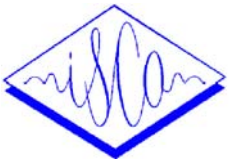

EUROSPEECH '97
5th European Conference on Speech Communication and Technology
Rhodes, Greece, September 22-25, 1997

The Incorporation of Path Merging in a Dynamic Network Recogniser

Simon Hovell

Speech Technology Unit, BT Laboratories,
Martlesham Heath,
Suffolk, England.

simon.hovell@bt-sys.bt.co.uk

10.21437/Eurospeech.1997-66

ABSTRACT

In this paper, the incorporation of *path merging* within BT's dynamic speech recognition architecture[1] is discussed. One of the disadvantages of dynamic network generation is the size of the network generated. This is to a large extent due to the creation of many duplicate network portions. The use of a path merging strategy can redress this problem to some extent. This paper discusses the theory behind path merging, demonstrating a 22% speed improvement on a typical recognition task for no loss in top-*N* accuracy.

1. INTRODUCTION

Modern tasks often involve very large vocabularies, comprising many thousands of words and high perplexities. One way of handling the computational requirements of these tasks has been to use dynamic network recognisers [2]. Static network recognisers load a predefined finite state network at the commencement of recognition. This network describes fully all possible utterances that can be recognised. A dynamic network recogniser however, creates a network during recognition. This allows the use of grammars that cannot be described by finite state networks. At any given time, the network can be extended where necessary by adding extra words, subword units, or phrases. Similarly, those parts of the network not in use can be disassembled and need no longer be considered. This approach ensures that only the minimum amount of network necessary is ever in existence at any given time.

Unavoidably, a significant amount of processing is involved in the dynamic creation and destruction of the network, and this can result in dynamic network recognisers having slower recognition times than equivalent static network recognisers for small tasks. When applied to large natural language tasks however, the cost of dynamic network creation becomes trivial when compared with the memory savings made.

This paper concentrates on the concept and use of path merging and presents some theory and applications.

2. Path Merging

Dynamic network extension can result in a tree network with many branches that only differ slightly from each other. A considerable saving in both memory and computational efficiency can be made if similar branches are merged together. However, if this merging is implemented without due consideration, then the quality of the top-N output from an N-best recogniser will be degraded.

The simplest form of path merging is "instant merging", in which all partial hypotheses which reach a word boundary are propagated into a single set of following models. Only the best partial hypothesis up to the merge point is continued, and other hypotheses can be stored for subsequent use in traceback. Although the propagation of the best hypothesis is guaranteed, the final N-best output may be sub-optimal, as other hypotheses may have to compete directly with (and hence be destroyed by) the winning hypothesis. The root of the problem observed above lies in the potential for alternate hypotheses to have different time segmentation at the merge point.

Associated with each partial path through the network are a number of hypotheses, each representing a different time alignment. Figure 1 shows an example of this displaying two hypotheses for the model sequence *A*-*C*. At frame 21, the two hypotheses converge, and the winning hypothesis *H1* destroys the alternate hypothesis $H1'$. Figure 2 shows the time alignment of the second best hypothesis in the network, *H2* (corresponding to a different partial path, with the model sequence *B*-*C*), which enters model *C* at the same time as $H1'$ (frame 14). If instant

path merging is being implemented and the same instance of model *C* is shared by both partial paths, then *H2* will be stored as a link from *H1′*, and will thus be lost when *H1′* is destroyed.

.

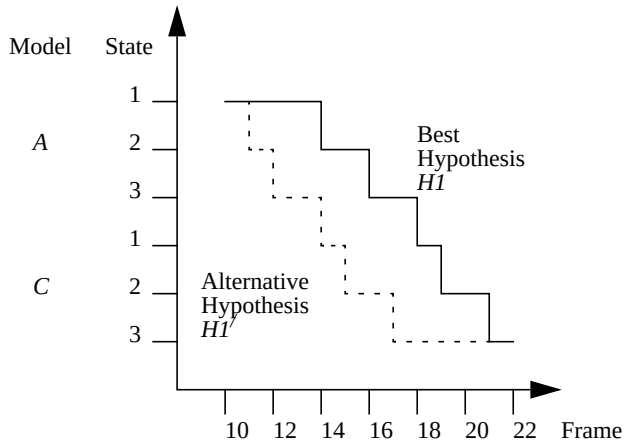

Figure 1: Two Hypotheses in the partial path *A*-*C*

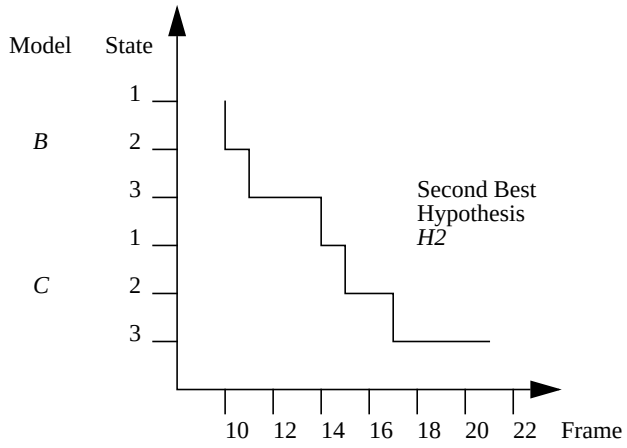

Figure 2: Hypothesis in the partial path *B*-*C*

An optimal *N*-best output can be assured if differing hypotheses are merged only when their time segmentation has become aligned. In the above examples, it can be seen that this has happened by the time the hypotheses *H1* and *H2* exit model *C*.

In the example, the difference in score between *H1* and *H2* varies with time as the hypotheses pass through different state sequences. This variation ceases at the point when the two hypotheses become time aligned. In general, if two partial paths *P1* and *P2* are merged only when all hypotheses associated with them are time aligned, then at the merge point the score offset between each hypothesis in *P1* propagated onward from that point and the corresponding (poorer scoring) stored hypothesis in *P2* will have become constant. Therefore, the best possible alternate hypothesis *P2H2* will be stored and linked to the best possible propagated hypothesis *P1H1*, so an optimal top-*N* is assured.

An interim step towards establishing the point at which two partial paths may be merged is to split each partial path through the network into two parts, *recent* and *distant*. The recent section is defined such that the segmentation of two partial hypotheses which have similar recent path sections will be aligned by the time they have each traversed the recent path section. In practice, this is caused by the acoustic match between the incoming speech data and the recent path segment imposing a particular time alignment on all hypotheses traversing the recent path segment, regardless of their distant history.

From this definition, it is evident that the merging of only those partial paths with identical recent sections will ensure the optimal *N*-best output of a recogniser. The remaining difficulty is to establish exactly the requisite length of the recent section. Schwartz and Austin [3] used a 1-word length, with encouraging results, although the variation in word length may be the cause of some error.

Generally, this minimum length is usually more than satisfied due to constraints imposed by the language model: The language model of choice is often an *n*-gram statistical one. When implementing path merging in conjunction with an *n*-gram language model, the recent section of different partial paths must also equal or exceed the span of the language model (*n*-1 words) before path merging between two hypotheses can take place. This ensures matching penalties on the resulting lexical tree shared by both paths.

During recognition, the best hypothesis traversing a given partial path is usually preceded by many poorer scoring hypotheses, corresponding to time segmentations with very short durations in each of the

HMMs within the partial path. Where a pruning algorithm is in operation, the most advanced hypotheses in two different partial paths with identical recent segments may be slightly misaligned. There is therefore a case for relaxing the time constraint and merging partial paths with initial hypotheses that reach word boundaries at slightly different times.

3. RESULTS

Measurements were carried out on a connected digit task taken from BT's subscriber [4] UK telephony speech database. Measurements were performed on an HP730 unix workstation. The test comprised five recognition runs:

- Dynamic network, no path merging
- Dynamic network, time synchronous path merging
- Dynamic Network, path merging time constraint relaxed to a 7 frame margin
- Static network, alternate hypotheses stored at merge points
- Static network, no alternates stored at merge points.

The dynamic network used a bigram language model in which all words were weighted equally except the probability of "double" followed by </s> (end of sentence) which was penalised heavily. The static network used is shown in Figure 3.

Speed measurements for the first four tests are presented in Figure 4 (there was no significant speed difference between the two static network runs). It can be seen that using the time synchronous path merging criteria, a 20% reduction in processor load was achieved, from 60.6% cpu utilisation to 48.3% utilisation. Relaxation of the path merging time constraint resulted in a further 5% drop in computational requirement, resulting in a total reduction of 25%. This compares with a static network based recogniser running on the same task with a 24.7% cpu utilisation. Clearly, for tasks where only top-1 output is necessary, a static network based recogniser still markedly outperforms the equivalent dynamic system, although the relative computational expense of using a a dynamic recogniser for small tasks is much reduced.

The experiments showed that the use of path merging as constrained by the length of the language model (i.e. a one word recent history) resulted in no reduction in top-*N* accuracy. However the employment of instant path merging as described earlier in this paper (i.e. the storing of alternate hypotheses at merge points within the static recogniser) resulted in a significant drop in top-*N* accuracy. This can be seen in Figure 5. Figure 5 also shows the equivalent accuracy for a recogniser in which no alternate hypotheses are stored (except for those traced back from each leaf node in the static network) Clearly, instant path merging is a dramatic improvement on this, and may be an acceptable compromise where low computational burden is an overriding consideration. This is particularly relevant when incorporating a recogniser in to a natural spoken language system.

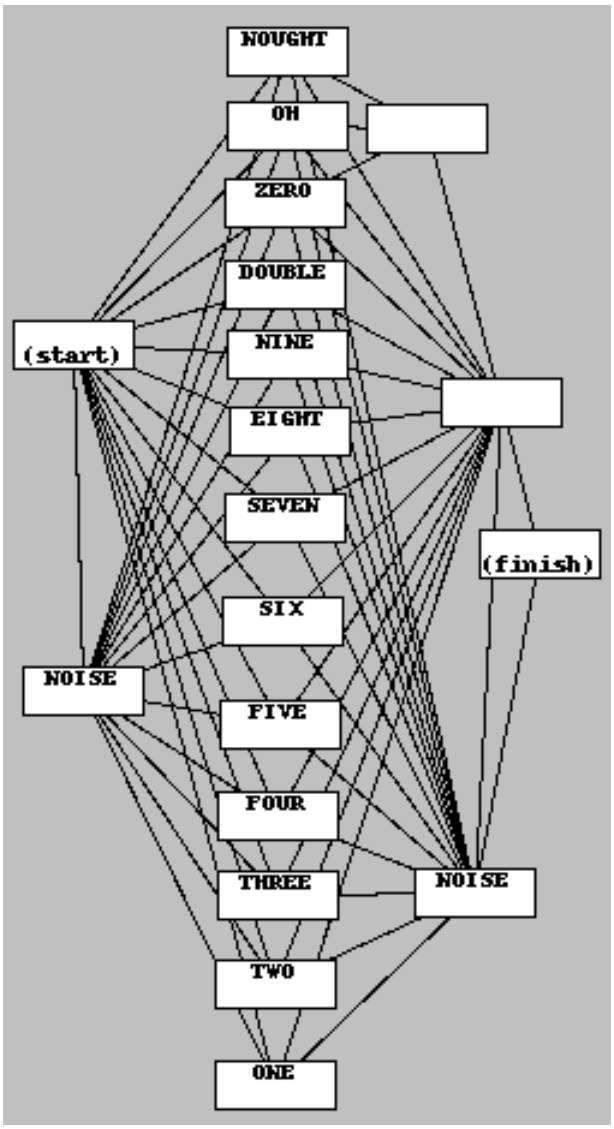

Figure 3: Network used for static recognition runs

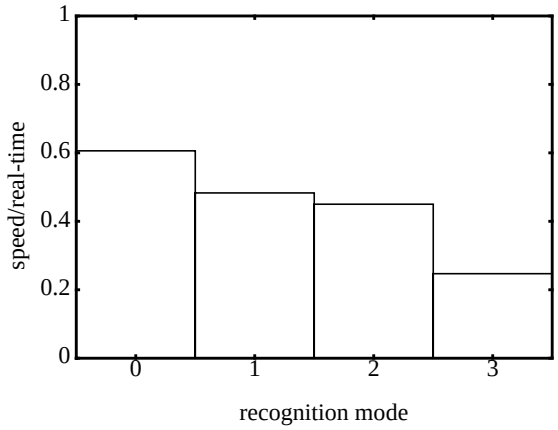

Figure 4: performance of different recognition modes:

- 0:Dynamic, no merging
- 1:Dynamic, time synchronous merging
- 2:Dynamic, 7-frame time constraint
- 3: Static (with or without stored alternates)

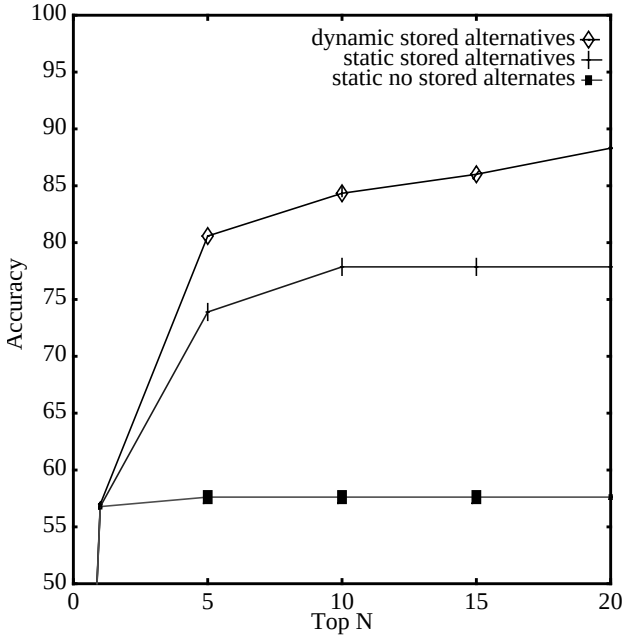

Figure 5: Top-N output for varying recognition modes

4. CONCLUSION

The theory of path merging has been examined, and the difficulties in its application whilst maintaining an admissible top-*N* result have been examined. It has been shown that the use of a path merging strategy can result in a 25% performance improvement with no loss in top-1 or top-*N* accuracy. Whilst an equivalent static network recogniser will still outperform a dynamic network recogniser, this improvement means that dynamic network recognisers can perform competitively on small tasks and may be considered for commercial applications

5. REFERENCES

1. Hovell S. A., Ringland S. P. A. and Ollason D. G. “A Modular Architecture for Speech Recognition”, Proc. 1995 IEEE ASR Workshop, pp 189-190, Snowbird, December 1995

2. Woodland P. C., Odell J. J., Valtchev V. and Young S. J., “Large Vocabulary Continuous Speech Recognition Using HTK”, Proc. ICASSP 94,pp. 125-128, 1994.

3. Schwartz R. and Austin S., “A comparison of Several Approximate Algorithms for Finding Multiple (N_BEST) Sentence Hypotheses”, Proc. ICASSP 91, pp 701-704, 1991.

4. Simons, A. D. and Edwards, K., “Subscriber - A Phonetically Annotated Telephony Database”, Proc. I.O.A. Vol 14 part 6 (1992), pp 9-16