EUROSPEECH '97
5th European Conference on Speech
Communication and Technology
Rhodes, Greece, September 22-25, 1997

A STATIC LEXICON NETWORK REPRESENTATION FOR CROSS-WORD CONTEXT DEPENDENT PHONES[*]

Kris Demuynck, Jacques Duchateau and Dirk Van Compernolle [†]

K. U. Leuven - ESAT., Kardinaal Mercierlaan 94, B-3001 Heverlee, Belgium
E-mail: Kris.Demuynck@esat.kuleuven.ac.be

ABSTRACT

To cope with the prohibitive growth of lexical tree based search-graphs when using cross-word context dependent (CD) phone models, an efficient novel search-topology was developed.

The lexicon is stored as a compact static *network* with no language model (LM) information attached to it. The static representation avoids the cost of dynamic tree expansion, facilitates the integration of additional pronunciation information (e.g. assimilation rules) and is easier to integrate in existing search engines. Moreover, the network representation also results in a compact structure when words have alternative pronunciations, and due to its construction, it offers partial LM forwarding at no extra cost.

Next, all knowledge sources (pronunciation information, language model and acoustic models) are combined by a slightly modified token-passing algorithm, resulting in a one pass time-synchronous recognition system.

1. INTRODUCTION

In recent large vocabulary continuous speech recognisers a lexicon tree is used to reduce the search-effort [5, 2]. Furthermore, considerable effort was put into the use of cross-word CD phone models as they provide better recognition results than word-internal context dependent or context independent models. The use of cross-word context dependent phone models with a static lexicon tree however results in an tremendous growth of the tree. When the lexicon tree is combined with LM information to form a complete search-graph (word conditioned lexical trees), the structure becomes even bigger.

In the literature, different solutions are proposed to cope with this problem:

- In [3], the number of cross-word connections is limited by exploiting the back-off component of the bigram LM. When trigrams are to be used, only that part of the search space that showed to be interesting during the bigram based recognition is searched (see also [1]). This results in a two pass recogniton system.
- Some systems (e.g. [4]), rely on even more passes to incorporate all different knowledge sources into the search.
- An alternative approach is described in [6]: the search graph is dynamically created around those search paths that are in the search beam. This approach is more flexible, but requires additional computations during the recognition.

In our recogniser we solve the problem through the use of an efficiently compacted static lexicon network combined with an almost complete decoupling of lexicon, acoustic models and LM information.

The remainder of this text is organised as follows: First a trivial compact lexicon representation for left CD phones is given. Next the same ideas are applied to right contexts and some results of the lexicon compression are presented. In the next section, the algorithms involved in the network construction are given. Finally, the effects of the new structure regarding the search algorithm are described and some interesting consequences and extensions are presented.

2. OPTIMISATIONS FOR LEFT CD UNITS

If a single static lexicon tree with (cross- and intra-word) left CD phones and with no LM information attached to it is used, some simple extensions suffice to limit the increase in memory requirements with respect to the context independent (CI) lexicon tree (see figure 1). This is due to the compaction of the lexicon at the word beginnings (provided by the tree structure) and the introduction of dummy START nodes: from each word-final phone only a *single* arc is needed to connect the word ending, through a dummy left context dependent start node, with *all* correct word beginnings.

[*] This research was supported by IWT Research Contract 940044, entitled Nerex.

[†] Currently with Lernout & Hauspie Speech Products.

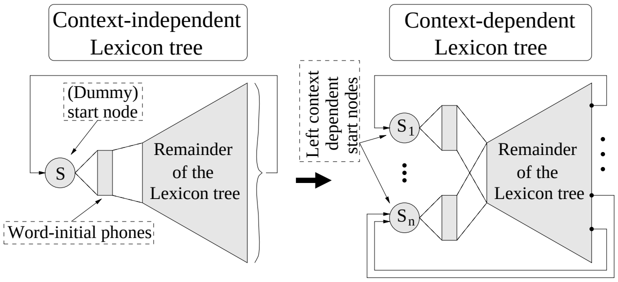
Context-independent Lexicon tree
(Dummy) start node
S
Remainder of the Lexicon tree
Word-initial phones
Left context dependent start nodes
Context-dependent Lexicon tree
S1
Sn
Remainder of the Lexicon tree

Figure 1: Left context dependent phones can be handled efficiently through the use of dummy START nodes.

If also right contexts are considered, and if we can compact the word endings and insert dummy end nodes, one may expect that the memory increase for right CD phones can be tempered also.

3. A LEXICON NETWORK

When both the word beginnings and the word endings are compacted, a word in the lexicon is described as a shared prefix, a non common mid-part and a shared suffix. The resulting structure is no longer a tree, but a network having the following properties:

- **minimal-branching:** In order to minimise the search effort during recognition the fan-out of all nodes must be minimal: no node whatsoever may be followed by two or more identical CD phones. This gives the network the same optimal properties as a lexicon tree.
- **labeled arcs:** As no word specific information can be stored in the shared parts and as the mid-part can be as short as a single arc, the word identity must be stored as a label on one of the arcs in the non common mid-part of the word. To get an optimal compression of the common suffix tree, the word identity must be stored on the first non shared arc.
- **directed:** All information concerning the words (pronunciation and identity) is confined to that part of the network between the START and END nodes. This part of the network has a left to right topology and is loop free (cf. directed graphs). The loop-back information points from the END nodes to the START nodes and is also loop free.

The lexicon construction so far limits the increase in *nodes* when converting the lexicon from a CI to a CD description, but there is still an excessive growth in *arcs* in the following two cases (see figure 2):

- if only the last phone of a word can be merged.
- if the word occurs as a whole in the beginning of another word (e.g. inflections of words).

Both problems were resolved by introducing dummy LINK nodes, fulfilling the same role as the dummy START and END nodes mentioned earlier: dispatch all incoming arcs to multiple successor nodes.

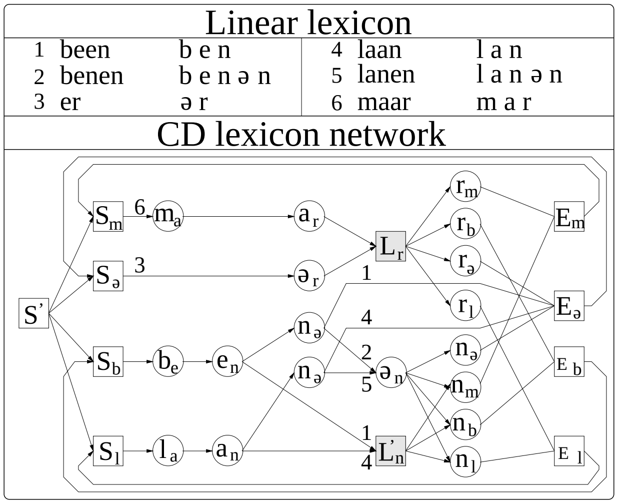

Figure 2: LINK nodes help to reduce the number of arcs in some special cases (example in Dutch).

4. RESULTS OF THE COMPRESSION

Table 1 shows the effectiveness of the network representation for a large Dutch lexicon. The lexicon contains a single pronunciation per word and uses a phone set of size 44. The linear representation consist of 31818 nodes for the 5000 words and 894261 nodes for the 100000 words. We assumed distinct models for *all* possible CD phones. For the given examples (5000 or 10000 words), 1193 or 1618 biphones and 23052 or 40176 triphones actually occurred (cross-word or intra-word).

words	CI phones	left CD	right CD	triphones
Lexicon Tree (as in figure 1)				
5k	13442	14414	198406	/
	18441	32913	383405	/
100k	291358	292711	4491317	/
	391357	418663	8691316	/
Lexicon Network (as in figure 2)				
5k	5894	8124	9375	36264
	10834	15612	29056	101446
100k	123772	135712	166990	227959
	222010	237662	468410	675920

Table 1: Number of nodes (top value) and arcs (bottom value) needed to represent the 5000 and 100000 most frequent Dutch words.

Table 2 gives more detailed information on the number of nodes and arcs needed in the lexicon representation for the WSJ0 task, using the 918 CD phones that occurred at least 250 times in the SI-84 train sentences. The lexicon contains 10% words with multiple transcriptions. We also give the average fan-in (f_i) and fan-out (f_o) for the different node types. By inserting a certain type of dummy node, $f_i \times f_o$ arcs are replaced by $f_i + f_o$ arcs. The above numbers can thus be used to estimate the impact of the dummy nodes.

node-type	5k-closed	20k-open	64k
normal	11344	32927	88732
	2.5→2.4	2.4→2.5	2.4→2.6
START	213	224	242
	22.0→9.4	40.0→9.9	62.8→10.1
END	417	605	1030
	8.4→10.1	18.4→11.9	35.8→13.7
LINK	659	1652	4362
	6.9→11.6	8.4→12.3	8.8→12.3
arcs	41125	112363	303838

Table 2: Size of the lexicon network using triphones for the 5k, 20k and 64k nvp word sets in the WSJ0 task. The second line contains the average number of input → output arcs for the given type of node.

5. CONSTRUCTING THE NETWORK

To construct the lexicon network the following steps are taken:

1 Add all words to a standard tree structure, advance the word identity labels when necessary. Multiple words that have the same pronunciation are treated as a single ’super’ word.
2 Compact the word endings (inverse tree structure), starting from the word final nodes. Note that for the inverse tree the above mentioned property of minimal-branching does not hold: two identical predecessor nodes can not be merged if the involved arcs contain different labels. This step is not necessary, but will speed up the conversion to a CD description.
3 Connect all word final nodes to a dummy END node, and connect the END node with the START node.
4 Mark all nodes as being context independent and start the recursive routine that converts the network to a CD description.
5 Clean up the LINK nodes that have no effect.

Converting the network to a CD description:

1 Search a context dependent variant that matches the current phone (check left and right contexts).
2 Make new derivates of the current phone to take care of the non matching contexts (see figure 3).
First the left context is handled by the recursive split routine, starting with the most distant matching predecessor nodes.
Next, the right context is handled, again starting with the most distant nodes. The complex case (see figure 3) occurs if the current node is not a START or LINK node, the central node is not the current node and the central node has more than one predecessor.
Going back or forth multiple levels in the network is required if long-span contexts are used (e.g. quintphones), or if dummy nodes are encountered.

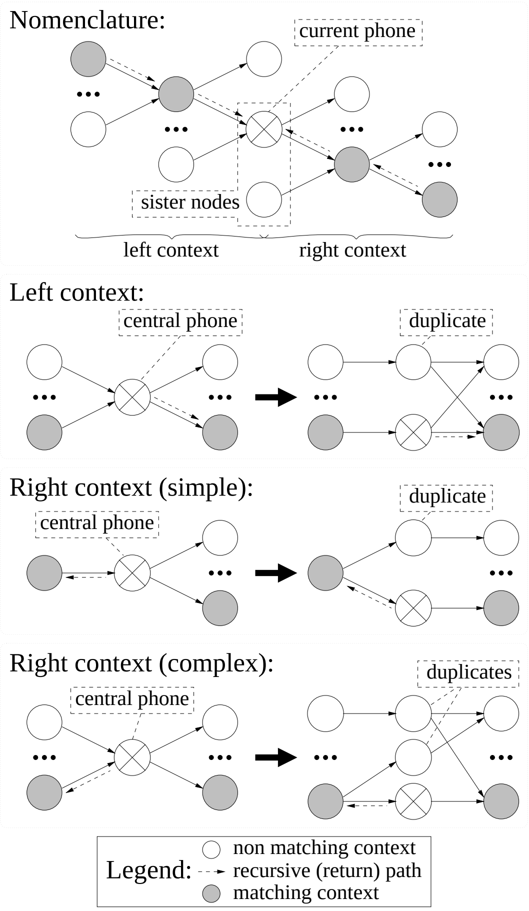

Figure 3: Isolate the correct contexts before converting the central phone to its CD variant

3 Do the same for all children that are not yet handled and are not an END node (recursive).
4 Mark the current phone as context dependent.
5 Try to compact the newly created CD unit:
If the node has only one descendent
  then try to merge with an equivalent sister-node (i.e. same CD phone, also one descendent and an identical label on the outgoing arc).
  else group the outgoing arcs per label and insert a LINK node in front of each group. Next, scan for equivalent LINK nodes (i.e. LINK nodes having the same group of descendents) and merge them.

In our current implementation, LINK nodes are only inserted if there are at least two nodes that refer to *exactly* the same set of successor nodes. We expect still a considerable decrease in the number of arcs if LINK nodes are also inserted before large common subsets of successor nodes.

6. THE SEARCH ALGORITHM

In our approach, the three basic knowledge sources (the pronunciation information, the LM and the acoustic models) are separated, i.e. they are not combined altogether in a single search-graph. Therefore, a modified search-algorithm is needed.

Basically, we still use a time-synchronous beam-search algorithm based on the token-passing paradigm [6]. The token still accumulates the observation scores and collects the trace-back information. But instead of referring directly to a point in the search-graph, multiple references to the different knowledge sources are needed. In our system, the tokens contain the following references:

- **The lexicon node:** gives the location of the token in the lexicon network, thus making the link between token and the pronunciation information.
- **The HMM node:** refers to the acoustic models. As the nodes in the lexicon network refer to phones, an extra index is needed to distinguish between the different states in a phone.
- **The word identity.** The word identity equals -1 at the beginning of the lexicon network and is filled in whenever a labeled arc (see section 3) is traversed. This reference in fact makes that the tokens traverse the lexicon network as if it was a lexicon tree.
- **The LM state:** incorporates the LM into the search. The LM state determines uniquely the impact of the LM on the score. For a word based bigram this is the predecessor word, for a finite state grammar it is the current state and for long-span LM's it is the word context that can still have an effect on the LM score.

The search algorithm is straightforward: on each input frame we update the scores in all tokens and propagate copies of the tokens to all successor nodes as indicated in the acoustic models (using the HMM node reference) or in the lexicon (using the lexicon node reference). Note that if the successor node is a LINK node, the token should be propagated to the next level. All tokens are organised in a hash table to allow fast checking for recombination: for any two tokens that contain the same references, only the best has to be retained.

7. CONSEQUENCES AND EXTENSIONS

As the word identity is known right after the first arc non common with other words is traversed, the LM can be applied before the end phone of a word is reached. This offers some form of (partial) LM forwarding. In the common prefix part of the network (see section 3), the unigram probabilities can be used as a first estimate of the LM cost [7]. We use the labels on the arcs to refer to the most probable word (unigram) that can be reached from that arc.

Alternative pronunciations are handled most efficiently by the network structure. Contrary to the tree structure, multiple local alternatives (e.g. USES: YUW[S/Z][AH/IH] Z) give no exponential growth in the lexicon representation.

Other pronunciation information such as inter-word assimilation rules can be pre-compiled in the lexicon network using similar algorithms (section 5) and give, except for the larger search space when introducing new pronunciation variants, no additional search cost. Duration models can be added in a straightforward way: we just have to add an extra reference in the tokens indicating the start time of the phone or HMM-state.

8. CONCLUSIONS

In this paper, a memory efficient search topology that is able to handle all available knowledge sources (intra- and cross-word context dependent phones, long-span language models, assimilation rules, ...) in a single time-synchronous recognition pass, is described. This is achieved by combining a compact static lexicon network with a knowledge conditioned token-passing algorithm.

REFERENCES

1. X. Aubert and H. Ney. Large vocabulary continuous speech recognition using word graphs. In *Proc. ICASSP*, volume I, pages 49–52, Detroit, May 1995.

2. L. Bahl, S.V. De Gennaro, P.S. Gopalakrishnan, and R.L. Mercer. A fast approximate acoustic match for large vocabulary speech recognition. In *Proc. EUROSPEECH*, volume I, pages 156–158, Paris, September 1989.

3. J.-L. Gauvain, L.F. Lamel, G. Adda, and M. Adda-Decker. Speaker-independent continuous speech dictation. *Speech Communication*, 15:21–37, October 1994.

4. F. Kubala, A. Anastasakos, J. Makhoul, L. Nguyen, R. Schwartz, and G. Zavaliagkos. Comparative experiments on large vocabulary speech recognition. In *Proc. ICASSP*, volume I, pages 561–564, Adelaide, April 1994.

5. H. Ney, R. Haeb-Umbach, B.-H. Tran, and M. Oerder. Improvements in beam search for 10000-word continuous speech recogniton. In *Proc. ICASSP*, volume I, pages 9–12, San Francisco, March 1992.

6. J.J. Odell. *The Use of Context in Large Vocabulary Speech Recognition*. PhD thesis, University of Cambridge, U.K., March 1995.

7. V. Steinbiss, B.-H. Tran, and H. Ney. Improvements in beam search. In *Proc. ICSLP*, volume IV, pages 2143–2146, Yokohama, September 1994.