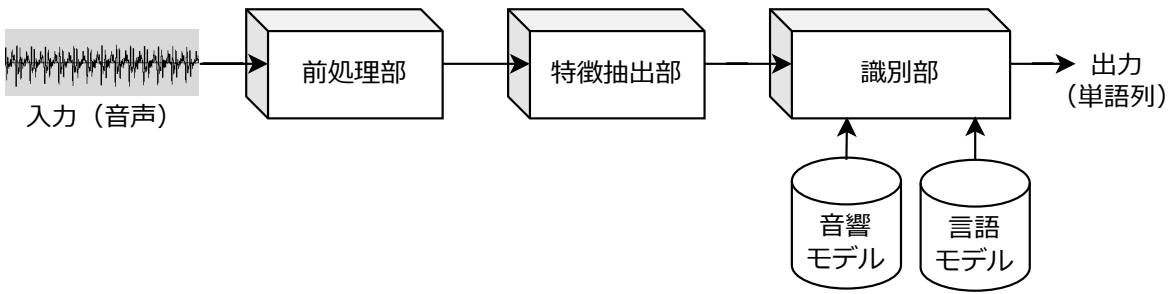
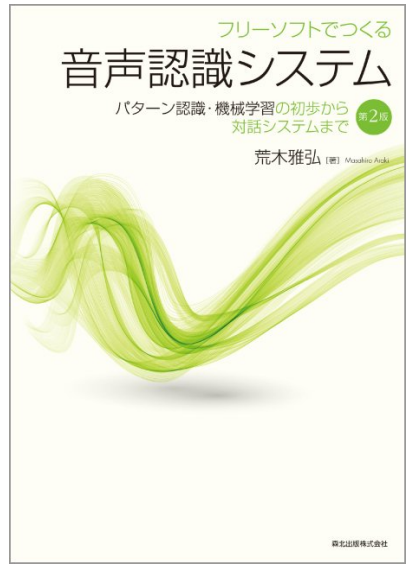
10-15. 実践編 ー 音声認識・音声対話 ー



- 10 声をモデル化してみよう
- 11 HTKを使って単語を認識してみよう
- 12 文法規則を書いてみよう
- 13 統計的言語モデルを作ろう
- 14 連則音声認識に挑戦しよう
- 15 会話のできるコンピュータを目指して



- 荒木雅弘:『フリーソフトでつくる音声認識システム(第2版)』(森北出版, 2017年)
- スライドとJupyter notebook
- サポートページ

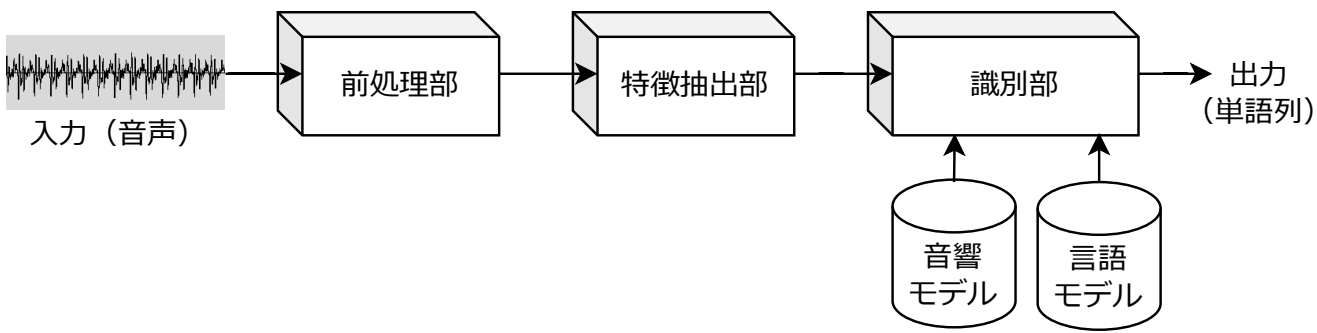
10. 声をモデル化してみよう

10.1 連続音声の認識

- 入力系列 $\boldsymbol{x}$ のもとで事後確率を最大にする単語列 $\hat{\boldsymbol{w}}$ を認識結果とする

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} P(\boldsymbol{w}|\boldsymbol{x}) = \arg\max_{\boldsymbol{w}} p(\boldsymbol{x}|\boldsymbol{w})P(\boldsymbol{w})$$

$p(\boldsymbol{x}|\boldsymbol{w})$：音響モデル、$P(\boldsymbol{w})$：言語モデル

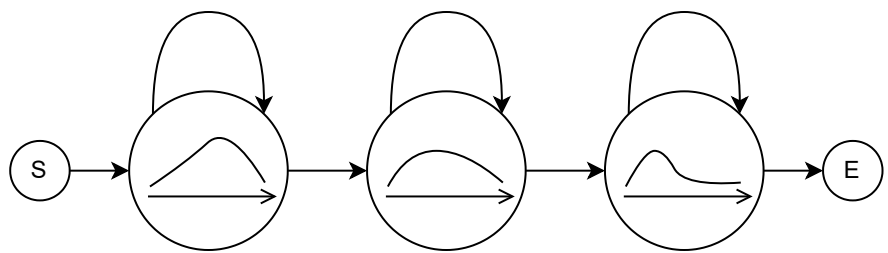


10.1 連続音声の認識

- $p(\boldsymbol{x}|\boldsymbol{w})$: 音響モデル
  - 音素 m を発声したときに特徴ベクトル系列 $\boldsymbol{x}$ が観測される確率を $p(\boldsymbol{x}|m)$ とする
  - 音素系列 $m_1, \ldots, m_i$ と単語 w との対応は単語辞書を用いる
  - 最大の確率を与える単語系列 $\boldsymbol{w}$ は**ビタビアルゴリズム**によって求めることができる
- $p(\boldsymbol{w})$: 言語モデル
  - ドメインやタスクが狭く限定されたものであれば、文法規則を用いることができる
  - 特に入力文が限定されなければ、大規模なコーパスから**統計的言語モデル**を作成する
- 探索
  - すべての可能な単語列に対して音響モデルと言語モデルのスコアを求めることは現実的ではないので、**探索**によって最適な単語列を求める

10.2 音響モデルの作り方

- $p(\boldsymbol{x}|m)$: 音素毎の音響モデル
  - 音素の前半・中盤・後半で特徴ベクトルがかなり異なる
    - 母音の過渡現象(例:/a/-/o/-/i/ の /o/ の音の *入り-定常-出*)
    - 子音の構成(例:/n/ の *先行母音からの入り-鼻音-後続母音への出*)
  - 話者や話速の違いで、1音素当たりの特徴ベクトル数がかなり異なる
  - 自己遷移を持ち、各状態で混合分布によって特徴ベクトルを出力するleft-to-right型オートマトンで各音素をモデル化
    - どの特徴ベクトルが、どの状態から出力されたかが隠れているので隠れマルコフモデル (HMM; Hidden Markov Model)とよぶ

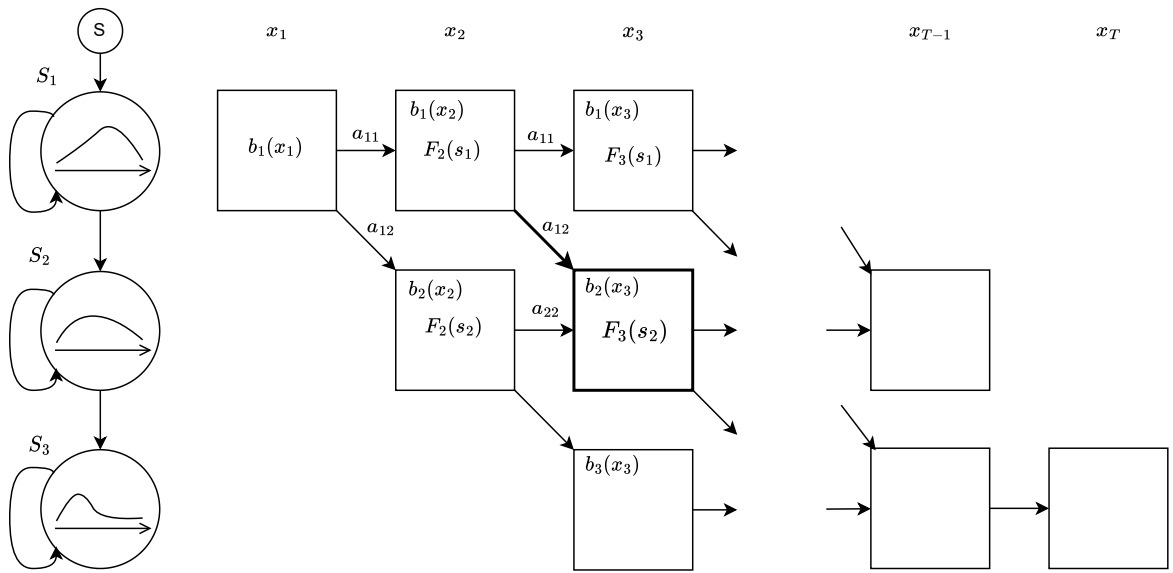


10.3 音響モデルの使い方 (1/3)

- $p(\boldsymbol{x}|m)$ の計算
  - HMMの状態数をN、特徴ベクトルの系列長をTとすると、すべての可能な系列に対する確率の計算量は $O(N^T)$
  - 状態のマルコフ性を仮定し、**動的計画法**を用いると、計算量は $O(N^2T)$
  - 正確な確率値は排反事象の和となるが、それを最も高い確率値となる系列を求める問題に置き換える(→ ビタビアルゴリズム)
  - このことによって、音素HMMを並列に連結したHMMにおいて、最尤の音素系列を高速に求めることができる

10.3 音響モデルの使い方（2/3）

- 動的計画法の計算に用いる**状態空間表現**（＝トレリス）
  - ビタビアルゴリズム：ある時刻のある状態でのスコアを求める際に、最大値を与える経路の情報だけを利用する



$$F_3(s_2) = \max\{F_2(s_1) + a_{12}, F_2(s_2) + a_{22}\} + b_2(x_3)$$

10.3 音響モデルの使い方（3/3）

- トレリス上のビタビアルゴリズムによる確率計算と最尤経路の決定
  - for t = 1 to N:
    - 時刻 t-1 から遷移可能なすべての状態について
      - 前状態のスコアと遷移確率の和が最大のものに対して、その状態での特徴ベクトルの出現確率を足し、その状態のスコアとする
        - 確率値は対数を取り、かけ算を足し算に置き換えておく
      - 最大のスコアを与えたパスを記録
  - 最終状態から最大のスコアを与えたパスを逆向きにたどったものが**最尤経路**
- 動的計画法のポイント
  - 問題全体を部分問題に分割
  - 一度行なった部分問題の計算結果を保存することによる計算の効率化

10.4 音響モデルの鍛え方

- Baum-Welch アルゴリズム
  - HMMの状態遷移確率や出力確率をEMアルゴリズムで求める

1. HMMのパラメータ(状態遷移確率と出力確率)をランダムで初期化
2. Eステップ

   現在のパラメータを使用して、各状態における観測データの期待値を計算
3. Mステップ

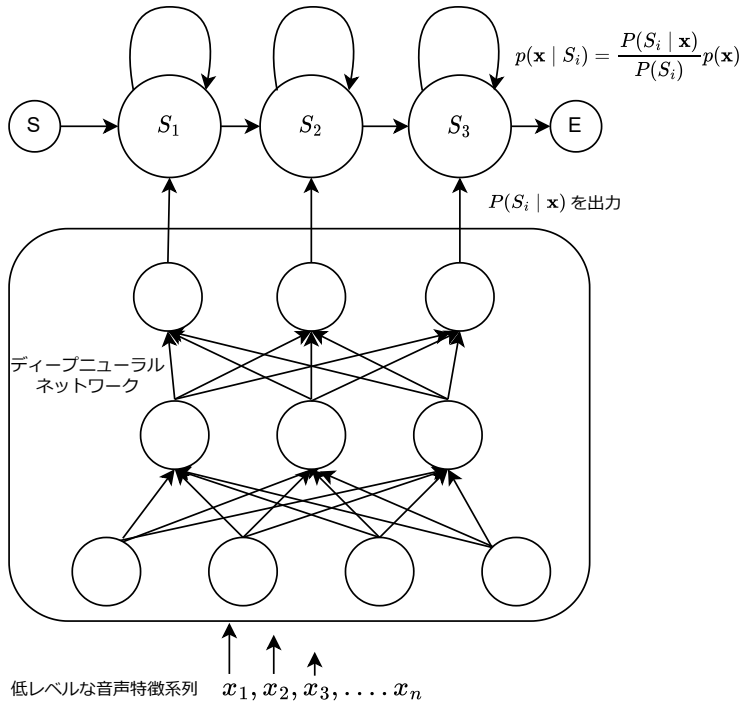
   Eステップで計算された期待値を使用して、HMMのパラメータを最尤推定
4. 収束判定

   パラメータの変化が十分小さくなった、または最初に設定した繰り返し回数に達した場合、

   アルゴリズムを終了
5. ステップ2から4を繰り返す

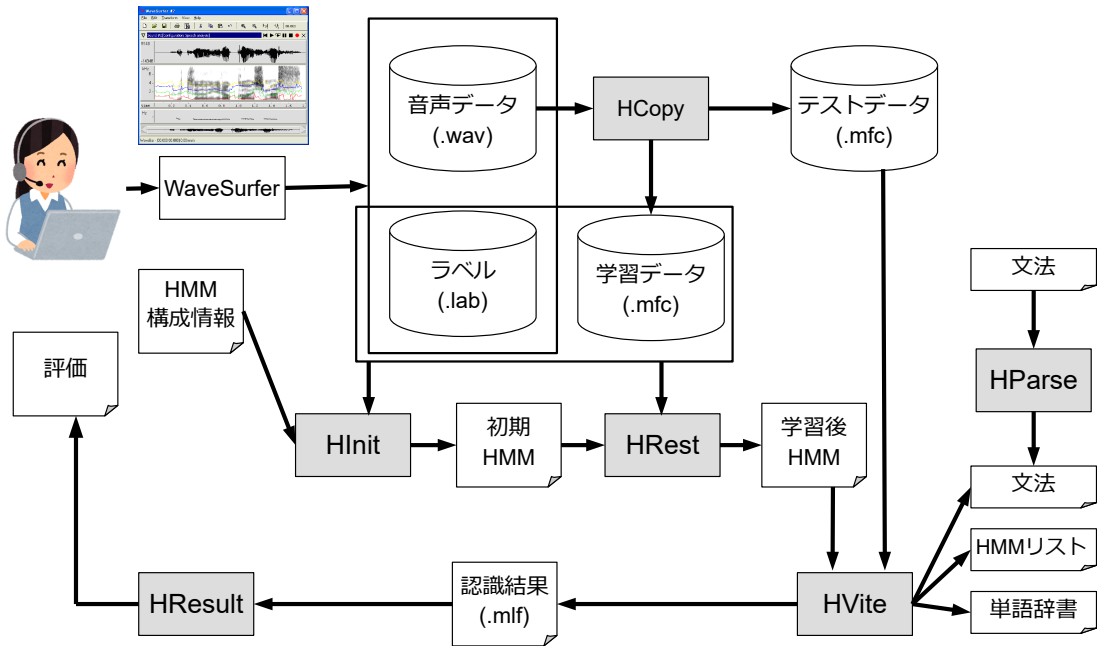
10.5 実際の音響モデル

- ディープニューラルネットワークを用いた音響モデル（DNN-HMM）
  - 混合分布の代わりに、DNNの出力に基づいて $p(\boldsymbol{x}|S_i)$ を計算



11. HTKを使って単語を認識してみよう

- HTK：音声認識の研究･開発のためのツールキット
  - 注) HTK の最終リリースは2016年



12. 文法規則を書いてみよう

- 言語モデルとしての文法
  - 単語から文を構成する規則を文法として記述
  - 大規模語彙の音声認識精度向上に伴い、現在では文法による方法はあまり使われない
- 文法の書き方
  - 記号の集合
    - 非終端記号は文・句・単語集合、終端記号は単語
  - 規則の書き方
    - 左辺は 非終端記号 1つ、右辺は 非終端記号 または 終端記号 の列
  - 文法の種類
    - 文脈自由文法：右辺に任意の記号列（大半の場合、処理効率のよい正規文法に変換可能）
    - 正規文法：右辺が終端記号 + 非終端記号、または終端記号

13. 統計的言語モデルを作ろう (1/2)

- $P(\boldsymbol{w})$: 言語モデル
  - 単語列 $\boldsymbol{w}$ の生成確率

$$P(\boldsymbol{w}) = P(w_1, \ldots, w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \ldots P(w_n|w_1, \ldots, w_{n-1})$$

  - N-gram近似：単語の生起確率は直前の $N-1$ 単語にのみ依存すると仮定
    - $N = 3$ のとき

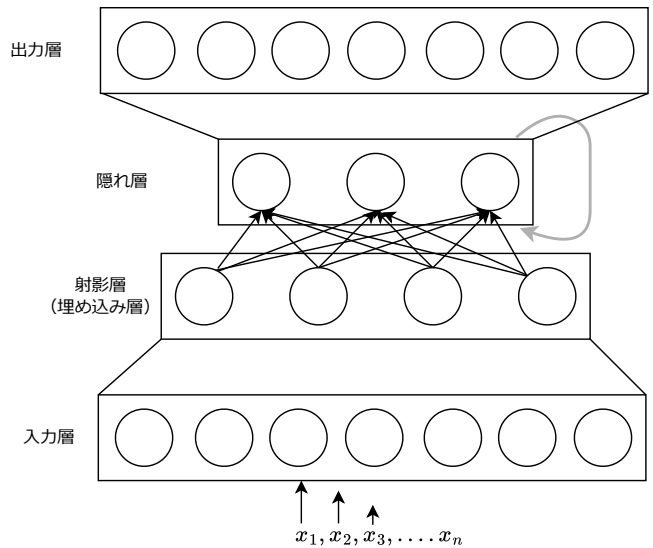
$$P(w_1, \ldots, w_n) = P(w_1)P(w_2|w_1)\prod_{i=3}^{n} P(w_i|w_{i-2}, w_{i-1})$$

    - N-gram確率はコーパス(文例集)での出現頻度から推定

$$P(w_i|w_{i-2}, w_{i-1}) = \frac{Count(w_{i-2}, w_{i-1}, w_i)}{Count(w_{i-2}, w_{i-1})}$$

13. 統計的言語モデルを作ろう（2/2）

- リカレントニューラルネットワークによる言語モデル
  - 入力層：語彙数 $|V|$ 次元の one-hot ベクトル
  - 出力層：語彙数 $|V|$ 次元ベクトルで、各次元が対応する単語の生起確率を表す
  - 学習：コーパス中の次単語を正解として学習



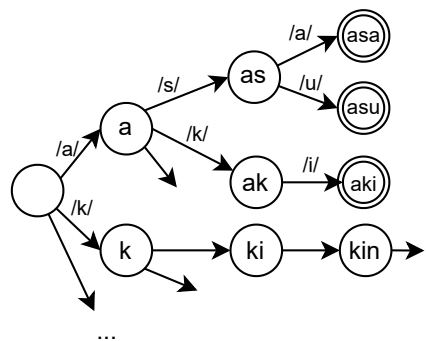
14. 連続音声認識に挑戦しよう

14.1 基本的な探索手法 (1/5)

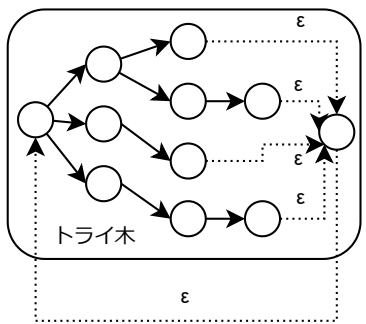
- 単語ラティスの作成
  1. 単語と音素列の対応を記した単語辞書からプレフィックスを共有するトライ木を作成
  2. トライ木のエッジを HMM に置き換えた単語連接 HMM を作成
     - 各単語の最終状態から全単語の初期状態へ、入力なしで遷移可能な ϵ-遷移を加える
  3. 単語連接 HMM 上のトレリス探索で上位 k 番目までの単語系列から単語ラティスを作成

単語	音素列
朝	a s a
明日	a s u
秋	a k i
北	k i t a
昨日	k i n o o
…	…

単語辞書



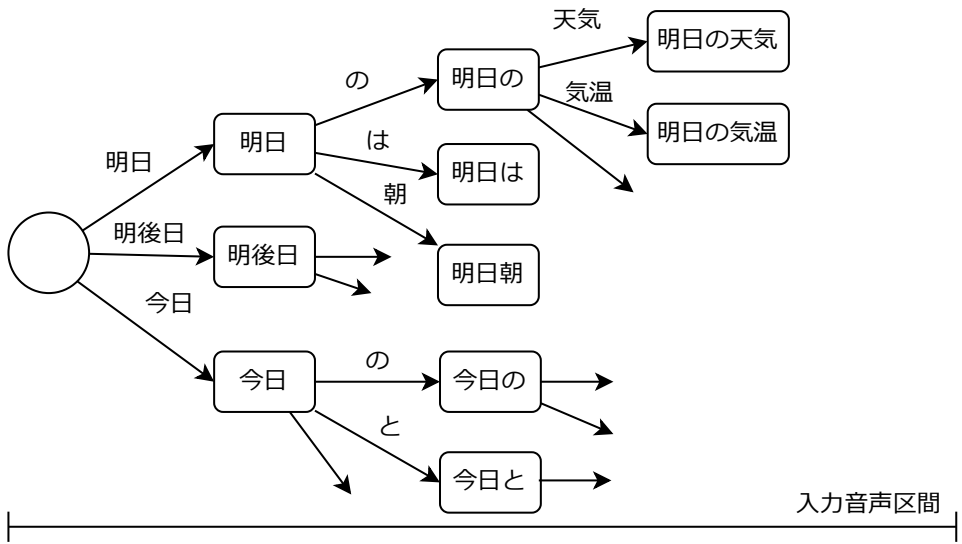
トライ木



単語連接HMM

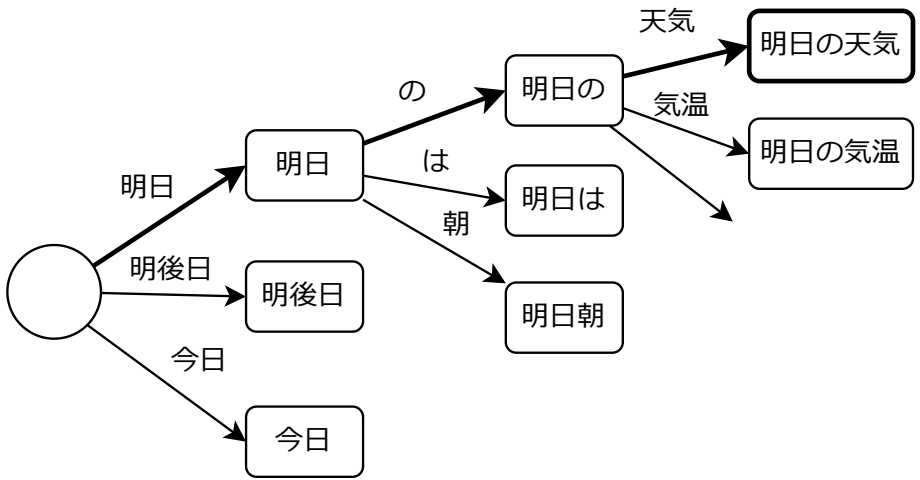
14.1 基本的な探索手法（2/5）

- 単語ラティス：連続音声認識における**状態空間表現**
- 探索の目標
  - 状態空間中で入力音声区間全体をカバーするスコア最大の単語系列を求める



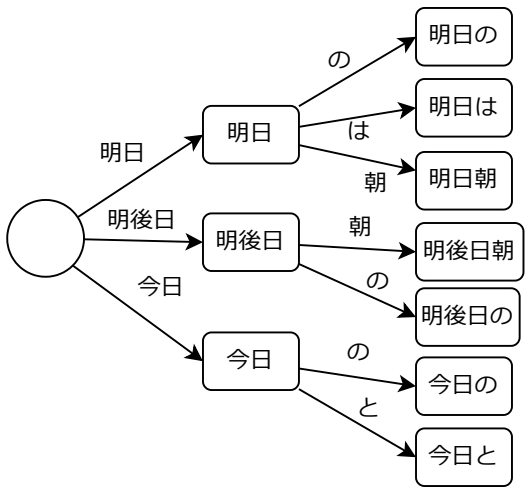
14.1 基本的な探索手法（3/5）

- 縦型探索（深さ優先探索）
  - ノードの展開時に最もスコアの高い枝を選んで進む
  - 目標状態でない、もしくは目標状態に到達できないことがわかったら、直近の分岐に戻り次の候補を探す



14.1 基本的な探索手法（4/5）

- 横型探索（幅優先探索）
  - ノードの展開時にすべての枝について解かどうかを調べる
  - 目標状態が含まれないことがわかったら、一段だけ枝を伸ばす



14.1 基本的な探索手法（5/5）

- 縦型探索のアルゴリズム

```
1. 初期状態をオープンリストに入れる。クローズドリストを空に初期化する。
2. while オープンリスト ≠ []:
     オープンリストから先頭要素 s を取り出し、クローズドリストに s を追加
     if s = 目標状態:
       s を解として探索終了
     else:
       s を展開し接続先のノードをオープンリストの先頭に追加（スタック構造）
```

- 横型探索のアルゴリズム
  - 上記アルゴリズム最終行の操作を「末尾に追加（キュー構造）」に変更
- いずれのアルゴリズムでも見つかった解が最適解であることは保証されない

14.2 ヒューリスティック探索 (1/3)

- ヒューリスティック探索手法
  - これまでの経路のスコアやこれからのスコアの見積を活用して効率的に準最適解を探索

関数	説明
$g(s)$	初期状態からノード s までの経路のスコア
$h(s)$	ノード s から目標状態までの経路のスコア
$f(s)$	ノード s を経由したときの最適経路のスコア $f(s) = g(s) + h(s)$

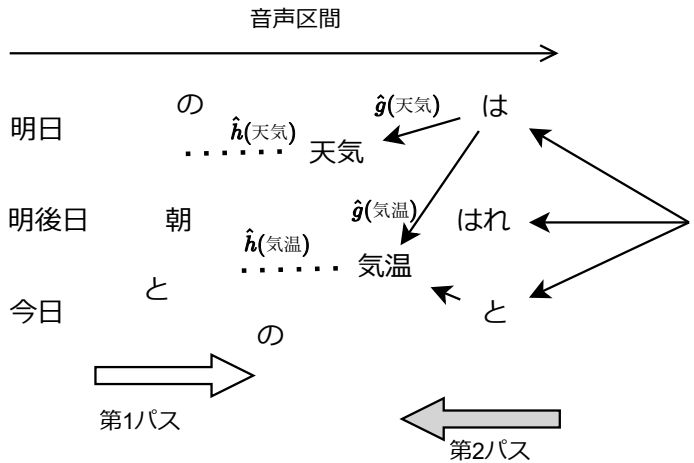
- スコアを活用した探索
  - 見積値(関数記号にハット(^)記号を付けたもの)の高速な計算を導入する
  - 探索仮説をスコア上位のものに限定するビームサーチを用いる

14.2 ヒューリスティック探索 (2/3)

- さまざまなヒューリスティック探索
  - 最適探索
    - 縦型探索の方法でオープンリストにノードを追加した後、$\hat{g}$ の値の順にソートする
  - 最良優先探索
    - 最適探索のアルゴリズムに対して用いる関数を $\hat{h}$ とする
  - A^* 探索
    - 最適探索のアルゴリズムに対して用いる関数を $\hat{f}$ とする
    - オープンリストとクローズドリストについて、同一の状態 s に対してリスト内の $\hat{f}(s)$ よりもスコアの高いものが探索中に出現したら、その値と置き換える
    - A^* 探索の特徴
      - 常に $\hat{h}(s) > h(s)$ (すなわち楽観的な予測)なら、最適解が見つかることが保証されている
      - 音声認識での探索においては、$\hat{h}$ をどうやって見積るかが問題

14.2 ヒューリスティック探索 (3/3)

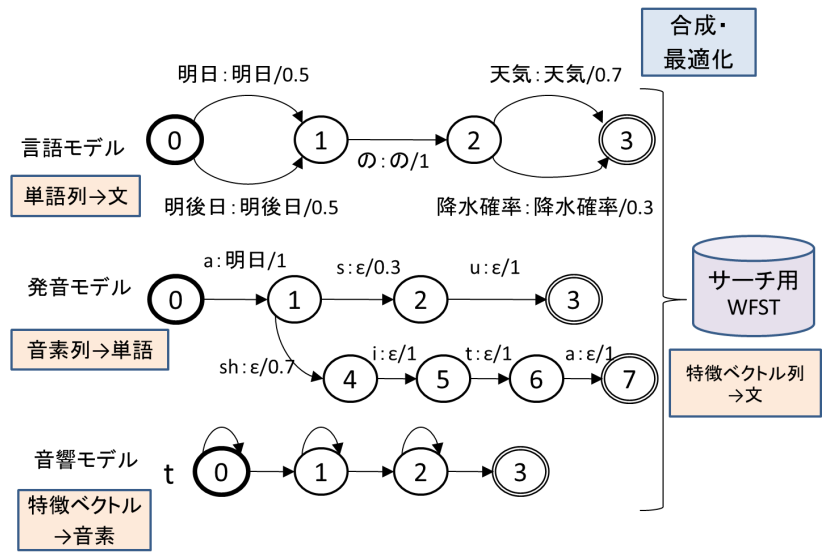
- 音声認識エンジン Julius での解法：2パス探索
  - 1パス目：単純な音響モデルと2-gram言語モデルで高速なトレリス計算
  - 2パス目：入力と逆方向に詳細なモデルで A* 探索



  - 注) Julius の最終リリースは 2020年

14.3 WFSTによる探索手法

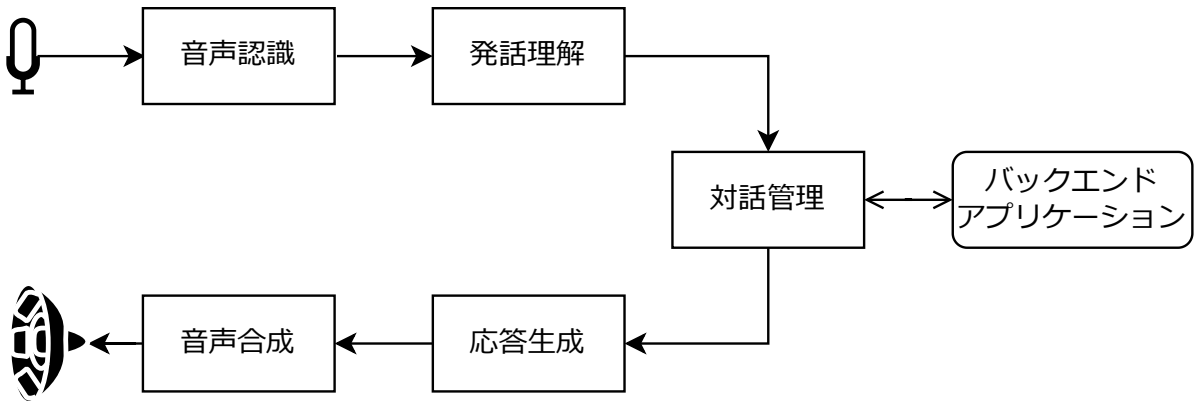
- WFST（weighted finite state transducer）による探索
  - 音響モデル・単語辞書・言語モデルを WFST に変換し、それらを合成・最適化した上で探索
  - 特徴ベクトル系列を入力とし、文としてもっともらしい単語列が出力される



15. 会話のできるコンピュータを目指して

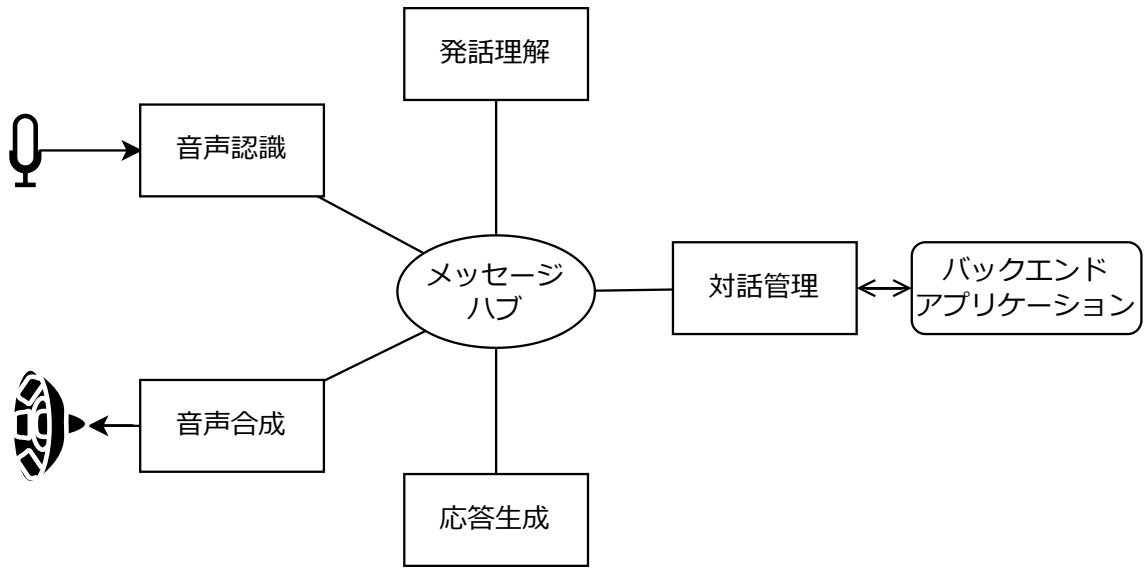
15.1 音声対話システムの構成 (1/2)

- 逐次的処理
  - ひとまとまりの音声入出力を仮定
  - モジュールの逐次結合で構成



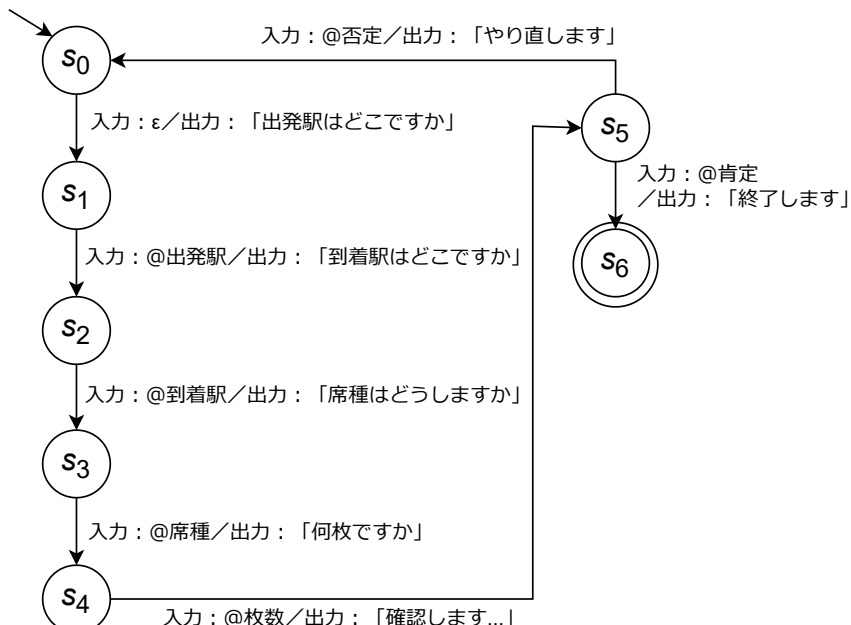
15.1 音声対話システムの構成（2/2）

- 漸進的処理
  - 相槌など短い音声入出力にも対応可能
  - 非同期メッセージの交換で動作



15.2 対話管理の方法

- 有限状態トランスデューサ(FST)による対話のモデル化
  - 入力:ユーザ発話 or イベント
  - 出力:システム発話／行為



15.3 音声対話エージェント

- MMDAgent-EX
  - 音声認識、音声合成、対話管理などを統合した音声対話エージェント
  - 対話シナリオは FST で記述
  - 入力：イベント
    - 音声認識結果
    - 合成音声出力終了
    - タイマー
  - 出力：コマンド
    - 合成音声、動作、画像の出力、webページの表示
  - 注）MMDAgent-EX の最終リリースは 2021年

まとめ

- 音声認識の定式化
  - 音響モデル：隠れマルコフモデルで構成
  - 言語モデル：文法または統計的言語モデルで構成
  - 探索：ビームサーチ、A^* 探索、WFST による探索
- 音声対話システム
  - 逐次的処理と漸進的処理の2種類の構成
  - 簡単なタスク指向対話のモデル化にはFSTが用いられる