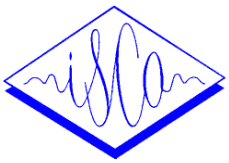

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

# EXTENDING THE GENERATION OF WORD GRAPHS FOR A CROSS-WORD M-GRAM DECODER

*Christoph Neukirchen, Xavier Aubert, Hans Dolfing*

Philips Research Laboratories, Weisshausstr. 2, 52066 Aachen, Germany
Email: {Christoph.Neukirchen, Xavier.Aubert, Hans.Dolfing}@philips.com

## ABSTRACT

This paper introduces a method for constructing word graphs in the extended decoding framework of $m$-gram language models ($m > 2$) and cross-word HMMs. The generation of word hypotheses contained in the graph relies on a word $m$-tuple boundary optimization extending the word-pair approximation. Two variants of graph generation are proposed: the first one fully encodes the cross-word and LM constraints used in the search into the graph structure which leads to compact sized graphs. The second method constructs word graphs with lower order constraints compared to those used in the search, resulting in larger graphs with lower graph error rates. Results are presented from systematic experiments carried out on the 5k WSJ and the 64k NAB tasks.

## 1. INTRODUCTION

Word graphs are commonly used in speech recognition systems to compactly represent many alternative word and phrase hypotheses that can be rescored in a multiple-pass strategy [4, 5, 6, 7].

In our previous decoding framework [7, 1] based on within-word acoustic models and bigram language models (LMs), the word-pair approximation [9] has been assumed for constructing word graphs. Under this assumption, the dependence of the time boundary between two words is limited to the immediate predecessor word, which appears to be valid in most cases [1]. This boundary optimization is actually a result of the dynamic programming (DP) search strategy that is conditioned on the predecessor word. Such implicit bigram boundary optimization generates a word-pair hypothesis at most once for any ending time and thus keeps the graph size small by avoiding redundant phrase arcs.

An alternative way to construct word graphs is based on the within-word DP recombination of time-conditioned hypotheses [5, 8]. By construction, this method does not rely on the word-pair approximation. In general, the time-conditioned method produces many redundant word hypotheses that have to be subsequently reduced by an explicit boundary optimization step at the word level.

The recent progress made in time-synchronous beam search allows to integrate complex knowledge sources like cross-word HMMs and $m$-gram LMs ($m > 2$) in the first pass of the decoder [4, 2]. It will be shown that the usage of such longer span context modeling and the corresponding search space constraints that have to be considered in the decoder have clear implications on the properties of the resulting word graphs, namely, on the size of the graph and the quality of the hypotheses within the graph. The strategy presented here for constructing the word graph in this framework, is based on the word $m$-tuple assumption and keeps track of those hypotheses that are (locally) most probable, even if they do not survive the recombination process at the word level.

## 2. WORD-HISTORY CONDITIONED DP DECODING

The word graph generation is integrated in a LVCSR one-pass decoder [2] that uses cross-word phonetic HMMs and $m$-gram LMs. The decoding strategy is based on the time-synchronous, left-to-right DP beam search. The DP works on partial hypotheses that are conditioned on their predecessor word history according to the LM $m$-gram order. The word lexicon is structured as a phonetic prefix tree.

To describe the generation of hypotheses, we introduce the following definitions (cf. [7]):

- ${}^{d}w^{c}$ : a word $w$ in the cross-word left and right phonetic contexts $d$ and $c$. The final phoneme of a word $w$ is given by $r(w)$; the initial phoneme is given by $l(w)$. The vocabulary size is $N_w$, the number of different phonemes is $N_p$.
- $\mathbf{W}^{c}$ : sequence of any $N$ words $\mathbf{W}^{c} = (w_1, \ldots, w_N^{c})$ that is followed by a word $w_{N+1}$ with $l(w_{N+1}) = c$.
- $S_m(\mathbf{W}^{c})$ : the $m$-gram LM-state of a sequence $\mathbf{W}^{c}$ is given by the $(m-1)$ most recent words augmented by the phonetic context $c$: $S_m(\mathbf{W}^{c}) = (w_{N-m+2}, \ldots, w_N^{c})$.
- $h({}^{d}w^{c}; \tau, t)$ : probability $p(\mathbf{X}_{\tau+1}^{t} | {}^{d}w^{c})$ that word ${}^{d}w^{c}$ generates the acoustic vectors $\mathbf{X}_{\tau+1}^{t} = x(\tau+1), \ldots, x(t)$.
- $G(\mathbf{W}^{c}; t)$ : joint probability $p(\mathbf{X}_1^{t} | \mathbf{W}^{c}) \cdot P(\mathbf{W})$ of generating the acoustic vectors $\mathbf{X}_1^{t}$ *and* a word sequence $\mathbf{W}^{c}$ with ending time $t$, including the LM probability.
- $H(S_m; t)$ : joint probability of generating the acoustic vectors $\mathbf{X}_1^{t}$ *and* a word sequence with the final $(m-1)$ words given by $S_m$ at ending time $t$.

Two DP optimization steps done during decoding are important for the properties of the generated word graph: within-word recombination and LM-state recombination.

### 2.1. Within-word hypotheses recombination

The DP optimization at the HMM-state level within the words $w$ in the lexicon tree is conditioned on the predecessor hypotheses' LM-states $S_m(\mathbf{W}^{c})$. In the cross-word acoustic model case, only the words $w$ that are phonetically compatible with the condition $S_m(\mathbf{W}^{c})$ have to be considered, i.e. $l(w) = c$.

The within-word recombination is applied to partial hypotheses $\mathbf{W}$ being in an identical LM-state $S_m(\mathbf{W})$ but having entered the tree at different starting times $\tau$. When the DP reaches a tree leaf (for word $w^{d}$) this results in an extension of the preceding hypothesis $\mathbf{W}$ to the new hypothesis $\tilde{\mathbf{W}}^{d} = (\mathbf{W}, w^{d})$ and in an implicit optimization of the last word boundary, i.e. the tree re-entering time

$$G(\tilde{\mathbf{W}}^d;t) = P(w|S_m(\mathbf{W}))\cdot \max_{\tau}\{H(S_m(\mathbf{W}^{l(w)});\tau)\cdot h(^{r(\mathbf{W})}w^d;\tau,t)\} \quad (1)$$

Here, the m-gram LM probability $P(w|S_m(\mathbf{W}))$ of the word $w$ given the preceding words in $\mathbf{W}$ is included.

From Eq. 1 follows that at each fixed time $t$ any truncated sequence of $m$ words (including the phonetic context $d$) can occur only once and the recent word boundary is optimized accordingly. So *before* the subsequent LM-state recombination is invoked, all new partial sentence hypotheses $\tilde{\mathbf{W}}^d$ differ (at least) in their $(m+1)$-gram LM-state $S_{m+1}(\tilde{\mathbf{W}}^d)$. The maximum number of new partial sentence hypotheses at time $t$ is $N_p\cdot(N_w)^m$.

Furthermore, the dependence of the word boundary between $\mathbf{W}$ and $w^d$ in $\tilde{\mathbf{W}}^d$ ending at $t$ is confined to the identity of the $m$ most recent words in $\tilde{\mathbf{W}}^d$ as given by Eq. 1:

$$\tau(\tilde{\mathbf{W}}^d;t) = \tau(S_{m+1}(\tilde{\mathbf{W}}^d);t) = \operatorname*{argmax}_{\tau}\{H(S_m(\mathbf{W}^{l(w)});\tau)\cdot h(^{r(\mathbf{W})}w^d;\tau,t)\} \quad (2)$$

### 2.2. LM-state recombination

At each time $t$, the LM-state DP optimization is invoked for all new word-end hypotheses. This word-end optimization recombines all partial sentence hypotheses $\tilde{\mathbf{W}}^d$ that are equivalent with respect to their $m$-gram LM-state (i.e. sharing the same last $(m-1)$ words):

$$H(S_m(\mathbf{W}^c);t) = \max_{\substack{\tilde{\mathbf{W}},d \\ S_m(\tilde{\mathbf{W}}^d)=S_m(\mathbf{W}^c)}}\{G(\tilde{\mathbf{W}}^d;t)\} \quad (3)$$

Efficient DP recombination can be realized by hashing the hypothesized LM-states $S_m(\mathbf{W}^c)$ [2].

Thus, *after* LM-state recombination at any fixed time $t$ all surviving partial sentence hypotheses $\mathbf{W}^c$ differ (at least) in their LM-state $S_m(\mathbf{W}^c)$. The maximum number of surviving hypotheses is $N_p\cdot(N_w)^{m-1}$.

Due to the combinatorial complexity, the average number of distinct LM-states after LM-state recombination typically increases when moving to longer span LMs and cross word acoustic models. In practice, this increase is moderate due to the improved beam pruning based on sharper knowledge sources [2].

## 3. WORD GRAPH GENERATION

The word graph to be generated is structured as an acyclic, directed, weighted graph defined over a set of edges and nodes. An edge in the graph corresponds to a single word hypothesis $w$ containing the word acoustic score and its LM score. The graph edges are connected by nodes which are associated with the word boundary time $t$. In addition, the nodes are used to ensure the proper structural $p$-gram constraints in the graph structure: all partial paths ending in the same graph node must have an identical $(p-1)$ word history. Thus, each node corresponds to a unique $p$-gram LM-state and there may be several nodes at time $t$ for $p>1$,

### 3.1. Word graph algorithm

The time-synchronous generation of the word graph relies on the word hypotheses information provided by the $m$-gram decoder. The following algorithm constructs a $p$-gram constrained word graph in a single pass with $p \le m$:

1. At each time $t$ consider all word $m$-tuples and phonetic contexts $c$: $\underbrace{(u,\ldots,v,w^c)}_{m\ words}$
   By Eq. 1, at time $t$ each word $m$-tuple and context is generated by the decoder at most once *before* the LM-state recombination. The beam search strategy limits the generated hypotheses $(u,\ldots,v,w^c)$ to the most probable ones.
2. For each $(m+2)$-tuple $(u,\ldots,v,w;c;t)$ keep track of the word boundary $\tau(u,\ldots,v,w^c;t)$ (provided by Eq. 2).
   Create a graph edge that contains:
   - the current word $w$
   - the word acoustic score $h(^{r(v)}w^c;\tau(u,\ldots,v,w^c;t),t)$
   - the word $p$-gram LM-score $P(w|S_p(u,\ldots,v))$
3. For each individual pair of time and $p$-gram LM-state $(t;S_p(u,\ldots,v,w^c))$ create a graph node. In the case $p=m$ the $m$-gram LM-state is provided by the decoder as a by-product of LM-state recombination (Eq. 3). For $p<m$ the $p$-gram LM-states must be stored in a separate hash table.
4. Link the graph edge of $(u,\ldots,v,w;c;t)$ with
   - the start node $(\tau(u,\ldots,v,w^c;t);S_p(u,\ldots,v^{l(w)}))$
   - the end node $(t;S_p(u,\ldots,v,w^c))$
5. Word graph management:
   In general, the within-word recombination and the beam pruning strategy prevents hypotheses from being expanded in the DP. Unexpanded hypotheses can cause dead paths in the word graph that never reach the final node in the graph. A garbage collection removes the nodes and edges being part of dead partial paths to reduce memory requirements.

### 3.2. Word graph properties

When connecting the word hypotheses edges as in step 4 using the boundaries obtained from Eq. 2 it is assumed that the word $m$-tuple approximation is true for sentences within the word graph.

Furthermore, the following graph properties are valid:

- The maximum number of nodes (p-gram states) at each time $t$ is $N_p\cdot(N_w)^{p-1}$.
- The maximum number of incoming edges per node (i.e. the number of hypotheses resulting from Eq. 1 with identical p-gram LM-state) is $(N_w)^{m-p+1}$.
- Due to the variation in word ending times the number of outgoing edges per node is not limited

For $p=m$, the $m$-tuple-based boundary optimization (Eq. 2) ensures that each sentence hypothesis is contained in the graph at most once. For $p<m$ redundant sentence hypotheses are possible. To remove such redundant paths, graph optimization algorithms can be applied [3]. In the current implementation only identical edges sharing the same start and end nodes are discarded.

In Fig. 1 an example of word hypotheses in a bigram and a trigram decoder is shown along with the generated parts of the word graphs. By using longer span LMs (here: trigram vs. bigram) the LM-state recombination is delayed and a larger number of distinct LM-states will occur, in general. In the corresponding $m$-gram constraint graph this leads to the generation of more nodes and to lower branching factors (assuming an equal number of word hypotheses). When a partial hypothesis is not further expanded due

to within-word recombination (see Fig. 1) or to pruning, the resulting dead path will be typically longer for the trigram graph. So, the final trigram word graph contains fewer edges and has smaller branching factors compared to the bigram graph. The situation is even more drastic when beam pruning is applied: the sharper the probabilities in the longer span LMs, the more aggressive the pruning will work. This leads to a further reduction of hypotheses in the $m$-gram graph.

The resulting word graph density can be increased by using a lower order $p$-gram constraint with $p < m$ ($p = 1$ in Fig. 1). In this case the original number of different graph nodes is reduced and the branching factor is increased accordingly. Typically, it reduces the length of dead paths and keeps more edges alive in the graph, even for large $m$. The resulting $p$-gram word graph contains more sentence hypotheses due to the larger amount of edges and to the higher branching factor.

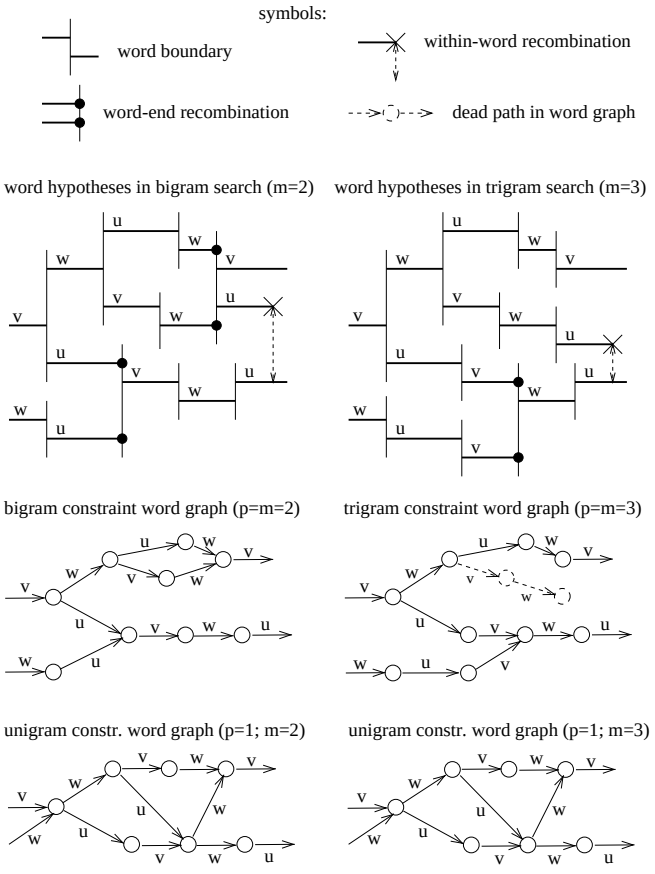


Figure 1: Generation of $m$-gram and unigram constraint word graphs using bigram ($m = 2$) and a trigram ($m = 3$) decoding.

## 4. EXPERIMENTAL RESULTS

Experimental results are presented for a 5K and for a 64K vocabulary task. The triphone HMMs are trained gender-dependently on the WSJ0+1 database. The 5K task consists of the male speakers of the Nov'92 and Nov'93 evaluation and development sets (776 sentences, 13113 spoken words, OOV: 0.16%). The 64K task consists of the female speakers in the Nov'94 development and evaluation sets of the North American Business (NAB) corpus (315 sentences, 7770 spoken words, OOV: 0.73%). In all experiments wide beam widths are used in the search to generate large graphs.

The decoder search space is specified by the following quantities: the average number per time frame of active triphone *arcs*, of *word-end* hypotheses (including the cross-word phonetic fan-out contexts), and of *LM-states* after recombination. Word-end hypotheses are the potential edges in the graph and the nodes are derived from the LM-states. The *word error rates* (WER) are given for first-pass decoding and for LM-rescoring experiments.

To specify the quality of the generated graphs the following definitions are used: The size of the word graph is given as the *word graph density* (WDG), which is the ratio between the total number of edges in the graph and the number of actually spoken words. The number of alternative hypotheses at each node is given by the *word graph branching factor* (WGBF), which is the ratio between the number of edges and the number of nodes in the graph. The *graph word error rate* (GER) is the word error rate of the sentence through the word graph that best matches the spoken sentence.

### 4.1. Word graph generation

The generation of word graphs using different kinds of acoustic models and LMs is illustrated in Table 1 for the 5K task and in Table 2 for the 64K task. Comparing the $m$-gram decoder search space when moving to longer span LMs and to cross-word HMMs demonstrates that the number of generated word end hypotheses and the number of different LM-states increases accordingly. Due to the better knowledge sources the top-best WER is improved. The decreasing ratio between the number of word ends and LM-states directly corresponds to the observed decrease (shown in part a) of the WGBF in the $m$-gram graphs for larger $m$. Although the decoder generates more word-ends for longer span context models, the sizes (WGD) of the resulting $m$-gram word graphs become significantly smaller. As explained in Sec. 3.2, this apparently paradoxical situation can be attributed to the increasing average length of dead path in addition to the stronger pruning. When increasing the $m$-gram order, the resulting graphs contain less phrases of small probabilities. With the lower graph density and the smaller branching factor, the number of sentence hypotheses in the graph is reduced which results in a significant increase in the graph word error rate (GER) in spite of improved WER.

In part b of Table 1 and Table 2 the generation of unigram constraint word graphs ($p = 1$) using a $m$-gram decoding is shown. Due to the *early* unigram-based recombination in the graph nodes, the effect of stronger beam pruning on the graph size (WGD) is much less severe compared to part a. The larger word graph sizes and the better branching factors result in a significant improvement in GER compared to the $m$-gram graphs. Furthermore the GERs that come close to the OOV rate are hardly influenced when moving to longer span context models.

### 4.2. Word graph rescoring

Table 3 shows rescoring results by applying $n = 2, \ldots, 6$-gram LMs [1] to the $m$-gram word graphs generated in Table 1 part a. The rescoring WERs on the original unpruned graphs (Table 3 part a) are quite insensitive to the $m$-gram order (and to the corresponding WGD and the GER) of the graphs. This indicates: (i) the top-best rescoring sentence hypotheses are contained in almost all graphs;

[1] $4 \ldots 6$-gram LMs provided by D. Klakow, Philips Research

| ac. model | within-word HMM | | | cross-word HMM | |
|---|---|---|---|---|---|
| LM order | m=2 | m=3 | m=4 | m=2 | m=3 |
| arcs | 1940 | 2234 | 2451 | 3209 | 3452 |
| word ends | 114 | 135 | 126 | 1354 | 1459 |
| LM-states | 38 | 83 | 88 | 625 | 952 |
| WER | 8.15% | 5.91% | 5.28% | 6.85% | 5.08% |
| a) original m-gram constraint word graph (p=m) | | | | | |
| WG constr. | p=2 | p=3 | p=4 | p=2 | p=3 |
| WGD | 1118.2 | 347.6 | 228.7 | 770.5 | 231.2 |
| WGBF | 7.5 | 3.3 | 2.7 | 4.9 | 3.0 |
| GER | 0.43% | 0.72% | 0.85% | 0.72% | 1.08% |
| b) unigram constraint word graph | | | | | |
| WG constr. | p=1 | | | | |
| WGD | 2283.6 | 2031.1 | 1614.4 | 1978.3 | 1909.9 |
| WGBF | 61.9 | 56.2 | 46.7 | 51.7 | 50.8 |
| GER | 0.27% | 0.28% | 0.29% | 0.32% | 0.30% |

Table 1: 5K WSJ task: search space and properties of m-gram (a), and unigram (b) word graphs obtained from decoding with bi-, tri-, and fourgrams and cross-word and within-word HMMs.

| a) p=m | | | b) p=1 | | |
|---|---|---|---|---|---|
| WGD | WGBF | GER | WGD | WGBF | GER |
| m=2, word ends=311, LM-states=100, WER=12.54% | | | | | |
| 1896.8 | 9.2 | 1.62% | 5341.6 | 155.3 | 0.95% |
| m=3, word ends=497, LM-states=299, WER=9.47% | | | | | |
| 712.5 | 3.9 | 2.02% | 5676.6 | 163.0 | 0.97% |

Table 2: 64K NAB task: search space and properties of generated m-gram (a), and unigram (b) constraint word graphs obtained from within-word HMM decoding with bigram and trigram LMs

(ii) the influence of different word boundaries in the graphs on the performance is small.

In the rescoring WER, no difference was observed when using the original $m$-gram word graphs instead of unigram constraint graphs (with better GERs as shown in Table 1 part b). This emphasizes the above mentioned situation: the top-scoring sentence hypotheses are already contained in the $m$-gram word graphs, so the additional hypotheses in the unigram graphs do not contribute to the first-best result.

In part b of Table 3 rescoring results on pruned $m$-gram word graphs are shown. By application of forward-backward pruning [4] the WGDs are reduced to 6.0 and the WGBF to 1.6 approximately. Now, there is a clear advantage for the word graphs that are generated by using longer span context models since edges are pruned based on these enhanced context knowledge sources: both the GER and the rescoring top-best WER for the different LMs is improved when more detailed models are used in the word graph generation. Note that the strong graph pruning based on the $p$-gram probabilities can improve the WER for $n$-grams with $n < p$.

## 5. CONCLUSION

An extension of the generation of word graphs using long span LMs and cross-word HMMs has been described and evaluated on LVCSR tasks. The graph generation algorithm is based on the word $m$-tuple assumption for word boundaries. The resulting graphs contain the full contextual model information and they become more compact when the range of context dependence is increased. When there is interest in dense word graphs with extremely low GER, the method of falling back to lower order graph constraints has been proposed.

| ac. model | within-word HMM | | | cross-word HMM | |
|---|---|---|---|---|---|
| graph order | p=2 | p=3 | p=4 | p=2 | p=3 |
| a) unpruned word graph | | | | | |
| WGD | 1118.2 | 347.6 | 228.7 | 770.5 | 231.2 |
| WGBF | 7.5 | 3.3 | 2.7 | 4.9 | 3.0 |
| GER | 0.43% | 0.72% | 0.85% | 0.72% | 1.08% |
| | n-gram rescoring WER | | | | |
| n=2 | 8.15% | 8.11% | 7.64% | 6.85% | 6.79% |
| n=3 | 5.86% | 5.91% | 5.89% | 5.08% | 5.08% |
| n=4 | 5.27% | 5.25% | 5.28% | 4.64% | 4.84% |
| n=5 | 5.23% | 5.22% | 5.24% | 4.73% | 4.77% |
| n=6 | 4.98% | 4.96% | 5.00% | 4.41% | 4.48% |
| b) pruned word graph (pruning thresh.=30000) | | | | | |
| WGD | 5.8 | 6.0 | 6.5 | 5.3 | 5.4 |
| WGBF | 1.6 | 1.6 | 1.6 | 1.6 | 1.5 |
| GER | 3.57% | 2.93% | 2.67% | 3.24% | 2.78% |
| | n-gram rescoring WER | | | | |
| n=2 | 8.15% | 6.41% | 5.57% | 6.85% | 5.53% |
| n=3 | 6.23% | 5.91% | 5.44% | 5.21% | 5.08% |
| n=4 | 5.76% | 5.15% | 5.28% | 4.98% | 4.84% |
| n=5 | 5.77% | 5.07% | 5.14% | 4.94% | 4.63% |
| n=6 | 5.68% | 4.93% | 4.91% | 4.83% | 4.56% |

Table 3: 5K WSJ task: rescoring results with n-gram LMs on m-gram constraint word graphs. In a) the original word graph is rescored, in b) a pruned graph is used.

## 6. REFERENCES


[1] X.L. Aubert, H. Ney, 'Large Vocabulary Continuous Speech Recognition Using Word Graphs', *Proc. of ICASSP, Detroit,* 1995, pp. 49–52.

[2] X.L. Aubert, 'One Pass Cross Word Decoding for Large Vocabularies Based on a Lexical Tree Search Organization', *Proc. of Eurospeech, Budapest,* 1999, pp. 1559–1562.

[3] M. Mohri, M. Riley, 'Weighted Determinization and Minimization for Large Vocabulary Speech Recognition', *Proc. of Eurospeech, Rhodes,* 1997, pp. 131–134.

[4] J.J. Odell, 'The Use of Context in Large Vocabulary Continuous Speech Recognition', *PhD Thesis, Engineering Department, Cambridge University,* 1995.

[5] M. Oerder, H. Ney, 'Word Graphs: An Efficient Interface between Continuous Speech Recognition and Language Understanding', *Proc. of ICASSP, Minneapolis,* 1993, pp. 119–122.

[6] H. Murveit, et.al. 'Large Vocabulary Dictation Using SRI's Decipher Speech Recognition System: Progressive Search Techniques', *Proc. of ICASSP, Minneapolis,* 1993, pp. 319–322.

[7] H. Ney, X.L. Aubert, 'A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition', *Proc. of ICSLP, Yokohama,* 1994, pp. 49–52.

[8] H. Ney, S. Ortmanns, I. Lindam, 'Extensions to the Word Graph Method for Large Vocabulary Continuous Speech Recognition', *Proc. of ICASSP, Munich,* 1997, pp. 1791–1794.

[9] R. Schwartz, S. Austin, 'A Comparison of Several Approximate Algorithms for Finding Multiple (N-Best) Sentence Hypotheses', *Proc. of ICASSP, Toronto,* 1991, pp. 701–704.