ONE PASS CROSS WORD DECODING FOR LARGE VOCABULARIES BASED ON A LEXICAL TREE SEARCH ORGANIZATION

Xavier L. Aubert

Philips Research Laboratories Aachen, Weißhausstraße 2, D-52066 Aachen, Germany
E-mail: {aubert}@pfa.research.philips.com

10.21437/Eurospeech.1999-133

ABSTRACT

This paper describes the new Philips Research decoder that performs large vocabulary continuous speech recognition in a single pass for cross-word acoustic models and an m-gram language model (with m up to 4) as opposed to our previous technique of multiple passes. The decoder is based on a time-synchronous beam search and a prefix tree structure of the lexicon. Cross-word transitions are treated dynamically. A language-model look-ahead technique is applied on the bigram probabilities.
On a variety of speech data, reduced error rates are obtained together with significant speed-ups confirming the advantage of an early use of all available knowledge sources. In particular, the search effort of a one-pass trigram decoding is only marginally increased compared to bigram and the integration of cross-word triphones improves the overall accuracy by typically 10% relative.

1. INTRODUCTION

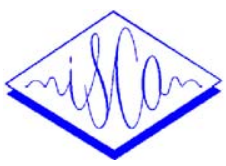
This paper presents an extension of the bigram search algorithm published in [1] to m-gram language models with $m > 2$ and cross-word acoustic models, for the purpose of large vocabulary continuous speech recognition.
As opposed to the technique of successive decoding passes, the present algorithm proceeds in one single pass by integrating all available knowledge sources, similar to the decoder described in [2]. This one-pass decoder is thus meant to replace the multiple step strategy we have been using in the past, where a bigram language model (LM) and within-word (WW) models are first applied to produce a word lattice which is subsequently rescored with a trigram [3, 4] and where cross-word models are typically applied on N-Best lists [5].
The decoder is based on a time-synchronous left-to-right beam search technique with a prefix tree structuring of the lexicon and relies on a list organization of word-conditioned partial hypotheses [1].
The generalization to longer-span m-gram language models simply follows from a proper definition of word end nodes that depend on their $m-1$ predecessor word history and from the use of a hash table to insure the efficiency of the recombination stage [7, 8]. A related topic which has also been addressed in several other systems [2, 9] concerns the early use of language model constraints. In the present work, bigram scores are smeared over the lexical tree following the method described in [6]. This leads to significantly enhanced pruning capabilities, compared to the smearing of unigram scores we used before [3, 10].
The extension to cross-word (CW) contexts involves (1) a data-driven expansion of the fan-out arcs at word-end, (2) a modified definition of the word end node taking account of the fan-out right context and (3) a fast selection of compatible root arcs for the next word startups. The handling of "unseen" context-dependent (CD) states is achieved with decision trees.
In the current design, the part of the search network that is *statically* expanded and stored is reduced to a single prefix tree structure of the lexicon, called the generic lexical tree, without multiple (fan-in) arcs in the first generation and without multiple (fan-out) arcs at word-ends. These cross-word specific aspects are treated *dynamically* with little overhead as explained in section 4.
This new decoder has been tested on a variety of speech data ranging from clean dictation to spontaneous broadcast recordings [11] and has shown its ability of performing accurate one-pass decoding for large vocabularies. In this study, three different setups have been considered with a vocabulary of 5K, 20K and 64K, respectively. Results show that the integration of cross-word models provides significant gains in accuracy and that the whole computational effort is only marginally increased when decoding is performed with a trigram instead of a bigram, confirming the advantages of an early use of all available knowledge sources.

2. SEARCH SPACE REPRESENTATION

Let N_W be the vocabulary size and N_P the number of distinct context-independent (CI) phonemes. Given the transcription of a word w, $BP(w)$ denotes its first phoneme. The subset of words starting with a given phoneme p is written as $W(p) = \{w \in Lex | BP(w) = p\}$ and there are N_{BP} such subsets. $W(*)$ thus means all words in the lexicon.

2.1. M-Gram Language-Model Constraints

The use of a probabilistic m-gram LM means that the search network is fully branched at the word level and that the word probabilities depend on their $m-1$ predecessors. The dynamic programming principle imposes to keep track of the individual $m-1$ word histories until the optimization step can take place at the next word ending. This leads to the definition of word-end recombination nodes that are made dependent on the last $m-1$ words. When using within-word (WW) acoustic models, each word end node is further connected to the whole set of words in the lexicon. This actually means re-entering the lexical prefix tree structure at its root which has been sometimes referred to as "using word-conditioned copies of the lexical tree". However, this is just a mental view, the lexical tree structure being only stored once.
As an example, for decoding with a trigram and within-word models, there is a total of $(N_W \times N_W)$ word pair nodes each one linked to the whole lexicon:



6th European Conference on Speech Communication and Technology (EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

$$WW\ 3G\ Node\ \{u,v\} \rightarrow w\ :\ P(w|u,v)\ \ \forall\ w\ \in W(*)$$

2.2. Cross-Word Contextual Constraints

When performing a left-to-right time-synchronous search with CD cross-word models, multiple contexts have to be considered at word end to anticipate for the next successor words. This leads to the so-called fan-out expansion taking place at word-endings, where the last phoneme arc is given several instances, each one with another right conditioning context. In the absence of context tying, the number of fan-out arc instances equals N_{BP}, the number of phonemes occuring at word start in the lexicon. Figure 1 illustrates the general cross-word transition pattern.

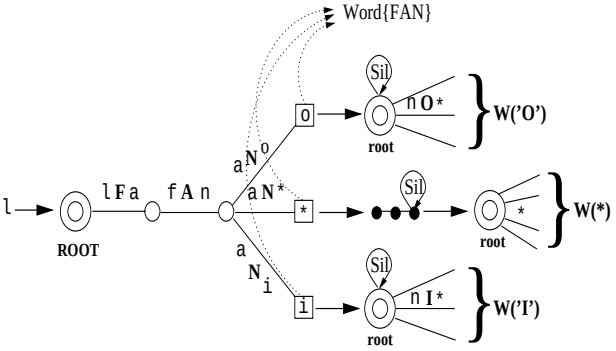


Figure 1: Cross-Word Transitions with Optional Pauses

It would however be prohibitive -besides being useless- to proceed to a static expansion of these fan-out arcs in the lexical tree as their total number ($N_W \times N_{BP}$) largely exceeds the number of arcs in the whole tree. For example, in a 64K task there are about 3 million fan-out arcs in total. In this decoder, fan-out arcs are handled dynamically when a word-end arc is activated by the beam search engine, as explained in section 4.

Another key difference with respect to within-word contexts is that a particular fan-out arc may only be followed by words whose first phoneme matches the (right) output context. This has a clear implication on the word-end recombination scheme since the set of word successors now depends on the particular fan-out arc instance. Consequently, the definition of a word-end node is augmented with the identity of the fan-out right conditioning context. For example, when using a trigram and CW models word end nodes are now defined as :

$$CW\ 3G\ Node\ \{u,v;r\} \rightarrow w\ :P(w|u,v)\ \ \forall\ w\ \in W(r)$$

Concerning the next word startups, a CW word-end node is linked to the subset of compatible successors given by W(r) and the left context needed to specify the triphones at the beginning of next words is provided by the last phoneme of the last word in the LM history. Both the selection of admissible CW successors and the triphone expansion are also done dynamically "on the fly".

Last, optional pauses may occur between two consecutive words. Depending on the silence duration, the word transition can be treated as coarticulated or not, the latter case implying that any word might follow (See figure 1).

2.3. Generic Lexical Tree Structure

As a consequence of this decoder's design, the only structure that is *statically* expanded and stored is the generic lexical tree which is constructed from CD expanded transcriptions using the "wild-card" symbol at the begin and end of each word. Allophones that have identical state sequences consecutive to tying are merged.

Table 1: Characteristics of Generic Lexical Trees

Task Size	5K	20K	64K
#Base Words	4987	19980	64737
#Lex. Entries	5608	22370	69975
#Generations	16	16	18
Total #Arcs	18637	67143	198256
#Arcs 1st Gen.	457	612	827

Some characteristic figures of the generic lexical tree structures used in this study are given in table 1. About 12% of the lexical entries are pronounciation variants. Particularly relevant is the slow increase of the arc number in the first generation as a function of the vocabulary.

3. DATA DRIVEN BEAM SEARCH

Decoding proceeds time-synchronously by extending the most promising paths according to the underlying search space representation described above. Each active word-end node in turn activates the admissible arcs of the first-generation and the next within-word arcs are further proposed by the tree structure. Active paths are recorded in lists based on a triple hierarchy : word-end nodes, arcs and states [1]. Only the DP quantities (score, backpointer etc.) that are relevant to the still active paths are stored. When a word-end arc is reached, hypotheses related to each fan-out arc instance are dynamically inserted in the lists (see next section). The whole propagation process is controlled by a beam pruning technique. Peaks in terms of large number of active states are handled by the so-called histogram pruning technique [10] making possible to work with fixed size lists.

A word hypothesis is generated when a tree leaf is reached. Recombination at the word level is efficiently solved by associating a bijective hash index to each word-end node. For a trigram node $\{u, v;\ r\}$, it is defined as

$$H(u,v;\ r) = M_P * (M_W * u + v) + r, \quad u,v,r > 0,$$

with the constants $M_W > N_W$ and $M_P > N_P$, which allows to easily retrieve both the LM history and acoustic context by successive modulo and division operations. This method is fairly general in its principle (apart from secondary range problems) and has been successfully applied so far up to a fourgram LM using a hash table of moderate size.

4. DYNAMIC HANDLING OF CROSS-WORD TRANSITIONS

The fan-out arcs of a particular word end are very efficiently handled by a "cloning" mechanism as all these fan-out arcs actually share most of their attributes in common (start-time, entry-score, back-pointer and word-identity) but for the right conditioning context and the identity of the mixture states. In addition, a particular form of

LM look-ahead pruning is applied on these fan-out arcs as described in section 5, before inserting new paths in the search lists.
Concerning optional pauses between consecutive words, two cases are considered as shown in figure 1. The first one occurs for the fan-out arc labeled with a "wild-card" right context : a "long" silence must follow before re-entering the global tree at its root. This non-coarticulated word transition is controlled by imposing a minimum duration to the pause. The second case is the cross-word coarticulated transition following fan-out arcs with a specific right context where a "short" pause might still be inserted provided it is not longer than a specified duration.
When re-entering the lexical tree, two actions have to be performed. First, the selection of the tree-root arcs whose phone label matches the right fan-out context. This is made especially simple when the words of the lexicon are sorted on their first phoneme. Next, the mixture state indices have to be identified for the instantiated left context given by the last phone of the preceeding word using the decision trees and a look-up table.

5. SMEARING OF LANGUAGE MODEL PROBABILITIES

A well known problem when using a prefix tree is that word identities are only known at the tree leaves. Postponing the use of the language model probabilities up to this point is disadvantageous since (1) the LM predictive capabilities are delayed and (2) the DP accumulated scores incur clear discontinuities at word-ends, both factors affecting the pruning efficacy.
The solution consists in distributing the LM scores across the lexical tree by factorizing the word probabilities such that they can be applied incrementally at each phone arc, a process we call "LM smearing" [10]. Smearing the exact m-gram LM scores appears computationally expensive due to the dependency on the $m-1$ predecessors. Therefore in [3] a simplified approach has been applied by smearing unigram scores that can be easily pre-processed and stored [10].
In the present work, bigram scores are smeared following an approach similar to [6]. This involves a compact tree configuration and a LM cash strategy in conjunction with fast access to the LM proba's such that any needed partial bigram scores can be made available on demand.
A situation of particular interest occurs at the CW fan-out arcs since the identity of the word v being produced is already known as well as the subset $W(r)$ of words that may follow. Defining the upper bound over the bigram probabilities of joining the next admissible word,

$$\hat{P}(w^*|v,r) = MAX_{\forall w \in W(r)} \; P(w|v),$$

r being the fan-out right context, an additional language model look-ahead pruning is performed before activating a given fan-out arc instance in the search lists.
As shown in our results (See table 4 in section 6), when bigram-scores are smeared instead of unigram-scores, the active search space is significantly reduced and less search errors occur, however at the expense of significant memory overhead.

6. RESULTS

All decoding results presented in this section have been obtained with the standard Philips acoustic modeling [12] whose main features can be summarized as follows :

- MFCC analysis followed by LDA in a 35 dim. space
- Mixtures of Laplacians with single pooled deviation
- Tied triphone states handled with decision trees.

This new decoder was first applied in the Philips Hub-4 evaluation of Nov'98 as described in [11]. In this study, three different setups have been considered with a vocabulary of respectively 5K, 20K and 64K. Acoustic HMMs have been trained gender-dependently on WSJ0+1 (142 females & 142 males). Likelihood computations are sped up using standard techniques. All Word Error Rates (WER) presented here have been computed with a simple Levenshtein metric, not taking account of possible splits and merges. or equivalent pairs like "I'm" and "I am".
The 5K task has been evaluated on a large amount of data (approx. 3 hours of recordings) made of the development and evaluation sets of Nov'92 and Nov'93, a total of 38 speakers, 1468 sentences and 24630 spoken words. The Out-Of-Vocabulary (OOV) rate is 0.37% with the 5K official word list.
Both the 20K and the 64K setups are evaluated on the North American Business corpus using the development and evaluation sets of Nov'94. This represents about 1,7 hours of speech of 40 speakers, with a total of 626 sentences and 15573 spoken words. For the 20K setup, the official CMU trigram has been applied and the OOV rate equals 2.55%. For the 64K setup, we used a Philips trigram LM trained over 238 million of words from the NAB corpus, the OOV rate being reduced to 0.64%.

Table 2: Word Error Rate for the 5K WSJ Task Bigram to Fourgram, WW to CW Models

LM Range	WW	CW	Rel. CW Gain
Bigram	6.95 %	6.05 %	-12.9 %
Trigram	4.90 %	4.23 %	-13.7 %
Fourgram	4.65 %	4.01 %	-13.9 %
Rel. LM Gain	-32.9 %	-33.7 %	-

Table 2 contains the WER obtained on the 5K task when using longer span (word-based) language models, from bigram to fourgram and when applying within-word (WW) or cross-word (CW) triphone models. Comparing the error rates obtained with language models of increasing range, these results show a relative WER reduction of 29% for the trigram LM versus the bigram while the fourgram brings another 5% improvement versus trigram. Referring to our previous two-step strategy that relies on the word-pair approximation [3], the accuray achieved with the one pass trigram decoder appears slightly higher, by about 2% relative.

Table 3: Search Effort for 5K Task with WW Models

LM Range	Av.# of Active States, Arcs, Nodes	Rel. CPU Measure
Bigram	5980, 1710, 25	100
Trigram	6400, 1830, 27	106
Fourgram	7240, 2095, 34	114

Some typical figures concerning the search effort are given in table 3, in terms of average number of active states, arcs and word-end nodes that are processed in the search lists

for the various language models. The rightmost column contains a relative measure of the CPU time needed for performing the whole decoding including the log-likelihood computations. It can be seen that the increase of the overall decoding cost is relatively small when applying a trigram or a fourgram instead of a bigram.

Table 4: Search Effort for 20K Trigram Decoding Smearing of Bigram versus Unigram Scores

Smeared LM	Unigram	Bigram	Ratio
# States	20700	9100	2.3
# Arcs	6050	2700	2.2
# Nodes	67	49	1.4
WER (%)	13.06%	13.05%	-
Rel. CPU	100	73.7	-26%

The impact of LM score look-ahead is clearly observed by comparing the search efforts needed for performing a 20K trigram decoding either with unigram smearing [10] or bigram smearing [6]. The figures presented in table 4 have been obtained after adjusting the beam widths such that both runs achieved almost the same error rates. In this conservative setup, bigram smearing reduces the overall decoding costs by one quarter but further savings can be obtained when accepting some slight degradation.

Table 5: Word Error Rate on the NAB'94 Task for 20K or 64K Trigram, WW or CW Models

Vocab. & LM	WW	CW	Rel. CW Gain
20K Trigram	13.05%	11.85%	-9.2 %
64K Trigram	10.07%	9.10%	-9.6 %
Rel. LM Gain	-22.8%	-23.2%	-

Finally, the NAB'94 task has been decoded using trigram LMs for 20K and 64K words respectively, looking again at the impact of cross-word triphones versus within-word. As can be observed in table 5, when the vocabulary is enlarged to 64K, the number of errors drop by more than one fifth consecutive to the reduction of the OOV words, from 2.55% to 0.64%. On the other hand, cross-word decoding brings an improvement of about 10% compared to the corresponding within-word accuracies.

7. CONCLUSION

A one-pass cross-word decoder has been described and evaluated on several large-vocabulary continuous-speech recognition tasks with up to a fourgram LM. The general architecture is based on a (single) generic lexical tree structure, the cross-word transitions being dynamically treated in the course of a data driven left-to-right beam search algorithm. Though the underlying search network is by no means "optimal" in terms of minimal branching, coupling a prefix tree with powerful pruning strategies including the look-ahead of bigram scores, leads to a flexible and efficient recognition engine.

8. ACKNOWLEDGEMENT

I would like to thank all my colleagues at Philips Research in Aachen, especially Hans Dolfing with whom I discussed many software issues, Dietrich Klakow who provided the language models and Reinhard Blasig who managed to get a fast access to the LM probabilities !

9. REFERENCES

[1] Ney, H., Haeb-Umbach, R., Tran, B.-H. & Oerder M., "Improvements in beam search for 10000-word continuous speech recognition, in Proceedings of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing, San Francisco, CA, pp. 13-16, March 1992.

[2] Odell J.J., Valtchev V., Woodland P.C. & Young S.J., "A One Pass Decoder Design for Large Vocabulary Recognition" in Proceedings of ARPA Spoken Language Technology Workshop, Plainsboro, NJ, pp. 405-410, March 1994.

[3] Aubert X., Dugast C., Ney H. & Steinbiss V., "Large Vocabulary Continuous Speech Recognition of Wall Street Journal Corpus", in Proceedings of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Adelaide, Australia, pp. 129-132, April 1994.

[4] Aubert, X. and Ney, H., "Large Vocabulary Continuous Speech Recognition using Word Graphs" in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Detroit, MI, 1995, pp. 49–52, May 1995.

[5] Beyerlein, P., Aubert X., Haeb-Umbach R., Klakow D., Ullrich M., Wendemuth A. and Wilcox P., "Automatic Transcription of English Broadcast News" in Proceedings of DARPA Broadcast News Transcription and Understanding Workshop", Lansdowne, Virginia, Feb. 8-11 1998.

[6] Ortmanns S., Ney H. & Eiden A., "Language-Model Look-Ahead for Large Vocabulary Speech Recognition" in Proceedings of the ICSLP 1996, Philadelphia, p. 2095-2098, October 1996.

[7] Ortmanns S., Ney H. & Lindam, I., "A Comparison of Time Conditioned and Word Conditioned Search techniques for Large Vocabulary Speech Recognition", in Proceedings of the ICSLP 1996, Philadelphia, p. 2091-2094, October 1996.

[8] Ortmanns S., Ney H. & Aubert, X., "A word graph algorithm for large vocabulary continuous speech recognition" in Computer Speech and Language (1997) 11, 43-72.

[9] Alleva, F., Huang, X. & Hwang, M.-Y., "Improvements on the pronunciation prefix tree search organization", Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Atlanta, GA, pp. 133-136, May 1996.

[10] Steinbiss V., Tran B.-H., Ney, H. : "Improvements in Beam Search", in Proceedings of the Inter. Conf. on Spoken Language Processing, Yokohama, Japan, pp. 2143-2146, Sep. 1994.

[11] Beyerlein, P., Aubert X., Haeb-Umbach R., Harris M., Klakow D., Wendemuth A., Molau S., Pitz M. & Sixtus A., "The Philips/RWTH System for Transcription of Broadcast News", elsewhere in this Eurospeech'99 proceedings.

[12] Haeb-Umbach R., Aubert X., Beyerlein, P., Klakow D., Ullrich M., Wendemuth A. and Wilcox P., "Acoustic Modeling in the Philips Hub-4 Continuous-Speech Recognition System" in Proceedings of DARPA Broadcast News Transcription and Understanding Workshop", Lansdowne, Virginia, Feb. 8-11 1998.