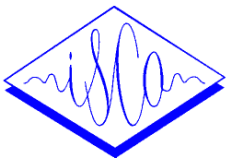

3rd European Conference on
Speech Communication and Technology
EUROSPEECH'93
Berlin, Germany, September 19-23, 1993

EFFICIENT ENUMERATION OF SENTENCE HYPOTHESES IN CONNECTED WORD RECOGNITION†

Víctor M. Jiménez*, Andrés Marzal** and Enrique Vidal
e-mail: vjimenez@dsic.upv.es

Dpto. de Sistemas Informáticos y Computación.
Universidad Politécnica de Valencia.
Cno. Vera s/n. 46071, Valencia (Spain).

ABSTRACT

A new algorithm is presented for the search of the N best paths in weighted graphs applied to the computation of the N Best Sentence Hypotheses in Continuous Speech Recognition based on the One-Stage procedure.

It is also shown how Beam Seacrh techniques can be integrated in the algorithm leading to a very efficient sentence hypotheses enumeration algorithm.

Keywords: *N Best Sentences, N Shortest Paths, Beam Search, Connected Word Recognition.*

1. INTRODUCTION

In [MM89] and [Ste89], a correct algorithm for the search of the N Best Sentence Hypotheses (BSH) in Connected Word Recognition was presented. The need to compute the N-best partial sentence hypotheses at *every node* of the "computation graph" underlying the well-known *One-Stage* algorithm [Ney84] made it impractical for real applications due to the resulting prohibitive time complexity.

Since then, much effort has been devoted to defining adequate heuristics to reduce the time requirements of this approach, sacrificing the correctness (optimality) of the algorithm [MM89], [Ste89], [SC89], [LMNS91], [SA91], [NT91], [Ste91].

More recently, two new approaches have appeared which correctly compute the N-best sentence hypotheses: the *Tree-Trellis* algorithm [SH90], [SH91], and a backtracking-like technique for Dynamic Programming recurrences whose application to the *Two-Level* algorithm [Sak79] was presented in [MV92].

In this paper, this backtracking-like technique is presented as general procedure for the computation of N shortest paths in directed networks and is applied to the *One Stage* procedure resulting in a new correct algorithm for the enumeration of the N-BSH. The time complexity of the new approach is comparable to that of the most efficient heuristics referred to above, while strictly guaranteeing the correctness of the solution [JM92]. On the other hand, with this new approach the generation of new sentence hypotheses does not require any additional "Viterbi alignment" (as the *Tree-Trellis* algorithm does). This, along with the fact that it is not necessary to decide in advance the number of sentence hypotheses to generate —the hypotheses can be sequentially generated when required and the computation time cost is strictly related to the number of sentences actually needed—, leads to a very efficient procedure.

† This work has been partialy supported by CICYT under contract TIC-1026/92-CO2.
* Supported by a grant from the Conselleria d'Educació i Ciència de la Generalitat Valenciana.
** Supported by a grant from the Ministerio de Educación y Ciencia of the Spanish Government.

2. A NEW ALGORITHM FOR N BEST SENTENCE HYPOTHESES ENUMERATION

The Connected Word Recognition problem can be stated as follows: let $a = a_1 a_2 \ldots a_{|a|}$ be a given input acoustic sequence whose symbols (*frames*) belong to an alphabet or vector space, and let W be a "dictionary" composed of models of words; find the sequence of words from W which modelizes the input sequence a with minimum "distortion". The so-called N Best Sentence Hypotheses enumeration problem consists of finding the N sequences of models that best describe the acoustic sequence.

Most commonly used Connected Word Recognition algorithms deal with an underlying graph in which there is a correspondence between paths and sentence hypotheses. This permits a formulation of the problem in terms of the search of the shortest path in the graph. Consequently, the problem of finding the N best sentence hypotheses can be stated in terms of the search of the N shortest paths (with different associated sentence hypotheses).

2.1 A New Algorithm for N Shortest Paths Computation

Given a weighted graph $G = \langle V, E \rangle$, the algorithm of Figure 1 computes the N shortest paths proceeding in two stages [MJ92]. In the first stage the weight of the shortest path from the initial node to all other nodes in the graph is computed by means of any known "one-to-all" shortest path algorithm. In a second stage, every new path is generated by tracing back the preceding one.

The basic idea of the tracing back step can be stated as follows: if the n-th best path at a node v comes from the m-th best one at a predecessor node u, then the $n+1$-th best path at node v is generated by *recursively* asking for the $m+1$-th best at u; and then, choosing the best among it and all the latest available paths at predecessor nodes of v.

The recursive nature of this approach does not lead to high computation costs. Every recursive call generates, at most, a new recursive call, allowing a simple and efficient implementation. The computation of every new path implicitly visits, at most, the nodes of the previous path. That is, it is computed in time order of the minimum between the number of nodes of the graph and the length of the previous path.

On the other hand, the paths are sequentially generated when required and the computation cost is strictly related to the number of paths actually needed. The cost of generating every new path is negligible when compared to the cost of the first stage.

The main disadvantage of this algorithm is that its space requirement is proportional to the size of the graph. Though this could be critical in certain cases, we will later show how this problem can be minimized in a practical situation.

2.2 Application to *N* Best Sentence Hypotheses Enumeration

The method that has been presented in the previous section can be applied to the problem of obtaining the N Best Sentence Hypotheses for any modelization of words and distortion criterion that leads to a formulation in terms of search of shortest paths in a graph.

In particular, we will assume that the dictionary is composed of Hidden Markov Models (HMM) and the distortion will be the (negative log-)likelihood of the sequence of states that most probably describes the input acoustic sequence (Viterbi score).

The well-known One-Stage algorithm [Ney84] can be used to efficiently find the best sentence hypotheses. The nodes in the underlying graph, which is called *Trellis*, are labelled with a record $\langle f,m,s\rangle$ where f is the index of a frame of a, m identifies a model of W and s is one of the states of the model. We can distinguish two kinds of arcs: *intra* and *intermodel*.

- Intramodel arcs link nodes $\langle f,m,s\rangle$ with others of the form $\langle f+1,m,s'\rangle$, where (s,s') corresponds to a valid transition in model m. The weight of this arc is (the negative logarithm of) the probability of transition between both states multiplied by the probability of emission of symbol a_{f+1} in state s'.
- Intermodel arcs are of the form $(\langle f,m,s\rangle,\langle f+1,m',s'\rangle)$, where m and m' are two models of W, s is the final state of model m and s' is the initial state of model m'. Its weight is (the negative logarithm of) the probability of transition between both models by the probability of emission of symbol a_{f+1} in state s'.

The initial node is $\langle 0,-,-\rangle$ and can transit to all nodes of the form $\langle 1,m,s\rangle$, where m is any model and s is its initial state, with the probability of emission of a_1 in s. The final state is $\langle |a|+1,-,-\rangle$ and is the destination of arcs whose origin are nodes of the form $\langle |a|,m,s\rangle$, where m is any model and s is its final state and probability 1.

The existence of a (stochastic) regular grammar for sequences of words would restrict the possible transitions between models and would affect the dictionary by forcing it to include one model per arc of the grammar [Ney84].

Therefore, when applied to the N-BSH problem, the first stage of the algorithm presented in Section 2.1 is the *One-Stage* algorithm [Ney84], [ACM*92] and every new sentence hypothesis is obtained by means of the tracing back procedure described in the previous section. If different paths had associated different sentence hypotheses, this step would require visiting $|a|$ nodes in the worst case. But in the Trellis this is not the case: different paths may have identical associated sequences of words and differ only in the underlying segmentation of a. As different (partial) sentence hypotheses are required, it may be necessary to iterate at each node until the new sentence is actually different (or no new sentences can be obtained).

```
Algorithm N-Shortest Paths   // Returns the weight of the N
                                shortest paths of a graph
  Input   G = ⟨V,E⟩          // V=nodes set, E=arcs set
          d: E ↦ R           // weighting function
          s,t: V             // s: initial node, t: final node
          N: N               // Number of paths
  Output  D̂: array [1...N] of R   // Weight of the N paths
var
  D: array [V,1...N] of R    // D[v,n]: weight of n-th best path
                                from the initial node to v
  B: array [V,1...N] of V    // B[v,n] points to the node from
                                which the n-th best path comes
  n: array[V] of N           // Number of paths computed at
                                every node
  a: array[E] of N           // Number of last best path
                                coming from an arc
  procedure Next(v: V)
  begin
    if ∄(u,v) ∈ E then
      D[v,n[v]+1] := +∞
    else
      if n[B[v,n[v]]] < a[B[v,n[v]],v]+1 then
        Next( B[v,n[v]])
      end if
      (u,v) := arg min_{(u',v)∈E} D[u',a[u',v]+1] + d(u',v)
      D[v,n[v]+1] := D[u,a[u,v]+1] + d(u,v)
      B[v,n[v]+1] := u
      a[u,v]++
    end if
    n[v]++;
  end
begin
  First Stage:
  Compute D[v,1], B[v,1], and set n[v] to 1 for all v in V
  Second Stage:
  for all (u,v) in E do
    a[u,v]:=0; a[B[v,1],v]:=1;
  end for all
  for i:=2 to N do
    Next(t)
    D̂[i] := D[t,i]
  end for
end
```

Figure 1: *N* Shortest Paths algorithm

3. BEAM SEARCH

The previous N-BSH algorithm is very efficient in time and allows real-time implementations. But the algorithm may have quite severe space requirements in large Connected Word Recognition tasks due to the huge size of the underlying graph: the best path ending at every node of the Trellis must be kept in order to compute the N best sentence hypotheses.

However, in practice, the distortion of the N best sentence hypotheses at every frame of a is close to the best one even for relatively large values of N. This fact is illustrated in Figure 2, where the distortion of the N-th best sentence at every frame is shown for values of N up to 80 for a typical utterance of the task that will be described in the next section. Sentence hypotheses whose distortion significantly differs from the optimal one at a given frame, can be discarded without affecting the correctness of the solution. Thus, only those nodes whose best path survives this pruning criterion must be maintained in memory. The heuristic threshold α that determines the maximum allowed discrepancy between distortions must be fixed experimentally.

This technique of selection of nodes is known as Beam Search [Low76]. The algorithm for the computation of N best sentence hypotheses incorporating Beam Search in its first stage in order to determine the nodes that will be taken into account in the second stage, will be called Beam Search N-BSH algorithm.

The heuristic threshold can be reduced even more than the value necessary to preserve the correctness of the algorithm, resulting in a lower computational complexity: smaller values of α will decrease the number of survivor nodes of the Trellis and the execution time will decrease as well. The compromise between computational complexity and correctness of the Beam Search N-BSH algorithm should be determined experimentally.

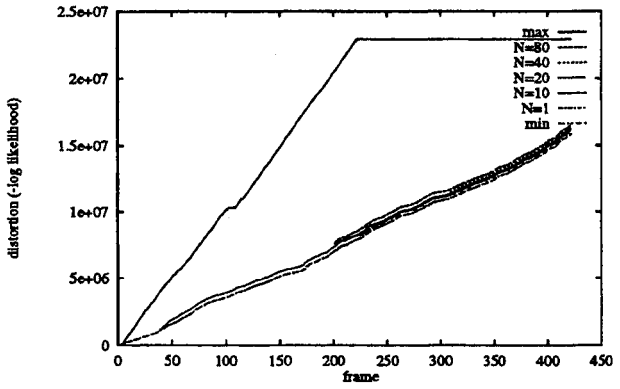

Figure 2: Distortion of the N-th best sentence hypotheses at every frame for a given input acoustic sequence. The curve "max" shows the maximum distortion at every frame and "min" the minimum one.

4. EXPERIMENTAL RESULTS

Experiments were performed on forty-eight continuous speech queries to a geographical database uttered in Spanish by four speakers [DRP*93]. The dictionary was composed of sixty Hidden Markov Models describing words. Each of them was formed by properly concatenating 5-state discrete Hidden Markov Models from a set of twenty-eight phonemes. These phoneme models were trained with sentences from a different task that were uttered by speakers not involved in the test set. The height of the Trellis was 4500 states and the average length of the acoustic sequences was 450 frames.

The Beam Search N-BSH algorithm was run taking into account a simple template grammar that allows for a quite flexible query syntax.

The objective of the first experiment was to determine a value of the parameter α that constitutes a good compromise between average space savings and correctness of the solution. The "algorithmic" correctness was measured as the percentage of the actual N best sentence hypotheses that survive for a given value of α.

The space reduction was measured as the percentage of nodes of the Trellis that are pruned. Analogously, time reduction was calculated as the percentage of time that is saved with respect to the original N-BSH algorithm.

Figure 3 shows the results of this experiment averaged over all the test sentences. It can be seen that there is a value of the heuristic threshold ($\alpha=8\text{x}10^5$) with a satisfactory compromise between correctness and computation effort: on average, 84% of the correct sentences were obtained with only 13% of survivor nodes and 28% of the time. Another interesting value is $\alpha=1.6\text{x}10^6$, that halves the computational complexity while still providing 100% of the correct N best sentence hypotheses.

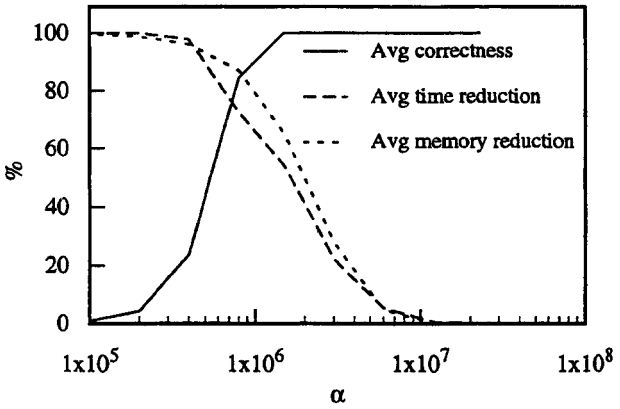

Figure 3: Correctness of the solution versus computation savings for different values of α and N=40.

The average time used to generate the N best sentence hypotheses for the original algorithm ($\alpha=+\infty$) and the chosen value of the heuristic threshold ($\alpha=8\text{x}10^5$) is shown in Figure 4 for different values of N. It is remarkable how the first few sentence hypotheses are generated with computing time that is just a fraction of the time required to compute the best one.

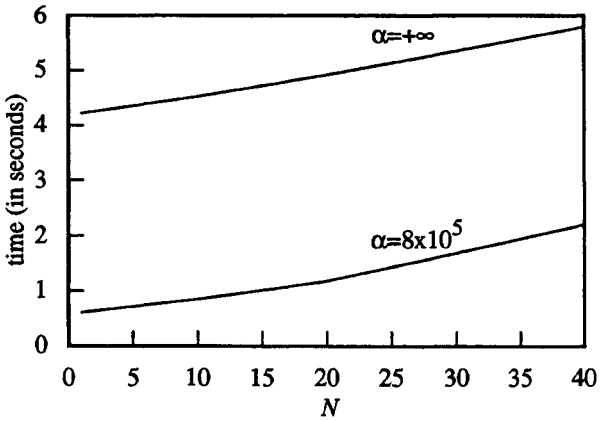

Figure 4: Actual computing time for different values of N and α.

On the other hand, from a practical point of view we are interested in how often the sentence actually uttered is among the N sentence hypotheses generated by the algorithm. Figure 5 shows that for the chosen value of the heuristic threshold ($\alpha=8x10^5$) the utility of the algorithm is not noticeably affected. Figure 6 shows this same behaviour for different values of α.

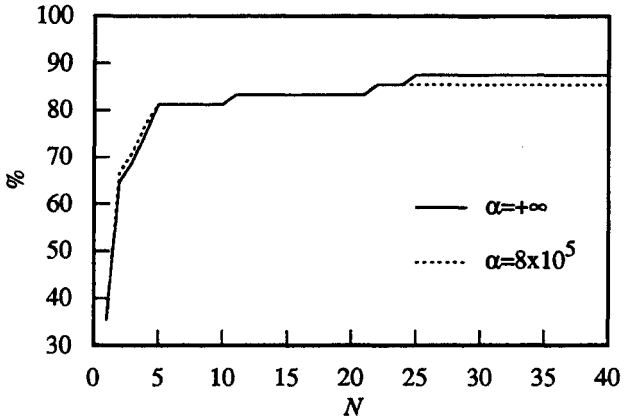

Figure 5: Percentage of times that the actually uttered sentence was found among the N best sentence hypotheses for the N-BSH algorithm ($\alpha=+\infty$) and Beam Search N-BSH algorithm ($\alpha=8x10^5$).

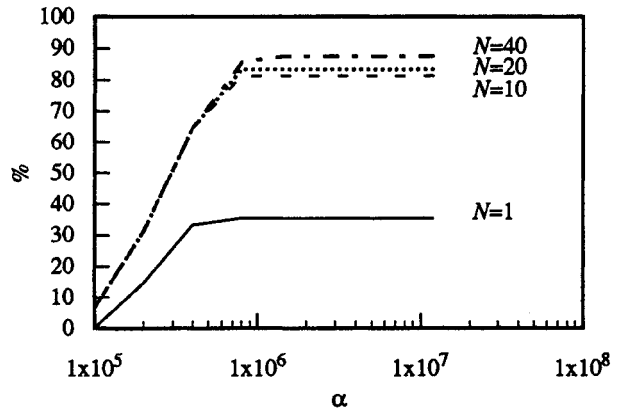

Figure 6: Percentage of times that the actually uttered sentence was found among the N best sentence hypotheses for different values of α.

5. CONCLUSIONS

A new algorithm that correctly computes the N shortest paths in a graph has been presented. This algorithm has been applied to a Connected Word Recognition task and has shown to be efficient: once the best sentence hypothesis has been computed (by means of the One-Stage algorithm) several sentence hypotheses can be obtained in real time, making its integration in dialog systems or its interaction with higher level (syntactic, semantic...) modules possible.

On the other hand, it has been shown how to reduce the space (and time) complexity of the algorithm without practically affecting its recognition rate.

ACKNOWLEDGEMENTS

The authors wish to thank Ismael Ripoll for his help in the processing of experimental data and to Juan Andrés Sánchez for training the phoneme HMMs.

REFERENCES

[DRP*93] J. E. Díaz, A. J. Rubio, A. M. Peinado, E. Segarra, N. Prieto and F. Casacuberta. "Development of task oriented Spanish Speech Corpora". In these Proceedings.

[JM92] V. Jiménez and Andrés Marzal. "Un Nuevo Algoritmo para la Búsqueda de las N Mejores Decodificaciones en Reconocimiento de Palabras Conectadas, basado en el Algoritmo de Un Paso". In *Actas del V Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes* pages 180–187, 1992. (In Spanish).

[LMNS91] E. Lleida, J. B. Mariño, C. Nadeu and J. Salavedra. "Demysillable-based HMM Spotting for Continuous Speech Recognition". In *Proceedings of ICASSP*, 1991.

[Low76] B. T. Lowerre. "The HARPY Speech Recognition System". Ph. D. Thesis, Carnegie Mellon Univ. Tech. Report, Dept. of Computer Science, 1976.

[MJ92] A. Marzal and V. Jiménez. "Un estudio comparativo de algoritmos clásicos y nuevas aproximaciones para la búsqueda de los N mejores caminos en un grafo". Technical Report DSIC-II/27/92. Dept. S.I.C. Universidad Politécnica de Valencia, Spain. 1992. (In Spanish).

[MM89] J. B. Mariño and E. Monte. "Generation of Multiple Hypothesis in Connected Phonetic Unit Recognition by a Modified One Stage Dynamic Programming Algorithm". In *Proceedings of EUROSPEECH*, pages 408–411, 1989.

[MV92] A. Marzal and E. Vidal. "A N Best Sentence Hypotheses Enumeration Algorithm with Duration Constraints based on the Two Level Algorithm". In *Proceedings of the 11th IAPR International Conference on Pattern Recognition*, 1992..

[Ney84] H. Ney. "The use of the One Stage Dynamic Programming Algorithm for Connected Word Recognition". *IEEE Trans. on ASSP*, 32(2):263–271, 1984.

[NT91] P. Nowell and H. S. Thompson. "An efficient implementation of the N-Best Algorithm for Lexical Access". In *Proceedings of EUROSPEECH*, pages 667-670, 1991.

[SC89] R. Schwartz and Y. L. Chow. "The N-Best Algorithm: an Efiicient Procedure for Finding Top N Sentence Hypotheses". In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 199–202, 1989.

[SA91] R. Schwartz and S. Austin. "A Comparison of Several Algorithms for Finding Multiple (N Best) Sentence Hypotheses". In Proceedings of ICASSP, pages 701–704, 1991.

[SH90] F. K. Soong and E. F. Huang. "A Tree-Trellis Based Fast Search for Finding the N Best Sentence Hypotheses in Continuous Speech Recognition". In *Proceedings of DARPA Speech and Natural Language Workshop*, pages 709–712, 1990.

[SH91] F. K. Soong and E. F. Huang. "A Tree-Trellis Based Fast Search for Finding the N Best Sentence Hypotheses in Continuous Speech Recognition". In *Proceedings of ICASSP*, pages 705–708, 1991.

[Sak79] H.Sakoe: "Two-Level DP-Matching — A Dynamic Programming-Based Pattern Matching Algorithm for Connected Word Recognition". In *IEEE Transactions on ASSP*, (27):6, pp.588-595 , 1979.

[Ste89] V. Steinbiß. "Sentence Hypotheses Generation in a Continuous Speech Recognition System". In *Proceedings of EUROSPEECH*, pages 51–54, 1989.

[Ste91] V. Steinbiß. "A Search Organization for Large-Vocabulary Recognition Based on N-Best Decoding". In *Proceedings of EUROSPEECH*, pages 1217–1220, 1991.