EXTENSIONS TO THE WORD GRAPH METHOD FOR LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

H. Ney, S. Ortmanns, I. Lindam

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

ABSTRACT

This paper describes two methods for constructing word graphs for large vocabulary continuous speech recognition. Both word graph methods are based on a time-synchronous, left-to-right beam search strategy in connection with a tree-organized pronunciation lexicon. The first method is based on the so-called word pair approximation and fits directly into a word-conditioned search organization. In order to avoid the assumptions made in the word pair approximation, we design another word graph method. This method is based on a time conditioned factoring of the search space. For the case of a trigram language model, we give a detailed comparison of both word graph methods with an integrated search method. The experiments have been carried out on the North American Business (NAB'94) 20,000-word task.

1. INTRODUCTION

This paper studies some questions arising in the context of word graphs for large vocabulary continuous speech recognition [1]. In particular, we study two word graph methods in detail. In addition to the method based on the widely used word pair approximation, we present another approach, in which we try to avoid assumptions such as the word pair approximation. The novel contributions of this paper are:

- To improve the word pair approximation, we describe another word graph method which tries to avoid any type of approximation. The method is based on a time conditioned factoring of the search hypotheses.
- We give an exact comparison of the two word graph methods with an integrated search method in combination with a trigram language model. This comparison is needed, because for short predecessor words the quality of the word pair approximation is questionable.
- We present experimental results on the North American Business (NAB'94) 20,000-word task. For the case of a trigram language model, the recognition experiments demonstrate that the integrated search method leads only to a slight improvement over the word graph methods in recognition accuracy.

2. DEFINITION OF WORD GRAPHS

2.1. Problem Specification

The basic idea of a word graph is to represent all word sequence hypotheses whose scores are very close to the locally optimal hypothesis in the spirit of beam search. Unlike the n-best approach for finding the n-best sentences, the word graph method results in more compact representation of word and sentence alternatives.

The goal of constructing a word graph can be summarized as follows. In principle, we consider all n-best sentences with a huge value for n. All these sentences are now to be represented by a word graph with the following properties: 1) The acoustic scores of each sentence should not be affected. 2) The word boundaries should be preserved. 3) In general, the word graph may contain more than the original n-best sentences, however, with worse scores. 4) The word graph should have a minimum number of word arcs.

2.2. Word Boundary Function

To describe the generation of a word graph, we introduce the following definitions:

> $h(w;\tau,t)$ = probability that word w produces the acoustic vectors $x_{\tau+1}...x_t$.
>
> $H(w_1^n;t)$ = (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 $H(w_1^n;t)$ can be computed from the scores $H(w_1^{n-1};\tau)$ and $h(w;\tau,t)$ by optimizing over the unknown word boundary τ:

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

where we have used the probability $Pr(w_n|w_1^{n-1})$ of a general language model. To describe the construction of a word graph, we introduce a formal definition of the word boundary $\tau(w_1^n;t)$ between the word hypothesis w_n ending at time t and the predecessor sequence hypothesis w_1^{n-1}:

$$\tau(t;w_1^n) \quad := \quad \arg\max_{\tau} \left\{ H(w_1^{n-1};\tau) \cdot h(w_n;\tau,t) \right\} \quad .$$

In the case that we are given an m-gram language model, we could further simplify the search and recombine partial sentence hypotheses if they do not differ in their final $(m-1)$ words [5]. However, we do not exploit this property for the word graph construction since we want to consider the most general case.

3. WORD PAIR APPROXIMATION

In the word pair approximation, the crucial assumption is that the dependence of the word boundary $\tau(t; w_1^n)$ can be confined to the final word pair (w_{n-1}, w_n) [9]. This assumption can be expressed by the equation:

$$\tau(t; w_1^n) = \tau(t; w_{n-1}^n) \quad .$$

By taking this property into account, we obtain the following algorithm for the word graph construction [5] which fits directly into a word-conditioned search organization:

- At each time frame t, we consider all word pairs (v, w). Using a beam search strategy, we limit ourselves to the most probable hypotheses $(t; v, w)$, i.e. word pair (v, w) with ending time t.
- For each triple $(t; v, w)$, we have to keep track of:
  - the word boundary $\tau(t; v, w)$
  - the word score $h(w; \tau(t; v, w), t)$
- At the end of the speech signal, the word graph is constructed by tracing back through the bookkeeping lists.

Given a word graph and an m-gram language model, the second-pass of the word graph method can be carried out at the sentence level using a left-to-right dynamic programming algorithm as described in [1]. The word graph generated by the acoustic recognition process can be very large. To reduce the size of the word graph, pruning methods can be applied to the word graph without affecting the word error rate. The pruning of the word graph is based on the usual concept of beam search: hypotheses with a score relatively close to best hypothesis are retained as active, the others are pruned. This pruning method can also be combined with histogram pruning to limit the maximum number of word hypotheses per time frame.

From the viewpoint of the word pair approximation, the extensions to be presented in the next section can be interpreted as follows. For a given speech signal to be recognized, there are two conditions which deserve more detailed considerations:

- There are regions in the acoustic signal such that the partial sentence hypotheses do *not require* the word pair approximation. In these cases, the word graph can be further condensed.
- There are regions in the acoustic signal and partial sentence hypotheses such that the word pair approximation is *not sufficient*. In these cases, a word triple or even higher approximation for the word graph is needed.

4. IMPROVED WORD GRAPHS

4.1. Concept: Time Conditioned Search

In this section, we present another approach to word graphs, which avoids the word pair approximation. This method is based on a time synchronous beam search strategy using time conditioned copies of the lexical prefix tree [7]. To describe the time conditioned one-pass search algorithm, we define the following quantity as introduced in [4]:

$Q_\tau(t, s)$:= overall score of the best path up to time t that ends in state s of the lexical prefix tree with starting time τ.

$H(w; t)$:= (joint) probability of generating the acoustic vectors $x_1 ... x_t$ and a word sequence with final word w and ending time t.

To start up a new tree, i.e. new words, we initialize this quantity as follows:

$$Q_{t-1}(t-1, s) = \begin{cases} \max_u H(u; t-1) & \text{if } s = 0 \\ 0 & \text{if } s > 0 \end{cases}$$

The state $s = 0$ is used for initialization. The usual dynamic programming recursion for the word interior is:

$$Q_\tau(t, s) = \max_\sigma \{ \, q(x_t, s|\sigma) \, Q_\tau(t-1, \sigma) \, \} \; .$$

To formulate the recombination across word boundaries for a bigram language model, it is useful to introduce an additional auxiliary quantity for an efficient implementation:

$$\hat{H}(v, w; t) \; := \; \max_\tau \left\{ \frac{H(v; \tau)}{\max_u H(u; \tau)} \cdot Q_\tau(t, S_w) \right\} \; ,$$

where S_w denotes the terminal state of word w in the lexical tree. To select the best predecessor word for each pair $(w; t)$, i.e. word w with ending time t, two optimization steps have to be performed. The first step is to maximize over all possible starting times τ of word w. This optimization results in a list of predecessor words v of word w. In the equation above, a normalization is necessary since each tree hypothesis is started with the best predecessor word. Second, we have to maximize the hypothesis score over the predecessor words v for each w:

$$H(w; t) = \max_v \left\{ \, p(w|v) \cdot \hat{H}(v, w; t) \right\} \; .$$

The best of the scores $H(w; t)$ is used to start up the new tree for time $(t + 1)$. For the subsequent word graph construction, we compute the scores $h(w; \tau, t)$ as

$$h(w; \tau, t) \; = \; \frac{Q_\tau(t, S_w)}{\max_u H(u; \tau)} \quad .$$

It should be mentioned that these word scores $h(w; \tau, t)$ define already what could be called a time conditioned word lattice. However, this word lattice is much too big and contains multiple paths for the same sentence because the optimization over the word boundaries has not been done yet. The method presented in the next subsection is aimed exactly at removing these redundant paths. The same principle was used for the approach presented in [6].

4.2. Algorithmic Details

Using the time conditioned search strategy, a word graph can be generated in a time-synchronous fashion. To simplify the description, we assume for the moment that the following two operations are performed subsequently:

- *Generation of the sentence hypothesis tree.* Using the word hypothesis scores $h(w; \tau, t)$, we extend the partial sentence hypotheses and compute the score $H(w_1^n; t)$.

This includes the word boundary optimization over the unknown word boundary τ for each partial sentence hypothesis w_1^n. As usual in beam search, this is done for all sentence hypotheses that survive within the beam. These sentence hypotheses are organized in the form of a tree the arcs of which are the word hypotheses with specific start and end times. When constructing this sentence tree, a *purging* operation is useful to reduce the memory requirements by removing dead partial hypotheses. Evidently, this tree represents all sentence hypotheses; however, it is only practical for short sentences, say less than a sentence length of 10 words. To further reduce the memory requirements, a first step towards a graph representation is done as follows. At time frame t, we merge all tree nodes that are not reachable any more from any current state hypothesis $Q_\tau(t,s)$.

- *Generation of the word graph.* In practice it is desirable to produce a word graph with a minimal word graph density. Starting with the tree (or graph) of sentence hypotheses, the final word graph is constructed by merging all tree nodes with identical associated times into a single node. Each word arc of the sentence hypothesis tree is copied and added to the word graph. If for the same pair of nodes and for the same word there are multiple arcs, only one arc is retained. It is guaranteed that this word graph contains all sentence hypotheses of the original tree. However, there might be additional sentence hypotheses which in the original tree did not survive within the beam due to the language model score.

In reality, the above two operations are carried out in parallel for each time frame.

5. EXPERIMENTAL RESULTS

5.1. Word Graph Quality Measures

To specify the quality of a word graph, we introduce the following definitions:

- *Size of the word graph.* For a spoken sentence, the *word graph density* (WGD) is defined as the total number of word graph edges divided by the number of actually spoken words. Similarly, the *node graph density* (NGD) and the *boundary graph density* (BGD) denote the number of nodes and of different word boundaries, respectively, per spoken word.
- *Graph word error rate.* The graph word error rate (GER) is computed by determining that sentence through the word graph that 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 and gives a better measurement of the word graph quality than the graph sentence error rate. Note that the graph word error rate (GER) is to be distinguished from the standard *recognition word error rate* (WER).

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 the 10 male and 10 female speakers. 199 of the spoken words were out-of-vocabulary words.
- In all the recognition experiments, we used the official 20 000-word trigram language model for the NAB'94 task [8].
- The training of the emission probability distributions of the Hidden Markov models was performed on the WSJ0 and WSJ1 training data [2].

5.3. Recognition Experiments

As baseline results, we use the speech recognition results obtained for the integrated search method in combination with a trigram language model [7]. The recognition results are given in Table 1. Table 1 shows the search space, which is given in terms of the average number (per time frame) of active states, of active arcs and the recognition word error rate. It can be seen that by increasing the average number of active states per time frame from 4 714 to 86 772 the word error rate is reduced from 14.8% to 13.9%.

To test and compare the two word graph methods, we carried out a series of recognition experiments. Table 2 summarizes the results. The table consists of two parts, namely part (a) for the word graph method based on the word pair approximation and part (b) for the improved word graph method. The table was produced in the following way. For each of the two word graph methods, a conservatively large word graph was constructed using a bigram language model with a test set perplexity of 198.1. Then the size of the word graph was reduced by applying a pruning operation using a pruning threshold f_{lat}. For this resulting word graph, the table reports the size of the word graph in terms of the word graph density (WGD), the graph word error rate (GER) and the recognition word error rate (WER), for both of which the number of word deletions (DEL) and insertions (INS) is also given. For the recognition test, a full search through the word graph was performed using a trigram language model (perplexity of 130.2).

For the word graph method using the word pair approximation (Table 2a), on average, the acoustic search space (when computing the initial word graph) consisted of 27 672 active states, 7 674 active arcs and 115 active trees per time frame during the first pass of the two-pass search strategy and results in a word error rate of 16.5%. When varying the word graph density from 1415.9 to 10.7, the graph error rate is increased from 4.3% to 6.8%. At the same time, there is no observable effect on the recognition word error rate which is invariably 14.3%. This result is to be compared with a word error rate of 13.9% for the

Table 1. Recognition results for the integrated method (trigram language model with $PP = 130.2$) as a function of the search space.

Average number of active			Recognition word error rate [%]	
States	Arcs	Trees	DEL / INS	WER
4714	1392	29	1.7 / 2.9	14.8
18734	5430	70	1.6 / 2.8	14.2
48940	13877	112	1.6 / 2.8	14.1
67541	18764	130	1.6 / 2.7	14.0
86772	23688	145	1.6 / 2.7	13.9

Table 2. Recognition results on the NAB'94 H1 development data (trigram language model with $PP = 130.2$; OOV rate: 2.7%): a) word graph method using the word pair approximation (word graph generation: 27672 states, 7674 arcs, 115 trees), b) improved word graph method (word graph generation: 28017 states, 7998 arcs, 37 trees).

		Graph density			Graph word error rate [%]		Recognition word error rate [%]	
Method	f_{lat}	WGD	NGD	BGD	DEL / INS	GER	DEL / INS	WER
a)	150	1415.9	175.9	19.3	0.1 / 0.5	4.2	1.6 / 2.7	14.3
	90	460.4	77.4	12.3	0.2 / 0.5	4.3	1.6 / 2.7	14.3
	80	269.2	51.8	9.8	0.2 / 0.6	4.5	1.6 / 2.7	14.3
	70	137.4	31.4	7.4	0.3 / 0.7	4.8	1.6 / 2.7	14.3
	50	25.2	9.0	3.8	0.5 / 0.9	5.8	1.7 / 2.7	14.3
	40	10.7	4.8	2.7	0.7 / 1.1	6.8	1.7 / 2.7	14.3
	30	4.5	2.8	2.0	1.1 / 1.3	8.4	1.8 / 2.6	14.6
	20	2.4	1.8	1.6	1.5 / 1.7	10.6	2.0 / 2.5	14.8
	10	1.6	1.4	1.4	2.1 / 2.2	13.8	2.3 / 2.4	15.6
	1	1.3	1.3	1.2	2.3 / 2.6	16.3	2.3 / 2.6	16.4
b)	70	525.1	363.2	13.4	0.3 / 0.6	4.1	1.6 / 2.7	14.0
	65	330.3	231.8	10.1	0.4 / 0.6	4.4	1.6 / 2.7	14.0
	60	203.3	145.1	8.0	0.4 / 0.6	4.5	1.6 / 2.7	14.0
	50	71.5	53.0	4.9	0.6 / 0.8	5.1	1.6 / 2.7	14.0
	40	24.8	19.3	3.1	0.5 / 0.9	6.0	1.7 / 2.6	14.2
	30	8.8	7.4	2.1	0.9 / 1.1	7.2	1.7 / 2.6	14.4
	20	3.6	3.2	1.6	1.5 / 1.4	9.1	1.9 / 2.5	14.7
	10	1.9	1.8	1.2	2.2 / 1.8	12.3	2.3 / 2.5	15.6
	5	1.6	1.5	1.2	2.4 / 1.9	13.5	2.5 / 2.6	16.2

integrated search method, which however involves a much higher computational cost.

For the improved word graph method using time conditioned search hypotheses (Table 2b), the acoustic search space consisted of 28 017 states, 7 998 arc and 37 tree hypotheses per time frame on average when computing the initial word graph. As before, the size of the word graph was reduced by varying the pruning threshold f_{lat}, and both the word graph error rate and the recognition word error rate were measured. Comparing the results in the two parts of Table 2 for a fixed word graph density, we can see that the improved word graph method leads to both better recognition word error rates and better graph word error rates. Furthermore, looking at Table 1, we see that there is only a slight degradation of the recognition word error rate, namely 14.0% instead of 13.9%. We attribute this marginal degradation to the small size of the beam that was used to construct the initial word graph.

6. SUMMARY

We have presented an improved method for word graph construction, which avoids the widely used word pair approximation. The experimental results performed on the 20000-word NAB task confirm that the method offers clear advantages over the word pair approximation.

REFERENCES

[1] X. Aubert, H. Ney: Large Vocabulary Continuous Speech Recognition using Word Graphs. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Detroit, MI, pp. 49-52, May 1995.

[2] C. Dugast, R. Kneser, X. Aubert, S. Ortmanns, K. Beulen, H. Ney: Continuous Speech Recognition Tests and Results for the NAB'94 Corpus. Proc. ARPA Spoken Language Technology Workshop, Austin, TX, pp. 156-161, January 1995.

[3] H. Ney, D. Mergel, A. Noll, A. Paeseler: Data Driven Organization of the Dynamic Programming Beam Search for Continuous Speech Recognition. IEEE Trans. on Signal Processing, Vol. SP-40, No. 2, pp. 272-281.

[4] H. Ney: Search Strategies for Large-Vocabulary Continuous-Speech Recognition. NATO Advanced Studies Institute, Bubion, Spain, June-July 1993, pp. 210–225, in A.J. Rubio Ayuso, J.M. Lopez Soler (eds.): 'Speech Recognition and Coding – New Advances and Trends', Springer, Berlin, 1995.

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

[6] 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.

[7] 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.

[8] R. Rosenfeld: The CMU Statistical Language Modeling Toolkit and its Use in the 1994 ARPA CSR Evaluation. Proc. ARPA Spoken Language Technology Workshop, Austin, TX, pp. 47-50, January 1995.

[9] R. Schwartz, S. Austin: A Comparison of Several Approximate Algorithms for Finding Multiple (N-Best) Sentence Hypotheses. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Toronto, pp. 701-704, May 1991.