音声言語情報処理 11－4
（1996．5．27）

A* 探索に基づく大語彙連続音声認識

李 晃伸　河原 達也　堂下 修司
京都大学 工学部 情報工学教室
〒606-01 京都市 左京区 吉田本町

あらまし

本研究では，大語彙の条件下における A* 探索に基づく連続音声認識について検討する．単語対 HMM はヒューリスティックとして強力な言語モデルであるが，その処理量は語彙の大きさに比例し，大語彙のタスクにおいては破綻する．これに対し，単語対 HMM の木構造化及びヒューリスティック計算におけるビームサーチの導入を提案する．また，パープレキシティの増加による最適解のスタックあふれを防ぐために一仮説当たりの展開仮説数を制限することを提案する．各手法に対して語彙数が 5000 のタスクにおいて文認識実験を行い，有効性を検証した．合わせて，大語彙におけるビームサーチと A* 探索の比較を行った．

和文キーワード　大語彙, 連続音声認識, A* 探索, 木構造, ビームサーチ

Large Vocabulary Continuous Speech Recognition Based on A* Search

Akinobu Lee　Tatsuya Kawahara　Shuji Doshita
Department of Information Science, Kyoto University
Sakyo-ku, Kyoto 606-01, Japan
e-mail: ri@kuis.kyoto-u.ac.jp

Abstract

In this paper, we study large vocabulary continuous speech recognition based on A* search. Although word-pair HMM is a powerful linguistic model as heuristics, the size increases proportional to vocabulary size, and the cost for computing heuristics become too expensive to perform search. To avoid this, we propose composing the word-pair HMM as a tree structure and introducing beam search technique on heuristic computation phase. we also discuss introducing the idea of local stack to A* search to decrease the possibility for optimum sentence hypotheses to overflow from hypothesis stack. Next we report experimental sentence recognition result for each methods on 5000 word task. Search performance of heuristic beam search at large vocabulary compared with A* search is also mentioned in this paper.

Keywords　large vocabulary, continuous speech recognition, A* search, tree-structure, beam search

1 はじめに

近年，音声認識の分野において，これまでと比べてより大語彙のタスクに対する研究が盛んに行われている．大語彙向けの連続音声認識アルゴリズムに関しても様々な研究機関において研究が進められている [1, 2, 3, 4, 5]

我々の研究室では，会話音声の認識において最適解を得ることが保証される A* 探索に基くアプローチをとっている．これまでに，250 単語程度の小語彙のタスク設定の下で，定型的な発話に対してほぼ最適解を得ることができている [6].

本研究では，この A* 探索に基づく連続音声認識の，大語彙の条件下での実現について検討する．具体的には，単語の組合わせ爆発による計算の破綻を防ぐため，探索のヒューリスティックとして用いる単語対 HMM の木構造化，ヒューリスティック計算のビームサーチ化，そして探索における一仮説当たりの展開仮説数の制限について検討する．そして語彙数が 5000 のタスクにおいて文認識実験を行い各手法を評価する．

2 章で大語彙連続音声認識について概観し，本研究の位置づけを行う．3 章では，A* 探索を用いた連続音声認識手法について解説し，その大語彙での問題点を指摘する．そして 4 章で A* 探索の大語彙化手法を幾つか説明し，5 章で各手法の評価実験を行った結果を報告する．

2 大語彙連続音声認識

2.1 探索問題としての連続音声認識

本研究では，文 (1 文連続発声) を音素認識器の連結によって認識することを考える．現在最も有力な音素認識手法である HMM を用いる場合，文仮説は音素 HMM の連結でモデル化される．各文仮説のスコア (尤度) は，基本的には，その仮説を表現する HMM の状態を，許された遷移にしたがって時間軸に展開した状態空間 (これを HMM トレリスという) に対して Viterbi アルゴリズムを適用することにより求められる．

探索の方法には，探索の方向に着目すると大きく分けて，入力音声の先頭を始点として探索を行う left-to-right 探索と，入力音声中の最も確らしい単語をまず見つけ出し，そこから前後に探索を行う island-driven 探索がある．一般的に island-driven 探索は制御が困難であるため，構文的制約を探索に組み入れることは難しい．よって本研究では，left-to-right 探索を考える．この場合，文は時間方向を深さとする単語列の木のパスとして表現され，文認識は木探索の問題となる．この木に対して HMM トレリスが展開されることになる．

このように連続音声認識は，与えられた制約の元で，単語列の木において最もスコアの高い文候補のパスを見つける探索問題として定式化される．

2.2 大語彙認識の問題点

大語彙における連続音声認識では，出現し得る文の数は極めて膨大であり，その中から最適な文を現実的な処理量で探索することが要求される．そのためには音節・語彙・構文・意味等の様々なレベルの言語的制約を探索に用いることで，探索空間を縮小することが重要である．

音節制約は語彙数によらず一定であるという大きな利点があるが，音節の連鎖のみでは制約が弱すぎるので，単語・文節の認識は可能であるが [7]，これのみでは連続音声認識には適さない．

語彙・構文レベルの制約は有効であるが，語彙の増大に伴う単語の組み合わせ爆発のため，語彙に比例する処理は，大語彙では破綻する．

また大語彙では，語彙中の単語の類似度が増加し，単語間の弁別性が弱くなるので，単語レベルでの認識誤りが増大するが，本研究では対象としない．

3 単語対制約をヒューリスティックとする A* 探索

3.1 A* 探索法

A* 探索は best-first 探索の一種であり，評価値の最も高い仮説を展開することによって探索を進める．仮説の評価値には，未探索部分のスコアのヒューリスティックな推定値を加える．すなわち，仮説 n について，その評価値 $\hat{f}(n)$ を次のように定義する．

$$\hat{f}(n) = g(n) + \hat{h}(n) \tag{1}$$

ただし，$g(n)$ は既に展開された仮説のスコア，$\hat{h}(n)$ は未展開部分の推定スコアである．このとき最適解が必ず得られるようにするためには，推定スコア $\hat{h}(n)$ を実際のスコア $h(n)$ より厳しくしない，つまり

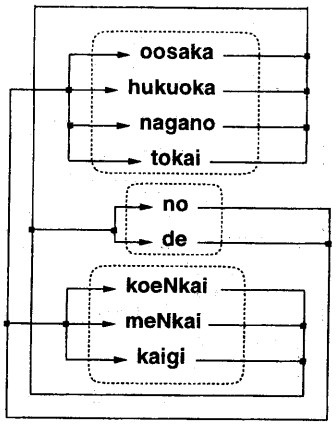
$$|\hat{h}(n)| \leq |h(n)| \tag{2}$$

という条件 (A* 実行可能性条件) が必要である．また，できる限り無駄な仮説を展開することなく最適解を早く見つけるためには，この推定スコアができるだけ実際の値に近いことが望ましい．

3.2 単語対制約によるヒューリスティック

left-to-right 探索でのヒューリスティックは，部分文仮説の未探索の部分に対応する評価値の期待値である．認識処理全体を 2 パス構成とし，まず全ての部分文仮説に共通の言語制約を用いて探索方向とは逆向きに認識処理を行い，その途中段階の結果を保存しておく．次にその結果をヒューリスティックとして用いて探索を行うことにより，未探索部分の評価値を高精度に推定することができる [8].

語彙制約と構文制約を使用して，ヒューリスティック計算のための認識処理を行う．ただし，探索における全て



(注) 点線は同一カテゴリを表す

図 1: 単語対 HMM

の部分文仮説に対するヒューリスティックを与える必要があるので，探索に用いる文法から，単語間の接続に関する情報のみを抽出して用いる．このモデルを単語対 HMM と呼ぶ(図 1)．これは，語彙制約に加えて文脈自由文法の完全なスーパセットである単語対制約を表現しており，強力な制約である．また，探索に用いる制約よりも弱いため，A* 探索の実行可能性条件を満たす．

3.3 パージングアルゴリズム

探索の具体的アルゴリズムは，以下のようになる．

1. 入力音声に対して，探索方向と逆向きに単語対 HMM を用いて認識処理を行い，後ろ向きトレリスを生成．
2. 文の先頭に出現し得る全ての単語に対して，その 1 単語からなる文仮説を生成し，その評価値を計算して，スタックへ．
3. スタックから最も評価値の高い仮説 n を取り出す．
4. 仮説 n が文法で受理されるならば，最適解として探索終了．
5. 仮説 n に構文的に続き得る単語を予測し，そのすべてに対して，その単語を接続した新しい文仮説を生成し，評価値を計算して，スタックへ入れる．ステップ 3 へ．

このアルゴリズムは，最適解が得られた後もそのまま探索を続行することで，第 N 候補まで正しく求める N-best アルゴリズムになる．

なお，本研究では簡単のため left-to-right 探索を考えるが，等価な構文規則では，日本語の性質によって，left-to-right より right-to-left に展開する方が展開仮説数が減少できる [9]．よって実時間に近い処理を目指す場合，right-to-left 探索を行う方が有利である．

3.4 大語彙における A* 探索の有効性と問題点

A* 探索とビームサーチを比較する．1 パスのビームサーチでは，探索に用いる制約のパープレキシティに処理が比例する．パープレキシティの大きい大語彙のタスクにおいては，常にビーム幅いっぱいの仮説を扱わねばならず，処理量が大きい．また，最適解が局所的に低いスコアをとる場合に枝刈りされてしまう危険性がある．これに対して A* 探索のような 2 パスの探索戦略では，探索に用いる制約と，ヒューリスティック計算に用いる制約とのパープレキシティの差に探索処理量が依存する．この差が小さければ，推定スコア $h(n)$ は実際のスコアに近づき，より効率よく最適解を得ることができる．従って，探索に用いる制約のパープレキシティが大きくても，この差が小さければ探索の処理量は少なくて済む．

しかし，ヒューリスティックとして単語対 HMM を用いる場合，その大きさは語彙数に比例するので，第 1 パスの処理量が爆発し，実質的に探索が不可能になる．

また，A* 探索においては，未探索部分の推定スコア $h(n)$ を加えて文仮説の評価値とするためスコアの時間依存度は小さいが，推定スコアは実際の値より緩い値なので，A* 探索においても，仮説の評価値はある程度探索の時間的進度に依存する．

A* 探索においても現実には文仮説総数 (スタック幅) には制限がある．大語彙において文法のパープレキシティが増大したとき，最適解となるはずの部分文仮説が，より短い仮説の優先的展開によって，スタックからあふれる可能性が A* 探索においても存在する．

4 A* 探索の高効率化

前章で述べた問題点に対して，大語彙の条件下で効率よく A* 探索を行うための手法を幾つか提案する．各提案手法と認識処理の関係を図 2に示す．

4.1 単語対 HMM の木構造化

ヒューリスティック計算における，単語対 HMM 上の探索方向に合わせて，単語の末尾もしくは先頭から，他の単語と同じ音素を持つ部分のみを共有することで，ヒューリスティック力を全く変えずに HMM の状態数を減らすことができる．これによってヒューリスティックモデルは部分的に木構造で表現される．

ただし，異なった構文制約を持つ単語同士でその先頭もしくは終端に当たる状態を共有すると，単語対 HMM が構文制約を表現できない．そのため，構文上違った制約を

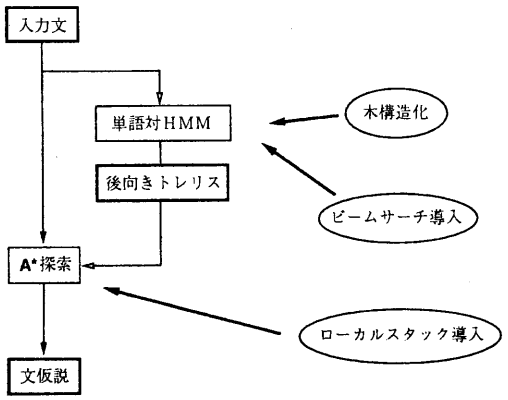
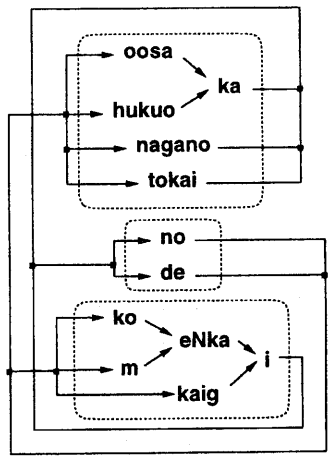


図 2: A* 探索による音声認識と各手法の関係



(注) 点線は同一カテゴリを表す

図 3: 単語対 HMM の木構造化

持つ単語同士，すなわち文法上のカテゴリが違う単語同士は状態を共有できない (図 3).

木構造化による状態数の減少の度合いは，単語の先頭もしくは終端からの音素単位での類似度に依存する．よって，大語彙では状態共有の割合は高まり，語彙の増加に比べて単語対 HMM の増大を抑えることができると考えられる．

4.2 ヒューリスティック計算におけるビームサーチ

探索で現れ得る全仮説についての推定スコアを得るために，ヒューリスティック計算においてフルサーチを行う場合，そのコストは単語対 HMM の大きさに比例する．そこでフルサーチを止めてビームサーチを行うことを検討する．この場合，将来的にスコアの高くなるヒューリスティックの経路が局所的に低くなった時に枝刈りされてしまう可能性が生じる．

また，枝刈りされることによって計算されないノードが生じるが，探索においてそのノードに対応するヒューリスティックスコアを使用する可能性があるので，足切りされたノードにはなんらかの値を変わりに代入しておく必要がある．このため推定スコアが実際のスコアより厳しくなる可能性があるので，A* 実行可能性条件を満たさず，厳密には A* 探索とはいえない．

ビーム幅の設定方法について，次の 2 つの方法を比較する．

ノード数による設定 前フレームでの計算結果から上位 b 番のスコアを持つノードを選び出し，それを枝刈り値とする．ヒューリスティック計算の最初から最後までビーム幅を一定に保つことができるが，1 フレーム計算毎に上位 b 個を選出するためにソートが必要である．ヒープソートを用いれば，N 個のノードから上位 b 個を選びだす計算量は，$O(b \log N)$ である．大語彙になるにつれて，1 フレームごとのソートによる処理コストは相対的に小さくなる．

スコアの比による設定 しきい値の算出には，前フレームでのスコアの最大値からの比を用いる．ソートが不必要なためしきい値算出の計算コストは小さいが，ビーム幅は不定である．比は α(定数) 倍とする方法が一般的であるが，単語対 HMM を用いたヒューリスティック計算では，計算の初期段階ではスコアのばらつきが大きく，計算が進むにしたがって比が一定範囲に収まってくるという傾向が認められる (図 4)．これより，しきい値の算出式を次のように

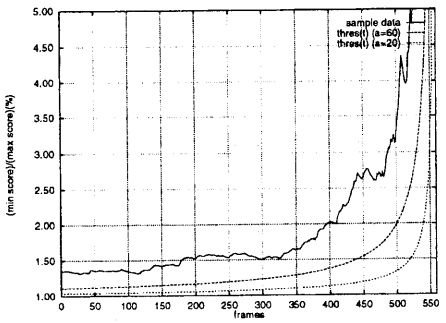


図 4: ヒューリスティック計算におけるノードのスコア比率 (MIN/MAX) の推移

表 1: タスクの仕様

	G1	G2	G3
語彙数	228	831	5019
非終端記号数	67	83	83
書き換え規則数	328	949	5137
パープレキシティ	17.2	28.3	75.6

定めた (図 4中の破線).

$$thres(t) = maxscore(t) \times (1.0 + \frac{a}{peseqlen - t}) \quad (3)$$

($peseqlen$: 入力音声のフレーム長)

ここで a は定数である. 以下この a をしきい値定数と呼ぶこととする.

4.3 探索時における展開仮説数の制限

パープレキシティの増大による最適解のスタックあふれに対して, 一仮説からの展開仮説数を制限する方法を考える. これはビームサーチにおけるローカルスタック幅の設定にあたる.

ローカルスタック幅を狭くしすぎると, かえってローカルスタックにおける最適解の枝刈りが起こる可能性が生じる. 最適解が失われる可能性があるため, 厳密には A* 探索とはいえなくなる.

5 実験的評価

前章で提案した各手法を不特定話者連続発生の文音声の認識実験で評価した. タスクドメインは "計算機による個人スケジュールの管理" であり, 発話はスケジュールの登録・変更・問い合わせに関するものである. 8 名の話者によって発声された 50 種のサンプル文を用いた.

タスクは, 語彙数の違う 3 種類を用意した. それぞれについて記述した文法の仕様を表 1に示す. 作成方法は, まず人間対人間, 人間対機械の対話サンプルなどを元に文法を作成し (G1), それに有意な単語を 800 語まで人間の手で付け加えた後 (G2), 日本語かな変換システム Wnn4 に付属の基本辞書から品詞に合わせて無作為に抜き出した単語を, 各カテゴリに追加した (G3).

木構造化の効果 まず, 各文法での木構造化による単語対 HMM の縮小率およびヒューリスティック計算時間を比較した. 結果を表 2 に示す. 制約としての強さを全く変えることなく状態数を半分程度に縮小できた. 語彙数の違いによる縮小率の大きな変化が認められないが, これは平均単語長が G1 で 7.5 音素, G2 で 8.7 音素, G3 で 7.6 音素と比較的長く, 単語の長さに対して共有できる音素の数が少なかったためと見られる. なお G3での木構造化をしない場合のトレリス計算時間は, 実装上の都合 (メモリ不足) から現段階では測定できなかった.

表 2: 単語対 HMM の木構造化の効果

文法	ノード数	トレリス計算時間
G1	3857 (58%)	1.6 (38%)
G2	15969 (57%)	9.5 (40%)
G3	70490 (50%)	58.2

(注 1) 計算時間は各サンプルの平均, 単位は秒
(注 2) 括弧内は木構造化しない場合の値に対する百分率

ヒューリスティック計算におけるビームサーチの効果 次に, 提案した各手法による認識実験の結果を表 3に示す. 使用した文法は G3 である. なお, ビームサーチにおける幅設定や制限する展開仮説数は, 何回かの予備実験の後, 認識率が下がりはじめる付近の値を用いた.

ヒューリスティック計算のビームサーチ導入については, ビーム幅を一定に保つ方法が, ビーム幅を単語対 HMM の大きさに比べてかなり狭く設定しても認識率が変わらず, 有効であることが示された. スコアの比による枝刈りは, 幅を小さく保つことが難しく, 安定した結果を得るのが難しい.

ローカルスタック導入による認識率の改善はほとんど見られない. これは, ヒューリスティックとして用いた単語対制約がかなり強力なため, 最適解がスタックからほとんどあふれることがなかったことを示している. なお, 副次的な効果として文仮説スタックに対する操作回数の減少による処理効率の改善がわずかながら認められる.

複数手法を組み合わせた評価 以上の各手法を組み合わせた結果, 10-best での単語認識率は 64.0%, 平均実行時間は 111.8 秒という結果を得た. A* 探索では現在, 探索終了までに一定数以上の仮説が展開されたら探索失敗と判断しているため, 探索失敗の検知に時間がかかる. 探索失敗の場合を除いた平均実行時間は 76.6 秒であった.

ビームサーチとの比較 最後に A* 探索とヒューリスティックビームサーチとの比較を行う. この手法では, 各仮説のスコアに A* 探索と同様に未探索部分の推定スコアを加えて評価する. 結果は表のようになった.

ビームサーチのほうが探索失敗が顕著に少なく, 単語認識率は高い. これは A* 探索が best-first 探索であるために, 似た仮説が展開され続けて探索が前に進まず結局解が得られない場合があるのに対し, ビームサーチは幅優先に近い形で探索を強制的に 1 単語づつ進めるため, とにかく何らかの解が得られるためである. しかし, ビームサーチは常に幅いっぱいの仮説を扱う必要があるため, 計算量は A* 探索よりも多い.

6 おわりに

単語対制約をヒューリスティックとする A* 探索に基づく, 大語彙の条件下での連続音声認識について検討し

表 3: 認識実験結果

手法		認識率 (%)		平均実行時間 (sec.)			失敗
heuristic 計算	探索	単語	文	第1パス	1-best	10-best	(%)
全探索	A*	58.0 (69.1)	36.8 (59.8)	58.2	109.0	155.0	15.3
ビーム (数)	A*	53.1 (63.6)	35.0 (56.3)	29.6	86.0	127.8	18.0
ビーム (比)		57.9 (69.0)	36.5 (59.5)	45.7	96.9	142.7	15.3
全探索	A*(展開仮説数制限)	58.2 (69.3)	36.8 (58.5)	58.4	100.7	138.3	13.8
ビーム (数)	A*(展開仮説数制限)	53.3 (64.0)	35.0 (55.5)	29.7	77.5	111.8	16.8
全探索	ビーム	64.6 (77.1)	37.5 (59.8)	58.5	—	198.7	—

(注 1) 文法は G3 を用い，木構造化を行っている.
(注 2) 認識率の項目の括弧内は 10-best の値.
(注 3) 平均実行時間はすべて SPARC Station 20 HS21(125MHz) 上にて測定.

た.

ヒューリスティック言語モデルである単語対 HMM の木構造化・ヒューリスティック計算におけるビームサーチの導入によって，語彙の増加に対して計算量が抑えられることが示された．一仮説ごとの展開仮説数を制限する手法は，A* 探索においては認識率の改善にほとんど影響しないが，スタック操作減少のメリットがあることが示された.

ヒューリスティックスコアを用いたビームサーチと A* 探索を比較したところ，ビームサーチの方が探索の失敗が少なく，最適解が得られなくても何らかの解を出すことで A* 探索より高い単語認識率を示した．しかし処理効率に関しては A* 探索のほうが良い.

今後は，ヒューリスティック計算において音響的に類似した単語をマージすることや，一定長以上の長さの単語をクリップすることで状態の共有度を高める手法，統計的言語モデルへの応用について検討していく.

参考文献

[1] L.R.Bahl, Gennaro, S., P.S.Gopalakrishman and R.L.Mercer: A Fast Approximate Acoustic Match for Large Vocabulary Speech Recognition, Vol. 1, No. 1, pp. 59–67 (1993).

[2] P.S.Gopalakrishnan, L.R.Bahl and R.L.Mercer: A Tree Search Strategy for Large-Vocabulary Continuous Speech Recognition, *Proc. IEEE-ICASSP*, pp. 572–575 (1995).

[3] Murveit, H. et al: Large-Vocabulary Dictation using SRI's Decipher Speech Recognition System: Progressive Search Techniques, *Proc. IEEE-ICASSP*, Vol. 2, pp. 319–322 (1993).

[4] Li, Z., Kenny, P. and O'Shaughnessy, D.: Searching with a transcription graph, *Proc. IEEE-ICASSP*, Vol. 1, pp. 564–567 (1995).

[5] 野田喜昭, 鵜峨山茂樹: 前向き尤度を用いた A* ビーム探索による HMM-LR 音声認識, 電子情報通信学会技術研究報告, SP94-23 (1994).

[6] 河原達也, 松本真治, 堂下修司: 単語対制約をヒューリスティックスとする A* 探索に基づく会話音声認識, 電子情報通信学会論文誌, Vol. J77-D-II No.1, pp. 1–8 (1994).

[7] 門前聖康, 好田正紀: HMM-LR による文節音声認識における Viterbi-best first サーチの検討, 電子情報通信学会技術研究報告, SP93-109 (1993).

[8] Soong, F. K. and Huang, E.-F.: A tree-trellis based fast search for finding the n best sentence hypotheses in continuous speech recognition, *Proc. IEEE-ICASSP*, pp. 705–708 (1991).

[9] 北岡教英, 河原達也, 堂下修司: 格構造を利用した right-to-left A* 探索に基づく会話音声認識, 電子情報通信学会技術研究報告, SP93-19 (1993).