HIGH QUALITY WORD GRAPHS USING FORWARD-BACKWARD PRUNING

Achim Sixtus[1] *and Stefan Ortmanns*[2]

[1] Lehrstuhl für Informatik VI, RWTH Aachen – University of Technology, 52056 Aachen, Germany
[2] Lucent Technologies – Bell Labs., Murray Hill, NJ 07974, USA
sixtus@informatik.rwth-aachen.de, ortmanns@research.bell-labs.com

ABSTRACT

This paper presents an efficient method for constructing high quality word graphs for large vocabulary continuous speech recognition. The word graphs are constructed in a two-pass strategy. In the first pass, a huge word graph is produced using the time-synchronous lexical tree search method. Then, in the second pass, this huge word graph is pruned by applying a modified forward-backward algorithm. To analyze the characteristic properties of this word graph pruning method, we present a detailed comparison with the conventional time-synchronous forward pruning. The recognition experiments, carried out on the North American Business (NAB) 20 000-word task, demonstrate that, in comparison to the forward pruning, the new method leads to a significant reduction in the size of the word graph without an increase in the graph word error rate.

1. INTRODUCTION

In this paper, we present a different approach to the word graph forward pruning technique [7]. This approach is based on the paradigm of the forward-backward algorithm [1] and is therefore referred to as *forward-backward word graph pruning* since the key concept is to eliminate word graph hypotheses according to their forward and backward probabilities [2]. Unlike the method described in [2], we use a trigram language model to construct the word graph and during the following pruning process, in order to obtain a more compressed word graph. The word graph is based on a time-conditioned structure [4, 5], where for each word arc a list of possible predecessor arcs is kept for fast graph traversing. In addition, a sentence hypothesis tree as described in [4] is used for an efficient calculation of the forward and backward probabilities. The novel contributions of this paper are:

- Starting from the calculation of the forward and backward probabilities for an m-gram language model, we derive the word graph pruning criterion;
- We then describe the implementation details for a trigram language model which is incorporated into the lexical tree search method [4, 6];
- Finally, we report on recognition experiments for the 20 000-word NAB task and give an experimental comparison of the forward-backward pruning with the forward pruning. As the results show, the forward-backward pruning offers clear advantages in terms of word graph density and graph word error rate over the forward pruning method for very sparse word graphs.

2. FORWARD-BACKWARD WORD GRAPH PRUNING

Since the word graph produced by the acoustic recognition process can be very large, it is essential to use pruning methods for generating compact word graphs. Unlike the forward pruning, where at each time frame only the most promising word hypotheses are retained in a one-pass strategy [7], the forward-backward pruning consists of two passes after generating a huge word graph. The pruning of the word graph is based on the usual concept of beam search, but with respect to the forward and backward score of a specific word hypothesis [2]. Strictly speaking, for every word graph arc representing a word hypothesis $(w;\tau,t)$ with starting time $\tau+1$ and ending time t and the corresponding acoustic word score, we compute the overall score of the best path passing through this specific arc. Word arcs with a score relatively close to the global best path are kept in the word graph, the others are pruned. Therefore, we have to compute the following scores to obtain the best path through a given arc hypothesis $(w;\tau,t)$:

1. *The forward score* is defined as the overall score of the best word sequence that starts at the first time frame of the spoken sentence and ends at time t in the word arc w;
2. *The backward score* is defined as the overall score of best partial word sequence that starts at the time frame $\tau+1$ and ends at the last time frame of the spoken sentence.

Using these two scores we are able to compute the overall score of the best path traversing a given arc. In the following we lay the ground for the calculation of the forward and backward score and the derivation of the pruning criterion when using a trigram language model.

2.1. Computation of the Forward Score

To compute the forward score of a given arc hypothesis $(w;\tau,t)$, we use the following quantities [4]:

$h(w;\tau,t)$ = probability that word w produces the acoustic vectors $x_{\tau+1}...x_t$.

$G(w_1^n;t)$ = (joint) probability of generating the acoustic vectors $x_1...x_t$ and a word sequence $w_1...w_n$ with ending time t.

The score $G(w_1^n;t)$ can be computed from the scores $G(w_1^{n-1};\tau)$ and $h(w;\tau,t)$ by optimizing over the unknown word boundary τ:

$$G(w_1^n;t) = \max_\tau \left\{ Pr(w_n|w_1^{n-1}) \cdot G(w_1^{n-1};\tau) \cdot h(w_n;\tau,t) \right\} ,$$

where we have used the probability $Pr(w_n|w_1^{n-1})$ of a general language model. Using an m-gram language model it is sufficient to distinguish partial word sequence hypotheses only by their final (m-1) words since we can recombine word sequence hypotheses that do not differ in their final (m-1) words [3]. Therefore we use the following definition [7]:

> $H(v_2^m;t)$ = (joint) probability of generating the acoustic vectors $x_1...x_t$ and a word sequence with the ending sequence $v_2...v_m$ and ending time t.

As described in [7] the computation of $H(v_2^m;t)$ can be derived from $H(v_1^{m-1};\tau)$ as follows:

$$\begin{aligned} H(v_2^m;t) \quad &= \\ \max_{w_1^n} &\left\{ Pr(w_1^n) \cdot \max_{s_1^t} \left\{ Pr(x_1^t, s_1^t | w_1^n) \right\} : w_{n-m+2}^n = v_2^m \right\} \\ = \max_{v_1} &\left\{ p(v_m | v_1^{m-1}) \cdot \max_{\tau} \left\{ H(v_1^{m-1};\tau) \cdot h(v_m;\tau,t) \right\} \right\} . \end{aligned}$$

2.2. Computation of the Backward Score

The computation of the backward scores is performed in a similar way to the computation of the forward scores. For the same reasons as during the computation of the forward scores using an m-gram language model it is sufficient to distinguish the word sequences only if they differ in their first (m-1) words. Therefore, we introduce the following definition:

> $\tilde{H}(v_m^{m+m-2};\tau)$ = (joint) probability of generating the acoustic vectors $x_{\tau+1}...x_T$ and a word sequence with the start sequence $v_m...v_{m+m-2}$ and start time $\tau+1$.

The term $\tilde{H}(v_m^{m+m-2};\tau)$ can be computed fairly straightforward from $\tilde{H}(v_{m+1}^{m+m-1};t)$ and $h(v_m;\tau,t)$. This can be expressed by:

$$\begin{aligned} \tilde{H}(v_m^{m+m-2};\tau) \quad = \quad \max_{v_{m+m-1}} &\left\{ p(v_{m+m-1} | v_m^{m+m-2}) \right. \\ &\left. \cdot \max_t \left\{ \tilde{H}(v_{m+1}^{m+m-1};t) \cdot h(v_m;\tau,t) \right\} \right\} . \end{aligned}$$

2.3. Trigram Language Model

For simplification we rewrite the above equations for the case of a trigram language model, which is used during the acoustic recognition to build a word graph and during the pruning process. The arc of the word graph under consideration is denoted by $w := w_n$, whereas (w_{n-2}, w_{n-1}) denotes the immediate predecessor word pair and (w_{n+1}, w_{n+2}) the successor word pair of word w_n. Thus, we assume the partial word sequence $w_{n-2}^{n+2} := w_{n-2}, w_{n-1}, w_n, w_{n+1}, w_{n+2}$. Using this word order we can rewrite the recursion equations for the forward and backward scores as follows. For the forward score $H(w_{n-1}^n;t)$ we obtain:

$$\begin{aligned} H(w_{n-1}^n;t) \quad &= \\ \max_{w_{n-2}} &\left\{ p(w_n | w_{n-2}^{n-1}) \cdot \max_{\tau} \left\{ H(w_{n-2}^{n-1};\tau) \cdot h(w_n;\tau,t) \right\} \right\} ; \end{aligned}$$

where the backward score $\tilde{H}(w_n^{n+1};\tau)$ can be computed as:

$$\begin{aligned} \tilde{H}(w_n^{n+1};\tau) \quad &= \\ \max_{w_{n+2}} &\left\{ p(w_{n+2} | w_n^{n+1}) \cdot \max_t \left\{ \tilde{H}(w_{n+1}^{n+2};t) \cdot h(w_n;\tau,t) \right\} \right\} . \end{aligned}$$

2.4. Computation of the Path Score

Using the forward and the backward scores of any given arc w in the word graph, the score of the best path passing through this arc can be computed as the product of the terms $H(w_{n-1}^n;t)$ and $\tilde{H}(w_n^{n+1};\tau)$. Since the acoustic score of the word arc is incorporated twice during this computation, the product has to be divided by $h(w_n;\tau,t)$. Finally, one language model score, namely $p(w_{n+1}|w_{n-1}^n)$, is missing in the score of the best path going through this specific arc. Thus, we get the following equation for the computation of the score of the best path traversing this specific arc w_n with ending time t:

$$\begin{aligned} Q(w_n;\tau,t) \quad &= \\ \max_{w_{n-1},w_{n+1}} &\left\{ \frac{H(w_{n-1}^n;t) \cdot \tilde{H}(w_n^{n+1};\tau)}{h(w_n;\tau,t)} \cdot p(w_{n+1}|w_{n-1}^n) \right\} . \end{aligned}$$

2.5. Forward-Backward Pruning Criterion

For each arc $(w;\tau,t)$ in the word graph the score of the best path traversing this arc ($Q(w;\tau,t)$) is computed as described above. Denoting the best scoring path by:

$$Q_{max} \quad = \quad \max_{(w;\tau,t)} \{Q(w;\tau,t)\}$$

an arc is pruned if the following unequation is fulfilled:

$$Q(w;\tau,t) \quad < \quad f_{Lat\_fb} \cdot Q_{max} \ ,$$

where $f_{Lat\_fb} < 1$ is the so-called *forward-backward pruning parameter*.

3. FORWARD PRUNING

To compare the efficiency of forward-backward pruning with the so-called time-synchronous forward pruning strategy, we give a short review of this standard pruning method. Unlike the forward-backward pruning this pruning technique considers only the forward score and is only applied to word hypotheses with the same ending time t. The scores of these hypotheses are compared to the best partial word sequence at time t [7]:

$$H_{max}(t) \quad = \quad \max_{w,v} \{H(v,w;t)\}$$

Thus, a word hypothesis (w,τ,t) is pruned, if

$$H(v,w;t) \quad < \quad f_{Lat} \cdot H_{max}(t)$$

$f_{Lat} < 1$ denotes the *word graph pruning parameter*.

Both word graph pruning methods can be further combined with histogram pruning, limiting the number of surviving word hypotheses with the same ending time to a maximum number of hypotheses.

4. IMPLEMENTATION DETAILS

Using the word-conditioned lexical tree search method in combination with a trigram language model, a huge word graph is generated in a time-synchronous one-pass strategy. During this process, the forward pruning method as described in Section 3 can be

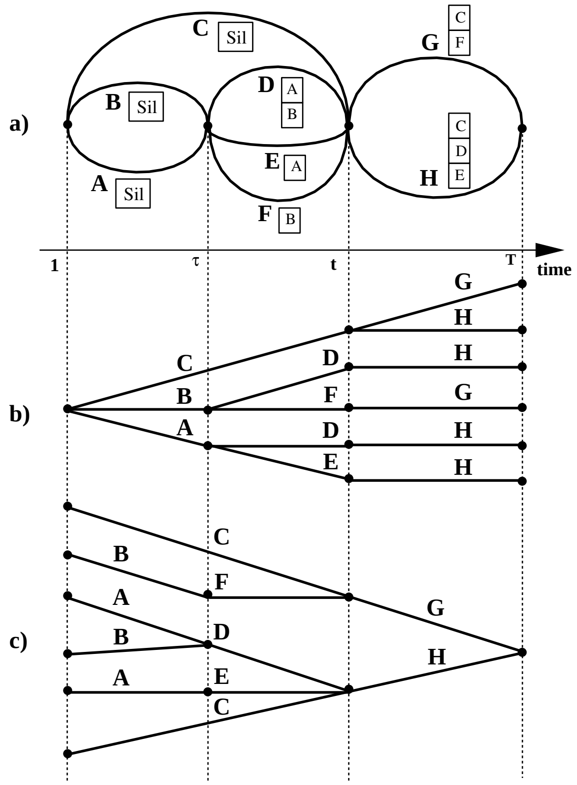

Figure 1: a) Example for a time-conditioned word graph using predecessor-word lists; b) Word hypothesis tree for the forward score calculation; c) Word hypothesis tree for the backward score calculation

employed, i. e. at each time frame we store for each of the most probable word end hypotheses $(w;\tau,t)$ the word identity w, the corresponding starting time $(\tau+1)$ and ending time t, the acoustic word score $h(w;\tau,t)$ and the immediate predecessor word v. This strategy results in a word-conditioned word graph [3, 7, 8]. In a subsequent transformation step, the final word graph is constructed by merging all nodes of the word graph with identical associated times into a single node. If there are multiple arcs with the same word and with the same ending and starting time, only one is retained in the word graph [4, 5]. Furthermore, for each word arc a list of predecessor words is kept. This list is needed to speed up the word boundary optimization and the pruning process using a sentence hypothesis tree [4]. An example of such a time-conditioned word graph is shown in Fig. 1 a).

Using this time-conditioned word graph structure, the forward-backward pruning can be efficiently applied. For this, in the first step we build a temporary sentence hypotheses tree to calculate the forward scores, namely $H(v,w;t)$ (see Fig. 1 b). For each time t all arcs of the word graph ending at this time are added to the tree. A word graph hypothesis with a start time $\tau+1$ is connected to all arcs in the tree, which end at time τ and, of course, whose word identity is contained in the list of possible predecessor words of w. Then, the word boundary optimization is applied [4]. Arcs which do not survive this maximization are removed from the tree. Next, the backward scores can be computed vice versa. Thus, the sentence hypothesis tree is constructed backwards (Fig. 1 c). In the last step, only those word arcs are kept in the word graph which satisfy the forward-backward pruning criterion. It should be mentioned that the computational effort for the calculation of the forward and backward score and the generation of the compressed word graph after pruning is negligible.

5. EXPERIMENTAL RESULTS

5.1. Word Graph Quality Measures

To specify the quality of a word graph, we use the following definitions [4]:

- *Size of the word graph.* For a spoken sentence, the *word graph density* (WGD) is defined as the total number of word graph arcs divided by the number of actually spoken words. The *node graph density* (NGD) is defined as the total number of different words ending at each time frame divided by the number of actually spoken words. Finally, the *boundary graph density* (BGD) denotes the number of different word boundaries, i.e. different start times, respectively, per spoken word.
- *Graph word error rate.* The graph word error rate (GER) is computed by determining that sentence through the word graph which best matches the spoken sentence. The matching criterion is defined in terms of word substitutions (SUB), deletions (DEL) and insertions (INS). This measure provides a lower bound of the word error rate for this word graph.

5.2. Test Condition

The experimental condition for the recognition experiments can be summarized as follows:

- NAB'94 H1 development test set including 310 sentences with 7387 spoken words from 10 male and 10 female speakers. 199 of the spoken words were out-of-vocabulary words.
- In all the recognition experiments, a trigram language model with a perplexity of 121.8 on the test set was used. Furthermore, we used $216\,000$ gender independent Gaussian mixture densities in the acoustic recognition process, which were trained on the WSJ0 and WSJ1 training data.

5.3. Recognition Experiments

To test and compare the two pruning methods we created a huge word graph with an integrated trigram search. The initial GER of the word graph was 4.44 % at a WGD of 105.08. Then we used forward pruning on the one hand and forward-backward pruning on the other hand to reduce the size of the word graph with different values for f_{Lat} and f_{Lat_fb}. The results are shown in Table 1 for the forward pruning and Table 2 for the forward-backward pruning (F_{Lat} and F_{Lat_fb} denote the logarithm of f_{Lat} and f_{Lat_fb}). Comparing the results of the two pruning methods we see, that forward-backward pruning leads to smaller word graphs. For example, considering a WGD of 8.03 for the forward-backward pruning we obtain a GER of 5.58%. The forward pruning only leads to a GER of 6.85% at a WGD of 8.23. Thus, with forward-backward pruning we obtain an absolute reduction in GER of nearly 1.3% even at a slightly compressed WGD. In addition to both Tables, Fig. 2 illustrates the efficiency of the forward-backward pruning

Table 1: Results of the forward pruning method.

F_{Lat}	WGD	NGD	BDG	GER [%]
500.00	105.07	45.70	10.54	4.44
300.00	99.74	45.12	10.20	4.44
250.00	89.11	43.69	9.50	4.45
200.00	71.87	40.85	8.39	4.51
180.00	59.31	35.55	7.42	4.62
160.00	46.52	29.34	6.35	4.81
140.00	33.74	22.40	5.20	5.08
120.00	22.86	15.96	4.20	5.40
100.00	14.07	10.31	3.29	5.96
80.00	8.23	6.36	2.59	6.85
60.00	4.78	3.89	2.10	7.89
40.00	2.97	2.55	1.76	9.39
20.00	2.05	1.85	1.54	10.79
10.00	1.77	1.65	1.45	11.63
5.00	1.64	1.56	1.40	11.90
1.00	1.54	1.48	1.36	12.12

Table 2: Results of the forward-backward pruning method.

F_{Lat_fb}	WGD	NGD	BDG	GER [%]
750.00	97.07	43.11	9.96	4.51
500.00	82.47	39.92	9.96	4.53
300.00	39.98	23.52	7.34	4.63
250.00	27.55	17.21	6.00	4.78
200.00	16.95	11.26	4.58	4.95
180.00	13.45	9.17	4.03	4.98
160.00	10.51	7.36	3.54	5.28
140.00	8.03	5.78	3.08	5.58
120.00	6.06	4.49	2.66	5.92
100.00	4.58	3.51	2.32	6.44
80.00	3.48	2.77	2.03	7.16
60.00	2.68	2.23	1.79	8.11
40.00	2.09	1.82	1.59	9.67
20.00	1.65	1.51	1.42	11.87
10.00	1.45	1.38	1.33	13.04
5.00	1.35	1.31	1.29	13.75
1.00	1.25	1.25	1.24	14.34

methods particularly with regard to small word graph densities. The Figure shows the GER over the WGD for the forward (solid line) and the forward-backward pruning method (dotted line).

6. SUMMARY

We have presented an efficient word graph pruning method for constructing compact word graphs, which is based on the forward-backward paradigm. The experimental results carried out on the NAB task show that this method leads to much smaller word graphs than the forward pruning method.

Acknowledgement. This research was partly funded by grant 01 IV 701 T4 from the German Ministry of Science and Technology (BMBF) as a part of the VERBMOBIL project. The views and conclusions contained in this document are those of the authors.

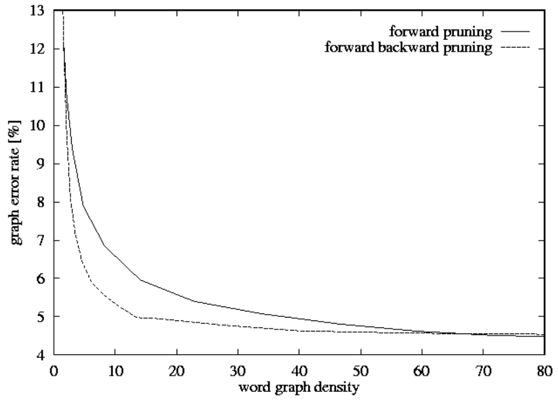

Figure 2: GER over WGD for forward pruning (solid line) and forward-backward pruning (dotted line).

7. REFERENCES

[1] S. Austin, R. Schwartz, P. Placeway: The Forward-Backward Search Algorithm, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Toronto, Canada, pp 697-700, May 1991.

[2] T. Kuhn, P. Fetter, A. Kaltenmeier, P. Regel-Brietzmann: DP-based Wortgraph Pruning, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Atlanta, GA, pp. 461-64, May 1996.

[3] H. Ney, X. Aubert: A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition, Proc. Int. Conf. on Spoken Language Processing, Yokohama, Japan, Vol. 3, pp. 1355-1358, September 1994.

[4] H. Ney, S. Ortmanns, I. Lindam: Extensions to the Word Graph Method for Large Vocabulary Continuous Speech Recognition, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Munich, Germany, pp. 1787-1790, April 1997.

[5] M. Oerder, H. Ney: Word Graphs: An Efficient Interface Between Continuous Speech Recognition and Language Understanding. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Minneapolis, MN, Vol.II, pp. 119-122, April 1993.

[6] S. Ortmanns, H. Ney, F. Seide, I. Lindam: A Comparison of Time Conditioned and Word Conditioned Search Techniques for Large Vocabulary Speech Recognition, Proc. Int. Conf. on Spoken Language Processing, Philadelphia, PA, pp. 2091-2094, October 1996.

[7] S. Ortmanns, H. Ney, X. Aubert: A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition, Computer, Speech and Language, Vol. 11, No. 1, pp. 43-72, January 1997.

[8] P. C. Woodland, C. .J. Leggetter, J. J. Odell, V. Valtchev, S. J. Young: The 1994 HTK Large Vocabulary Speech Recognition System, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Detroit, MI, pp. 573-576, May 1995.