LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION USING WORD GRAPHS

Xavier Aubert

Philips GmbH Research Laboratories Aachen
Weißhausstraße 2, D-52066 Aachen
Germany

Hermann Ney

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

ABSTRACT

We address the problem of using word graphs (or lattices) for the integration of complex knowledge sources like long span language models or acoustic cross-word models, in large vocabulary continuous speech recognition. A method for efficiently constructing a word graph is reviewed and two ways of exploiting it are presented. By assuming the word pair approximation, a phrase level search is possible while in the other case a general graph decoder is set up. We show that the predecessor-word identity provided by a first bigram decoding might be used to constrain the word graph without impairing the next pass. This procedure has been applied to 64k-word trigram decoding in conjunction with an incremental unsupervised speaker adaptation scheme. Experimental results are given for the North American Business corpus used in the November'94 evaluation.

1. INTRODUCTION

In the recent past, the use of word graphs or word lattices has become quite popular among the various search techniques applied to large vocabulary continuous speech recognition ([1], [2], [3], [4], among others). These developments have been stimulated by the need for dealing with still more detailed acoustic models, more complex language models, and vocabularies of still larger sizes, as many factors that can lead to order of magnitude increases of the potential search space. The main idea about word graphs is to come up with word alternatives in regions of the speech signal where the ambiguity of the recognition is high and to apply subsequently more elaborate knowledge sources within this narrowed-down search space.

We address the problem of using word graphs (or lattices) for the integration of long span language models (LM) and of more detailed acoustic models, in large vocabulary continuous speech recognition. A method for efficiently constructing the word graph is first reviewed [5] and then, two distinct ways of exploiting the word graph are presented. In both cases, the word graph is generated with a bigram LM using our standard one-pass algorithm based on word conditioned lexical trees [6].

The first graph search technique concerns the application of a long span LM in conjunction with the same acoustic models used for generating the word graph. The assumption of the so-called "word pair approximation" leads to a very efficient algorithm: the search can be performed at the phrase level by using *as such* the boundaries and acoustic scores provided by the bigram word graph. This algorithm achieves a good decoupling between acoustic and syntactic levels and has already been successfully applied to trigram LM decoding [3].

However the exact influence of the underlying word pair approximation is unknown and on the other hand, the use of different acoustic models implies that the word boundaries and scores have to be re-evaluated anyway. This concerns for example the integration of cross-word acoustic models or the use of an unsupervised speaker-adaptation scheme together with the best language model available.

Therefore we have been investigating a general graph-search procedure either to perform "full" trigram decoding or to efficiently implement more detailed acoustic models. In particular, a phonetic network is built up for each word arc to integrate cross-word triphones and alternative pronunciations.

To reduce the complexity of the search, several constraints imposed on the word graph have been investigated and the following results have been achieved:

- We show that the predecessor-word identity provided by a first bigram decoding might be retained to constrain the word graph without impairing the next pass. This leads to very low branching factors making it unnecessary to resort to backing-off when creating word copies for multiple trigram contexts.
- Using a full graph search technique, we are able to assess the impact of the "word-pair approximation" on the accuracy of trigram decoding. On typical Wall Street Journal (WSJ) data (5k and 20k vocabularies), a relative loss of only 1 to 2 % is observed when applying the phrase level search algorithm as opposed to the "full" trigram decoding.
- This word graph procedure has been applied to trigram LM decoding in conjunction with an incremental unsupervised speaker adaptation scheme for vocabularies of up to 64k words.
- Experimental results are given for the "North American Business" (NAB) corpus used in the November'94 evaluation.

0-7803-2431-5/95 $4.00

2. BIGRAM WORD GRAPH CONSTRUCTION

We briefly explain the word graph construction that is embedded in our one-pass bigram LM decoding [6]. More details can be found in [5].

The main idea is to keep track of word sequence hypotheses whose score is close to the locally optimal hypothesis, but that do not survive due to LM recombination, and to represent all these sequences by a graph in which each arc is a word hypothesis.

In the one-pass algorithm, we approximate the most likely word sequence by the most likely state sequence and apply dynamic programming to compute the probabilities

$$Pr(w_1...w_N) \cdot Pr(x_1...x_T|w_1...w_N)$$

in a left-to-right fashion and carry out simultaneously the optimization over the unknown word sequence. Here, $x_1...x_T$ is a time sequence of observed acoustic vectors and $w_1...w_N$ is a hypothesized word sequence.

When an m-gram language model $p(u_m|u_1^{m-1})$ is exploited in the course of the one-pass search, word sequence hypotheses are recombined as soon as they do not differ in their final $(m-1)$ words. Therefore to distinguish partial word sequence hypotheses, it is sufficient to consider only their final words u_2^m. The corresponding score is denoted by $H(u_2^m;t)$:

$$H(u_2^m;t) := \max_{w_1^n} \left[Pr(w_1^n) \cdot Pr(x_1^t|w_1^n) : w_{n-m+2}^n = u_2^m \right]$$

which gives the (joint) probability of generating the acoustic vectors $x_1...x_t$ and a word sequence with *ending* sequence u_2^m and *ending* time t.

To arrive at the dynamic programming (DP) recursion, we need to isolate the probability contributions of the last word hypothesis with respect to both the language model and the acoustic model. Hence we introduce:

$$h(w;\tau,t) := Pr(x_{\tau+1}^t|w) \ ,$$

the probability that w produces the $x_{\tau+1}...x_t$ vectors. Now, the score decomposition isolating the last word contributions can be visualized as follows:

$$\underbrace{x_1,\cdots,\cdots,\cdots,\cdots,x_\tau}_{H(w_{n-m+1}^{n-1}=u_1^{m-1};\tau)} \ \underbrace{x_{\tau+1},\cdots,\cdots,x_t}_{h(w_n=u_m;\tau,t)} \ \underbrace{x_{t+1},\cdots,x_T}_{\cdots}$$

and using the above definitions, we can write the dynamic programming equation at the word level:

$$H(u_2^m;t) = \max_{u_1} \Big[\ p(u_m|u_1^{m-1}) \cdot \max_{t'<t} \left[H(u_1^{m-1};t') h(u_m;t',t) \right] \Big] \qquad (1)$$

The boundary itself between u_{m-1} and u_m, for the word sequence with final portion u_1^m ending at time t, follows from a maximization operation:

$$\tau(u_1^m;t) := \arg\max_{t'<t} \left[\ H(u_1^{m-1};t') \ h(u_m;t',t) \ \right] .$$

When using a bigram LM, this equation implies that the dependence of the word boundary $\tau(u_1^m;t)$ will be confined to the final word pair u_{m-1}^m. This so-called "word pair approximation" had originally been introduced in [7] to efficiently calculate n-best sentences. The assumption that the other predecessor words have no effect on the boundary position of the ending word pair is surely satisfied if the word u_{m-1} is long enough but is questionable for a one-phone word. Assuming the word pair approximation, we have the following algorithm for bigram word graph construction:

- At each time t, we consider all active word pair hypotheses $u_1^2 = (v,w)$ for which w ends at t. The most probable word pairs are selected using a beam pruning strategy.
- For each triple $(v,w;t)$, we keep track of:
  - the (unique) word boundary $\tau(v,w;t)$
  - the word acoustic score $h(w;\tau(v,w;t),t)$
- At the utterance end, the word graph is constructed by tracing back through the book keeping lists.

3. M-GRAM PHRASE LEVEL SEARCH

The task is now to extract from the bigram word graph the "best" sentence according to a longer span ($m > 2$) LM. Assuming again the word pair approximation, this search can be performed at the phrase level i.e. using as such the boundary points and the word acoustic scores coming from the first bigram decoding pass. This leads to the following left-to-right search algorithm:

- For each time $t = 1,\cdots,T$ and each triple $(v,w;t)$:
  - Get boundary $\tau = \tau(v,w;t)$ and score $h(w;\tau,t)$
  - DP recursion for m-gram LM with $u_{m-1}^m = (v,w)$:

$$H(u_2^m;t) = \max_{u_1} \Big[p(u_m|u_1^{m-1}) \big[H(u_1^{m-1};\tau) h(u_m;\tau,t) \big] \Big] \qquad (2)$$

- The "best" sentence is obtained using back pointers.

In contrast to the general m-gram recursion (1), there is no expensive time optimization of the boundary between the two last hypothesized words. Instead, the segmentation points and acoustic scores included in the bigram word graph are injected which leads to a dramatic complexity reduction. For a trigram LM, the phrase level search represents less than 1% of the effort for constructing the bigram word graph. Table 1 presents recognition results obtained on two WSJ test-sets for a vocabulary of 45,000 words. In both cases, there are 0.35% 'OOV' words and nearly one fifth of the errors are recovered.

Table 1: From Bigram to Trigram LM for 45k Lex. WER=Word Error Rate (Del+Ins+Sub)

WSJ Test-Set	BIGRAM WER %	BIGRAM Perp.	TRIGRAM WER %	TRIGRAM Perp.	Rel.Err. Reduct.
Nov'92 Evl	11.9	219	9.8	146	-18%
Nov'93 Evl	16.4	233	13.4	146	-18%

Still the exact influence of the underlying word pair assumption is unknown. In Section 5, experimental results

obtained with a general graph search technique will give some evidence that the degradation caused by the word pair approximation is quite small.

4. FULL WORD GRAPH DECODING

4.1. Extraction of Syntactic Content

Our starting point consists of the bigram word graph described in Section 2. This is nothing but a time-structured list of word hypotheses consisting of word identity, start- and end-time, acoustic score and predecessor word identity. To represent all these word sequences by a graph data structure, the definition of a node has to be specified, each arc being a word hypothesis. Two cases have been considered.

In the general case a node is simply a time-mark. This means that all word hypotheses ending at time t are pointing to the same node and might be followed by any word starting at $t+1$ in the word graph.

However, the success of the search algorithm of Section 3 suggests that the predecessor word identity provided by the bigram decoding might be used to constrain the word graph without impairing the next pass. When this predecessor word dependence is to be kept, a node is defined as a pair {time, predecessor-Id}. In this "bigram-constrained" word graph, the predecessor information is thus used to restrict the connections between succeeding words. Application of this constraint is supported by the observation that if a particular word pair has a very small (bigram) LM probability, any m-gram ($m > 2$) including this word pair is likely to be also of very small probability.

On the other hand, we are no longer interested in the time and score informations as we now intend to perform a full decoding at the 10-ms level, possibly using different acoustic models. Instead, we want to get rid of all copies of words occurring in the same contexts at consecutive time frames since they do not bring anything new in terms of syntactic richness and they will only burden the graph search process. To eliminate these copies of words appearing in the same contexts, nodes that are closely spaced in time are merged using several reduction rules. This provides a very significant "compaction" effect.

4.2. Word Phonetic Networks

For each arc in the graph, a word model has to be specified in terms of elementary acoustic units. These are typically triphones conditioned on the left and right phonemes. When cross-word coarticulation effects are *explicitly* taken into account, the triphones at the begin and end of a word depend on the neighboring words as given by the graph structure. Therefore, multiple triphone instances are created at the initial and final position of a word model, the number of which depends on the local graph characteristics. Note that for the bigram-constrained word graph, there is only one predecessor context.

Alternative pronunciations are introduced by allowing the substitution and skip of particular phones. Cross-word dependent assimilation rules are also used to model "hard" pronunciation changes that occur at word juncture [8], for example when a phone is completely deleted like in " ... receive(d) the ...". As a result, a phonetic network is build up for each word hypothesis and inserted in the graph together with optional between-word silence models.

4.3. Viterbi Graph Search

Decoding proceeds from left to right using a time-synchronous search algorithm with a beam-pruning technique. However the word graph has first to be expanded with respect to all contextual constraints introduced by either the LM or the cross-word models. For an m-gram LM, words appearing in different contexts have to be duplicated to keep track of all hypotheses differing in their final $(m-1)$ words (Section 2). Consecutive word arcs are then connected with language transitions whose probabilities are given by the m-gram LM. In case of a trigram-LM for example, separate arc copies are made for each predecessor word and are recombined at the end of the succeeding word. This implies that if the word graph exhibits a *local* branching factor of b, with b arcs pointing to - and leaving each node, b^3 language transitions are requested which leads to a prohibitive number of arcs in the region of the sentence where the ambiguity of the speech signal is high.

So far, this problem has been solved mainly by relying on the back-off property of the LM, i.e. by duplicating an arc only if the corresponding m-gram has been taken explicitly into account by the LM (see a.o. [1], [4]). Our solution consists of two parts:

- First, the word graph is expanded dynamically on demand, that is, only when a word-end hypothesis is reached and kept active within the beam.
- Second, bigram-constrained word graphs are used that request only b^2 language transitions for a trigram LM since the predecessor dependence has already been integrated.

5. EXPERIMENTAL RESULTS

5.1. Impact of Word Pair Approximation and Predecessor Constraint

To get some measure of the accuracy loss introduced in trigram decoding either by the word pair approximation or by the predecessor constraint, several graph search strategies have been tested on the November'92 WSJ evaluation set (4 males, 4 females, 5k and 20k vocabularies).

We first generated word graphs of high density using our standard bigram-LM beam search, to insure that the spoken word sequences were included in the word graphs whenever possible, i.e. in the absence of Out-of-Vocabulary (OOV) words. The details of the acoustic modeling and training are described in [3]. Then, trigram decoding has been performed under 3 different search conditions, all other things being identical:

- First, we used the phrase level search algorithm relying on the word pair approximation (Section 3, Equation 2).
- Next, we applied the "full" graph decoding procedure with large beam widths to a general graph data structure obtained from the original bigram word graph.

- Third, the same procedure was applied to a bigram-constrained word graph that preserves the predecessor information of the original bigram decoding, however without time and score information.

Table 2 summarizes the recognition results at the word level, obtained with a trigram LM for 5k and 20k vocabularies. For each test condition, the Word Error Rate (WER%) is given together with both the average and maximum number of word arcs expanded per sentence in the course of the graph-search process.

Table 2: Trigram Results on the Nov'92 WSJ Test-Sets

Algorithm	WER%	Av.#Arcs	Max.#Arcs
	5k Closed-Vocabulary		
Word Pair	4.90%	-	-
General Graph	4.75%	5,000	108,000
Bigram Graph	4.75%	1,300	18,000
	20k Open-Vocabulary		
Word-Pair	11.9%	-	-
General Graph	11.8%	7,000	114,000
Bigram Graph	11.8%	1,400	14,300

The following conclusions can be drawn:

- The word pair approximation introduces a relative degradation of less than 2% and we did observe that essentially short words are affected.
- Compared to general word graphs, bigram-constrained word graphs achieve the same precision.
- The number of arcs expanded during search is drastically reduced in the last case due to the very low branching factors of bigram-constrained word graphs.

5.2. 64k-Word Trigram and Speaker Adaptation

Using the full graph search procedure, we can combine trigram decoding with incremental speaker adaptation. The principle of incremental unsupervised speaker adaptation amounts to update the acoustic models after each spoken sentence by using the alignment between the speech signal and the *recognized* word sequence. The success of such a scheme depends partly upon the correctness of the recognition, hence the interest for taking the best available LM.

This technique has been applied to the North American Business (NAB) corpus which contains read articles taken from several newspapers with an unlimited vocabulary. To achieve a high coverage, a vocabulary of 64k words has been taken. Table 3 gives some information about the two test sets used for the Nov'94 ARPA evaluation. Both sets include 10 male and 10 female speakers each having uttered 15 sentences of about 25 words.

Table 3: NAB'94 Coverage & Perplexity for 64k Vocab.

Set	#Words	% OOV	BI-Perp	TRI-Perp
Dev	7,387	0.53	230.0	137.2
Evl	8,186	0.79	231.3	137.6

In our system, the acoustic models are based on mixtures of continuous densities and so far, the adaptation scheme concerned only the mean vectors [9], the mixture weights being kept fixed. Table 4 summarizes the 64k-word recognition results.

Table 4: NAB'94 Recognition Results for 64k Vocab.

Set	BI-WER	WG-D	TRI-WER	#Arcs	RT
Dev	14.7%	38	11.7%	6.2k	1.8
Evl	14.8%	108	11.5%	24.5k	3.2

The density of the word graphs is expressed in terms of the average number of word hypotheses per spoken word (WG-D). Note that for the evaluation set we used much larger beam widths to minimize the risk of search errors. The average number of arcs expanded per sentence during the trigram graph-search is also given together with the real-time factor (RT) of the decoding on a DEC Alpha workstation. About 75% of the CPU time is actually devoted to the log-likelihood computations. Speaker adaptation brings a relative improvement of about 5% while a trigram reduces the errors by about 20% with respect to a bigram LM.

6. REFERENCES

[1] Murveit H., Butzberger J., Digalakis V., Weintraub M., "Large-Vocabulary Dictation using SRI's Decipher Speech Recognition System: Progressive Search Techniques.", *Proc. ICASSP'93*, Minneapolis, MN, USA, Vol. II, pp 319-322, 1993.

[2] Oerder M., Ney H., "Word Graphs: An Efficient Interface Between Continuous-Speech Recognition and Language Understanding", *Proc. ICASSP'93*, Minneapolis, MN, USA, Vol. II, pp 119-122, 1993.

[3] Aubert X., Dugast C., Ney H., Steinbiss V., "Large Vocabulary Continuous Speech Recognition of Wall Street Journal Corpus.",*Proc. ICASSP'94*, Adelaide, Australia, Vol. II, pp. 129-132, 1994.

[4] Gauvain J.L., Lamel L.F., Adda G., Adda-Decker M., "The LIMSI Continuous Speech Dictation System: Evaluation on the ARPA Wall Street Journal Task.", *Proc. ICASSP'94*, Adelaide, Australia. Vol. I, pp. 557-560, 1994.

[5] Ney H., Aubert X., "A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition", *Proc. ICSLP*, Yokohama, Japan, pp 1355-1358, 1994.

[6] Haeb-Umbach R., Ney H., "Improvements in Time-Synchronous Beam Search for 10000-Word Continuous Speech Recognition.", IEEE Trans. on Speech and Audio Processing, Vol. 2, pp 353-356, April 1994.

[7] Schwartz R. and Austin S., "A Comparison of Several Approximate Algorithms for Finding Multiple (N-BEST) Sentence Hypotheses", *Proc. ICASSP'91*, Toronto, Canada, pp 701-704, 1991.

[8] Giachin E.P., Rosenberg A.E. and Lee Chin-Hui, "Word Juncture Modeling using Phonological Rules for HMM-Based continuous Speech Recognition", Computer Speech and Language, Vol 5, pp 155-168, 1991.

[9] Dugast C., Aubert X. and Essen U., "An Algorithm for Unsupervised Incremental Speaker Adaptation of Continuous Mixture Parameters", to be published.