文法カテゴリ対制約を用いたA* 探索に基づく大語彙連続音声認識パーザ

李 晃伸　河原 達也　堂下 修司

京都大学大学院 情報学研究科 知能情報学専攻
〒606-8501 京都市 左京区 吉田本町
e-mail: ri@kuis.kyoto-u.ac.jp

あらまし 大語彙の文法ベースの連続音声認識では，探索時に仮説の単語ネットワークが爆発する，また文法による単語の絞り込みも不十分である．これに対して (1) 元の文法から抽出した単語 (カテゴリ) 対制約によってネットワークの大きさを抑え，かつ (2) 文法カテゴリごとに単語辞書を木構造化することで，効率的に強力なヒューリスティック計算を行なう．さらに (3) この第1パスの結果をインデックス化しそれを元に展開単語を絞り込むことで，大語彙で効率の良いA* 探索を実現する．認識実験の結果，提案手法を用いた汎用連続音声認識パーザ Julian は，標準的な1パスビーム探索に比べて計算量がはるかに少なくて済み，また複雑なネットワークに対しても安定して動作することが示された．最終的な性能は，800語のタスクでほぼ実時間で97.1%，5,000語のタスクでは実時間の2倍強で91.2% の単語認識精度を達成した．

キーワード 大語彙，連続音声認識，パーザ，有限状態文法，A* 探索

Large Vocabulary Continuous Speech Recognition Parser based on A* Search using Grammar Category-Pair Constraint

Akinobu Lee　Tatsuya Kawahara　Shuji Doshita

Graduate School of Informatics
Kyoto University, Kyoto 606-8501, Japan
e-mail: ri@kuis.kyoto-u.ac.jp

Abstract In grammar-based LVCSR, the expanded word hypothesis network grows enormously and the grammar only is not enough for word prediction. We propose an efficient two-pass search algorithm by (1)using word(category)-pair constraint as heuristics and (2)tree-organizing the word lexicon for each grammar category, both to represent the whole network in a compact loop structure. Futhermore, (3)the survived words on the first pass are indexed to eliminate candidates to be accessed on the second pass. Experimental results show that our depeloped portable FSA-based parser, named Julian, shows greater performance in costs and stability than a typical one-pass beam decoder. Finally, our parser archieved a word accuracy of 97.1% with almost real time processing on a 800-word task, and 91.2% with 2.5 times the real time on a 5000-word task.

key words large vocabulary, CSR, parser, FSA, A* search

1 はじめに

数千語から数万語の大語彙を対象とする連続音声認識は困難な問題の一つであり，特に効率よく最適解を見つける探索アルゴリズムが要求される．

大語彙の連続音声認識で近年盛んに用いられているのは，単語N-gramに代表される統計的言語モデルに基づくアプローチ[1][2][3]であるが，統計的に言語モデルを推定するには大量の整ったテキストコーパスが必要であるため，新聞記事等の大規模コーパスから汎用的なディクテーションシステムを作成することが多い．一方，情報検索や予約システムなどの音声インタフェースにおいては，タスクやドメインを反映した言語モデルが必要である．

しかし，タスクごとに大量のデータを収集しラベリングを行なうのは大変な労力を要するので，統計的なモデルの構築は容易でない．また地名や商品名のように統計的なモデルにあまり意味がない場合もある．このため人手で文法や語彙を指定するほうがタスクに特化したモデルの構築が容易であり，語彙の入れ替えなどの変更が簡単に行なえる利点がある．

本論文では，大語彙の条件下における文法ベースの音声認識（パージング）アルゴリズムを研究の対象とする．パージングについては，これまで主に数百から千語程度の小中語彙での検証しか行なわれていない[4][5][6]．

大語彙のタスクでは，単語辞書が語彙に比例して巨大化する．また文法では単語パープレキシティの増大に伴って仮説ネットワークが爆発する．このため単純な1パスのビーム探索ではビーム幅を非常に大きくとる必要があり，効率が悪い．

これに対して，単語対制約をヒューリスティックとするA*探索[4]を大語彙へ適用することを考える．中語彙のタスクでは解の最適性を重視してヒューリスティック計算を行っていたが，大語彙においてはこのヒューリスティック計算を候補の予備選択と位置づけ，全体として効率の良い探索アルゴリズムの実現を目指す．処理は2パスで構成され，第1パスでは，単語対制約を用いて探索空間である仮説ネットワークをコンパクトに抑える．その認識結果に基づいて第2パスで再探索を行なうことで，効率よく高精度な解が得られる．さらに効率化のために，第1パスのビーム探索化，文法のカテゴリ単位での木構造化，および第2パスでの音響的照合に基づく展開単語の絞り込みを導入する．

以下，まず大語彙における問題点，および従来手法である1パスビーム探索と単語対制約をヒューリスティックとするA*探索について述べる．そしてA*探索を大語彙で適用するための手法について述べ，具体的アルゴリズムを示す．これを実装した大語彙連続音声認識パーザ Julianの仕様を述べた後，1パスビーム探索と5,000語のタスクにおける認識実験による比較評価の結果を報告する．

2 文法に基づく大語彙連続音声認識

文法に基づく連続音声認識の大語彙における問題点，および探索法として1パスビーム探索と基本的なA*探索について述べる．

一般に連続音声認識は，与えられた言語的・音響的制約のもとで，最も確率の高い単語列からなる文候補を見つけ出す探索問題として定式化される．言語制約として文法を用いる場合の探索空間は，その文法および単語辞書をオートマトンとして展開した単語仮説のネットワークとなる．有限状態文法を用いる場合は，全ての規則をオートマトンとしてあらかじめ展開しておくことが可能である．また文脈自由文法の場合は，探索時にパーザにより次単語を逐次的に予測しつつ，必要に応じて展開する．

2.1 大語彙の問題

一般に大語彙においては，パープレキシティの増大から探索時に多くの仮説を扱う必要がある．特に文法ベースの連続音声認識においては，以下の要因から単語仮説ネットワークが爆発し，探索が困難となる．

文法カテゴリの多重性 単語N-gramモデルが1,2単語の履歴しか考慮しない単純なモデルであるのに対して，文法では文全体に対する履歴を考慮した制約が与えられる．そのため大語彙においては仮説ネットワークの大きさが爆発的に増大しやすい．図1に文法と展開後の仮説ネットワークの例を示す．なお文法は単語のカテゴリのみを記述してあり，実際には各パス上でカテゴリに含まれる全単語が展開される．

このように文法には，コンテキストが異なるものの局所的には同じ構文規則が頻繁に含まれるため，同じ単語が類似した位置に何度も展開される（図1のEVENTやPLACE等）．特に大語彙のタスクでは，地名や商品名といった特定のカテゴリに大量の単語が登録される傾向が強いが，このカテゴリが多重に展開されると仮説ネットワークの増大が著しい．

文法においてカテゴリが重複せずに，展開後のネットワークが小さくなるようなコンパクトな文法を意識して記述することも可能であるが，文法を作成するアプリケーション開発者が，常に文法の専門家であるとは限らない．また展開後のネットワーク（オートマトン）の最小化問題はNP困難であり，コンパイラ等

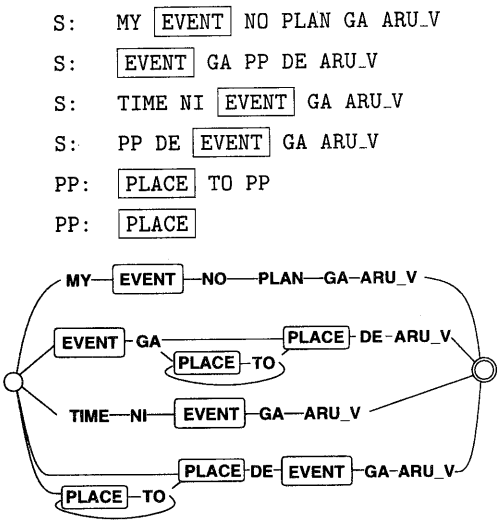

図 1: 文法から展開される単語仮説ネットワーク

による自動最小化も限界がある．このように文法ノードの多重性は避けがたい問題である．

単語辞書の巨大化　認識用の単語辞書は語彙サイズに比例して巨大化する．このためこれを文法に従って展開した単語仮説ネットワーク全体の大きさも，これに比例して増大する．

これに対して，同じプレフィックスを持つ単語同士で状態を共有することで，単語辞書の大きさを抑えることが考えられる [7]．この木構造化は語彙が大きくなるほど効果があり，単語 N-gram ベースの大語彙連続音声認識では不可欠である．しかし文法を用いたパージングの場合は，異なる文法ノード上の単語同士で単語の先頭（＝木の根ノード）を共有するとネットワーク上で文法制約を表現できないという問題が生じる．

不十分な候補の絞り込み　文法ベースの探索では，単語 N-gram と異なり，文法制約から展開単語を決定的に絞り込むことができる．しかし語彙サイズが大きくなるにつれて，特に地名などのように特定の文法カテゴリの語彙が大きい場合は，この絞り込みが十分に機能しない．展開のたびに多くの単語を照合する必要から，計算コストは文法の単語パープレキシティに比例して増大する．

これらの問題は，特に語彙サイズの増大に伴い顕著になり，複合的に作用して探索を困難にする．

2.2　1 パスビーム探索

一般に広く用いられている 1 パスのビーム探索法 [8][5] においては，探索時点までのスコアに基づいて枝刈りをしながら探索を進める．そのため局所的なノイズや類似性の影響を受けやすく，最適解が探索途上で失われないようにするためにはビーム幅を大きくする必要がある．

また探索の幅優先的性質から，処理量は文法のパープレキシティに比例する．特に前述のような文法の多重性によってあちこちの文法ノードに同じ単語が出現すると，評価値の高い同じ単語がビームの上位を占めるため，これもビーム幅を大きくさせる要因となる．

2.3　A* 探索

これに対して 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で探索を進める．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，$g(n)$ は既に展開された区間の評価値，$\hat{h}(n)$ は未展開部分のヒューリスティックな推定値である．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* 適格性）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 $\hat{h}(n)$ ができるだけ実際の値に近いことが望ましい。

トリートレリス探索 [9] においては，$h(n)$ を求めるために，探索の前処理として探索とは逆方向に認識処理を行い，そのトレリス上の評価値を保存しておく．そして探索時にこの評価値をヒューリスティックとして用いる．つまり認識処理は，ヒューリスティックの計算とパージングの 2 パスから構成される．

我々はこれまでに，このヒューリスティックの制約として単語対文法を用いることを提案した [4]．これは元の文法から単語間の接続に関する情報のみを抽出したものである．元の文法のサブセット（言語的にはスーパセット）であり，探索に用いる制約よりも弱いため A* 適格性を満たす．また単純な単語連接や音素連接の制約よりも強力であり，良いヒューリスティックとなる．

このA*探索は，これまでに小・中語彙のタスクにおいて有効性が示されている[4][10]．しかしA*探索において適格性を厳密に満たした探索を行うには，パージング中に出現しうる全単語について，あらかじめヒューリスティック評価値を計算しておく必要がある．そのため第1パスでは基本的に全探索となり，大語彙ではヒューリスティックの計算量からこのまま適用することは不可能である．

3 大語彙におけるA*探索の実現

A*探索を大語彙で実現するためには，ヒューリスティックの計算量の増大が大きな問題となる．これは1パスのビーム探索における計算量増大の問題と基本的に同じであるが，異なるのは，A*探索ではマルチパスで処理を行うことにより，第1パスで近似を導入しても，第2パスでそのエラーの回復が行えることである．すなわち，ヒューリスティック計算のプロセスを単語候補の予備選択と見なして，コンパクトな制約によって候補の絞り込みを行い，その結果に基づいて第2パスで再探索することによって最終的に高精度で効率も良い探索を実現する．ヒューリスティックが厳密なA*適格性を満たさなくなるが，十分な候補を後段へ残せば実際の認識において大きな問題にはならないと考えられる．

具体的には，ヒューリスティック計算におけるビーム幅の設定のほかに，2.1節で述べた大語彙での各問題点に対して以下のアプローチをとる．

3.1 単語対制約による束ね効果

単語対制約をヒューリスティックのための制約として用いることで，ネットワークの増大を大幅に抑えられる．1パスビーム探索では，巨大な探索ネットワークを一回で走査するため探索効率が悪い．一方単語対制約による文法ネットワークは単純なループで表現されるため，文法上で異なるノードの同一単語がネットワーク上に複数現れることがない．またどのような文法からも安定して語彙サイズに比例したノードからなるコンパクトなネットワークが得られる．またこの単語対制約による認識処理結果をヒューリスティックとして再び第2パスで文法による探索を行なうことで，最終的には同じ制約による結果が得られる．

単語対制約によって同じ仮説単語を1つにまとめることは，仮説の束ね効果の一種といえる．束ね処理に関しては，これまでに1パスのフレーム同期ビーム探索において，伊藤ら[11]が音韻レベルで，渡辺ら[12]が単語レベルで提案している．どちらも近傍に出現した同一の音韻もしくは単語の照合を1回のみ行い，他の候補はその結果を再利用することで近似するものである．しかしこれらは束ね処理と探索を別々に行うのに対して，単語対制約はあらかじめ仮説ネットワークそのものをコンパクト化するため，特別な機構は必要なく単純に実現できる．また認識結果が束ねによる誤差を含むのに対して，A*探索では第2パスの再探索でより正確な結果を得ることが可能である．

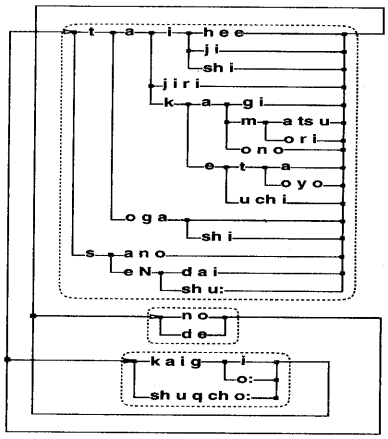

図2: 木構造化された文法カテゴリ対ネットワーク

3.2 単語辞書のカテゴリ単位の木構造化

単語辞書に関しては，文法のカテゴリごとに別々に木構造化を行なうことで，単語対ネットワークを保ったままで単語辞書の大きさを抑えられる[13]．つまり，制約の表現を単語対から文法カテゴリ対に変える．カテゴリごとに木構造化した文法カテゴリ対制約のネットワークの例を図2に示す．

これによる探索空間の削減は，特に地名などのように1カテゴリに多数の単語が登録される場合に効果が大きい．逆にカテゴリが多数設定され1カテゴリ当たりの単語が少ない場合は，木構造化の効果は小さいが，その分文法による絞り込みの効果が大きくなるので，全体として処理量に大きな差はない．

3.3 音響的照合による絞り込みの導入

大語彙においては文法による単語予測のみでは十分に候補を絞れない．これに対して，第1パスのヒューリスティック計算の結果のトレリスに，フレームごとにビーム内に残った単語のインデックスを付加する[3]．第2パスの探索における仮説の展開の際には，文法制約に加えてこの予備選択された単語のイン

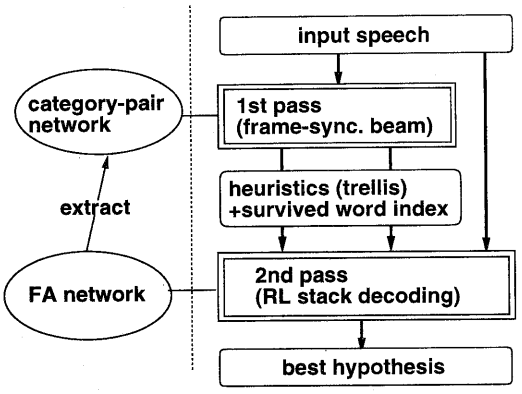

図 3: 探索アルゴリズムの構成

デックスを参照することで，高速に照合を行える．

地名などのように，そのカテゴリだけで語彙のほとんどを占める場合に，この音響的照合からの絞り込みは特に効果的であると考えられる．

3.4 パージングアルゴリズム

この A* 探索に基づく大語彙連続音声認識は，図3に示すように2パスで構成される．第1パスは探索の前処理であり，文法カテゴリ対制約に従って全入力に対してフレーム同期に left-to-right に認識処理を行い，途中の各フレームごとに，ビーム内に残った単語の終端の評価値およびそのインデックスをトレリスの形で保存しておく．第2パスでは文法に従ってスタックデコーディング探索を行うが，その際探索を逆の right-to-left 方向に進め，仮説の未探索部分のヒューリスティックとして，対応する第1パスのトレリスを接続する．

以下に探索（第2パス）の具体的アルゴリズムを示す．ここで部分文仮説を $W_n = w_n, w_{n-1}, ..., w_1$（$w_1$ が文末）と表し，入力長を T とする．時刻 T から t までの文仮説 W_n の後向きの尤度を $g(W_n, t)$，時刻 t に単語 w で終わる前向きヒューリスティックを $h(w, t)$，文仮説 W_n の評価値を $f(W_n)$ とそれぞれ表す．また第1パスで得られた，フレーム t においてビーム内に残った単語の集合を $index(t)$ とする．

1. 文末に出現し得る各単語 w_1 について，1単語からなる部分文仮説 W_1 を生成し，以下の評価値を計算して仮説スタックに入れる．

$$f(W_1) = h(w_1, T)$$

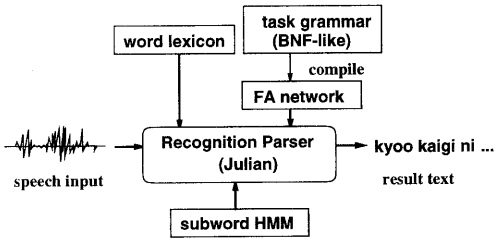

図 4: 連続音声認識パーザ Julian を用いた認識システム

2. 仮説スタックから評価値の最も高い部分文仮説 W_n を取り出す．W_n が受理状態にありかつ入力始端に達していれば，解として出力し探索を終了．

3. 仮説 W_n の最終単語 w_n の後向きトレリスを展開する．

$$g(W_n, t) = \begin{cases} \max_{t'}\{\beta(w_n, t, t') + g(W_{n-1}, t')\} & \text{if } n > 1 \\ \beta(w_n, t, T) & \text{if } n = 1 \end{cases}$$

ただし，$\beta(w_n, t, t')$ は単語 w の t' から t までの後向き尤度である．

4. 文法制約上，仮説 W_n に接続し得る単語のうち，第1パスのインデックスに残っていたもののみについて，W_n に接続して新たな仮説 W_{n+1} を生成する．新たな仮説の評価値は次のように求める．

$$f(W_{n+1}) = \max_{t:w_{n+1} \in index(t)} \{h(w_{n+1}, t) + g(W_n, t)\}$$

5. 生成した全ての W_{n+1} を仮説スタックに入れる．(2) へ戻る．

4 大語彙連続音声認識パーザ Julian

前節の探索アルゴリズムを，大語彙に適用可能な汎用の連続音声認識パーザとして実装した．京都大学における名称は Julian である．これに文法および音響モデルを与えることで，任意のタスクドメインに対する認識システムが構成できる．Julian を用いた連続音声認識システムの構成例を図4に示す．

文法と単語辞書は別々に用意する．文法は，単語のカテゴリを終端記号としてBNF形式で記述する．単語辞書は，各文法カテゴリごとに，それに属する単語の名前およびサブワード（音素）列を列挙する．文法は専用コンパイラによって有限状態オートマトン（FA）に変換される．BNF形式では文脈自由文法のクラスまで記述可能であるため，コンパイル時に正規文法のクラスに収まるかどうかがチェックされる．再帰性は右再帰性のみ扱える．また文法カテゴリ対文法は，パーザの実行時に自動抽出される．

音響モデルは，HTK[14]のフォーマットに対応している．使用可能なHMMの型は連続混合型で，モデル数・全状態数・混合数について任意のモデルを扱える．またtriphoneなどの音素環境依存モデルを扱うこともできる．このとき環境依存性は，扱いの簡便さから第1パスでは単語内のみ扱い，第2パスで単語間まで考慮する．

このようなインタフェースにより，Julianは高いポータビリティを実現しており，様々なアプリケーションに容易に適用することができる．

5 実験的評価

同一の音響モデルおよび等価な文法を用いて，等しい言語的・音響的制約のもとで認識実験を行ない，探索アルゴリズムの精度と効率を調べる．

比較対象として，フレーム同期の1パスビーム探索を行なう標準的なソフトウェアであるHTKの連続音声認識モジュールHViteを用いた．

5.1 実験条件

タスクドメインは個人スケジュール管理であり，スケジュールの登録・削除・変更・問い合わせに関する発話を認識対象とする．テストセットとして男性8名によって発声された50種のサンプル文の計400サンプルを用いた．文の平均発話長は3.2秒であり，一文当り平均6.2単語からなる．文例を図5に示す．

タスク文法として，任意フレーズの繰り返しを許すフレーズ単位の構文制約PGと，フレーズの文中での位置を固定的に記述した文単位の構文制約SGの2種類を記述した．またそれぞれについて語彙サイズの異なる2種類の単語辞書を用意した．この計4つの文法の諸元を表1に示す．なおFA状態数とは，文法を有限状態オートマトンに変換した後のノード数である（単語のカテゴリ単位）．

PG1/SG1はタスクドメインで典型的に出現し得るフレーズを人間が1つづつ登録した文法であり，PG2/SG2はそこに辞書から抽出した名詞（特に地名・時間を表す名詞）や動詞を追加して語彙を増加させたものである．SG1,SG2は構文制約を発話のパターンごとに文単位で記述してあるため，変換後の文法ノードはPG1,PG2に比べて10倍以上多い．

- 明日の2時から3時まで、二研で音声研究会を開きたい。
- 金曜日の予算会議の場所を小会議室に変更したい。
- 今日のセミナーは何時からですか？
- では、16時からにします。

図5: 評価に用いた文例

表1: タスク文法の諸元

grammar	lexicon size	word perp.	num of FA nodes
PG1	806	91.6	257
PG2	4439	257.1	257
SG1	833	28.7	5061
SG2	5023	76.0	5061

音響モデルはmonophoneおよびtriphoneのHMMを用いる．日本音響学会研究用連続音声データベースの男性話者30人分と日本音響学会新聞記事読み上げ音声コーパスのうち男性話者100人分の発声データで学習した不特定話者HMMを用いる[15]．monophone HMMの総状態数は147, triphone HMMの総状態数は2110であり，それぞれ1状態あたり16混合分布を持つ．

5.2 評価基準

評価値として，認識率とビーム幅を用いる．ここでビーム幅とは，1フレームごとにViterbi計算を行なうHMMの（平均）状態数と定義する．

HTKについては，音素モデル数によるビーム設定値をHMM状態数に換算した値を用いる．実際は使用した音素モデルが全て3状態からなるので，上記オプションで設定した値を3倍した値がここでのビーム幅となる．

2パス探索のJulianについては，そのビーム幅$\hat{b}$を次式のように定める．

$$\hat{b} = b_1 + \hat{b_2} \qquad (3)$$

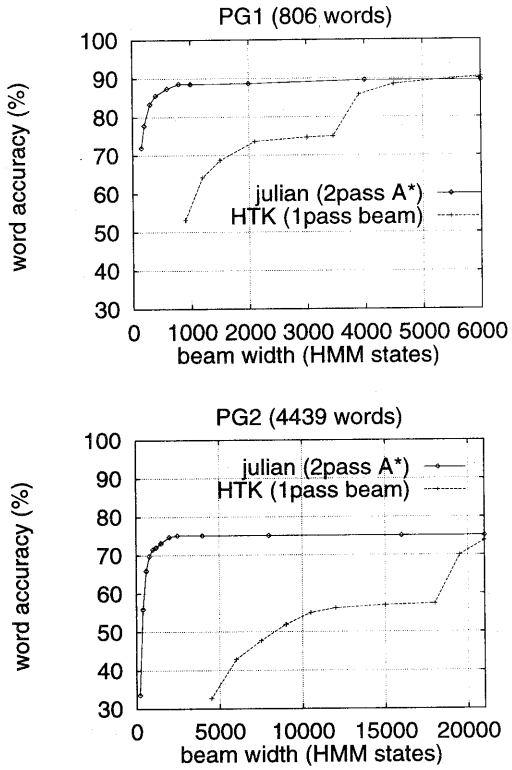

図 6: 探索性能の比較： Julian 対 HTK

ここで b_1 は第 1 パスのヒューリスティック計算でのビーム幅であり，$\hat{b_2}$ は第 2 パスの計算量から換算した仮想的なビーム幅である．$\hat{b_2}$ は，バージングの際の全ての仮説に含まれる HMM 状態数から以下の式によって正規化した値を求め，その全サンプルでの平均値を求める．ここで *npop* はある解が得られるまでに仮説が展開された回数，*nword* はその得られた文仮説の単語数，そして *avg_state* は 1 単語当たりの平均状態数である．

$$\hat{b_2} = \frac{npop}{nword} \times avg\_state \qquad (4)$$

5.3 1 パスビーム探索との比較

文法 PG1 および PG2 における一位候補の単語認識精度を，ビーム幅 $\hat{b}$ ごとに図 6に示す．なお，音響モデルは monophone モデルを用いている． Julian では，HTK に比べてはるかに小さいビーム幅で同等の認識精度が得られた．1 パスビーム探索の HTK では，枝刈りの精度が良くないのに加えて，パープレキシティの増大に伴って単語ネットワークが組み合わせ的に巨大化するため，十分な精度を得るにはビーム幅をかなり大きくする必要があった．一方 A* 探索の Julian では，第 1 パスでよりコンパクトなカテゴリ対ネットワークを用いることでビーム幅を抑えることができた．また強力なヒューリスティックによる深い先読みが行えるため，第 2 パスではほとんどのサンプルで best-first に解が求められた．第 2 パスのビーム幅 $\hat{b_2}$ は第 1 パスの幅によらず 24 から 26 と，第 1 パスに比べてはるかに少ない計算量で済んだ．

語彙数に関しては，PG1，PG2 とも語彙サイズにほぼ比例した大きさのビーム幅が必要であることがわかる．このことからさらに大語彙ではこの差はより顕著であろうと考えられる．

文法 SG を試したところ，HTK はネットワークの爆発から動作が困難となり，安定した結果が得られなかった．一方 Julian は安定して動作し，SG1 で 97.1%，SG2 で 91.2% の単語認識精度を示した．一般に BNF などの文法記述からコンパクトな文法ネットワークを生成するのは容易ではないが，カテゴリ対文法は単純なため元文法の複雑さによらず安定して抽出できる．このため Julian はどのような文法に対してでも安定して動作すると考えられる．

次に処理効率を比較する．PG1,PG2 で，認識率がほぼ飽和するビーム幅における平均処理時間を表 2に示す．Julian は HTK に比べて，はるかに少ない処理時間で同じ認識率を得ることができた．

表 2: 1 サンプルあたりの平均処理時間

grammar	avg. time (sec.) Julian	HTK
PG1	3.2 (1000)	10.5 (4500)
PG2	7.2 (2500)	44.2 (21000)

CPU: UltraSPARC 300MHz
RT: 3.2 sec.　括弧内はビーム幅

5.4 音素環境依存モデルの効果

音素環境依存モデル (triphone) を導入した場合の，認識精度および処理コストの変化について調べた．Julian で文法 SG1,SG2 を用いたときの認識精度および処理コストを表 3に示す．triphone での認識精度は，monophone に比べて 1% 弱の改善にとどまった．一方 triphone はメモリを多く必要であり，

表 3: 音素環境依存モデルを用いた認識性能

AM	grammar SG1(833)	SG2(5023)
mono	97.1 / 3.6 / 9	91.2 / 8.2 / 12
tri	96.5 / 9.5 / 26	91.8 / 18.3 / 29

注 1：単語認識精度 (%)/ 平均処理時間 (sec.)/ 動作に必要なメモリ量 (MB)

またキャッシュが効きにくいことから出力確率計算の時間がかかり，処理時間も 2 倍以上となった．

6 おわりに

文法に基づく連続音声認識において，A* 探索を大語彙の条件下で実現し，評価を行った．5,000 語レベルのタスクによる認識実験の結果，文法カテゴリ対制約をヒューリスティックとする A* 探索は

1. 元文法から導出したコンパクトな文法カテゴリ対制約を用いた効率のよい探索
2. その処理結果をヒューリスティック（先読み情報）とした高精度で best-first な再探索

の 2 パスの組合わせによって，単純な 1 パスのビーム探索に比べて，小さいビーム幅で高精度な探索を行えることが示された．

語彙や複雑さの異なるいくつかの文法で比較した結果，語彙が大きくなるほど両者の性能の差が顕著に現れることが確認された．また文法カテゴリ対制約によるヒューリスティック計算は，タスク文法の規模や複雑さに対して頑健であり，どのような文法でも安定して動作することが示された．

実装した大語彙連続音声認識パーザ Julian は，語彙数 800・パープレキシティ 28.7・monophone モデル という条件で，ほぼ実時間の処理で 97.1% の単語認識精度を示し，また 5,000 語のタスクでも高い認識精度を示した．今後は種々の音声インタフェースなど，他の音声分野への応用が期待できる．

謝辞 音響モデルは IPA の「日本語ディクテーション基本ソフトウェア 97 年度版」のものを使用した．

参考文献

[1] 西村雅史, 伊東伸泰. 単語を認識単位とした日本語ディクテーションシステム. 電子情報通信学会論文誌, Vol. J81-D-II No.1, pp. 10–17, 1998.

[2] 松岡達雄, 大附克年, 森岳至, 古井貞熙, 白井克彦. 新聞記事データベースを用いた大語い連続音声認識. 電子情報通信学会論文誌, Vol. J79-D-II No.12, pp. 2125–2131, 1996.

[3] 李晃伸, 河原達也, 堂下修司. 単語トレリスインデックスを用いた大語彙連続音声認識エンジン JULIUS. 信学技報, SP98-3, 1998.

[4] 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.

[5] 北研二, 川端豪, 斎藤博昭. Hmm 音韻認識と拡張 lr 構文解析法を用いた連続音声認識. 情報処理学会論文誌, Vol. 31, No. 3, pp. 472–480, 1990.

[6] 伊田政樹, 中川聖一. 音声認識におけるビームサーチ法と a* 探索法の比較. 電子情報通信学会技術研究報告, SP96-12, 1996.

[7] J.W.Klovstad and L.F.Mondshein. The CASPERS linguistic analysis system. In *IEEE Sympo. Speech Recognition*, pp. 234–240, 1974.

[8] 中川聖一. 文脈自由文法のフレーム同期型構文解析法による連続音声認識. 電子情報通信学会論文誌, Vol. J70-D, No. 5, pp. 907–916, 1987.

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

[10] Douglas B. Paul. An efficient A* stack decoder algorithm for continuous speech recognition with a stochastic language model. In *Proc. IEEE-ICASSP*, volume 1, pp. 25–28, 1992.

[11] 伊藤克亘, 速水悟, 田中穂積. 音素文脈依存モデルと高速な探索手法を用いた連続音声認識. 電子情報通信学会論文誌, Vol. J75-DII, No. 6, pp. 1023–1030, 1992.

[12] 渡辺隆夫, 吉田和永, 畑崎香一郎. バンドルサーチ法を用いた連続音声認識の高速化. 電子情報通信学会論文誌, Vol. J75-DII, No. 11, pp. 1761–1769, 1992.

[13] 李晃伸, 河原達也, 堂下修司. A* 探索に基づく大語彙連続音声認識. 情報処理学会研究報告, 96-SLP-11-4, 1996.

[14] S.Young, J.Jansen, and J.Odell D.Ollason P.Woodland. *The HTK Book*, 1995.

[15] 武田一哉, 峯松信明, 伊藤彰則, 伊藤克亘, 宇津呂武仁, 河原達也, 小林哲則, 清水徹, 田本真詞, 荒井和博, 山本幹雄, 竹沢寿幸, 松岡達雄, 鹿野清宏. 大語彙日本語連続音声認識研究基盤の整備 –汎用音素モ デルの作成–. 情報処理学会研究報告, 97-SLP-18-3, 1997.