特集 音声情報処理技術の最先端

重み付き有限状態トランスデューサによる音声認識

堀 貴明
NTT コミュニケーション科学基礎研究所
hori@cslab.kecl.ntt.co.jp

塚田 元
NTT コミュニケーション科学基礎研究所
tsukada@cslab.kecl.ntt.co.jp

音声認識技術の進展に伴い，数万単語以上の語彙を扱う音声認識が汎用のパーソナルコンピュータ上で動作するようになった．この背景には，これまでに培われた音声認識アルゴリズムの効率化に関する数々の研究成果がある．しかし，その一方で音声認識のアルゴリズムは複雑化し，システムの保守・拡張が困難になる傾向があった．近年，このような従来手法に代わるアプローチとして，重み付き有限状態トランスデューサ（Weighted Finite-State Transducer: WFST）による音声認識が注目を集めている．WFSTは確固たる理論に支えられ，拡張性の高いシステムの構築と従来手法を凌ぐ高速な音声認識を実現する．本稿では，WFSTによる音声認識を紹介し，従来の音声認識との相違点を明らかにする．さらに，WFSTによってもたらされる音声認識の今後の展開について述べる．

■音声認識におけるデコーダ技術

音声認識は，人間の発声した音声信号をその発声内容に対応する言語表現に変換する処理である．その処理の中では，音声信号処理，音響モデル，言語モデル等が複雑に組み合わされ，デコーダと呼ばれるプログラムがこれらを統括している．

デコーダの役割は，入力された音声信号に対して音響的にも言語的にも最も適合する言語表現を探し出すことである．現在主流となっている，確率モデルによる音声認識の枠組みに従えば，次の式を解くことになる．

$$\hat{W} = \underset{W}{\operatorname{argmax}}\ P(O|W)P(W) \qquad (1)$$

ここで，Oは入力音声，Wは任意の単語列である．$P(O|W)$は単語列Wの音響信号がOである確率（音響尤度）であり，音響モデルによって計算される．$P(W)$はWの言語としてのもっともらしさを示す確率（言語尤度）であり，言語モデルによって計算される．そして$\hat{W}$はこれら確率の積を最大とする単語列（音声認識結果）である．

デコーダに求められるのは，誤りなく，素早く，最適な単語列$\hat{W}$を探し出すことである．しかし，大語彙連続音声認識（一般に数万単語以上の語彙を対象とする文章の音声認識のことを指す）を考えた場合，わずか数単語の長さの単語列であっても，その候補数は莫大である．たとえば，5万単語を登録しているシステムで，3単語からなる単語列が発声されたとすると，その候補には$(5\text{万})^3$通りの可能性がある．実際には発声された単語の数も未知であるため，候補数はさらに多い．したがって，デコーダには多数の候補から効率的に最良候補を探し出す高度な探索技術が必要とされる．

一般的な音声認識システムは，音声単位（音素等）とそれらの音響的特徴を表す音響モデル，個々の単語の発音を表す単語発音辞書，文法規則や単語の連鎖する確率を与える言語モデルを備えている．多くの場合，音響

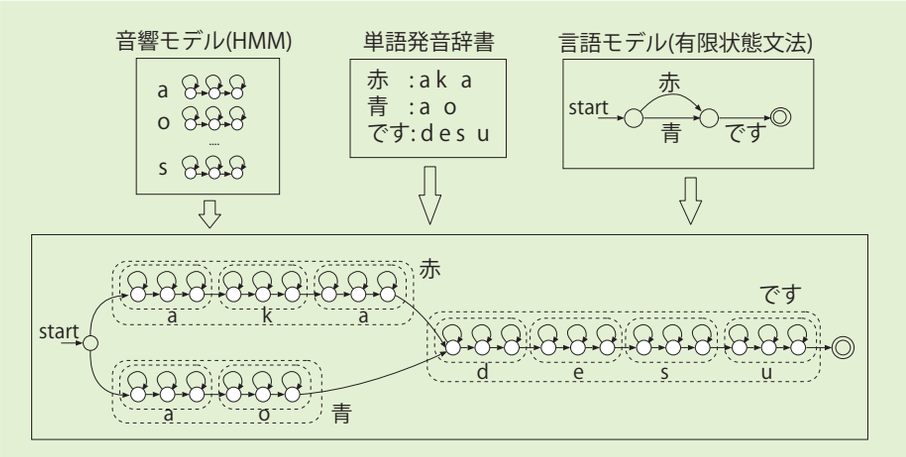

図 -1　音声認識における探索空間表現

モデルには隠れマルコフモデル（Hidden Markov Model: HMM），言語モデルには有限状態文法（正規文法）や N-gram が用いられる．これらのモデルを組み合わせることによって探索空間が構成され，それは HMM の状態をノードとするネットワークで表現される．図-1 は，HMM，単語発音辞書，有限状態文法によって探索空間が構成された例を表す．図に示すように，単語の中に音素列，音素の中に HMM が埋め込まれたネットワークとなっている．

この探索ネットワークが構築されれば，音声認識処理はネットワークの探索問題に帰着される．すなわち，入力音声信号に対して最も適合するネットワーク内の経路（最適状態遷移過程）を見つけ，その経路に対応する単語列を音声認識結果とする．したがって，式 (1) は次のように書き換えることができる．

$$\hat{W} \simeq \underset{W(S)}{\operatorname{argmax}} \prod_{t=1}^{T} P_A(o_t,\, s_t|s_{t-1})\, P_L(s_t|s_{t-1}) \qquad (2)$$

ここで，$o_1 \ldots o_T$ は入力音声信号の短時間スペクトルから得られる特徴ベクトルの時系列，Tは入力音声の長さ（フレーム数），$S=s_1 \ldots s_T$は時刻 1 ～ Tにおいて可能な任意の状態遷移過程（s_t は t 番目の入力フレームにおいて到達した状態）を表す．$P_A(o_t,s|r)$ は，HMM の状態 r から状態 s へ遷移し特徴ベクトル o_t が出力される確率（状態遷移確率×出力確率）である．$P_L(s|r)$ は状態 r から状態 s への遷移における言語モデルの確率であり，r,s が単語の境界である場合に適用される（境界以外は 1 とする）．$W(S)$ は，状態遷移過程 S に対応する単語列を表す．

$\hat{W}$ を求めるための最適状態遷移過程の探索には，動的計画法に基づくビタビアルゴリズムが有効である．しかし，基本的なビタビアルゴリズムでは，大きな探索空間を対象とする場合に計算量が膨大となり，大語彙連続音声認識をオンラインで動作させることは困難である．そのため，ビタビアルゴリズムの計算量を削減し，音声認識を高速化する手法がこれまで検討されてきた．

それらの検討には，探索ネットワークの冗長性を削減する工夫，探索途中での候補の枝刈り，音響尤度の近似計算，簡略化したモデルで候補を絞り込み詳細なモデルで最終結果を求めるマルチパス探索など，数々の手法がある．これらの手法を併用することで，大語彙連続音声認識が汎用のパーソナルコンピュータ上で動作する．しかし，数々の技術を併用することでデコーダのプログラムは複雑になる傾向があり，デコーダの保守・拡張には卓越したプログラミングスキルと多大な労力が必要とされる．また，どのような手法をどのように組み合わせれば最良か，その手法は他のタスクにおいても有効か，といった見通しが立ちにくいことも問題の 1 つになっている．その一方で，デコーダには，コンピュータの小型化やコスト削減，もしくは，より複雑なモデルを導入するために，常に高速化が求められる現実がある．

■ WFST による音声認識

近年，従来の音声認識に代わるアプローチとして，WFST による音声認識が注目を集めている[3]．WFST は従来のデコーダ技術の問題点の多くを解決する有効な手法であり，その特徴は次の 3 つに集約される．

(1) 個別に設計したモデルをオブジェクトとして組み合わせることで，複雑な音声認識処理を容易に実現できる．

(2) 探索ネットワークの構築とデコーダのプログラムとは切り離して考えることができるため，デコーダの保守・拡張が容易になる．

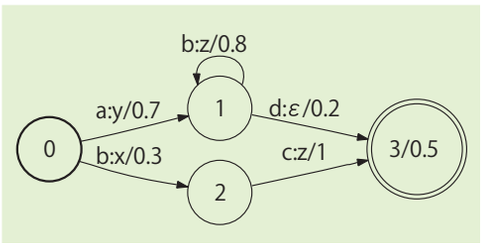

図 -2　重み付き有限状態トランスデューサ（WFST）

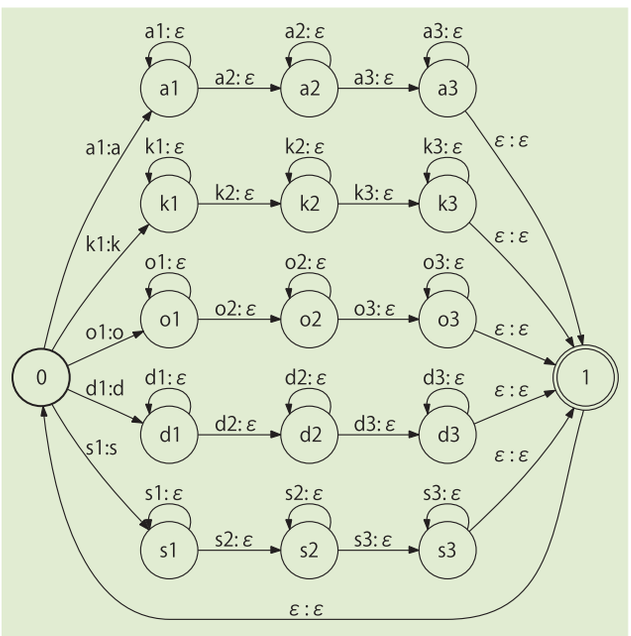

図 -3　音響モデル（HMM）を表す WFST

（3）WFSTには最適化演算が存在し，事前に最適な探索ネットワークに変換しておくことで，常に高速な音声認識が可能となる．

以降，WFSTによる音声認識とその効果について述べる．

WFSTとは

WFSTは，状態遷移機械のモデルとして広く知られる有限オートマトン（Finite Automaton）[1]の一種である．有限オートマトンは，アルゴリズムやモデルを状態遷移機械として表現し制御するための理論であり，これまで論理回路，暗号技術，データ圧縮，言語処理など，多くの分野に応用されてきた[2]．

有限オートマトンの最も基本的なモデルは，有限状態アクセプタ（Finite-State Acceptor : FSA）と呼ばれ，ある特定の記号列を受理するか否かを表す．FSAは状態と状態遷移の有限の集合によって記述され，各状態遷移に受理できる記号を持つ．入力記号列が与えられたとき，それを受理するような初期状態から終了状態に至る状態遷移過程が存在するときその記号列は受理される．

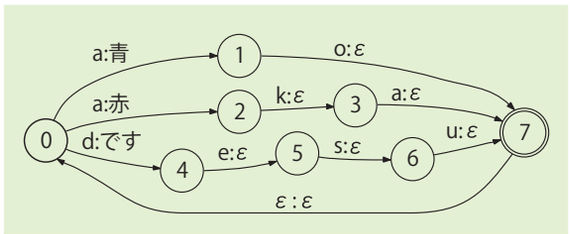

図 -4　単語発音辞書を表す WFST

FSAの拡張として，状態遷移において記号を受理すると同時に別の記号を出力する，有限状態トランスデューサ（Finite-State Transducer: FST）がある．FSTは記号列を受理し，それと同時に別の記号列を出力する記号列変換モデルである．さらに，状態遷移に対して重みを付与することで，コストや確率といった概念の導入を可能にしたモデルが重み付き有限状態トランスデューサ（WFST）である．

WFSTの一例を図-2に示す．ノードは状態を表し，ノード内には状態番号（または状態の名前）が記される．初期状態は状態0とする．また，二重線で囲まれた状態は終了状態を表す．図では状態3が終了状態であり，終了重み0.5を持つ．一方，アークは状態遷移を表し，入力記号，出力記号，重みが，“入力記号:出力記号／重み”のように記される．ε記号は，入力記号の場合は入力なしで状態遷移でき，出力記号の場合は何も出力しないことを表す．図-2のWFSTは，たとえば，“abbd”という記号列を“yzz”という記号列に変換する．

WFSTによるモデル表現

音声認識で用いられるモデルの多くはWFSTで記述することができる．たとえば，音響モデルのHMMは図-3のように表すことができる．入力記号はHMMの出力確率分布を表す記号（たとえば音素“a”の第1状態の分布“a1”），出力記号はそのHMMが表す音声単位（たとえば 音素“a”）である．WFSTの重みはHMMの状態遷移確率となる．そして，各音素のHMMを並列に接続し，状態1から0への遷移を加えることで，出力確率分布の記号列から音素列へと変換するWFSTとなる．HMMの出力確率は入力音声に対してオンラインで計算されるのでWFSTの記述からは除外されているが，認識処理の過程では重みとして累積される．

単語発音辞書は，音素列から単語列へと変換するWFSTとして，図-4のように表すことができる．このWFSTは，たとえば音素列“aoaka”をその発音に対応す

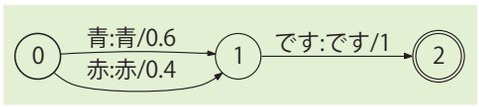

図 -5　言語モデル（有限状態文法）を表す WFST

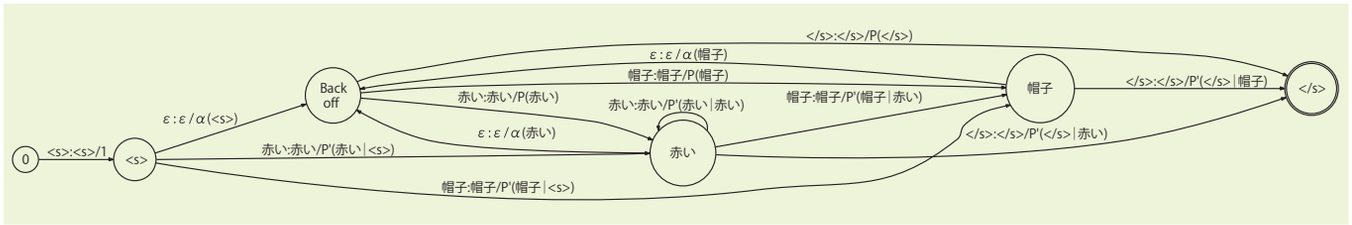

図 -6　*N*-gram（バックオフ・バイグラム）を表す WFST

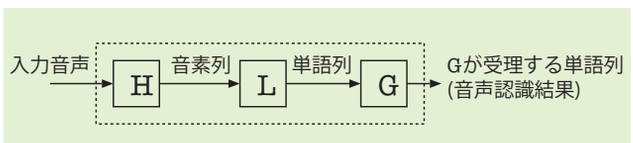

図 -7　音声認識の変換プロセス

る単語列“青赤”に変換する．

言語モデルは，図-1 で示したような有限状態文法であれば，**図-5** のように入力記号と出力記号が同一の WFST に書き換えることができる．

また，大語彙連続音声認識では言語モデルとして N-gram が用いられることが多い．N-gram は単語の連接のしやすさを $N-1$ 重のマルコフ過程で表すモデルであり，登録単語数を $|V|$ とするとき $|V|^N$ 個の確率値を持つ．したがって，WFST で表せば $|V|^N$ 個の状態遷移が必要となる．そのような WFST は非現実的であるため，WFST による音声認識では N-gram のバックオフを状態遷移の中に埋め込むテクニックを用いる．バックオフとは，N-gram 確率が学習データの不足等により精度よく推定できない場合，より精度よく推定された (N-1)-gram の確率を用いる方法である．たとえば，N=2 のバイグラムにおいて，バイグラム確率 $P(w|v)$ を推定するための単語列 vw の出現頻度 $C(vw)$ がある定数 c に満たないとき，$P(w|v)$ の代わりに (N-1)-gram に対するバックオフ係数 $\alpha(v)$ とユニグラム確率 $P(w)$ の積を用いる．

$$P(w|v)=\begin{cases} P'(w|v) & \text{if } C(vw)>c \\ \alpha(v)P(v) & \text{otherwise} \end{cases} \qquad (3)$$

ただし，$P'(w|v)$ は，バックオフを考慮して $P(\cdot|v)$ の総和が 1 になるように補正された値である．

これにより，$C(vw)>c$ であるようなバイグラムの状態遷移だけを作り，それ以外はユニグラムへと遷移させるため，状態遷移数は大幅に削減される．**図-6** は，バックオフ・バイグラムを WFST で表した一例である．“赤い”の状態から“帽子”の状態への遷移においてバイグラム確率 P'(帽子|赤い) が重みとして割り当てられている．一方，P'(赤い|帽子) に対応する“帽子”→“赤い”の状態遷移は存在しないが，“Back off”の状態を経由することで，α(帽子)P(赤い) の重みが適用される．

WFST によるモデルの統合

HMM，単語発音辞書，言語モデルは，音声から音素列へ変換する WFST，音素列から単語列へ変換する WFST，言語モデルに従う単語列を受理し出力する WFST で表せることを示した．これら WFST を H，L，G で表すと，音声認識は**図-7** のような H，L，G による逐次的な変換過程で表せる．

これらの変換に基づく音声認識は，式 (1) を O と W の独立性を仮定して書き換えた次の式で表せる．

$$\begin{aligned} \hat{W} &\simeq \underset{W}{\operatorname{argmax}} \sum_{\Psi} P(O|\Psi)P(\Psi|W)P(W) \\ &\simeq \underset{W}{\operatorname{argmax}} \{ \max_{\Psi} P_H(O_{\rightarrow}\Psi) \\ &\qquad P_L(\Psi_{\rightarrow}W)P_G(W_{\rightarrow}W) \} \end{aligned} \qquad (4)$$

ここで，$P_T(X_{\rightarrow}Y)$ は WFST T によって記号列 X を Y に変換するときの累積重み（確率），Ψ は任意の音素列を表す．この式は，入力音声 O から変換されるあらゆる音素列 Ψ，および，Ψ から変換されるあらゆる単語列 W を考慮して，全体の累積重みが最大となる単語列 $\hat{W}$ を求めることを意味する．

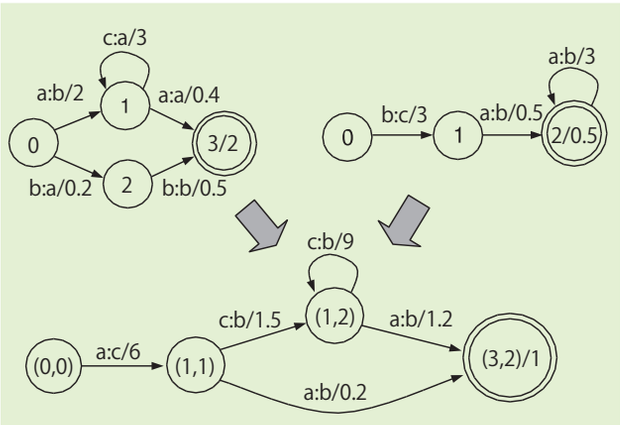

図 -8　WFST の合成（composition）

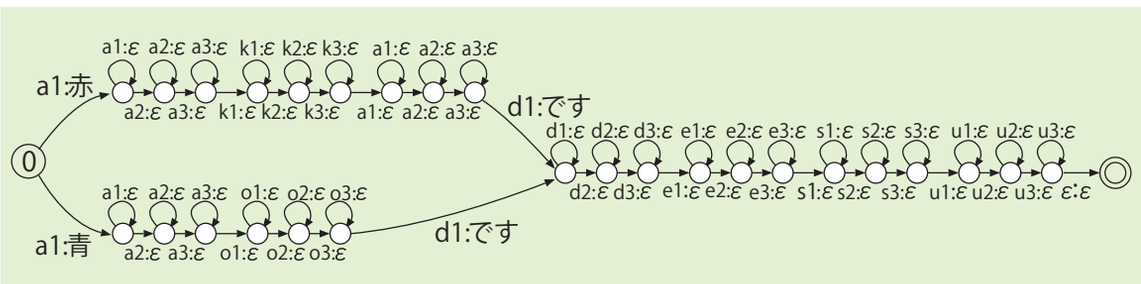

図 -9　WFST による探索空間表現

実際には，複数の WFST による変換をさまざまな入出力の可能性を考慮しながら，最終的な出力記号列を得るのは簡単なことではない．ところが，WFST にはそのような複数の WFST による変換過程を事前に統合し，1 つの WFST に置き換える演算が存在する．それは WFST の合成（composition）である．

合成の演算は，2 つの WFST を用いて 2 段階で適用される記号列変換を 1 回で行えるような WFST を生成する．具体的には，前段の WFST の状態遷移過程における出力記号列を受理する後段の WFST の状態遷移過程を求め，対応する状態および状態遷移を合成する．2 つの WFST を合成する例を図 -8 に示す．図下段の WFST は，上段 2 つの WFST を合成した結果である．状態内の数字のペアは，該当する状態が 2 つの WFST のどの状態から合成されたかを表している．

合成の演算を “∘” で表すと，音声認識の WFST は，

$$H \circ L \circ G \qquad (5)$$

のように求めることができる．この WFST は，HMM の出力確率分布の系列を言語モデルの制約を満たす単語列に変換する WFST となる．たとえば，図 -3，図 -4，図 -5 の WFST を合成した結果は図 -9 となる．これは図 -1 と基本的に同じネットワークである．このように，WFST では探索ネットワークの構築が合成の演算により簡単に実現される．

また，近年は前後の音素に依存した音素単位（トライフォン）を持つトライフォン HMM が広く利用されている．トライフォン HMM を導入するには，図 -10 のような WFST を用意する．図では，音素 /a/，/i/ および無音 /sil/ のみを仮定し，先行音素が /a/，後続音素が /sil/ である音素 /i/ のトライフォンを /a−i+sil/ のように表している．WFST の入力はトライフォンの系列であり，出力は通常の音素列となる．トライフォン同士は前後の関係を満たすような接続のみが許されている．この WFST を C とすると，

$$H \circ C \circ L \circ G \qquad (6)$$

のように合成した WFST を用いることでトライフォン HMM を用いた音声認識が可能となる．ただし，H はトライフォン HMM の WFST であり，トライフォンの記号列を出力する．

WFST による効率的探索

次に，WFST の最適化演算による探索効率の改善の効果について述べる．音声認識では，大規模なモデルを用いるほど高い精度を期待できるが，その反面，探索空間は大きくなり，計算量は増加する．また，従来法では，多くの場合，探索ネットワークを事前に構築せず，必要

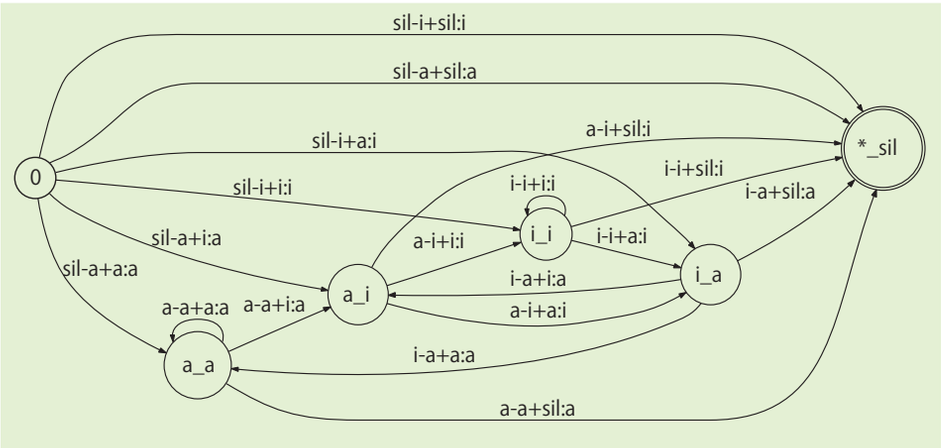

図 -10　トライフォン接続制約を表す WFST

	状態数	状態遷移数	実時間比
$H \circ C \circ L \circ G$	11,643,484	21,050,957	13.23
$H \circ C \circ opt(L \circ G)$	6,572,980	15,872,140	2.74
$H \circ opt(C \circ opt(L \circ G))$	3,748,476	9,843,837	1.36
$opt(H \circ opt(C \circ opt(L \circ G)))$	2,412,972	8,163,944	0.57

$opt(\cdot) = min(det(\cdot))$　($det(\,)$: 決定化，$min(\,)$: 最小化)

表 -1　WFST の最適化の効果

な部分だけをオンラインで生成し，探索処理を行ってきた．これは，ネットワークの論理サイズが，汎用コンピュータの物理メモリのサイズを超えてしまうためである．オンラインでネットワークを構築すれば，認識時に大きなオーバーヘッドとなって計算量がさらに増加する．このような問題を解決するのが，WFSTの最適化演算である．

最適化には，主としてネットワークを探索に適した構造に変換する決定化（determinization）とネットワークのサイズを小さくする最小化（minimization）がある．決定化とは，非決定性WFSTを決定性WFSTに変換する演算である．決定性WFSTとは，入力記号を受理するときに遷移先の状態がただ1つに定まるようなWFSTであり，それ以外は非決定性WFSTという．決定性WFSTは，次に遷移する状態が常に1つに決まるので，非常に効率よく探索を行える．最小化は，WFSTの状態数を最小にする演算であり，ネットワークの冗長性を削減する効果を持つ．これらの最適化演算により，高速に探索が行えるだけでなく，圧縮されたネットワークをメモリにすべて展開できるようになり，オーバーヘッドの問題をも回避できる．

表 -1 は，最適化演算を適用した場合のWFSTの状態数と状態遷移数，および音声認識のスピード（実時間比☆1）を表す．この結果は，日本語話し言葉コーパスの男性10講演の音声認識実験に基づいている．使用した計算機はIBM互換機，CPUはPentium4，クロック周波数2.8GHzである．表より，最適化が進むほどWFSTのサイズおよびスピードが改善されることが分かる．

ネットワークを小さくする工夫は，従来の音声認識でも検討されてきた．WFSTの最適化は，そのような工夫の多くを包含している．たとえば，単語発音辞書において接頭辞を共有させる木構造化があるが，これは決定化とほぼ同じ操作である．WFSTでは，さらに最小化やネットワーク全体に渡る最適化を行えるため，従来法を上回る最適化が可能となる．

■今後の展望と課題

WFSTでは，複数のWFSTによる記号列変換が合成という演算により1つのWFSTで実現されることをすでに示した．これは，複数の逐次的な変換処理の各々をWFSTで記述できれば，それらを合成して1つのWFSTによる変換処理として実行できることを意味する．たとえば，音声認識のWFSTに，その認識結果を処理する言

☆1 入力音声の長さ（時間）を認識処理にかかった時間で割った値．音声認識は一般に発話と並行して処理を進めるので，実時間比1のときは，発話終了直後に認識結果が得られることを意味する．

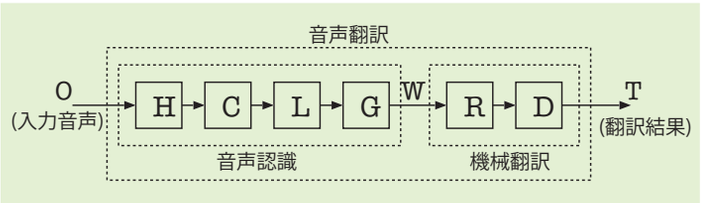

図-11　音声翻訳の変換プロセス

語処理のWFSTを合成すれば，音声認識と言語処理を同時に実行することができる．このときデコーダのプログラムは変更する必要がない．

この特徴を利用して，話し言葉の音声認識と同時に，言い直しやフィラー等の言語的揺らぎを正規化して書き言葉に翻訳する文整形処理，並びに音声認識誤りを排除しつつ重要単語を抽出する要約処理をWFSTに統合した音声要約法が提案され，その効果が示されている[4]．話し言葉の音声を認識して書き言葉に変換する過程は，音声翻訳のプロセスと見なせる．図-11に示すように音声認識の後に単語列を置換するWFST Rと翻訳先の言語モデルのWFST Dを追加し，これらを1つのWFSTに合成することで，音声翻訳のWFSTが構成される．このアプローチでは，音声認識の終了を待つことなく言語処理を実行できるため，発話終了とほぼ同時に要約結果が得られる．また，すべてのモデルによって総合的に最良の要約結果が得られることから音声認識の誤りも削減されるという利点がある．

ただし，合成の演算は2つのWFSTの間の入出力のあらゆる組合せを状態遷移で表現するため，非常に大きなWFSTが生成される可能性がある．たとえ最適化を行ったとしても十分に小さくなる保証はない．このような問題には，WFSTのオン・ザ・フライ合成（on-the-fly composition）が有効である．オン・ザ・フライ合成は，記号列を変換する過程で必要な部分だけを合成する方法であり，メモリ使用量を大幅に抑えることができる．実行時にはオン・ザ・フライ合成によるオーバーヘッドを伴うが，そのオーバーヘッドを最小限に抑え，効率的に探索する方法が提案されている[5]．この手法により従来の限界をはるかに超える語彙（180万単語）を扱うリアルタイム音声認識が実現されている．WFSTにおけるオン・ザ・フライ合成は，大規模な統合システムを動作させる上での現実的な手法として重要である．

今後，計算機パワーの向上はさらに進むことが予想される．しかし，大規模な音声認識を携帯電話やPDA等のより小さな計算機で動作させたい，1台の音声認識サーバでより多くのコールを同時に処理したい，といったニーズは今後もなくなることはない．このようなニーズに対し，WFSTは常に最適な音声認識を実現する有力なツールとして使われていくであろう．そして，WFSTは音声認識の枠を超えた新たな統合的アプリケーションへ応用されていくことが予想される．

ただし，WFSTは有限状態モデルであるため，表現できるモデルには制限がある．言語処理でよく用いられる再帰的な置換手続きはWFSTにとってむしろ不向きである．しかしながら，WFSTよりも強力なモデルが必要と考えられてきた分野であってもWFSTが適用できる可能性はある．たとえば，異言語間の機械翻訳が本稿で述べたアプローチで実現されつつある[6]．このようにWFSTの適用範囲を広げる努力を続けることで，将来，音声翻訳のような高度な音声言語処理がWFSTで実現される日も近いと考えられる．

参考文献

1）J.ホップクロフト，J.ウルマン（著），野崎昭博，高橋正子，町田　元，山崎秀記（訳）：オートマトン 言語理論 計算論 I，サイエンス社（1991）．
2）Roche, E. and Schabes, Y.: Finite-state Language Processing, MIT Press (1997).
3）Mohri, M., Pereira, F. and Riley, M.: Weighted Finite-state Transducers in Speech Recognition, Computer Speech and Language, Vol.16, No.1, pp.69-88 (2002).
4）Hori, T., Hori, C. and Minami, Y.: Speech Summarization using Weighted Finite-state Transducers, Proc. of Eurospeech2003, pp.2817-2820 (2003).
5）Hori, T., Hori, C. and Minami, Y.: Fast on-the-fly Composition for Weighted Finite-state Transducers in 1.8 Million-word Vocabulary Continuous Speech Recognition, Proc. of ICSLP2004 (To appear).
6）Tsukada, H. and Nagata, M.: Efficient Decoding for Statistical Machine Translation with a Fully Expanded WFST Model, Proc. of EMNLP2004, pp.427-433 (2004).

（平成16年7月13日受付）