

6 音声の認識と合成

6.1 音声認識

6.1.1 単語認識

音声認識は、コンピュータに入力された音声を文字へと変換する技術です。このような音声認識は、大きく2つに分類することができます。1つは、コンピュータにあらかじめ登録された単語やコマンドを認識するもので、これは、単語認識やコマンド認識と呼ばれます。もう1つは、コンピュータに対して、制限のない自由文を連続的に認識させるものです。これは、ディクテーションと呼ばれます。

単語認識を実現するには、まず、認識対象となる単語を認識するためのモデルとなる認識モデルをあらかじめデータベースに登録します。例えば、「東京」「千葉」「埼玉」などの音声特徴をデータベースに登録します。そして、入力された単語の音声特徴と、データベース中の単語の音声特徴との比較を行います。例えば、「千葉」と発声された場合、その音声特徴と最も類似度が高かったデータベース中の単語が選択され、認識結果となります。ただし、どの単語と比較しても類似度が十分に高くなかった場合には、認識すべき単語以外が発せられたと解釈します。これは、コマンド以外の言葉をコマンドと認識し、誤作動を起こさないために必要な処理です。また、コンピュータが処理を容易に行えるように、コマンドが発せられたのか、それ以外の音声が発せられたのかの区別を明示的に示す場合もあります。例えば、特定のキーを押しながら発した音声のみをコマンドとして認識するといったことが行われます。また、コマンドの前に、必ず、「コンピュータ」のように言って、それに続く音声のみをコマンドとして認識させるといったことも行われます。

認識の際に用いられる特徴量としては、例えば、音声波形をそのまま使う方法や、周波数分析により抽出した音素の情報などを利用する方法があります。ケプストラムと呼ばれる、周波数スペクトルをフーリエ逆変換して得られる特徴などもしばしば利用されます。ケプストラムの低ケプレンシー部分は、主に、周波数スペクトルの大まかなうねりの部分に対応するため、この部分を、単語認識を行う際の特徴量として利用することができます。

ここで、入力された音声の特徴パターンとデータベースに登録された音声の特徴パターンの比較について考えてみましょう。入力された音声の特徴パターンとデータベースに登録された音声の特徴パターンの比較を行う場合、単純に、そのまま2つのパターンを比較することはできません。例えば、「葵」という単語を発声する場合を考えてみましょう。「葵」という単語を発声する場合、ある人は、他の人よりも早口で話すかもしれません。また、ある人は、「お」の音を他の人に比べて長く発音するかも知れません。このため、2つのパターンの比較を行うには、時間軸方向で、部分的に伸び縮みさせながらパターンがどれほど類似しているかの比較を行う必要があります。このような2つのパターン間の照合を行うための手法としては、例えば、DPマッチングや隠れマルコフモデル（HMM）といった手法があります。

DPマッチングは、2つのパターンが与えられた場合、それらのパターンを非線形に伸縮し、それらのパターンの最適なマッチングを行う手法です。非線形とは、パターンをある部分は伸ばしたり、ある部分は縮めたりしてマッチングを行うということです。今、長さ*i*のパターンAと長さ*j*のパターンBを対応付けを考えます。横軸がパターンA、縦軸がパターンBとします。この時、例えば、パターンAとパターンBのはじめが一致しているとすると、両方の1番目と1番目が一致することになります。これは、座標が(1,1)の1つの点で表現することができます。また、パターンの最後が一致しているとすると、これは、パターンAの*i*番目とパターンBの*j*番目が対応していることになります。これは、座標(*i*,*j*)の1点で表現することができます。そして、パターンAとパターンBを部分的に伸縮した場合の対応付け結果は、この平面上の(1,1)と(*i*,*j*)を結ぶ経路として表現できます。対応付けを行う2つのパターン間であまり大きく離れた部分同士が対応しないようにする場合は、例えば、図6-1のようにパターン間の局所的な傾斜が1/2から2になる経路のみをとることを仮定します。パターン間の距離を表す $D(i,j)$ は、漸化式で表現できるため、この式を解いて $D(i,j)$ の最小値を求めることで、パターン間の最適な対応を求めることができます。

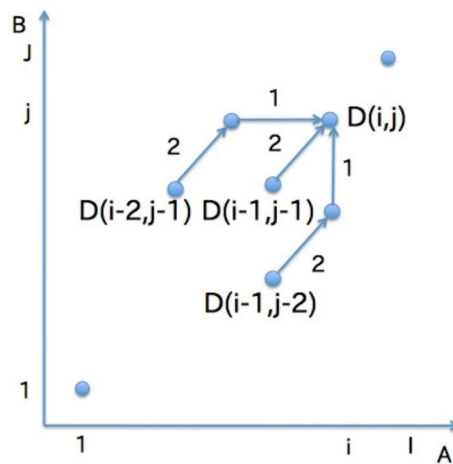


図6-1 DPマッチング

隠れマルコフモデル（HMM）は、パターンの非線形なマッチングを確率的なモデルによって行う方法です。例えば、図6-2のように、状態1、状態2、状態3の3つの状態があったとします。そして、初期状態の状態1から状態1に遷移する際にaが出力され、再び状態1に遷移してaが出力される...というように、ある状態からある状態に遷移する場合に、確率的にaやbといったシンボルを出力するとします。このようなモデルによって、入力されたパターンが生成される確率を求めることで、2つのパターンがどれほど類似しているかの比較を行うことができます。

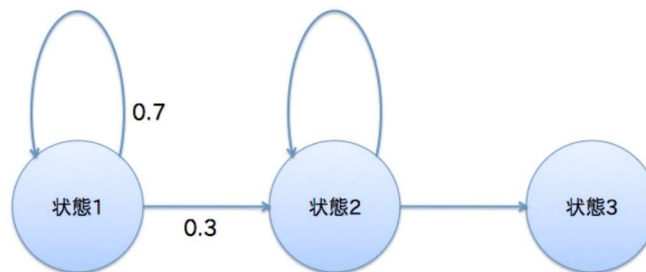


図6-2 隠れマルコフモデル

6.1.2 ディクテーション

ディクテーションは、自由文を連続的に認識させる処理です。ディクテーションは自由文を認識するため、その処理は単語認識に比べて難しくなります。このため、さまざまな経験則や統計的制約などにより認識結果の曖昧性を減らし、認識精度を向上させることが行われます。

認識の曖昧性を制限するためには、辞書が利用されます。例えば、都市名の「東京」の音声の認識結果として「とおきょう」のような認識結果が得られた場合に、そのまま「とおきょう」を認識結果とすれば認識は誤りということになります。辞書による制限は、あらかじめ認識用に用意しておいた辞書の中で最も類似している単語を探し、「とおきょう」と最も類似度の高い「とうきょう」を認識結果とするものです。辞書による語彙の制限を行う場合の問題点は、辞書中の単語数が多い場合や、類似した発音の単語が多い場合には、誤認識が起りやすくなることです。また、入力された単語のすべてが辞書に登録されているとは限らないため、辞書に登録されていない単語の認識が困難になる場合もあります。このため、辞書による語彙の制限を行う場合は、どのように辞書に登録されていない未知語を検出するかが問題となります。

また、認識結果の曖昧性の解消のため、文法的な制約により語彙の制限を行う方法もあります。1つの方法としては、2つの単語が接続するか否かにより語彙の制限を行う方法があります。この場合、例えば、「形容詞の後に名詞は続くが、助詞は接続しない」といった接続関係を語彙の制限として設定します。また、より詳細に、文全体の文法的な制約により語彙の制限を行う方法もあります。この方法では、処理が重くなるものの、文全体として、文法的に、より自然な認識結果を選択することができるという利点があります。

統計的な制約により語彙を制限する方法としては、N-gramと呼ばれる方法があります。N-gramは、(N-1)個の連続する単語の後に、どのような単語が続くかという条件付き確率を求めておくことで、認識結果となる語彙の制限を行います。例えば、I amの後の単語がaであるかareであるか区別できなかった場合を考えてみましょう。この時、統計的にaが出現する確率が高いものの、areは通常続かないことが分かっていたら、認識結果をaとすることができます。N-gramのうち、2つの連続する単語の関係を見るものはbigram、3つの連続する単語の関係を見るものはtrigramと呼ばれます。しかし、連続する単語の数を増やせば増やすほど、その処理は重いものになります。N-gramを使って語彙を制限する場合に必要な、それぞれの単語の続きやすさの確率は、多量の文書データから統計的に求めることができます。

また、ディクテーション処理を改善するものとして、話者適応があります。話者適応は、認識の際のモデルとして利用される認識モデルを話者に応じて最適化するものです。例えば、最初にプログラムを起動した際に、指定された文章を読むなどして、その人用に認識モデルの調整を行い、認識精度の向上を図るものです。音声認識には、このような特定話者に対するものと、話者を特定しない不特定話者に対する

音声認識があります。不特定話者の音声認識は、特定話者に対する音声認識に比べて認識精度が落ちる場合が多いものの、認識モデルの調整のためにあらかじめ文章を読むといった手間が不要であるという利点があります。

6.2 音声合成

次に音声合成について見てみましょう。音声を合成する音声合成は、さまざまな場面で実用化されています。例えば、時報案内や駅のアナウンスで音声合成が利用されています。また、スマートフォンに向かって「放送大学はどこ？」と話しかけると行き方を音声で案内してくれるサービスなども実現されています。このような音声合成を実現する手法としては、録音編集方式や規則合成方式などがあります。

録音編集方式は、あらかじめ単語や定型文などを録音しておき、それらを並べ替えるなどの簡単な編集を行って音声を合成する方式です。例えば、時報案内であれば、「10時」、「15分」、「です」といった音声をあらかじめ録音しておきます。そして、それらを繋ぎ合わせることで、「10時15分です」といった音声の合成を行います。録音編集方式の場合、あらかじめ録音した実際の音声を用いるため比較的音質が良いという利点があります。その反面、決まった内容の音声以外は合成できないという欠点があります。

規則合成方式は、基本的な音声単位の結合方法や韻律情報等の規則に基づき音声の合成を行います。特に、入力されたテキストデータに基づき音声を合成する方式は、テキスト音声合成（Text-To-Speech）と呼ばれます。規則合成方式で出力された音声は、人が直接話した音声とは異なります。このため、合成された音声は、録音編集方式に比べて、機械的で不自然に聞こえる場合も多くあります。しかしながら、合成できる音声の自由度が高く、録音をし直すこと無しに、任意の文章を音声化できるという利点があります。

日本語を合成する場合の音声単位としては、例えば、CVといった単位が利用されます。Cは子音で、Vは母音です。すなわち、「あ」や「か」といった音をあらかじめ録音しておいて、それらを並べ替えることで、「あか」といった音声を合成します。この場合、五十音に濁音「ば」や半濁音「ぱ」などを加えた、100個程度の波形データ等を蓄積しておけば、任意の音声を合成することができます。より自然な音声を合成するため、音声単位として、VCVすなわち（母音+子音+母音）や、CVCすなわち（子音+母音+子音）などが用いられる場合もあります。VCVの場合は、例えば、usaやagiという音をあらかじめ録音しておきます。このusa とagi という音を組み合わせることで、うさぎ（usagi）という音を作り出します。また、CVCの場合は、mik やkan という音をあらかじめ録音しておきます。このmik とkan という音を組み合わせることで、みかん（mikan）という音を作り出すことができます。ただし、これらの場合、用意する必要のある波形データ等は、数百から数千に増加します。一般には、より長い音声単位を用いて音声合成を行う方が音質が良くなります。このため、出現頻度が高い音声単位については、なるべく長い単位で蓄積しておき、出現頻度が低いものについては、短い単位で合成するといった方法も提案されています。

合成した音声をより自然なものにするためには、アクセントなどの韻律情報を付加することが必要になります。一般に、英語では、音の強弱によりアクセントを付加しますが、日本語では、音の高低によりアクセントを付加します。

ここで、モーラという言葉が出てきます。モーラとは、一定の時間的長さをもった音の単位で、おおむね仮名1文字に相当します。日本語の場合、アクセントのあるモーラの位置により、アクセントはいくつかの型に分類できます。日本語のアクセントは、通常、第1モーラは低く、第2モーラからアクセントのあるモーラまで高く、それ以降は低くなります。すなわち、音が下がる1つ前がアクセントの位置ということになります。例えば、3型では、最初が低く、2番目、3番目のモーラが高くなり、4番目以降のモーラは低くなります。「おぎなう」は、4番目の「う」で音が下がりますので、3型になります。同様に、2型では、最初が低く、2番目のモーラが高くなり、3番目以降のモーラは低くなります。「あきらか」は2型の例です。一方、1型では第1モーラが高く、それ以降のモーラは低くなります。「おんせい」は1型の例です。0型では、最初が低く、2番目以降のモーラが高くなり、アクセントが最後まで下がりにません。「じょうほう」は0型の例になります。

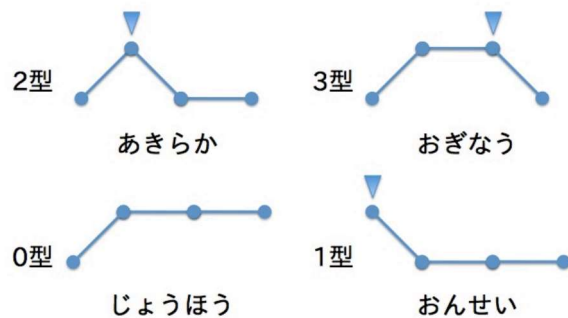


図6-3 日本語アクセントの型

アクセントは、単語のつながりによっても変化します。例えば、「歩く」は、もともと2型のアクセントです。しかし、そこに「せる」が続き「歩かせる」となった場合、アクセント型は4型のように変化します。このようなアクセント型の変化には規則性があるため、これらを辞書化しておくことにより、文節のアクセントを決めることができます。

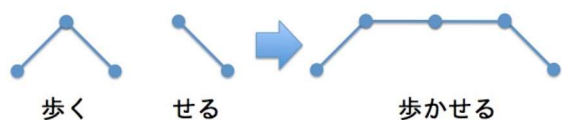


図6-4 文節のアクセントの変化