. However, to a first approximation, we can separate the two types of probability distributions in Bayes' decision rule and confine ourselves to the probability that the language model produces for a sequence of (test or training) words  $w_n$ ,  $n = 1, \dots, N$ . To normalize this prior probability with respect to the number  $N$  of words, we take the inverse of the  $N$ -th root and obtain the so-called corpus perplexity [1]:

$$PP := [Pr(w_1 \dots w_N)]^{-1/N} \quad .$$

Inserting the decomposition into conditional probabilities of Eq. (2) and taking the logarithm, we obtain the so-called log-perplexity:

$$\log PP = -\frac{1}{N} \sum_{n=1}^N \log Pr(w_n | w_1 \dots w_{n-1}) \quad . \quad (3)$$

## 2.2. Smoothing Method

To obtain a non-zero probability mass for unseen events, we review the method of *absolute discounting*. The basic idea of the absolute discounting [6] model is to leave the high counts virtually unchanged. We consider a word bigram or word trigram and denote it by  $(h, w)$  where  $h$  stands for *history*.

The count  $r = N(h, w)$  with which the event  $(h, w)$  has been observed in the training data is likely to change in another set of training data of the same size  $N$ . However, we can expect the difference to be small. Typically we would expect to see values like  $r - 1, r, r + 1$ . To take this type of variability into account, we introduce an *average* and therefore *non-integer* count offset which is assumed to be independent of the count value  $N(h, w)$ . Observing the normalization constraint, we obtain the model of absolute discounting:

$$p(w|h) = \begin{cases} \frac{N(h, w) - b}{N(h)} & \text{if } N(h, w) > 0 \\ \frac{b \cdot [W - n_0(h)]}{N(h)} \frac{\beta(w|\bar{h})}{\sum_{w': N(h, w')=0} \beta(w'|\bar{h})} & \text{if } N(h, w) = 0. \end{cases} \quad (4)$$

Here we have used the following symbols:

$W$ : vocabulary size, typically 20000 words;

$N$ : size of the training corpus, typically several millions of words;

$N(h, w)$ : the number of observations of the event  $(h, w)$  in the training corpus;

$N(h)$ : the number of observations of the history  $h$  in the training corpus;

$b$ : the discounting parameter, which here is assumed to be history independent;

$n_0(h)$ : the number of words that were *not* observed as successor words of the history  $h$ ;

$\beta(w|\bar{h})$ : the more generalized distribution for a generalized history  $\bar{h}$ , to which we *back off* [5]. E.g. for a word trigram  $(u, v, w)$  with  $h = (u, v)$ , we typically use the level of word bigrams as generalized events, i.e.  $\bar{h} = v$ .

The discounting parameter  $b$  is determined by the leaving-one-out method in combination with maximum likelihood estimation. There is no closed-form

solution, but the following approximation is sufficiently close in most cases [5]:

$$b \cong \frac{n_1}{n_1 + 2n_2},$$

where  $n_1$  is the number of ‘singleton’ events, i.e. events  $(h, w)$  with  $N(h, w) = 1$  and  $n_2$  is the number of ‘doubleton’ events, i.e. events  $(h, w)$  with  $N(h, w) = 2$ .

**Interpolation.** The smoothing method presented above is based on *backing-off*, which amounts to a strict *choice* between a *specific* and a *generalized* probability distribution. An alternative is to *add* the two probabilities distributions, where of course the normalization constraint must be satisfied. The advantage of interpolation over backing-off is that the computationally expensive renormalization can be avoided. In the experiments (e. g. see [5]), we always found that in principle the perplexities are virtually not affected. Thus we have the following *interpolation* for absolute discounting using the discounting parameter  $b_I$ :

$$p(w|h) = \max \left\{ 0, \frac{N(h, w) - b_I}{N(h)} \right\} + b_I \cdot \frac{W - n_0(h)}{N(h)} \cdot \beta(w|\bar{h}). \quad (5)$$

**Multi-Level Smoothing.** When smoothing a trigram model with a bigram model, we have to keep in mind that the generalized distribution itself requires smoothing. Thus the bigram itself is smoothed by a unigram which again may be smoothed by a zero-gram model, i.e. a uniform distribution over the words of the vocabulary. Thus, we can define the following levels for a trigram event  $(u, v, w)$ :

- the trigram level  $(u, v, w)$ , which defines the relative trigram frequencies as the level to start with;
- the bigram level  $(v, w)$ ;
- the unigram level  $w$ ;
- the zero-gram level if the unigram estimates are unreliable.

**Cache Model.** A simple method to include a short-term memory into a language model is by means of the so-called *cache* [3]. In a cache model, the probability of the most recent  $M$  words is increased as compared to the position independent unigram probability. A typical value of  $M$  is between 100 and 1000. For the following the history  $h_n = w_{n-M}^{n-1}$  consists of the  $M$  predecessor words of

Table 1: Perplexity ( $PP$ ) results and word error rates (WER) for three types of language models.

| Language model         | PP    | WER[%] |
|------------------------|-------|--------|
| Bigram: without cache  | 198.1 | 16.5   |
| Trigram: without cache | 130.2 | 14.3   |
| Trigram: with cache    | 113.2 | 13.8   |

$w_n$ . The cache model  $p_C(w_n|w_{n-M}^{n-1})$  is defined as:

$$p_C(w_n|w_{n-M}^{n-1}) = \frac{1}{M} \sum_{m=1}^M \delta(w_n, w_{n-m}), \quad (6)$$

with  $\delta(w, v) = 1$  if and only if  $w = v$  and  $\delta(w, v) = 0$  otherwise. The cache model is combined with the baseline model through *linear interpolation* (see [2]). For the resulting extended model we get:

$$p_E(w_n|w_{n-M}^{n-1}) = \lambda \cdot p_{tri}(w_n|w_{n-2}, w_{n-1}) + (1 - \lambda) \cdot p_C(w_n|w_{n-M}^{n-1}),$$

where  $\lambda$  is the so-called interpolation parameter.

### 2.3. Experimental Results

The effect of the perplexity on the word error rate in speech recognition is well known and has been confirmed in a large number of recognition experiments. As a specific result of our baseline language model and our recognition system, we show the recognition results obtained in a recent recognition experiment on a speech corpus from the Wall Street Journal task [7]. Table 1 shows the effect that the different types of language models have on the word error rate in speech recognition. Three types of language models were tested: a bigram model without cache, a trigram without cache and a trigram model with cache. For the integration of the language model into the recognition process, see [7]. For each of the three types of language models, Table 1 reports the language model perplexity (on the test set) and the word error rate. As can be seen, the improvement of the perplexity goes hand in hand with a consistent reduction of the word error rate.

## 3. WORD TRIGGERS

### 3.1. Methods

In this section we describe how long-distance dependencies can be included into our language model by so-called “trigger pairs”. A trigger pair is a long distance word pair. We restrict ourselves to trigger pairs, where both the triggered and triggering events are single words. For a vocabulary of size  $V$ , there are  $V^2$  possible word trigger pairs. We present only the main results of our work with trigger pairs here, for further details see [10].

$$p_{ab}(w|h) = \begin{cases} q(b|a) & \text{if } a \in h \text{ and } w = b \\ [1 - q(b|a)] \cdot \frac{p(w|h)}{\sum_{w' \neq b} p(w'|h)} & \text{if } a \in h \text{ and } w \neq b \\ q(b|\bar{a}) & \text{if } a \notin h \text{ and } w = b \\ [1 - q(b|\bar{a})] \cdot \frac{p(w|h)}{\sum_{w' \neq b} p(w'|h)} & \text{if } a \notin h \text{ and } w \neq b \end{cases}, \quad (7)$$

The basic approach to selecting the best trigger pairs is as follows. We consider the possible word trigger pairs one by one and extend the baseline language model  $p(w|h)$  by the word trigger pair under consideration. Then for each word trigger pair, we compute the perplexity improvement on a training corpus and select the best trigger pairs. A specific trigger pair is denoted by  $a \rightarrow b$ , where  $a$  is the triggering word and  $b$  the triggered word. The extended model  $p_{ab}(w|h)$  is defined as in Eq. (7), where  $q(b|a)$  and  $q(b|\bar{a})$  are two interaction parameters of the word trigger pair  $a \rightarrow b$ . For symmetry reasons, we have introduced a special interaction parameter  $q(b|\bar{a})$ , when  $a$  has not been seen in the history. The unknown parameters  $q(b|a)$  and  $q(b|\bar{a})$  will be estimated by maximum likelihood.

We consider the difference between the log-perplexity  $F_{ab}$  of the extended model  $p_{ab}(w|h)$  and the log-perplexity  $F_0$  of the baseline model  $p(w|h)$  on a corpus  $w_1, \dots, w_n, \dots, w_N$ . Choosing a unigram model  $p(w)$  as baseline model  $p(w|h)$ , we obtain after some rearrangements:

$$F_{ab} - F_0 = N(a; b) \log \frac{q(b|a)}{p(b)} + N(a; \bar{b}) \log \frac{1 - q(b|a)}{1 - p(b)} + N(\bar{a}; b) \log \frac{q(b|\bar{a})}{p(b)} + N(\bar{a}; \bar{b}) \log \frac{1 - q(b|\bar{a})}{1 - p(b)}, \quad (8)$$

where the counts  $N(\cdot, \cdot)$  are defined in a natural way, e.g.  $N(a, b)$  is the number of times the word  $b$  occurred in a corpus with  $a$  in its history. Formally we define:

$$N(a; b) = \sum_{n: a \in h_n, b = w_n} 1. \quad (9)$$

The other counts are defined accordingly. From Eq. (8) we derive the maximum likelihood estimates for the interaction parameter  $q(b|a)$  (similarly for  $q(b|\bar{a})$ ):

$$q(b|a) = \frac{N(a, b)}{N(a, b) + N(a, \bar{b})}. \quad (10)$$

Identifying the probability  $p(a; b)$  with the relative frequency  $N(a; b)/N$  and similarly for the other joint events  $(a; \bar{b}), (\bar{a}; b), (\bar{a}; \bar{b})$ , we obtain exactly the mutual information criterion as suggested in [4, 9]. In other words, this criterion is simply the improvement of the log-perplexity for a unigram model using the above backing-off model for the trigger pair  $a \rightarrow b$ . The trigger pairs selected by this criterion are called *unigram level triggers*.

The model defined by Eq. (7) might have a drawback due to the following observation. If  $w_n \neq b$  for a trigger pair  $a \rightarrow b$ , the baseline language model is always discounted by a factor  $[1 - q(b|a)]$  or  $[1 - q(b|\bar{a})]$  in the model defined by Eq. (7), which may hurt in terms of perplexity. As confirmed by the experimental results, there is another approach to integrating a trigger pair into a baseline language model. The idea is to use the trigger pair  $a \rightarrow b$  only in positions  $h_n$  where the baseline language model provides a poor probability. To give a quantitative formulation, we assume a linear interpolation of a unigram distribution  $\beta(w)$  and a specific language model  $p_S(w|h)$ , e.g. an unsmoothed trigram/cache model. For the baseline language model  $p(w|h)$ , we have then the form:

$$p(w|h) = (1 - \lambda) \cdot p_S(w|h) + \lambda \cdot \beta(w).$$

To incorporate a trigger pair  $a \rightarrow b$  into the baseline language model, we replace the unigram distribution  $\beta(w)$  by a new quasi-unigram distribution  $\beta_{ab}(w)$ , which is defined in a similar fashion as Eq. (7). The extended language model  $p_{ab}(w|h)$  is then given by linear interpolation:

$$p_{ab}(w|h) = (1 - \lambda) \cdot p_S(w|h) + \lambda \cdot \beta_{ab}(w|h) \quad .$$

As expressed by this equation, the trigger pairs are incorporated into the baseline model at the lowest possible level, namely at the level of the unigram distribution. Therefore these trigger pairs will be referred to as *low level triggers*. This term was chosen to distinguish these triggers from *high level triggers* [10] which are integrated at the highest level of the baseline language model as specified by Eq. (7).

Using a probability threshold  $p_0$ , we define the set of words whose probabilities cannot be improved by the trigger model for a given history  $h$ :

$$V(h) = \{w : p_S(w|h) > p_0\} \quad .$$

For the difference  $F_{ab} - F_0$  in the log-likelihoods, we

use the approximation:

$$\begin{aligned} F_{ab} - F &= \sum_{n=1}^N \log \frac{p_{ab}(w_n|h_n)}{p(w_n|h_n)} \\ &= \left[ \sum_{n: w_n \notin V(h_n)} + \sum_{n: w_n \in V(h_n)} \right] \log \frac{p_{ab}(w_n|h_n)}{p(w_n|h_n)} \\ &\cong \sum_{n: w_n \notin V(h_n)} \log \frac{p_{ab}(w_n|h_n)}{p(w_n|h_n)} \\ &\cong \sum_{n: w_n \notin V(h_n)} \log \frac{\beta_{ab}(w_n|h_n)}{\beta(w_n)} \end{aligned} \quad (11)$$

Applying the probability threshold  $p_0$  can also be interpreted as a sort of leaving-one-out or cross-validation approach in the following sense. Among the word bigrams and trigrams of the baseline language model, there are many *accidental* bigrams and trigrams whose probabilities tend to be overestimated. By leaving-one-out, these low-count events are affected most and we obtain more honest estimates of these events. It is easy to see that applying the probability threshold  $p_0$  produces a similar result because it excludes the low-count (i.e. low-probability) events too. As to the cache effect, we have to keep in mind that the cache effect is *not* related to the training corpus, and therefore applying a threshold  $p_0$  is, at least in spirit, equivalent to considering only those words for which there is no cache effect (see [10]).

The above approximation amounts to reducing the training corpus by considering only positions  $n$  with  $w_n \notin V(h_n)$ . Then the set of these positions can be processed as in the case of the unigram level triggers. In particular, the trigger interaction parameters  $q(b|a)$  and  $q(b|\bar{a})$  are estimated on the reduced corpus, too. In informal experiments, we found that the quality of the selection criterion could be improved by computing  $\beta(w)$  also only on the reduced corpus of all words  $w_n \notin V(h_n)$  rather than the whole corpus.

### 3.2. Experimental Results

The experimental tests were performed on the Wall Street Journal (WSJ) task [8] for a vocabulary size of  $W = 20000$ . We computed trigger pairs for two selection criteria:

- A: unigram level selection criterion in Eq. (8)
- B: low level selection criterion in Eq. (11).

For the baseline language model, there were three training corpora with sizes of 1, 5 and 38 million running words. Unlike the baseline language models, the word trigger pairs were *always* selected from the 38-million word training corpus. The low level triggers were computed on a reduced corpus (resulting in 1, 8 million positions). For these experiments, the baseline model was a bigram/cache model [5], where the

Table 2: List of best triggered words  $b$  for some triggering words  $a$  for the selection criteria A and B (training corpus of 38 million words) .

| $a$      | $b$   |
|----------|---|
| asked    | A: point replied Mr. The percent asked one seven eight<br>B: replied answered responded refused replies responses reply yes request requesting                                      |
| airlines | A: airlines airline air passenger fares carriers traffic flights miles continental<br>B: American's passengers Airlines' Eastern's United's hubs fares Northwest's carriers flights |
| Ford     | A: Ford Ford's cars auto Chrysler car G. Jaguar models M.<br>B: Ford's Dearborn Bronco Taurus Escort Chrysler's Tempo Mustang Thunderbird subcompact                                |
| love     | A: her love she point his I said dollars percent You<br>B: beautifully passion sweet sexy romantic hero pop lovers pale wit   |
| says     | A: says said point million dollars adds seven he five one<br>B: concedes explains adds agrees recalls asks insists acknowledges asserts predicts                                    |

bigram component and the cache component were linearly interpolated using a cache with a weight of 0.1. In the first part of this section, we present examples of the selected trigger pairs for the two selection criteria. These examples were chosen because we deemed them to be typical. In the second part, we present perplexity results for the combination of trigger pairs with a baseline language model.

**Examples of Trigger Pairs.** For a chosen set of triggering words  $a$ , Table 2 shows the best triggered words  $b$ . The words  $b$  are ordered by decreasing perplexity improvement. For each of the two selection criteria, the trigger pairs were taken from a list of the best 500 000 trigger pairs. The low level trigger pairs yield the best overall result of the two selection criteria. Some words produce very interesting trigger pairs, e.g. the triggering words “asked” and “says” in Table 2. These verbs mostly trigger verbs again, which even have the same tense. Additional interesting examples are the triggering words “airlines” and “Ford” in Table 2, where the triggered words are airlines or car models produced by the “Ford” company. The corresponding unigram level triggers look worse for the same verbs, but for some nouns as triggering words, the triggered words seem to be more general.

**Perplexity Results.** In the following we present perplexity results for the trigger pairs. The trigger pairs were selected as described before in this subsection and were used to extend the baseline language model  $p(w_n|h_n)$  which was a trigram model. To incorporate the selected trigger pairs along with the cache model  $p_C(w_n|h_n)$  into a full language model, we define the extended language model  $p_E(w_n|h_n)$ :

$$p_E(w_n|h_n) = (1 - \lambda_C - \lambda_T) \cdot p(w_n|h_n) + \lambda_C \cdot p_C(w_n|h_n) + \lambda_T \cdot p_T(w_n|h_n), \quad (12)$$

Table 3: Perplexity results for the combination of trigger pairs (1.5 million pairs) with a trigram/cache language model (1, 5 and 38 million training words).

| model                    | 1 Mio | 5 Mio | 38 Mio |
|--------------------------|-------|-------|--------|
| trigram with no cache    | 252   | 168   | 105    |
| trigram/cache            | 197   | 138   | 92     |
| + unigram level triggers | 191   | 135   | 91     |
| + low level triggers     | 180   | 128   | 87     |

where the history  $h_n = w_{n-M}^{n-1}$  consists of the  $M$  predecessor words of  $w_n$  and  $\lambda_C$  and  $\lambda_T$  are the interpolation parameters. The cache model is defined as in Eq. (6). The trigger model  $p_T(w_n|h_n)$  is defined as:

$$p_T(w_n|w_{n-M}^{n-1}) = \frac{1}{M} \sum_{m=1}^M \alpha(w_n|w_{n-m}) \quad .$$

The  $\alpha(b|a)$  are obtained by renormalization:

$$\alpha(b|a) = \frac{q(b|a)}{\sum_{b'} q(b'|a)},$$

where the interaction parameters  $q(b|a)$  are the maximum likelihood estimates given by Eq. (10). In the experiments, the history  $h$  was defined to start with the most recent article delimiter.

The baseline trigram model was based on absolute discounting in combination with interpolation. Although the interpolation parameters  $\lambda_C$  and  $\lambda_T$  in Eq. (12) could be trained by the EM procedure [2], they were adjusted by trial and error in informal experiments. For the different extended language models, the perplexity was computed on a test corpus of 325 000 words from the WSJ task. In an experiment, the trigram/cache baseline model was extended by two different sets of trigger pairs. The perplexity results are given in Table 3. For comparison purposes, the effect of the cache component in the baseline model was studied, too. Thus the Table 3 shows the perplexities for the following conditions: trigram with no cache, trigram/cache and its extensions using a set of 1.5 million unigram level trigger pairs, and a set of 1.5 million low level trigger pairs. By far the best results were obtained for the low level trigger pairs (criterion B in Table 2). As expected, the combination of the low level trigger pairs produced the best perplexity improvement using the model defined in Eq. (12). The problem with both selection criteria presented is that the combination of the selected trigger pairs into one global language model is not captured by any of the two. However, the low level criterion provides a better approximation to the use of the trigger pairs in Eq. (12).

#### 4. SUMMARY

In this paper, we first reviewed the basic methods in language modeling for speech recognition, namely  $m$ -gram word models and smoothing. As specific smoothing method, we studied the absolute discounting method. Often, the bigram and trigram language models are extended by the cache model, which provides a sort of short-term memory for the most recent words. We presented experimental results on the Wall Street Journal corpus, both in terms of perplexity and word error rates.

In the second part of the paper, we considered the problem of selecting word trigger pairs for language modeling. Instead of using some more or less arbitrary ad-hoc selection criterion, we presented a new method for finding word trigger pairs: given a baseline language model to start with, we extend it by including one word trigger pair at a time and compute the perplexity improvement of this extended model over the baseline model. This perplexity improvement is used to select the most important word trigger pairs. For the special case that the baseline language model is a unigram model, this new method results in the well-known mutual information criterion. The more interesting case is that the trigger pair under consideration is integrated at a lower level of the baseline language model, which leads to the so-called low level selection criterion. In the experimental tests, we found the following results:

1. The low level selection criterion produced intuitively better word trigger pairs than the usual mutual information criterion.
2. When combined with a full baseline language model, namely a trigram/cache model, the low level triggers reduced the perplexity from 138 to 128 for the 5-million training set and from 92 to 87 for the 38-million training set. In comparison, when using the conventional mutual information criterion, the perplexity improvements were significantly smaller.

#### REFERENCES

1. L. R. Bahl, F. Jelinek and 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. F. Jelinek: "Self-Organized Language Modeling for Speech Recognition", in A. Waibel and K.F. Lee (eds.): 'Readings in Speech Recognition', pp. 450-506, Morgan Kaufmann Pub., 1991.
3. 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.
4. R. Lau, R. Rosenfeld and S. Roukos: "Trigger-Based Language Models: A Maximum Entropy Approach", Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Vol. II, pp. 45-48, MN, April 1993.
5. M. Generet, H. Ney, and F. Wessel: "Extensions of Absolute Discounting for Language Modeling", Proc. of Fourth European Conf. on Speech Communication and Technology, pp. 1245-1248, Madrid, September 1995.
6. H. Ney and U. Essen: "Estimating Small Probabilities by Leaving-One-Out", Third European Conf. on Speech Communication and Technology, Berlin, pp. 2239-2242, September 1993.
7. S. Ortman, H. Ney and X. Aubert: "A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition", submitted to Computer, Speech and Language, September 1996.
8. D.B. Paul and J.B. Baker: "The Design for the Wall Street Journal-based CSR Corpus", Proc. of the DARPA Spoken Language Systems Workshop, pp. 357-361, February 1992.
9. R. Rosenfeld: "Adaptive Statistical Language Modeling: A Maximum Entropy Approach", Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, CMU-CS-94-138, 1994.
10. C. Tillmann and H. Ney: "Selection Criteria for Word Triggers in Language Modeling", Proc. of the Third Int. Colloquium on Grammatical Inference, Montpellier, September 1996.