確率的言語モデルによる自由発話認識に関する研究

博士（工学）
村上仁一
豊橋技術科学大学

平成 15 年 6 月 4 日

論文要旨

確率的言語モデルによる自由発話認識に関する研究

日本文音声入力においては、音声の持つ物理的特性に着目した音声認識装置の限界を克服するため、日本語の文法や意味を用いた自然言語処理を併用することの必要性が指摘されている。この場合の言語処理の方法として、多くの言語モデルがあるが、大きく分類してルールベースの言語モデルと確率ベースの言語モデルがある。

言語の確率ベースの研究を行なう場合、基本的には大量のテキストデータ量が必要である。英語ではデータベースの重要性が認識されていて古くからBrown corpus や AP corpus などがあるが、しかし日本語ではコンピュータに読み込める形式で利用できる大量のデータベースが最近まで存在していなかった。そのため、確率的な言語モデルの研究は最近まであまり報告されていなった。しかし、この状況も新聞記事が CD-ROM で提供されるようになり、国際電気通信基礎技術研究所（ATR）が各種対話データを販売するなど、状況が変化し始めている。

そこで、本論文では、日本語において N-gram モデルの有効性をシミュレーションや実際の音声認識実験などで定量的に示した。また自由発話認識に向けて、自由発話の言語的特徴や音響的な特徴を研究した。そして実際に自由発話認識にむけた文音声認識のアルゴリズムを提案し、認識実験の結果について述べている。

各章の内容は以下の通りである。

第 1 章では、本論文の目的、動機について述べた。

第 2 章では、音声認識システムを実現するために必要な要素技術について述べた。音声認識の基本として HMM が挙げられる。そして、そのパラメータを学習データに対して尤度を最大にする Baum-Wlech アルゴリズムがある。また、連続音声認識の基本アルゴリズムとして,tree-trellis サーチや Viterbi サーチ（one-pass DP）がある。ただし、実用的な認識アルゴリズムでは、計算量やメモリー量の減少が必須である。これらの要素技術について報告した。

第 3 章では言語をマルコフモデルで表現するときのデータ量と収束性について研究した。調査項目としては、主にエンロトピーとカバー率である。そして、全テキストデータの 98%はマルコフモデルで近似できるが、残り 2%が収束しないことを示した。

第 4 章では、日本語におけるかなや漢字、品詞の bigram および trigram の有効性を、新聞記事や医療用 X 線 CT の所見作成、ATR の国際会議の予約のタスクにおいて、連続分布 HMM と単語 trigram を使用して文認識について検討し、その有効性を示した。

第 5 章では、自由発話認識のアルゴリズムとその実験結果について述べた。

自由発話では間投詞や言い淀みや言い誤り、言い直しなどが頻繁に出現する。これらの間投詞や言い直しは文の全ての場所に出現する可能性がある。そこでこれらの単語をスキップすることで、自由発話の認識が可能になる。実際に、これを実現し、実験結果によりその有効性を示した。

第 6 章では、自由発話の特徴について言語的な面と音響的な面から研究した。この結果、対話文の 50% は「あのー」、「えーと」などの間投詞を含むこと。また、言い直しは約 10%に出現することが示された。また、４人の話者について朗読発話と自由発話の音響的な違いについて述べた。そして、自由発話は、朗読発話よりも発声が曖昧になるものの。各発話環境で音響モデルを学習すれば、あまり大きな音素認識率の差は無いことが示された。

第 7 章では、音声情報に含まれている韻律情報の情報量について述べた。韻律情報は F_0, 継続時間などの多くの要素から構成されているが、本章では、この中から特にアクセント句境界の位置およびアクセント核の位置の持つ情報量に焦点を当てて情報量を測定した。実験の結果、アクセント句境界の位置がアクセント情報が持つ情報量は 5.16bit であることが示された。

第 8 章では、異なる N 個の信号源より生成された信号系列が、どの信号源から生成されたのかを分割・識別する問題について述べた。そして、応用例として複数話者発話の識別をあげ、Ergodic HMM を用いた問題の解決方法を提示した。この実験の結果、複数話者発話の識別においては 341ms 程度の長時間窓分析した LPC ケプストラムを用いることにより、より良好な識別性能が得られること、および尤度の高いモデルを選択することにより平均識別率は向上することが得られた。

第 9 章では、Ergodic HMM を利用した確率付ネットワーク文法の自動学習について述べた。Ergodic HMM と確率つきネットワーク文法が類似した構造を持ち、同種のパラメータで表現される。また、大量のテキストデータを利用して HMM のパラメータを Baum-Welch アルゴリズムで学習できる。実際の会話から作成した単語列を Ergodic HMM に学習させて、確率つきネットワーク文法を自動的に抽出することを試みた。その結果、Ergodic HMM の構造は学習データの特徴をとらえた文法的特徴を示しており、単語を文中での機能によって分類して出力していることがわかった。さらに、得られた Ergodic HMM を言語モデルとして連続音声認識に用いた。この認識実験の結果、単語 bigram よりも高い性能が得られ、提案したアルゴリズムの有効性が示された。

第 10 章では、本論文の成果をまとめ、今後の研究課題について述べた。

ABSTRACT

Study of Spontaneous Speech Recognition based on Stochastic Language Modeling

There are two types of natural language modeling. One covers the class of deterministic models like network grammar or context free grammar, that exploit some known specific properties of the language. The other includes the class of statistical models like bigram or trigram in which one tries to characterize the statistical properties of the corpus. These statistical models include stochastic context free grammar and Markov process, a sort of non-deterministic finite state automaton.

A lot of text data is needed to study statistical language models. In English, the Brown corpus and AP corpus are well known. However, such sources for Japanese have only just been created. As one example, newspapers are now available on CD-ROM.

In this paper, we describe stochastic language modeling with emphasis on the bigram model and trigram model. We also describe the efficiency of these models for continuous speech recognition. Moreover, a spontaneous speech recognition system based on stochastic language models is described.

Chapter 1 describes the background, motivation, and special features of this study.

In Chapter 2, we outline some common speech recognition models and algorithms like one-pass Viterbi decoding, HMM, and the Baum-Welch algorithm.

In Chapter 3, we study N-gram modeling for language processing, especially in terms of entropy and cover rates.

In Chapter 4, we study the effectiveness of bigrams and trigrams of Kana, Kanji, and part-of-speech. And we carried out an experiment for newspapers and X-ray CT scanning reports, ATR international conference tasks, respectively.

In Chapter 5, we describe a spontaneous speech recognition algorithm based on word trigram models. Focusing on spontaneous speech recognition, we propose a skip phone procedure to handle the many filled pauses and false starts observed in spontaneous speech. Even though the proposed method employs a simple procedure, we obtain a 47.7% sentence recognition rate for spontaneous speech. Including semantically correct sentences, the sentence recognition rate is about 75%.

In Chapter 6, we present a preliminary study of spontaneous speech recognition, describing both the acoustic and linguistic characteristics of spontaneous speech. A preliminary study was done to compare spontaneous and

read speech. In hand-labeled spontaneous speech, the labeling uncertainty increased by about 50%. A phoneme recognition experiment yielded a two fold increase in the error rate. Filled pauses appeared in 40% of 11,000 sentences of spontaneous speech utterances and false starts were found in 10% of the sentences.

In Chapter 7, we investigate the amount of information contained by accents in speech signals. It is very difficult to measure the amount of information in accents because these concepts are not clearly defined. Therefore, Kana-Kanji translations are used. First, the number of Kanji candidates that are translated using syllable information is counted. Second, the number of Kanji candidates that are translated using syllable and accent information is counted. Their ratio indicates that the amount of information in accents is 5.16 bits. Although this quantity is small compared to Japanese syllables, it is important for speech recognition.

In Chapter 8, we consider signals that originate from a number of sources. As an application of a multiple signal source identification problem, an experiment is performed on unknown speaker identification. The results indicate that the model is sensitive to the initial values of the Ergodic HMM and that employing the long-distance LPC cepstrum is effective for signal preprocessing.

In Chapter 9, we investigate statistical network grammar using Ergodic Hidden Markov Model(HMM). HMM is very rich in mathematical structure so that language models are determined more precisely than that with stochastic network grammar or Markov processes. In this chapter, we develop a statistical network grammar automatically from about 4000 words using Ergodic HMM. The resultant model indicates that some grammatical features exist even though the process was automatic.

Finally, in Chapter 10, we summarize our work and describe open or further problems.

目 次

第1章　序論

本論文では N-gram モデルを使用した連続音声認識システムの概要と自由発話認識のための認識アルゴリズムと自由発話のための言語モデルについて述べる。

1.1　研究の背景

日本文音声入力においては、音声の持つ物理的特性に着目した音声認識装置の限界を克服するため、日本語の文法や意味を用いた自然言語処理を併用することの必要性が指摘されている [91]。特に大語彙を対象とする音声には発音の個人差や曖昧さの他に、同音異義語なども多数含まれるため、その認識においては音声の物理的特性が完全に生かされたとしても、なお絞り切れない曖昧さが残り、元の文を推定するには、言語解析や意味理解の技術が必要と考えられる。この場合の言語処理の方法として、多くの言語モデルがあるが、大きく分類してルールベースの言語モデルと確率ベースの言語モデルがある。

ルールベースの言語モデルとして、ネットワーク文法や文脈自由文法、unification 文法などがあげられる。これらの言語モデルの特徴として、意味情報を直接適用して文節を生成する点があげられる。しかし、実際には単語の代わりに単語の文法的カテゴリーや意味的カテゴリーが使用されるため、絞り込みの精度はこれらのカテゴリーの分解能に依存する。したがって複数の単語候補が同一のカテゴリーに属するような語彙の認識では、文節候補を絞り込むのは困難である [86]。また人間がルールを記述するため、文法を書く負荷が大きい。したがって文法のメインテナンスも困難である。そして、詳細なルールを書くことが困難であるため、これらの言語モデルでは非文を生成しやすい傾向がある。

確率ベースの言語モデルとして単語の N-gram や確率付きネットワーク文法、確率付き文脈自由文法などがあげられる。単語の N-gram は、非常に簡単なモデルで、例えば bigram は、直前の単語に対して現在の単語が接続する確率である。また trigram は、２つ前の単語と直前の単語に対して現在の単語が接続する確率である。言語モデルを N-gram のモデルとして扱った研究は古く、シャノン [70] に始まると思われる。彼は、言語を N-gram のモデルとして扱い、エントロピを測定した。その後、IBM の研究者たちは [7] は

N-gram モデルを音声認識の言語モデルとして使用し有効性を確かめた。現在 N-gram モデルは英語の文音声認識に使用する言語モデルの主流になっている。しかし、日本語において音声認識に N-gram を使用し有効性を確かめた論文は少ない [45]。

この原因の１つに、日本語の大量のテキストデータベースの欠如にあると思われる。trigram の値を精度よく求めるためには、基本的には大量のテキストデータ量が必要である。英語ではデータベースの重要性が認識されていて古くから Brown corpus や AP corpus などがある。これらのデータベースは形態素解析などの研究のために使用されている。しかし日本語ではコンピュータに読み込める形式で利用できる大量のデータベースが最近まで存在していなかった。そのため、確率的な言語モデルの研究は最近まであまり報告されていなった。

また、従来の音声認識システムの多くは丁寧に発声された音声を入力対象にしている。しかし、人間同士の対話には「あのー」「えーと」などの間投詞や、言い淀みや言い誤りおよび言い直しや倒置などが頻繁に出現する。このような音声でも認識できる、いわゆる自由発話の音声認識が、今後の重要な研究課題になると思われる。

1.2 本研究の目的

日本語において確率的な言語モデルの研究の遅延の１つの理由に、コンピュータに読み込める形で大量のテキストが存在しなかったことがあげられる。しかし、この状況も新聞記事が CD-ROM で提供されるようになり、国際電気通信基礎技術研究所（ATR）が各種対話データを販売する [10] など、状況が変化し始めている。

そこで、本論文では、日本語において N-gram モデルの有効性をシミュレーションや実際の音声認識実験などで定量的に示す。また自由発話認識に向けて、自由発話の言語的特徴や音響的な特徴を研究した。そして実際に自由発話認識にむけた文音声認識のアルゴリズムを提案し、認識実験の結果について報告する。

また、確率的な言語モデルとして N-gram モデルの他に確率つきネットワーク文法について報告する。確率つきネットワーク文法と Ergodic HMM は、出力シンボル確率を単語の出力確率に置き換えた場合、同一のパラメータを持つ。したがって、大量のテキストから Baum-Welch アルゴリズムを用いることで、確率つきネットワーク文法が自動的に獲得できる。

最後に、今後の自由発話認識に向けて役立つと思われる、韻率情報が持つ情報量と、複数話者が同時に発話している場合の Ergodic HMM を用いた話者識別方法について報告する。

1.3 論文の構成

本論文では、確率的言語モデルとして N-gram モデルと Ergodic HMM を基本モデルとして用いている。図 1.1 に、章構成を図示した。

章は次のように構成される。

1. 序論

   第 1 章では、この論文の背景や目的について述べた。

2. 連続音声認識システムに使用するアルゴリズム

   第 2 章では第 3 章移行の研究内容の理解を用意にするために音声認識の基本について述べた。2.1 節では HMM における Baum-Welch アルゴリズムについて述べた。2.2 節では連続音声認識のアルゴリズムについて述べた。そして 2.3 節では計算量およびメモリ量を削減方法について述べた。

3. 日本語の N-gram によるモデル化

   一般的に N-gram の確率値の信頼性は学習データが増加するにつれて向上すると考えられる。しかし、N-gram の確率値を計算するときに必要なデータ量に関する基礎的な報告は少ない。そこで第 3 章では、学習データの増加に伴うモデルのエントロピーの変化と新たな N-gram の組合せの出現頻度を研究した。

4. N-gram を用いた音声認識

   第 4 章では音声認識のための言語モデルとしての N-gram の有効性について述べた。始めに日本語として比較的入手しやすい新聞記事を選び、仮名および漢字および品詞の N-gram の有効性をシュミレーションで示した。また、実際に医療用文章の入力支援において、単語の bigram の有効性を認識実験を通して示した。最後に、国際会議の申し込みの文章において単語の trigram の有効性を示した。

5. 自由発話の音声認識

   第 5 章では自由発話認識のためのアルゴリズムとその実験結果について述べた。自由発話を認識するにあたって、特に問題になるのは、冗長語（間投詞）や言い淀み、言い直しである。これらの言語現象は、文の全ての場所に出現する可能性がある。一方連続音声認識アルゴリズムは、各時刻・各状態において最尤の単語系列を知るようにアルゴリズムを変更することができる。この特徴を生かして、自由発話において特徴的な間投詞や言い直しを、音響モデルでは音素系列として認識しながら言語モデルではこれらの単語をスキップすることにより、自由発話の音声が認識できるようになる。このアルゴリズムで自由発話音声認識の実験を行なった。この結果について述べた。

6. 自由発話音声における音響的・言語的な特徴

   第 6 章では自由発話の音響的、言語的特徴について述べた。人間同士のコミュニケーションでは、「あのー」、「えーと」などに代表される間投詞や、言い淀みや言い誤りおよび言い直しなどが頻繁に出現する。このような、いわゆる自由発話の音声認識が今後の重要な研究課題になると思われる。しかし、自由発話の認識を行なうために、従来から行なわれている朗読発話との違いを検討する必要がある。そこで、本章では朗読発話と自由発話の差を研究した。音声には音響的な面と言語的な面がある。そこで音響的な面から、融合ラベルの付与率と HMM による音素認識率などを調査した。次に言語的な面から、従来の朗読発声では出現しない冗長語と言い直しの出現頻度などを調べた。

7. 音声におけるアクセント情報の持つ情報量の考察

   自由発話音声認識のための基礎的な研究として、韻律情報の持つ情報量について研究した。第 7 章では、この結果について述べた。

   音声信号には音節の情報の他に様々な韻律情報を持っている。現在、韻律の認識が不安定なため、音声認識にあまり利用されていないが、将来的な自由発話の認識において有効なパラメータであると思われる。したがって、韻律の持つ情報量は、認識性能の向上を示す指標になるため興味深いものがある。たが、これらの情報の情報量を測定することは、かなり困難である。そこで、本論文は、韻律の情報の中でも比較的把握しやすいポーズおよびアクセント位置に着目し、この情報量を漢字仮名変換および仮名漢字変換を使用して測定する方法を提案した。そして実験によりポーズおよびアクセント位置の持つ情報量を定量的に測定した。

8. Ergodic HMM を用いた未知・複数信号源クラスタリング問題の検討

   自由発話音声認識のための別の基礎的な研究として、複数話者発話の識別問題を検討した。具体的な例としては、テープレコーダに収録された議事録を、話者ごとに発話内容を分類することに相当する。この問題は、未知・複数信号源クラスタリング問題という形で一般論化できる。第 8 章では、この結果について述べた。

   異なる N 個の信号源より生成された信号系列が、どの信号源から生成されたのかを分割・識別する問題を、未知・複数信号源クラスタリング問題とする。この問題は、音声処理分野に限らず言語処理などの分野でも重要なテーマである。本章では、未知・複数信号源クラスタリング問題の応用として複数話者発話の識別問題を検討した。一方、Ergodic HMM を複数話者発話の識別問題に利用した時、カテゴリーが話者に相当し、信号系列は状態から出力されるシンボル系列と考えることができる。したがって音声データから Baum-Welch アルゴリズムを用いて

パラメータの再推定を行ない Viterbi サーチをすることにより、各話者の発話区間を求めることができる。この実験結果について報告した。

9. Ergodic HMM を用いた確率付きネットワーク文法の自動獲得の研究

   この論文では、確率的言語モデルとして言語の N-gram モデル以外に確率つきネットワーク文法についても研究した。この結果を第 9 章で述べた。

   確率付き言語モデルとしては、N-gram の他に確率付きネットワーク文法がある。この確率付きネットワーク文法は言語モデルを状態遷移確率と単語（もしくは品詞）出力確率で記述している。一方、離散型 Ergodic HMM はパラメータとして状態遷移確率、シンボル出力確率、初期状態確率を持つ。したがって Ergodic HMM の出力シンボルを単語もしくは品詞とすれば、両者は等価となる。また Baum-Welch アルゴリズムを利用することにより、学習データの尤度が最大になるように各パラメータを計算できる。そこで、Ergodic HMM を想定し、単語列（もしくは品詞列）を入力データとし Baum-Welch アルゴリズムを用いることにより、確率つきネットワーク文法を自動的に獲得できる可能性がある。

   ここでは、始めに、品詞を入力として ErgodicHMM を Baum-Welch アルゴリズムを用いて学習した。そして、学習後の HMM を解析し、その形態と従来の言語学で使用されているネットワーク文法との類似性を研究した。次に単語列を Ergodic HMM に学習させ、学習後のパラメータを検討して Ergodic HMM が文法を学習しているかどうか検討した。また、得られた Ergodic HMM を連続音声認識のための言語モデルとして使用して有効性を確かめた。また、Ergodic HMM の状態数を増加させたときの Baum-Welch アルゴリズムの改良方法について述べた。そして状態数を増加させたときの認識性能を単語の bigram や trigram と比較した。

10. 結論

    第 9.2.7 章では本論文のまとめと今後の課題について述べた。

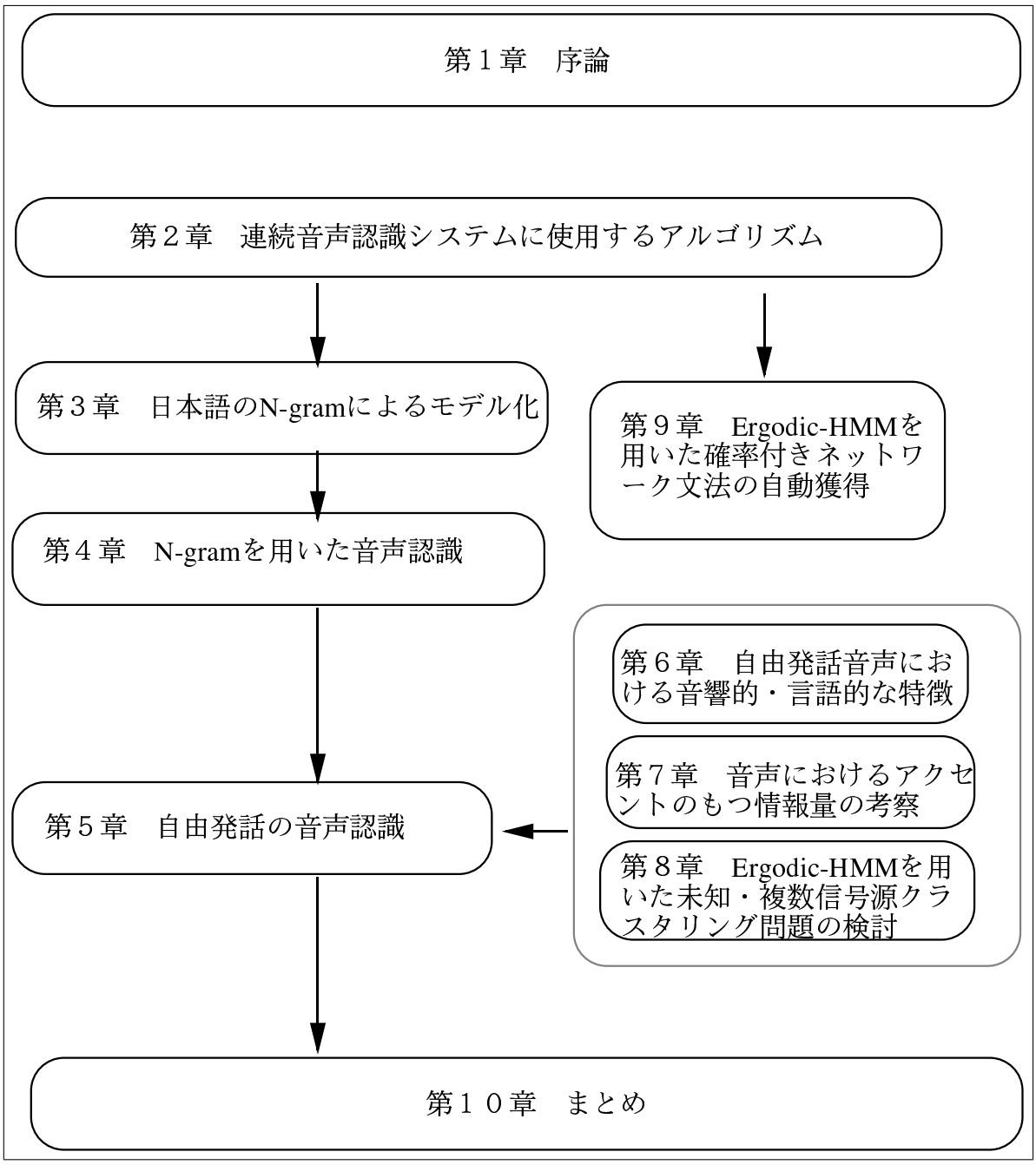

第１章　序論
第２章　連続音声認識システムに使用するアルゴリズム
第３章　日本語のN-gramによるモデル化
第９章　Ergodic-HMMを用いた確率付きネットワーク文法の自動獲得
第４章　N-gramを用いた音声認識
第６章　自由発話音声における音響的・言語的な特徴
第７章　音声におけるアクセントのもつ情報量の考察
第５章　自由発話の音声認識
第８章　Ergodic-HMMを用いた未知・複数信号源クラスタリング問題の検討
第１０章　まとめ

図 1.1: 章構成

第2章　連続音声認識システムに使用するアルゴリズム

2.1 HMM(Hidden Markov Model, 隠れマルコフモデル)

HMM は、不確定な時系列のデータをモデル化するための有効な統計的手法である [4]。HMM は、出力シンボルによって一意に状態遷移先が決まらないという意味での非決定性確率有限オートマトンとして定義される。出力シンボル系列が与えられても状態遷移系列は唯一に決まらない。観測できるのはシンボル系列だけであることから hidden(隠れ）マルコフモデルと呼ばれる [60]。

HMM はパラメータとして状態遷移確率、シンボル出力確率、初期状態確率を持つ。そして、シンボル出力確率の計算方法によって離散型 HMM と連続分布型 HMM に別れる。また、シンボル出力確率が状態で出力される Moore マシンと状態遷移で出力される Mealy マシンに分類できる。以下では、Mealy タイプの離散型 HMM について述べる [60]。なお、Moore タイプと Mealy タイプは相互に変換可能である。

T	: 観測系列の長さ
$o_1, o_2, ..., o_T$	: 観測系列
N	: 状態数
L	: 観測シンボルの数
$S = \{s\}$	: 状態集合
s_t	: 時刻 t の時の状態（番号）
i, j	: 状態番号
$v = \{v_1.v_2, ..., v_L\}$	: 出力可能なシンボル集合

と定義すると、このオートマトンは, 状態遷移確率 A, シンボル出力確率 B, 初期状態確率 π は、以下のように示される。

$$A = \{a_{ij} \mid a_{ij} = P(s_{t+1} = j \mid s_t = i)\} \qquad (1 \leq i, j \leq N) \tag{2.1}$$

$$B = \{b_{ij}(o_t) \mid b_{ij}(o_t) = P(o_t \mid s_{t-1} = i, s_t = j)\} \qquad (1 \leq i, j \leq N, 1 \leq t \leq T) \tag{2.2}$$

$$\pi = \{\pi_i \mid \pi_i = P(s_0 = i)\} \qquad (1 \leq i \leq N) \tag{2.3}$$

これらのパラメーターを用いて、HMM を次のように略記する。

$$\lambda = (A, B, \pi) \tag{2.4}$$

観測系列 O が

$$o_1, o_2, ..., o_T \qquad (o_t = v_k, 1 \leq k \leq L, 1 \leq t \leq T)$$

という系列を生成する過程は次のようになる。

1. 初期状態確率を π にしたがって決定する。
2. 次に遷移する状態（$s_{t+1} = j$）を現在の状態（$s_t = i$）と状態遷移確率 a_{ij} にしたがって決定する。
3. 状態遷移する際に出力するシンボルをシンボル出力確率 $b_{ij}(o_t)$ にしたがって決定する。
4. 2. に戻る

HMM には、ある状態から全ての状態に遷移できる全遷移型（Ergodic）モデルや、状態遷移が一定方向に進む left to right モデルがある。図 2.1 に簡単な HMM（left to right モデル）の例を示す。

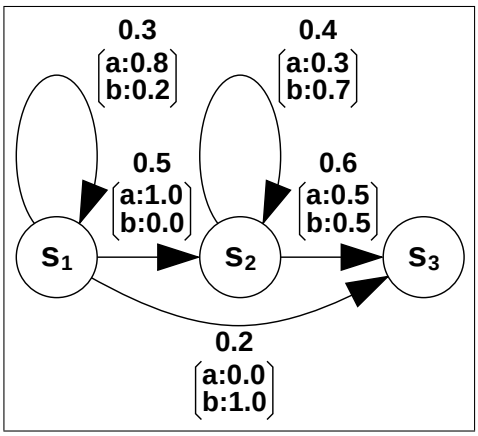


図 2.1: 3 状態 left-to-right HMM

この HMM は三つの状態で構成され、２種類のラベル a と b のみからなるラベル系列を出力する。初期状態確率は $\pi_1 = 1.0, \pi_2 = 0, \pi_3 = 0$、最終状態を S_3 とし、図のような遷移のみ行なうものとする。図において、0.3 などアークに添えられている数字は状態遷移確率を表し、[ ] 内の数字の上段はラベル a の出力確率、下段はラベル b の出力確率を表す。状態 s_1 を例にとれば、s_1 から状態 s_1 自身に 0.3 の確率で遷移し、遷移の際に 0.8 の確率で a を

出力し、0.2 の確率で b を出力する。他の状態、遷移についても同様である。ここで、ラベル系列が aab を出力する確率を考える。この HMM で許される状態系列において"aab" を出力する可能性のあるものは、$s_1 - s_1 - s_2 - s_3$ と $s_1 - s_2 - s_2 - s_3$ と $s_1 - s_1 - s_1 - s_3$ の 3 種類で、それぞれの確率は、

$$\begin{aligned} 0.3 \times 0.8 \times 0.5 \times 1.0 \times 0.6 \times 0.5 &= 0.036 \\ 0.5 \times 1.0 \times 0.4 \times 0.3 \times 0.6 \times 0.5 &= 0.018 \\ 0.3 \times 0.8 \times 0.3 \times 0.8 \times 0.2 \times 1.0 &= 0.01152 \end{aligned}$$

である。

よって、この HMM が aab を出力する確率は三つの合計、

$$0.036 + 0.018 + 0.01152 = 0.06552$$

となる。

HMM では状態系列に意味を持たないが、最尤の経路を推定することはできる。この例では、aab を出力する可能性がもっとも高い状態系列は、前記の計算から容易に $s_1 - s_1 - s_2 - s_3$ とわかる (2.1.7 参照)。

2.1.1 HMM の基本問題

HMM に関して重要な基本問題として次の五つが挙げられる [60][4]。

1. モデルの尤度評価
   観測系列 $O = o_1, o_2, ..., o_T$ と HMM,$\lambda(\pi, A, B)$ が与えられている時、モデル λ が O を出力する尤度 $P(O \mid \lambda)$ を求める。

2. モデルの推定
   学習用シンボル O を与えて尤度 $P(O \mid \lambda)$ が最大になるようにモデル λ のパラメータ π, A, B を推定する。

3. 最適状態系列の推定
   モデル λ がシンボル系列 O を出力する時の最も可能性の高い状態遷移系列を推定し、その系列に対する尤度を求める。

4. モデルの設計
   状態数や遷移先の種類などの HMM の構造を決定する。

5. 訓練用データの基準
   良いモデルを得るための訓練用データの量や質を決定する。

1. の解法を 2.1.3 節で、2. についての解法は 2.1.6 節で、3. については 2.1.7 節で方法を紹介する。4. 5. については、現在のところ経験則に依存している。

2.1.2 HMM の基本アルゴリズム

2.1.1 節で挙げた問題に対する基本的アルゴリズムを以下で紹介する [4] [60]。

2.1.3 Forward-Backward アルゴリズム

HMM $\lambda(\pi, A, B)$ が観測系列 $O = o_1, o_2, ..., o_T$ を生成する尤度 $P(O \mid \lambda)$ を求めるには、まず長さ T の全ての状態系列に対して、確率の計算を行なうことが考えられる。可能な状態系列 $S = s_0, s_1, ..., s_T$ が O を生成する確率を次のように定義する。

$$P(O \mid S, \lambda) = \prod_{t=1}^{T} P(O_t \mid s_t, \lambda) \tag{2.5}$$

各観測は、確率的に独立とみなして、

$$P(O \mid S, \lambda) = b_{s_0 s_1}(o_1) \cdot b_{s_1 s_2}(o_2) \cdot ... \cdot b_{s_{T-1} s_T}(o_T) \tag{2.6}$$

一方、状態系列 S の生成確率は次のようになる。

$$\begin{aligned} P(S \mid \lambda) &= \pi_{s_1} a_{s_1 s_2} a_{s_2 s_3} ... a_{s_{T-1} s_T} \\ &= a_{s_0 s_1} a_{s_1 s_2} ... a_{s_{T-1} s_T} \end{aligned} \tag{2.7}$$

したがって、観測系列 O のモデル λ における生成確率（尤度）は、

$$\begin{aligned} P(O \mid \lambda) &= \sum_{allS} P(O \mid \lambda) P(O \mid S, \lambda) \\ &= \sum_{all s_0 ... s_T} \pi_{s_0} a_{s_0 s_1} b_{s_0 s_1}(o_1) \cdot a_{s_1 s_2} b_{s_1 s_2}(o_2) \cdot ... \cdot a_{s_{T-1} s_T} b_{s_{T-1} s_T}(o_T) \end{aligned} \tag{2.8}$$

この方法による計算量は $\mathrm{O}(2TN^T)$ になり、実質的に計算不可能である。計算量を削減した実用的なアルゴリズムとして forward-backward アルゴリズムがある。

2.1.4 Forward probability

時刻 t の時に $o_1, o_2, ..., o_t$ という観測系列を出力して、状態 j にいる確率を次のように定義する。

$$\alpha_t(j) = P(o_1, o_2, ..., o_t, s_t = j \mid \lambda) \tag{2.9}$$

$P(O \mid \lambda)$ は $\alpha_t(j)$ の漸化式を次のように計算することによって求めることができる。

Forward probability

1. 初期化
   全ての状態 $j(1 \leq j \leq N)$ に対して

$$\alpha_0(j) = \pi_j \tag{2.10}$$

   とする。

2. 導出過程
   時間軸 $(t = 1, ..., T)$ に沿って、全ての状態 $j(1 \leq j \leq N)$ に対し、$\alpha_t(j)$ を次のように計算する。

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_{ij}(o_t) \tag{2.11}$$

3. 結果

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{2.12}$$

このアルゴリズムでは、直前のフレームにおける確率 $\alpha_{t-1}(i)$ から $\alpha_t(j)$ $1 \leq j \leq N)$ を求めている。

図 2.2 は、前記の図 2.1 の HMM がラベル系列 aab を出力する例に適応した例である。このように出力ラベル系列が対応する時間経過を横軸にして、各状態を縦に並べて状態遷移を示した図で考えると理解しやすい。$\alpha_t(j)$ はトレリス上の左上（初期状態）から右下（最終状態）に向かって順次求まる。この方法での計算量は $\mathrm{O}(N^2T)$ である。

また、$P(a, a, b \mid \lambda) = \alpha_3(3) = 0.06552$ となる。

2.1.5 Backward probability

forward probability が初期状態から前向きに計算するのに対して、backward probability は後向きに計算していく。モデル $\lambda(\pi, A, B)$ において、時刻 t に状態 i にいて、
以後 $o_{t+1}, o_{t+2}, ..., o_T$ を出力する確率を次のように表す。

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, ..., o_T, s_t = i \mid \lambda) \tag{2.13}$$

この $\beta_t(i)$ は以下の手順で求まる。

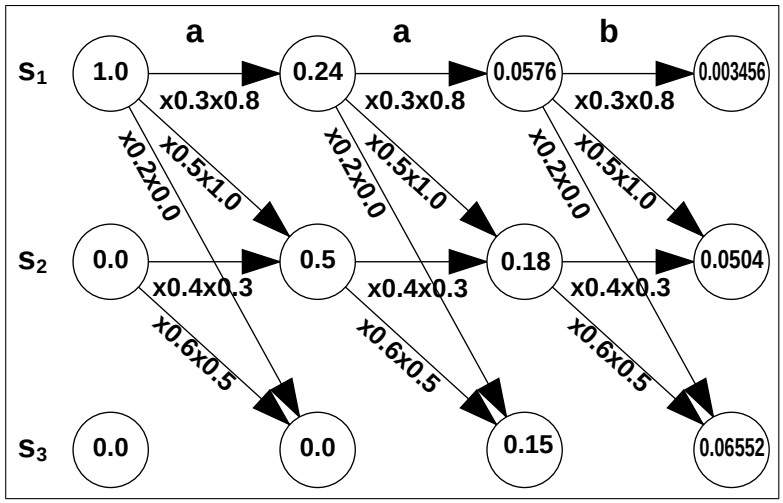


図 2.2: トレリス上の $\alpha_t(i)$ の計算

backward probability

1. 初期化
   全ての状態 $i(1 \leq i \leq N)$ に対して、

$$\beta_T(i) = 1 \tag{2.14}$$

   とする。

2. 導出過程
   時間軸 $(t = T-1, ..., 0)$ に沿って、全ての状態 $i(1 \leq i \leq N)$ に対し、β_t を次のように計算する。

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_{ij}(o_t) \beta_{t+1}(j) \tag{2.15}$$

3. 結果

$$P(O \mid \lambda) = \sum_{i=1}^{N} \pi \beta_0(i) \tag{2.16}$$

観測系列 O のモデル λ における尤度は forward probability を用いて計算できる。

2.1.6 Baum-Welch アルゴリズム

問題 2. で述べた観測系列の生成確率を最大にするモデル λ のパラメータの局所的最適値を求める方法として、Baum-Welch アルゴリズム (パラメータ再推定法) がある。

モデル λ が観測系列 $O = o_1, o_2, ..., o_T$ を生成する場合において、時刻 t で状態 i から状態 j に遷移する確率 $\xi_t(i,j)$ を次のように定義する。

$$\begin{aligned} \xi_t(i,j) &= P(s_{t-1} = i, s_t = j \mid O, \lambda) && (2.17) \\ &= \frac{\alpha_{t-1}(i) a_{ij} b_{ij}(o_t) \beta_t(j)}{P(O \mid \lambda)} \qquad (1 \le t \le T) && (2.18) \end{aligned}$$

ここで、シンボルの生成過程で、時刻 t で状態 j にいる確率 $\gamma_t(j)$ を定義する。

$$\begin{aligned} \gamma_t(j) &= P(s_t = j \mid O, \lambda) && (2.19) \\ &= \sum_{i=1}^{N} \xi_t(i,j) \qquad (1 \le t \le T) && (2.20) \end{aligned}$$

この $\gamma_t(i)$ と $\xi_t(i,j)$ とからモデル λ の再推定（$\lambda \to \bar{\lambda}$ ）を次のように行なう。

Baum-Welch algorithm

1. 初期状態確率

$$\bar{\pi}_i = \gamma_0(i) = \frac{\alpha_0(i)\beta_0(i)}{P(O \mid \lambda)} \qquad (1 \le i \le N) \tag{2.21}$$

2. 状態遷移確率

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T} \xi_t(i,j)}{\sum_{t=1}^{T} \gamma_{t-1}(i)} = \frac{\sum_{t=1}^{T} \alpha_{t-1}(i) a_{ij} b_{ij}(o_t) \beta_t(j)}{\sum_{t=1}^{T} \alpha_{t-1}(i) \beta_{t-1}(i)} \tag{2.22}$$

3. シンボル出力確率

$$\bar{b}_{ij}(o_t) = \frac{\sum_{t=1, s.t. o_t = v_k}^{T} \xi_t(i,j)}{\sum_{t=1}^{T} \xi_t(i,j)} = \frac{\sum_{t=1, s.t. o_t = v_k}^{T} \alpha_{t-1}(i) a_{ij} b_{ij}(o_t) \beta_t(j)}{\sum_{t=1}^{T} \alpha_{t-1}(i) a_{ij} b_{ij}(o_t) \beta_t(j)} \tag{2.23}$$

再推定された $\bar{\lambda}$ の評価は次のようになる。

1. $\bar{\lambda} = \lambda$ → （局所的な）収束状態

2. $P(O \mid \bar{\lambda}) > P(O \mid \lambda)$ → シンボル系列 O を出力するより最適なモデル λ を推定

Baum-Welch アルゴリズムは、学習データの尤度を最大にするようにパラメータを学習する。ただし、基本的には gradient 学習によるパラメータ収束の学習方法であるため、local maxmum の方向にしか学習は進まない。そのため初期値が重要になる。音響モデルでは通常 left-light モデルが使用されるためあまり問題にならないが、全ての状態が全ての状態に接続される Ergodic HMM では、この初期値が特に問題になる [4]。

2.1.7 Viterbi アルゴリズム

Viterbi アルゴリズムはモデル λ において最適な状態系列（最適経路）$S = s_1, s_2, ..., s_T$ と、この経路上での確率を求めるアルゴリズムである。

モデル λ において観測系列 $O = o_1, o_2, ..., o_T$ に対する最適な状態系列 $s = s_1, s_2, ..., s_T$ を求めるために、時刻 t で状態 i に至るまでの最適状態確率 $\delta_t(i)$ を定義する。

$$\delta_t(i) = \max_{s_1, s_2, ..., s_{t-1}} p(s_1, s_2, ..., s_t = i, o_1, o_2, ..., o_t \mid \lambda) \tag{2.24}$$

時刻 $t+1$ における最適状態の確率は次のように導出できる。

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] \cdot b_{ij}(o_{t+1}) \tag{2.25}$$

時刻 t 状態 i において生成確率を最大にする経路（状態遷移）を $\psi_t(j)$ 、最適経路の生成確率を P^* 、最適経路上の最終状態を s_T^* とすると最適経路、およびその生成確率は以下の手順で求まる。

Viterbi アルゴリズム

1. 初期化

$$\delta_0(i) = \pi_i \tag{2.26}$$
$$\psi_0(i) = 0 \qquad (1 \leq i \leq N) \tag{2.27}$$

2. 導出過程

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij} b_{ij}(o_t)] \tag{2.28}$$
$$\psi_t(j) = \underset{1 \leq i \leq N}{argmax} [\delta_{t-1}(i) a_{ij} b_{ij}(o_t)] \qquad (1 \leq t \leq T, 1 \leq j \leq N) \tag{2.29}$$

3. 結果

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \tag{2.30}$$
$$s_T^* = \underset{1 \leq i \leq N}{argmax} [\delta_T(i)] \tag{2.31}$$

4. 状態系列のバックトラック

$$s_t^* = \psi_{t+1}(s_{t+1}^*)(0 \leq t \leq T-1) \tag{2.32}$$

4. で求めた $s_0^*, s_1^*, ..., s_T^*$ が最適経路となる。前出の aab を出力するモデルに Viterbi アルゴリズム を用いた簡単な例を図 2.3 に示す。

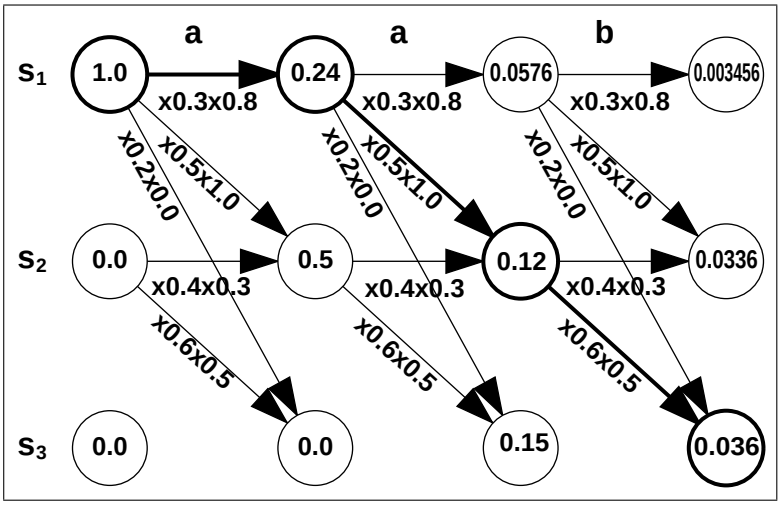


図 2.3: Viterbi アルゴリズムの適用例

2.1.8 スケーリング

forward-backward アルゴリズムによって、$\alpha_t(i)$ や $\beta_t(i)$ を求める際、入力系列が長いために、計算が進むにつれてこれらの値が小さくなり、計算機で扱える最小値よりも小さくなることがある（アンダーフロー）。特にシンボル出力確率が連続分布を仮定している連続分布型 HMM（2.1.9 章）において、この現象が顕著になる。このアンダーフローを回避するために、$\alpha_t(i)$ や $\beta_t(i)$ に適当な係数を掛けたスケーリングという方法が知られている [60] [4]。スケーリングを導入した forward-backward アルゴリズム を以下に示す。

1. スケーリングを適用した forward probability

   (a) 初期化
   全ての状態 $i(1 \leq i \leq N)$ に対して

   $$\alpha_0^*(i) = \alpha_0(i) = \pi_i \tag{2.33}$$

   $$\alpha_0'(i) = C_0\alpha_0^*(i) = C_0\alpha_0(i) \tag{2.34}$$

   とする。

   (b) 導出過程
   時間軸 $(t = 1, ..., T)$ に沿って、全ての状態 $i(1 \leq i \leq N)$ に対し、α_t' を次のように計算する。

   $$\alpha_t^*(i) = \sum_{j=1}^{N} \alpha_{t-1}'(j)a_{ji}b_{ji}(o_t) \tag{2.35}$$

   $$\alpha_t'(i) = C_t\alpha_t^*(i) = C_0C_1...C_t\alpha_t(i) \tag{2.36}$$

   とする。

   (c) 結果

   $$P(O \mid \lambda) = \prod_{t=0}^{T} C_t \tag{2.37}$$

   この式は積の形であるので、実際に計算する時はアンダーフローを回避するため対数で $P(O \mid \lambda)$ を算出する。

   $$\log P(O \mid \lambda) = \log\big(-\sum_{t=0}^{T} C_t\big) \tag{2.38}$$

但し、時刻 t におけるスケーリング係数 C_t は以下の式で求める。

$$C_t = \left[\sum_{i=1}^{N} \alpha_t^*(i)\right]^{-1} \tag{2.39}$$

これにより、全時刻において

$$\sum_{i=1}^{N} \alpha'_t(i) = 1 \tag{2.40}$$

となるため、forward probability のアンダーフローが回避できる。

2. スケーリングを適用した backward probability

   backward probability では forward probability で算出したスケーリング定数を用いる。

   (a) 全ての状態 $i(1 \leq i \leq N)$ に対して

   $$\beta^*_T(i) = \beta_T(i) = 1 \tag{2.41}$$

   $$\beta'_T(i) = C_T \beta^*_T(i) = C_T \beta_T(i) \tag{2.42}$$

   とする。

   (b) 時間軸 $(t = T-1, ..., 0)$ に沿って、全ての状態 $i(1 \leq i \leq N)$ に対し、β'_t を次のように計算する。

   $$\beta^*_t(i) = \sum_{j=1}^{N} a_{ij} b_{ij}(o_{t+1}) \beta'_{t+1}(j) \tag{2.43}$$

   $$\beta'_t(i) = C_t \beta^*_t(i) = C_t C_{t-1} ... C_t \beta_t(i) \tag{2.44}$$

   スケーリング法を用いて算出した $\alpha'_t(i)$、$\beta'_t(i)$ とスケーリングを用いない $\alpha_t(i)$、$\beta_t(i)$ の間には次式のような関係がある。

   $$\alpha'_t(i) = \prod_{\tau=1}^{t} C_\tau \alpha_t(i) \tag{2.45}$$

   $$\beta'_t(i) = \prod_{\tau=t}^{t} C_\tau \beta_t(i) \tag{2.46}$$

3. スケーリング適用時のパラメータの再推定

   スケーリングを行なう場合の各パラメータの更新式は次式となる。

   (a) 初期状態確率

   $$\bar{\pi}_i = \alpha'_0(i) \beta'_0(i) \qquad (1 \leq i \leq N) \tag{2.47}$$

   (b) 状態遷移確率

   $$\bar{a}_{ij} = \frac{\sum_{t=1}^{T} \alpha'_{t-1}(i) a_{ij} b_{ij}(o_t) \beta'_t(j)}{\sum_{t=1}^{T} \alpha'_{t-1}(i) \beta'_{t-1}(i) / C_{t-1}} \tag{2.48}$$

(c) シンボル出力確率

$$\bar{b}_{ij}(o_t) = \frac{\sum_{\substack{t=1 \\ s.t. o_t = v_k}}^{T} \alpha'_{t-1}(i) a_{ij} b_{ij}(o_t) \beta'_t(j)}{\sum_{t=1}^{T} \alpha'_{t-1}(i) a_{ij} b_{ij}(o_t) \beta'_t(j)} \tag{2.49}$$

2.1.9 連続分布型 HMM

HMM には離散型 HMM の他に連続分布型 HMM がある。連続分布型 HMM は、シンボル出力確率をガウス分布で表現したもので、離散型 HMM のシンボル出力確率は 0,0 から 1.0 までの値しかとらないのに対し、離散型 HMM のシンボル出力確率は 0 から $+\infty$ の値をとる。

連続分布型 HMM のシンボル出力確率 $b_j(O_t)$ は以下のように計算される。

$$b_j(O_t) = \sum_{m=1}^{M_j} \mathcal{C}_{jm} \mathcal{N}(O_t; \mu_{jm}, \Sigma_{jm}) \tag{2.50}$$

ただし、

M_j	...	状態 j における混合数
$\mathcal{C}_{jm}$	...	状態 j における混合数 m のときの重み
$\mathcal{N}(;\mu,\Sigma)$	...	平均ベクトル μ、共分散行列 Σ をもつ混合ガウス分布

$\mathcal{N}(;\mu,\Sigma)$ は以下の式で表現される。

$$\mathcal{N}(O;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp(-\frac{1}{2}(O-\mu)^t \Sigma^{-1}(O-\mu)) \tag{2.51}$$

ただし

n	...	観測行列の次元数
$(O-\mu)^t$	...	$(O-\mu)$ の天地行列
$\|\Sigma\|$	...	Σ の固有値
Σ^{-1}	...	Σ の逆行列

連続分布型 HMM における forward probability $\alpha_j(t)$ は以下のように計算される。

$$\alpha_j(t) = \big(\sum_{i=2}^{N-1} \alpha_i(t-1) a_{ij}\big) b_j(O_t) \tag{2.52}$$
$$1 < j < N, 1 < t \leq T$$

ただし

$$
\begin{aligned}
&\alpha_1(1) = 1 \\
&\alpha_j(1) = a_{1j} b_j(O_1) \\
&1 < j < N \\
&\alpha_N(T) = \textstyle\sum_{i=2}^{N-1} \alpha_i(T) a_{iN}
\end{aligned}
$$

連続分布型 HMM における backward probability $\beta_i(t)$ は以下のように計算される。

$$
\begin{aligned}
\beta_i(t) = \sum_{j=2}^{N-1} a_{ij} b_j(O_{t+1}) \beta_j(t+1) \\
1 < i < N, T > t \geq 1
\end{aligned}
\tag{2.53}
$$

ただし

$$
\begin{aligned}
&\beta_i(T) = a_{iN} \\
&1 < i < N \\
&\beta_1(1) = \textstyle\sum_{j=2}^{N-1} a_{1j} b_j(O_1) \beta_j(1)
\end{aligned}
$$

forward probability, backward probability から連続分布型 HMM における Baum-Welch アルゴリズムによるパラメータの再推定は以下の式で表現される。

$$
\begin{aligned}
\hat{a}_{ij} = \frac{\Sigma_{r=1}^{R} \frac{1}{P_r} \Sigma_{t=1}^{T_r - 1} \alpha_i^r(t) a_{ij} b_j(O_{t+1}^r) \beta_j^r(t+1)}{\Sigma_{r=1}^{R} \frac{1}{R_r} \Sigma_{t=1}^{T_r} \alpha_i^r(t) \beta_i^r(t)} \\
1 \leq r \leq R, 1 < i < N, 1 < j < N \\
P_r = \sum_r \big(P = \alpha_N(T) \ \ of \ \ r'thobservation \big)
\end{aligned}
\tag{2.54}
$$

$$
\hat{\mu}_{jm} = \frac{\Sigma_{r=1}^{R} \Sigma_{t=1}^{T_r} L_{jm}^r(t) O_t^r}{\Sigma_{r=1}^{R} \Sigma_{t=1}^{T_r} L_{jm}^r(t)} \tag{2.55}
$$

$$
\hat{\Sigma}_{jm} = \frac{\Sigma_{r=1}^{R} \Sigma_{t=1}^{T_r} L_{jm}^r(t) (O_t^r - \hat{\mu}_{jm})(O_t^r - \hat{\mu}_{jm})^t}{\Sigma_{r=1}^{R} \Sigma_{t=1}^{T_r} L_{jm}^r(t)} \tag{2.56}
$$

$$
\mathcal{C}_{jm} = \frac{\Sigma_{r=1}^{R} \Sigma_{t=1}^{T_r} L_{jm}^r(t)}{\Sigma_{r=1}^{R} \Sigma_{t=1}^{T_r} L_j^r(t)} \tag{2.57}
$$

ただし

$$
L_{jm}^r(t) = L_j^r(t) = \frac{1}{P_r} \alpha_j(t) \beta_j(t) \tag{2.58}
$$

$$
L_{jm}^r(t) = \frac{1}{P_r} U_j^r(t) \mathcal{C}_{jm} b_{jm}(O_t^r) \beta_j^r(t) b_{js}(O_t^r) \tag{2.59}
$$

$$U_j^r(t) = \begin{cases} a_{1j} & if \ \ t = 1 \\ \Sigma_{i=2}^{N-1} \alpha_i^r(t-1) a_{ij} & otherwise \end{cases} \tag{2.60}$$

2.1.10 HMM のエントロピー

$p(v_k \mid i)$ を状態 i から遷移する際に記号 v_k を出力する確率と定義して、HMM（λ）のエントロピーは次のようにして求められる [60]。

$$p(v_k \mid i) = \sum_{j=1}^{N} a_{ij} b_{ij}(v_k) \text{... 状態 } i \text{ でシンボル } v_k \text{ を生成する確率} \tag{2.61}$$

$$H(K \mid i) = -\sum_{k=1}^{K} p(v_k \mid i) \log_2 p(v_k \mid i) \text{...1 シンボル当たりのエントロピー} \tag{2.62}$$

$$H(\lambda) = \sum_{i=1}^{N} \omega_i H(K \mid i) \text{... モデル } \lambda \text{ のエントロピー} \tag{2.63}$$

ただし、

N	.....	HMM の状態数
K	.....	シンボルの数（種類）
a_{ij}	.....	状態 i から状態 j へ遷移する確率
$b_{ij}(v_k)$	.....	状態 i から状態 j へ遷移する際に v_k を出力する確率
ω_i	.....	状態 i の定常状態確率

定常状態確率 ω_i は以下の計算式から得られる [83]。

Ergodic HMM において、状態 S_i から遷移を開始し、n 回の遷移を繰り返した後に状態 S_j に達する確率 (n 次の遷移確率) を $a_{ij}^{(n)}$ と表すことにする。$a_{ij}^{(n)}$ には次の式が成立する。

$$a_{ij}^{(n+1)} = \sum_{\nu=1}^{N} a_{i\nu} a_{\nu j}^{(n)} \tag{2.64}$$

$$a_{ij}^{(n+m)} = \sum_{\nu=1}^{N} a_{i\nu}^{(n)} a_{\nu j}^{(m)} \tag{2.65}$$

n が大きくなるにつれて、$a_{ij}^{(n)}$ は一定値に近づき、その値は状態 S_j のみで決まり、出発点 S_i には無関係になることが証明できる [83]。すなわち、

$$\lim_{n \to \infty} a_{ij}^{(n)} = \omega_j \tag{2.66}$$

となる。ω_j は、十分な遷移のあとにおいて、任意の瞬間に、この過程が状態 S_j にある確率を表す。定常状態確率 ω_j には次の式が成り立つ。

$$\sum_{j=1}^{N} \omega_i = 1 \tag{2.67}$$

$$\sum_{i=1}^{N} \omega_i a_{ij} = \omega_j \tag{2.68}$$

したがって、定常状態確率確率 ω_j は、2.67 式と 2.68 式を解くことにより、遷移確率から求められる。

2.2 連続音声認識のアルゴリズム

連続単語認識アルゴリズムとして２段 DP や one-pass DP、Level building などのアルゴリズムが知られている。この中で tree-trellis サーチは全探索サーチで最も基本的なアルゴリズムといえる。ここでは tree-trellis サーチと Viterbi サーチ (one-pass DP) のアルゴリズムについて説明する。

2.2.1 tree-trellis サーチ

連続単語認識アルゴリズムとして最も基本的なアルゴリズムは、tree-trellis サーチである。

このアルゴリズムは、テストデータ全てに対して全ての可能性を計算するため、計算量、メモリ量は膨大になる。しかし、N 位までの累積尤度の単語列 (N-best リスト) を出力することができる。また、グリッドの選択において最尤なものを選ぶ方法 (Viterbi) とグリッドの尤度を足す方法 (trellis) の両者が選択できる。Trellis で計算をした場合、フレーム単位での状態の位置が明確にならないため、HMM を用いた音声認識において良く使用される duration control（状態継続時間の制限）は意味をもたない。また、任意の時間において単語を認識させることが可能なため、単語スポッタとしても動作が可能である。

またアルゴリズムにおいて各単語の HMM の最後の状態と後続する単語の最初の状態の遷移において任意の言語モデルの制約を加えることにより、音響モデルと言語モデルを簡単に結合することができる。つまり、言語モデルは単語 bigram に限らず CYK などの全ての left-right 型の言語モデルを採り入れることが可能である。

グリッドを Trellis で計算した場合の tree-trellis サーチのアルゴリズムを表 2.1 および図 2.4 に示す。なお、１文は n 個の単語から構成される ($L = w_0, w_1, w_2, \ldots, w_{n-1}$) と仮定した。

図 2.4 に、このアルゴリズムの簡略図を示す。この図では、認識語彙数を w_a と w_b の 2 単語で、単語の HMM は 4-state 3-loop で、状態は 0 から 2 までとする。縦軸は HMM の状態で、横軸は時間で、奥行きは語彙を示している。図中の①は時間 0 から時間 $t-1$ までの単語 w_a の状態 0、②は時間 0 から時間 $t-1$ までの単語 w_a の状態 1、③は時間 0 から時間 $t-1$ までの単語 w_a の状態 2、④は時間 0 から時間 t までの単語 w_a の状態 2、⑤は時間 0 から時間 $t-1$ までの連続 2 単語 w_a, w_a の状態 0、⑥は時間 0 から時間 $t-1$ までの連続 2 単語 w_a, w_b の状態 0、⑦は時間 0 から時間 t までの連続 2 単語 w_a, w_a の状態 0、⑧は時間 0 から時間 t までの連続 2 単語 w_a, w_b の状態 0、の累積尤度であるとする。

tree-trellis サーチの trellis 計算においては、単語の最初の状態 0 を意味するグリッド以外は、前時刻の同一状態を意味するグリッドの尤度と前時刻の

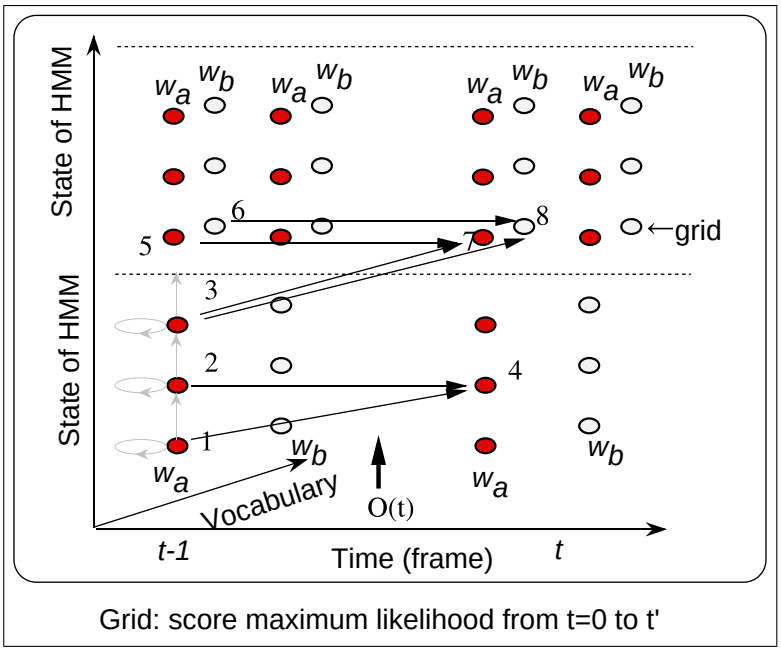


図 2.4: tree-trellis サーチのアルゴリズム

１つ前の状態のグリッドの尤度の２状態を加えて現時刻のグリッドの尤度を計算する。例えば、◯4は◯1の遷移と◯2から遷移の累積尤度の総和とする。

しかし、単語の最初の状態 0 を意味するグリッドは、前時刻の同一のグリッドの累積尤度とレベルが１つ下の単語の最終状態を意味するグリッドの累積尤度の総和とする。例えば、◯7は◯3の遷移と◯5の遷移の累積尤度の総和とする。

これを全グリッドに対して計算を行なう。

なお、このとき言語モデルの確率値を掛けることにより音響モデルと言語モデルが結合できる。例えば、◯7は◯3の遷移と◯5の遷移の累積尤度に単語 bigram$P(w_b|w_a)^\alpha$, (α: language weight) を掛けることによって、単語の bigram と単語の HMM が簡潔に結合できる。

2.2.2 Viterbi サーチ (one-pass DP)

Viterbi サーチ (one-pass DP) は各認識単語の最後の状態を意味するグリッドと単語の最初の状態を意味するグリッドの遷移において尤度の高い遷移を選択していく。認識単位を単語とした場合のアルゴリズムを表 2.2 に示す。

図 2.5 に、このアルゴリズムの簡略図を示す。この図では、認識語彙数を w_a と w_b の 2 単語で、単語の HMM は 4-state 3-loop で、状態は 0 から 2 までとする。縦軸は HMM の状態で、横軸は時間で、奥行きは語彙を示してい

る。(1)は時間 $t-1$ において単語 w_a 状態が 0、(2)は時間 $t-1$ において単語 w_b 状態が 1、(3)は時間 t において単語 w_a 状態が 0、(4)は時間 $t-1$ において単語 w_b 状態が 2、(5)は時間 $t-1$ において単語 w_a 状態が 2、(6)は時間 t において単語 w_b 状態が 2、を意味するグリッド（最大累積尤度）であるとする。

単語の最初の状態を意味するグリッド以外は、前時刻の同一状態と前時刻の１つ前の最大累積尤度の２遷移のうち、最大累積尤度の高い方を選択する。例えば、(6)は(2)の遷移と(4)から遷移の最大累積尤度の高い方を選択する。しかし、単語の最初の状態を意味するグリッドは、前時刻の最初の同一の最大累積尤度と各認識単語の最後の最大累積尤度の高い方を選択する。例えば、(3)は (1), (5), (6)の遷移の尤度の高い方を選択する。

なお、単語の bigram を利用するときは、(3)は(5)に bigram の値 $(p(w_a|w_a)^{\alpha})$ を掛けたものと(4)に bigram の値 $(p(w_a|w_b)^{\alpha})$, (α: language weight) の遷移の尤度の高い方を選択する。

これを全状態に対して計算を行なう。

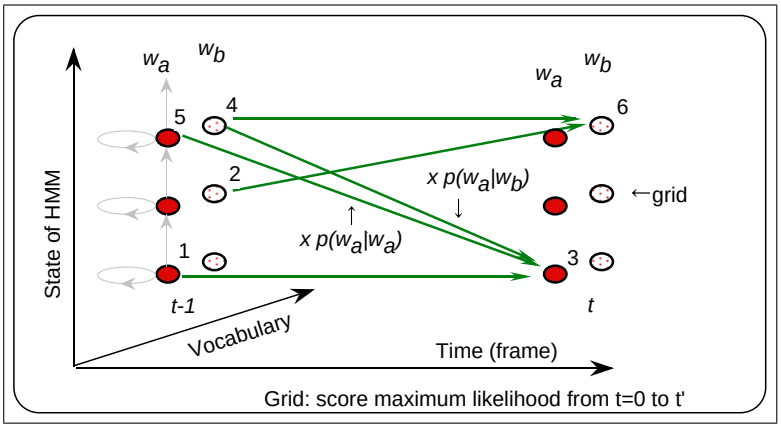


図 2.5: one-pass のアルゴリズム

2.2.3 tree-trellis サーチと Viterbi サーチの比較

上記で、tree-trellis サーチと Viterbi サーチのアルゴリズムについて述べた。この両者のアルゴリズムには各々長所短所がある。したがって認識性能と計算コストとメモリー量を考慮してアルゴリズムを選択する必要がある。表 2.3 にこれらのアルゴリズムの特徴についてまとめた。

なお、tree-trellis サーチと Viterbi サーチをグリッドを中心に考えると、tree-trellis サーチにおいて１単語ごとにマージするものを、Viterbi サーチと呼んでいることになる。したがってグリッドをオブジェクトと考えてプログラムを作成すると tree-trellis サーチと Viterbi サーチはほぼ同一のプログラムで作成

できる。また、音素の HMM は前後の音素環境を考慮する context dependent タイプが良く使用されている。グリッドを考えるとき、前後の音素環境も考慮しながらマージすることにより triphone のような context-dependent model も扱える。また、言語モデルとしてネットワーク文法や文脈依存文法などを利用するとき、過去の履歴に関して完全に一致する場合のみマージをすることで、left-right 型のネットワーク文法に当てはめることが可能である。

2.3 アルゴリズムの改良

現実のコンピュータにおいて 2.2.1 節や 2.2.2 節で紹介したアルゴリズムは、計算量およびメモリー量などの理由から実用的ではない。

この節では、実際に音声認識システムを動かすためのアルゴリズムの改良方法について説明する。

2.3.1 ビームサーチ

各フレームごとの尤度計算において、累積尤度の低い単語列は正解の単語列になる可能性が低いため、以後の探索から除外できる可能性が高い。そこで、フレームごとに最も高い累積尤度から正解の存在をおおよそ保証できる、ある個数（ビーム幅 b）のみ計算を続けることにより、計算量およびメモリ量が削減できる [68]。具体的には、すべての $w_n, .., w_0, i$ に対して表 2.1 、10) の式の計算のかわりに、最も高い累積尤度から、ある個数（ビーム幅 b）のみを計算する。したがって $G_t(w_1, ..., w_0, i)$ を記憶するメモリ量は、tree-trellis サーチでは O(認識語彙数 $^n\times$ 単語の状態数) が必要であるのに対し、ビームサーチでは O(ビーム幅 b) しか必要としないため大幅に削減できる。また、計算量もビーム幅の計算方法によって異なるが、同様な比率で削減できる。

2.3.2 ビームの絞り方

ビームの絞り方には、次の 2 つの方法がある。

1. 尤度の閾値

   尤度の閾値でビームを絞る方法は、計算量が少なくてすむためよく利用されている [40]。しかし、認識を行なう前に予め閾値を決めておかなければならないため、動作が不安定になることがある。

2. ビーム幅

   一定のビーム幅でビームを絞る方法は、フレームごとにソーティングが必要になる。そのため、計算量が増大する。だたし、始めにフレー

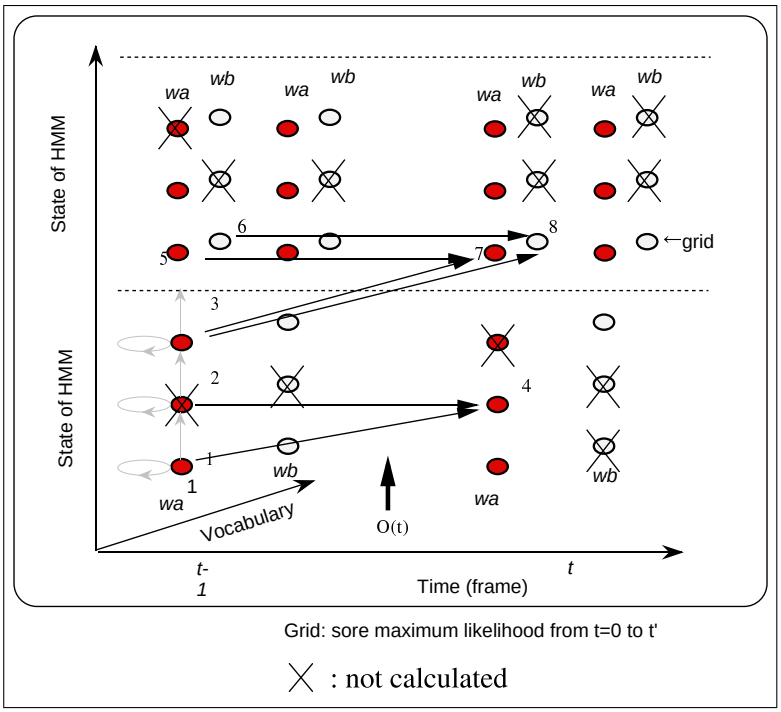


図 2.6: ビームサーチの計算方法

ムごとに最も高い最大累積尤度 (G_{max}) からビーム幅 b の最大累積尤度 (G_b) を計算し、次にこの尤度 (G_b) でビームを絞り込むことによって [85] フルソートと比較すると計算量が大幅に削減できる。

本論文では、histgram ソートと呼ばれる方法を採用している。これは、音声認識において正確なビーム幅 b を決めなくても、ある程度幅を持つ $\hat{b} = b \pm \delta$ でも問題が少ないという点に着目している。

計算方法を以下に示す。

1. フレームごとに尤度の最大値と最小値を計算する。
2. 最大値と最小値の間を適当な幅で区切り、尤度のヒストグラムを作成する。
3. 最大値からビーム幅 b の値に近い尤度 ($\hat{G}_b$) を計算する。
4. ($\hat{G}_b$) でビームの枝刈りを行う。

図 2.7 にビーム幅 4096 のときの例を示す。この場合はビーム幅は 4190 になっている。

このアルゴリズムは、尤度の閾値で枝刈りする方法とビーム幅で枝刈りする方法の中間のようなアルゴリズムと言える。

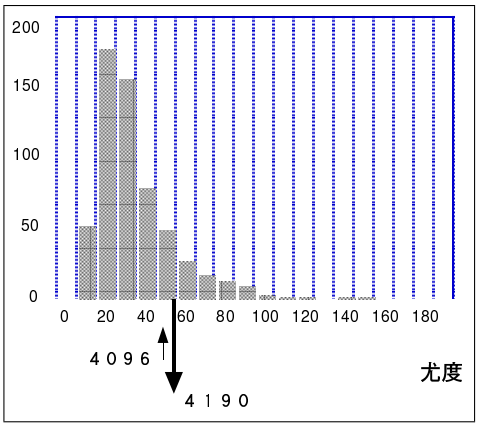


図 2.7: histgram sort

2.3.3 近接したフレームにおける言語モデルの類似性の利用

音声認識アルゴリズムにおいて単語と単語の境界の尤度計算するとき、言語モデルを使用した場合、パーザーを動かして単語仮説を生成させる必要がある。しかし、フレーム同期型のアルゴリズムでは、直前のフレームにおいて生成された文仮説候補は、現在のフレームの文仮説候補になる可能性が高い。この特徴を利用して、前フレームにおいて計算済みの単語仮説の確率値を再利用するようにすれば、フレームごとに単語仮説を計算する必要がないため、大幅に計算量が削減できる。なお、この方法は文献 [26] においても紹介されている。

2.3.4 単語 trigram の値の検索方法

言語モデルとして単語の trigram を利用する場合、単語 trigram の値を直接記憶すると［最大認識単語数 ^{3}］のメモリ量が必要である。しかし、サンプリングデータ中に存在する組み合わせをリスト構造で記憶することにより、trigram の値が 0 である組合せはメモリーに展開されないため、必要なメモリ量を削減できる。また、完全ハッシュアルゴリズム [1] を採用することにより、trigram の値を参照するための計算量は大幅に削減できる。

2.3.5 対数の加算の計算方法

音声認識において連続分布の HMM を使用したとき、計算のダイナミックレンジが大きく変化するため対数で計算する。そのとき対数の足し算が必要になる。このアルゴリズムとして線形補間の方法が報告されている [4]。本論文では、この方法と他に以下の方法を採用した。

$$
\begin{aligned}
A &= \log(a) \\
B &= \log(b) \\
C &= \log(a+b) \\
&= \log(a) + \log(1 + b/a) \\
&= A + \log(1 + \exp(B - A))
\end{aligned}
\tag{2.69}
$$

を利用して以下のように対数同士の足し算を計算する。

$$
\begin{aligned}
if \quad & (A \gg B) \quad A; \\
else\ if \quad & (B \gg A) \quad B; \\
else\ if \quad & (A \geq B) \quad A + \log(1 + \exp(B - A)); \\
else\ if \quad & (B \geq A) \quad B + \log(1 + \exp(A - B));
\end{aligned}
\tag{2.70}
$$

なお、この方法は HTK[15] において使用されている。

2.3.6 音素 HMM

表 2.4 に示したアルゴリズムは、連続単語認識アルゴリズムである。しかし、単語の HMM は一般的に音素の HMM を連結させて作成する。（例えば「通訳」という単語の HMM は/ts/, /u/, /y/, /a/, /k/, /u/ の計 6 音素の HMM が連結して作成する。）そこで認識単位を音素として全音素の HMM のシンボル出力確率を計算する。そして、各単語の、共通の音素のシンボル出力確率は共有する。これにより計算量が削減できる。

2.3.7 遅延言語処理

通常の認識アルゴリズムでは、各単語の先頭のグリッドは言語の遷移確率と接続する前の単語の最終状態のグリッドからの音響尤度を足した尤度と自己ループの尤度を比較して計算する。この時言語の遷移確率を遅らせて計算する。

例えば図 2.8 は、tree-trellis サーチにおいて単語の bigram を使用したときの図である。この図では、語彙 2 単語（w_a と w_b）で、連続 3 単語認識のときの grid を図示している。通常のアルゴリズムでは、grid 2 では grid 1 からの遷移に $p(w_a|w_a)$ の単語 bigram の確率を、grid 3 では grid 1 からの遷移に $p(w_b|w_a)$ を、grid 6 では grid 5 からの遷移に $p(w_a|w_a)$ を、grid 7 では grid 5 からの遷移に $p(w_b|w_a)$ をかける。

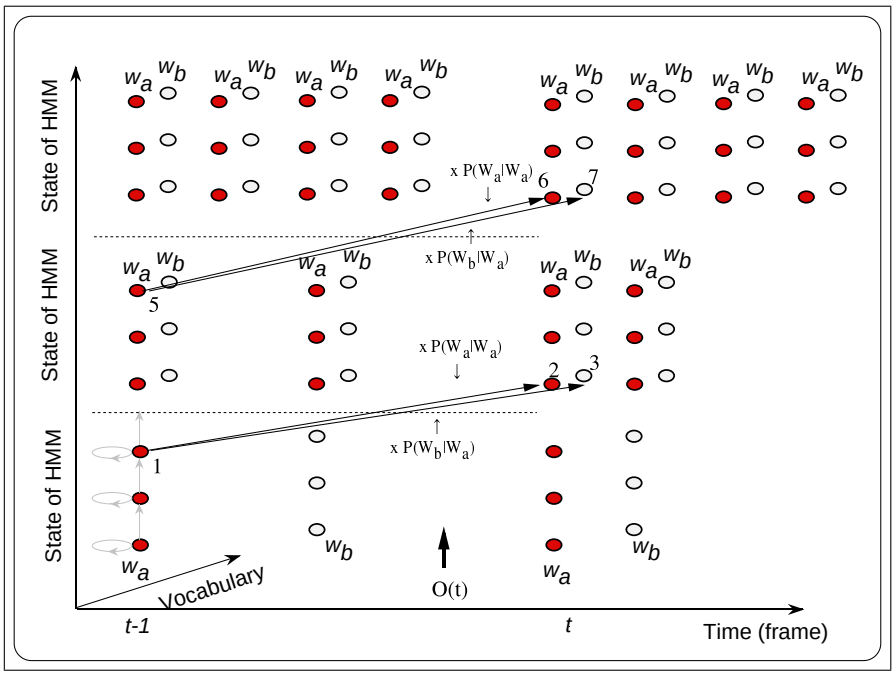


図 2.8: 通常の tree-trellis サーチ

遅延言語処理は、言語モデルの確率を、単語認識した後に音響尤度とかける。図 2.9 に、これを図示する。

図中、grid 2 は grid 1 からの遷移に $p(w_a|start)$ の単語 bigram の確率を、grid 3 は grid 1 からの遷移に $p(w_a|start)$ を、grid 6 は grid 5 からの遷移に $p(w_a|w_a)$ を、grid 7 は grid 5 からの遷移に $p(w_a|w_a)$ をかける。

この方法は、音響モデルで単語が認識されてから言語モデルが駆動される形で、言語モデルを 1 単語遅らせて計算するのと同等である。このため計算量が削減される。ただし、認識率は低下する。

ただし、この方法は本論文では使用していない。

2.3.8 Viterbi サーチにおける N-best サーチ

通常 Viterbi サーチ (one-pass DP) では、尤度が最大の 1 つの候補しか出力できない。しかし、最大累積尤度（グリッド）$G_t(w_0, i)$ を N 個用意することにより、1 回の forward サーチで N-best の単語列が出力できる [2]。図 2.10 に、単語 bigram を使用したときの例を示した。この図では語彙は (A,B,C,D) の 4 単語とし、 4-best の場合を示している。

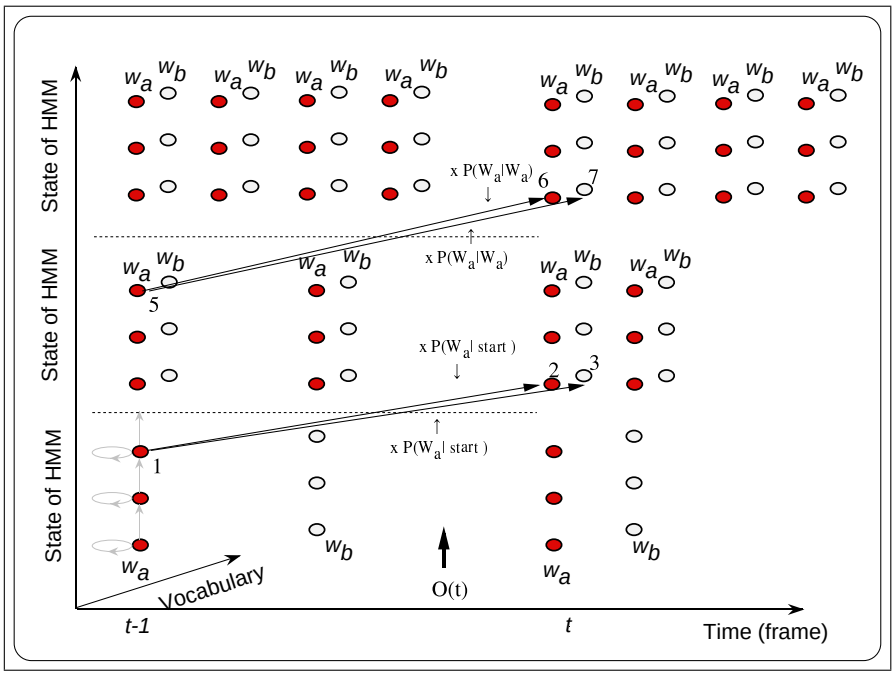


図 2.9: 遅延言語処理したときの tree-trellis サーチ

2.3.9 Viterbi サーチの経路計算

Viterbi サーチにおいて最尤の単語列の結果を得るアルゴリズムとして、2 つの方法が考えられる。

1. 最大累積尤度の計算終了後にトレースバック

   各時刻・各状態において、最大累積尤度を計算したときに、選択した経路を記憶しておく。そして尤度の計算が終了した後、トレースバックを行ない最尤の単語列を得る [40]。この方法は、各時刻・各状態において、選択した経路を記憶するために O(認識語彙数 $\times$ 音声データのフレーム数) のメモリ量が必要である。

2. 最大累積尤度と同時に計算

   各時刻・各状態において、最大累積尤度の計算と同時に、選択した経路を次の状態に渡す。図 2.11）に単語 bigram を使用した場合の例を示す。単語 trigram を使用したときもほぼ同様なアルゴリズムになる。このアルゴリズムにおいて必要なメモリ量は O (認識語彙数 $\times$ 文の単語数) である。ただし、この方法は、経路をコピーする必要があるため計算量は前の方法と比較すると、若干増加する。

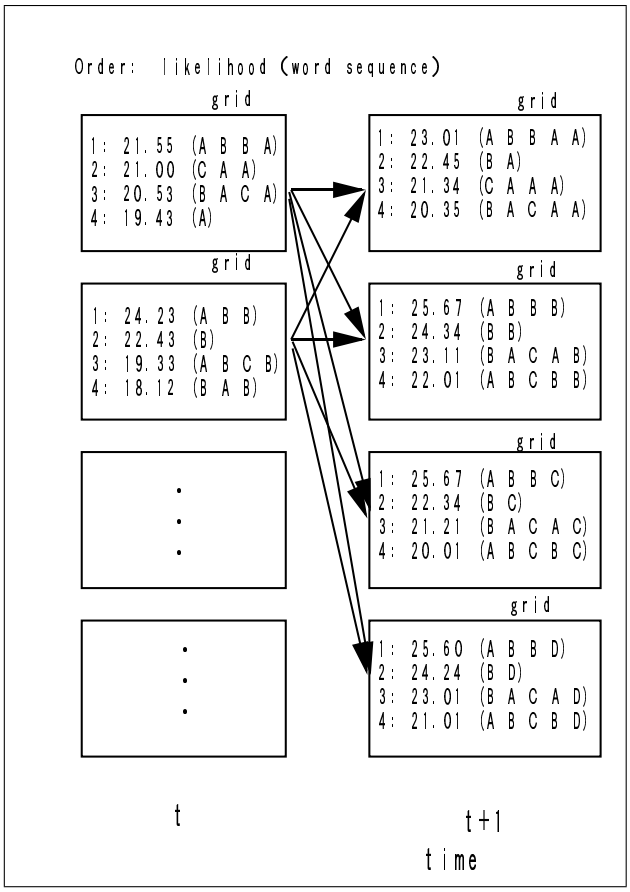


図 2.10: N-best の計算方法

前者は、計算量が少なくて済むため広く利用されている。後者は、前者と比較すると計算量は若干増加するが、多くの場合、文の単語数は音声データのフレーム数より少ないためメモリ量が削減できる。なお、このアルゴリズムは各時刻・各状態（$G_t(w_0, i)$）においてトレースバックをしなくても累積尤度が最大の単語列を知ることができる。

2.3.10 単語の trigram を使用したときの Viterbi サーチ

Viterbi サーチ (one-pass DP) は各認識単語の最後の状態と単語の最初の状態の遷移において trigram の確率を掛けることによって音響モデルと言語の trigram モデルが簡単に結合できる。ただし、 trigram は２つ前の単語が決定されて初めて現在の単語の出現確率が計算できるため Viterbi サーチのグリッドは、現在の単語と１つ前の単語の最大累積尤度を、つねに保持する必要がある。そのため bigram と比較すると、必要なメモリ量が大幅に増加す

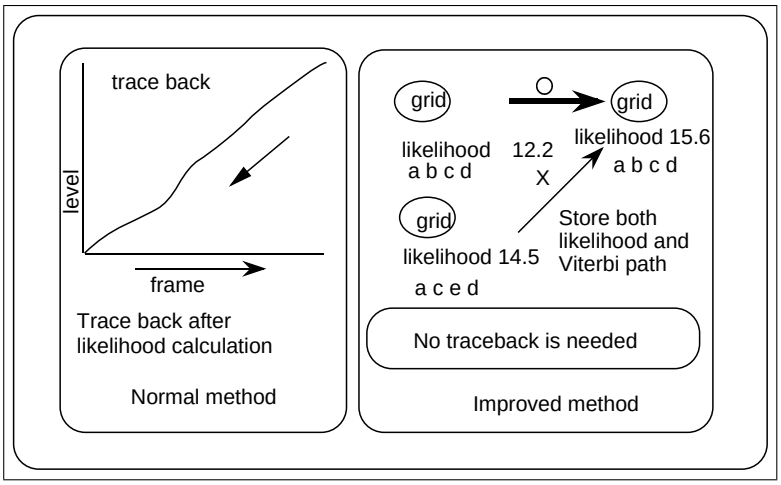


図 2.11: Viterbi サーチの経路計算方法

る。認識単位を単語とした場合のアルゴリズムを表 2.4 に示す。

図 2.12 に、このアルゴリズムの簡略図を示す。この図では、認識語彙数を W_1 と W_2 の 2 単語で、単語の HMM は 4-state 3-loop で、状態は 1 から 3 までとする。縦軸は HMM の状態で、横軸は時間で、奥行きは語彙を示している。

①は時間 $t-1$ において現在の語が w_2 で前の語が w_2 で状態が 0、②は時間 $t-1$ において現在の語が w_2 で前の語が w_2 で状態が 1、③は時間 t において現在の語が w_2 で前の語が w_2 で状態が 1、④は時間 $t-1$ において現在の語が w_2 で前の語が w_2 で状態が 2、⑤は時間 $t-1$ において現在の語が w_2 で前の語が w_1 で状態が 2、⑥は時間 $t-1$ において現在の語が w_1 で前の語が w_2 で状態が 0、⑦は時間 t において現在の語が w_1 で前の語が w_2 で状態が 0 までの最大累積尤度であるとする。

単語の最初の状態以外は、前時刻の同一状態と前時刻の１つ前の最大累積尤度の２遷移のうち、最大累積尤度の高い方を選択する。例えば、③は①の遷移と②から遷移の最大累積尤度の高い方を選択する。しかし、単語の最初の状態は、前時刻の最初の同一の最大累積尤度と各認識単語の最後の最大累積尤度に現在の単語に遷移する trigram の連鎖確率値を掛けたものから遷移の最大累積尤度の高い方を選択する。例えば、⑦は④に trigram の値 $(p(w_1|w_2,w_2)^\alpha)$ を掛けたものと⑤に trigram の値 $(p(w_1|w_1,w_2)^\alpha)$ と⑥の遷移の尤度の高い方を選択する。これを全状態に対して計算を行なう。

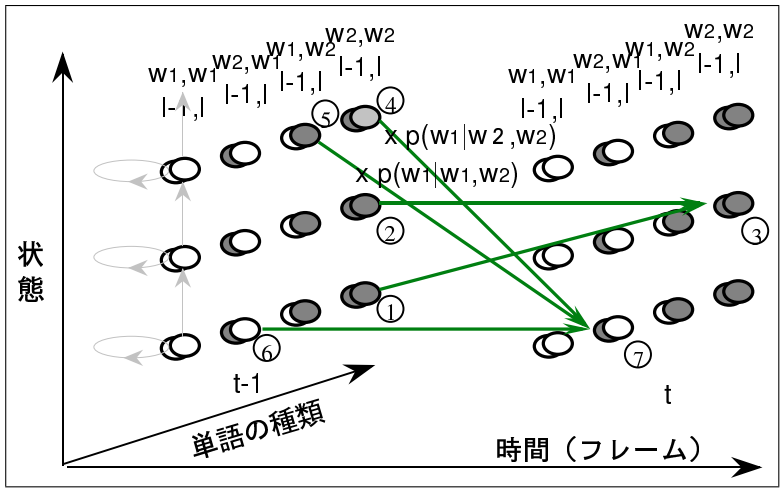


図 2.12: 単語の trigram を利用したときの Viterbi サーチ

2.4 まとめ

本章では、HMM の基本アルゴリズムと Baum-Welch アルゴリズムについて述べた。Baum-Welch アルゴリズムは、学習データの尤度を最大にするようにパラメータを学習する。ただし、基本的には gradient 学習によるパラメータ収束の学習方法であるため、local maxmum の方向にしか学習は進まない。そのため初期値が重要になる。

次に連続音声認識アルゴリズムとしてフレーム同期型の tree-trellis サーチと Viterbi サーチについて述べた。実際使用されている認識アルゴリズムは Viterbi サーチが多いと思われる。しかし、尤も基本的な tree-trellis サーチにも N-best 候補が出せるなどの長所がある。

最後に実用的な音声認識のためのアルゴリズムについて報告した。実際の tree-trellis サーチや Viterbi サーチのプログラムをインプリメントしてもメモリー量や計算量の関係から動かないことから実用的ではない。そこで、ビームサーチなどの様々な手法が使用される。本論文では、ビームの絞り方や、近接したフレームにおける言語モデルの類似性の利用、単語 trigram の値の検索方法などについて述べたが、この他にも様々な方法が提案されている [32]。

表 2.1: 連続単語認識における tree-trellis サーチのアルゴリズム

［定義］

l_w：単語 w における状態数
a_{ij}^w：単語 w における状態 s_i から状態 s_j への遷移確率
$b_j^w(v)$：単語 w の状態 s_j におけるベクトル v の出力確率
Q：語彙数
T：入力フレーム数
O_t：フレーム t における観測ベクトル
$G_t(w_n,..,w_0,i)$：単語 w_n から単語 w_0 までの状態 i での
フレーム t までの累積尤度

［初期化］

$w_0 = 0,...,Q-1$ において step1 を実行
1) $w_n = 0,...,Q-1$ において step3 を実行
.
.
.
2) $w_0 = 0,...,Q-1$ において step3 を実行
3) $G_0(w_n,...,w_0,0) = 0.0$

［ 単語内での計算］

$t = 0,1,..,T-1$ において step4 ∼ step8 を実行
4) $w_n = 0,...,Q-1$ において step7 を実行
.
.
.
5) $w_0 = 0,...,Q-1$ において step7 を実行
6) $i = 0,1,...,l_{w_0}-1$ において step7 を実行
7) $G_t(w_n,...,w_0,i) =$
$\Sigma(G_{t-1}(w_n,...,w_0,i) \times a_{i,i}^{w_0} \times b_i^{w_0}(O_t) +$
$G_{t-1}(w_n,...,w_0,i-1) \times a_{i-1,i}^{w_0} \times b_{i-1}^{w_0}(O_t))$

［単語境界の計算］

8) $w_n = 0,1,...,Q-1$ において step10 を実行
.
.
.
9) $w_0 = 0,1,...,Q-1$ において step10 を実行
10) $G_t(w_n,...,w_0,0) = \Sigma_{w_0}(G_{t-1}(w_n,w_{n-1},...,w_0,l_{w_0}-2)$
$\times a_{l_{w_0}-2,l_{w_0}-1}^{w_0} \times b_{l_{w_0}-1}^{w_0}(O_t) + G_t(w_n,...,w_0,0))$

表 2.2: Viterbi サーチのアルゴリズム

［定義］
l_w：単語 w における状態数 a_{ij}^w：単語 w における状態 s_i から状態 s_j への遷移確率 $b_j^w(v)$：単語 w の状態 s_j におけるベクトル v の出力確率 Q：語彙数 T：入力フレーム数 $O(t)$：フレーム t における観測ベクトル $G_t(w_0, i)$：単語 w_0, 状態 i での フレーム t までの最大累積尤度
［初期化］
$w_0 = 0, ..., Q-1$ において step1 を実行 1) $G_0(w_0, 0) = 0.0$ $start$ は文頭を意味
［ Viterbi サーチ］
$t = 0, 1, .., T-1$ において step2 $\sim$ step6 を実行 3) $w_0 = 0, ..., Q-1$ において step4 を実行 4) $i = 0, 1, ..., l_{w_0} - 2$ において step5 を実行 5) $G_t(w_0, i) =$ $\max(G_{t-1}(w_0, i) \times a_{i,i}^{w_0} \times b_i^{w_0}(O_t),$ $G_{t-1}(w_0, i-1) \times a_{i-1,i}^{w_0} \times b_{i-1}^{w_0}(O_t))$
［単語境界の計算］
7) $w_0 = 0, 1, ..., Q-1$ において step8 を実行 8) $\Delta = \max_{0 \le w_1 \le Q-1}(G_{t-1}(w_1, l_{w_1} - 2)$ $\times a_{l_{w_1}-2, l_{w_1}-1}^{w_1} \times b_{l_{w_1}-1}^{w_1}(O_t) \times P(w_0 \mid w_2, w_1)^\alpha$ もし $\Delta \ge G_t(w_0, 0)$ ならば $G_t(w_0, 0) = \Delta$

表 2.3: tree-trellis サーチと Viterbi サーチの比較

	tree-trellis サーチ	Viterbi サーチ
計算コスト	大きい	小さい
メモリ	大きい	小さい
グリッドの選択方法	Viterbi & trellis	Viterbi
N-best list	可能	アルゴリズムを改良して可能 ただし近似解になる。
言語モデルとの適合性	Left-Right 型の全ての言語モデルが可能	Left-Right 型の全ての言語モデルが可能。
ビームサーチとの適合性	良好	良好
音素モデルにおける	良好（ただし Trellis 計算では必要としない。）	良好
duration control との適合性	困難（必要ない）	比較的容易
スポッタとしての動作	可能	プログラムを改良すれば可能

表 2.4: 単語の trigram を用いた Viterbi サーチのアルゴリズム

［定義］

l_w：単語 w における状態数
a_{ij}^w：単語 w における状態 s_i から状態 s_j への遷移確率
$b_j^w(v)$：単語 w の状態 s_j におけるベクトル v の出力確率
$P(w_0|w_2, w_1)$ 単語 w_2, w_1 が出現したときに
w_0 に遷移する確率
Q：語彙数
T：入力フレーム数
O_t：フレーム t における観測ベクトル
$G_t(w_1, w_0, i)$：前単語 w_1, 単語 w_0, 状態 i での
フレーム t までの最大累積尤度
α ：音響尤度と言語の連鎖確率の結合値

［初期化］

$w_0 = 0, ..., Q-1$ において step1 を実行
1) $G_0(start, w_0, 0) = P(w_0|start, start)^\alpha$
$start$ は文頭を意味

［ Viterbi サーチ］

$t = 0, 1, .., T-1$ において step2,step6 を実行
2) $w_1 = 0, ..., Q-1$ において step3 を実行
3) $w_0 = 0, ..., Q-1$ において step4 を実行
4) $i = 0, 1, ..., l_{w_0} - 2$ において step5 を実行
5) $G_t(w_1, w_0, i) =$
$\max(G_{t-1}(w_1, w_0, i) \times a_{i,i}^{w_0} \times b_i^{w_0}(O_t),$
$G_{t-1}(w_1, w_0, i-1) \times a_{i-1,i}^{w_0} \times b_{i-1}^{w_0}(O_t))$

［単語境界の計算］

6) $w_1 = 0, 1, ..., Q-1$ において step7 を実行
7) $w_0 = 0, 1, ..., Q-1$ において step8 を実行
8) $\Delta = \max_{0 \le w_2 \le Q-1}(G_{t-1}(w_2, w_1, l_{w_1} - 2)$
$\times a_{l_{w_1}-2, l_{w_1}-1}^{w_1} \times b_{l_{w_1}-1}^{w_1}(O_t) \times P(w_0|w_2, w_1)^\alpha$
もし $\Delta \ge G_t(w_1, w_0, 0)$ ならば $G_t(w_1, w_0, 0) = \Delta$

第3章　日本語のN-gramによるモデル化

本章では言語モデルとしてN-gramを用いた場合の妥当性について考察した。調査項目として、学習データ量の変化に対するモデルのエントロピーとカバー率を調査した。

unigram・bigram・trigram・4-gramのエントロピーは次式によって計算できる。

$$unigram \quad \sum_i \quad p(w_i)\log[p(w_i)] \tag{3.1}$$
$$bigram \quad \sum_{i,j} \quad p(w_i,w_j)\log[p(w_j|w_i)] \tag{3.2}$$
$$trigram \quad \sum_{i,j,k} \quad p(w_i,w_j,w_k)\log[p(w_l|w_i,w_j)] \tag{3.3}$$
$$4-gram \quad \sum_{i,j,k,l} \quad p(w_i,w_j,w_k,w_l)\log[p(w_l|w_i,w_j,w_k)] \tag{3.4}$$

ここで

$p(w_i)$	...	モデルLにおける単語w_iの出現確率
$p(w_j\|w_i)$	...	単語w_iが出現したとき単語w_jに遷移する遷移確率
$p(w_i,w_j)$	...	モデルLにおける単語w_iと単語w_jが同時に出現する出現確率
$p(w_l\|w_i,w_j)$	...	単語w_iと単語w_jが同時に出現したとき 単語w_lに遷移する遷移確率

ただし、文集合モデルLを以下のように定義する。

$$L = \{w_k \mid w_k = w_1 w_2 \ldots w_k\}$$

本節ではエントロピーの他に“カバー率”も求めた。“カバー率”とは次のように定義する。

例えば“カバー率98%”が示す値は、学習データの中で98%をカバーするのに必要な最小のマルコフ連鎖確率の種類の数である。また“カバー率100%”が示す値は、学習データ量全てをカバーするのに必要なマルコフ連鎖の種類の数である。

評価はカバー率96%、カバー率98%、カバー率100%、およびエントロピーの合計4つの値で行なった。

図 3.2 から 3.8 は横軸は学習データ量で、縦軸は出現したマルコフ連鎖確率の種類の数およびエントロピーの値である。また図中における太い実線はカバー率 96%、太い断線はカバー率 98%、細い実線はカバー率 100%、細い断線はエントロピーを示している。また “Entropy” の横に示した値は、全学習データを利用したときのエントロピーの値である。

3.1 新聞記事

標準的な日本語として 1982 年 1 月 4 日から 3 月 30 日までの 74 日分の日経新聞の新聞記事を選んだ。この記事を日本語形態素解析プログラムで形態素解析を行ない、音節と品詞を自動的に付与した。そして、文節単位に区切り、このデータから音節および漢字仮名および品詞のマルコフモデルの収束性を研究した。ただし、過度の複雑さを避けるため、記号・外国語読み・数詞の文字が存在する文は文全体を削除した。データ量は漢字仮名の文字数にして約 170 万文字である。また、使用した日本語形態素解析プログラムの形態素解析の精度は単語認定率で約 95%である [46]。図 3.1 に新聞記事の一部を載せる。

> 大蔵省はことし四月から新銀行法が施行されるのに伴い、在日外銀の営業活動を日本の銀行同様に扱うとの基本方針を決め、これを盛り込んだ政令を二月中にも公布する。おもな内容は（１）企業向け貸し出しに対する大口融資規制を在日外銀にも適用し、五年間の猶予期間を設けるなどの配慮をする（２）利益準備金の積み立てを義務づけ、外銀に対する信頼を高める（３）邦銀の支店を買収することや現地法人化を認める――など。大蔵省はこれによって在日外銀に関する法的根拠が明確になるほか、在日外銀の国内活動がしやすくなり、欧米諸国の間に出始めているわが国の金融制度に対する不満を和らげるのに役立つとみている。（在日外国銀行は「きょうのことば」参照）

図 3.1: 新聞記事の例

3.1.1 新聞記事における音節のマルコフ連鎖確率の収束率

音節の unigram・bigram・trigram・4-gram の学習データ量に対するエントロピーおよびカバー率のグラフを図 3.2 に示す。音節の種類の数は、外来語を除き鼻濁音化したガ行を加え長音を 1 音節として 111 種類である。これらから以下のことがわかる。

1. エントロピーは比較的少ないデータで収束する。

2. カバー率 98%や 96%が収束するのに必要な学習データの量は、エントロピーを収束させるのに必要な学習データの量よりも多くのデータが必要である。

3. カバー率 100%は学習データを増やしても収束する傾向がみられない。これは、学習データを増加させるにともない、全体に占める割合は少ないが、新しい N-gram の組み合わせがたえず出現することを意味している。

4. エントロピーは unigram・bigram・trigram・4-gram になるにしたがい低下する。

3.1.2 新聞記事における漢字仮名文字のマルコフ連鎖確率の収束率

新聞記事における漢字仮名文字の学習文字数に対するエントロピーおよびカバー率のグラフを図 3.3 に示す。なお、使用した漢字仮名の種類は JIS 1 級、約 3000 種類に限定した。これらから以下のことがわかる。

1. 漢字仮名文字の場合、連鎖確率の値を収束させるためには音節の場合よりも大量のデータが必要である。

2. カバー率 98%,96%の収束に必要な学習データの量は、音節と同様にエントロピーの場合よりも多く必要である。

3. 漢字仮名と音節のエントロピーの値を比較すると、unigram と bigram においては、音節のエントロピーの方が低いが、trigram では漢字仮名文字のエントロピーの方が低い。漢字仮名の種類の数は音節の種類の数の約 30 倍もあることを考えると、漢字仮名文字の trigram による言語制約による効果は、音節と比較すると、かなり大きいと思われる。

3.1.3 新聞記事における品詞のマルコフ連鎖確率の収束率

品詞は、名詞・助詞などの機能的な分類の他に地名・人名・色の種類など意味的にも分類されていて、約 450 種類ある。学習データの量の変化に対する品詞のエントロピーおよびカバー率のグラフを図 3.4 に示す。これらから以下のことが示される。

1. 品詞は、音節や漢字仮名と比較すると小量のデータで収束する。

2. 音節や漢字仮名では、unigram,bigram,trigram になるにしたがいエントロピーは半減している。しかし、品詞の場合、unigram のエントロピーの値に対して bigram のエントロピーの値は約半減するが、bigram のエントロピーの値に対して trigram のエントロピーの値は、あまり減少しない。

3.2 X 線 CT 所見作成のデータ

次に日本語の専門的な文章の例として X 線 CT 所見作成の文章を研究した。全単語数は、71198 単語、語彙数は約 3000 語である。ただし、全体の認識性能を向上させるため文節出現率が高いものから上位 100 文節は単語として登録してある。図 3.5 に文の一部を載せる。特徴として以下の点があげられる。

1. 専門的な文章であるため、専門用語が多い。
2. 総語彙数は少ない。
3. 定型文が多い。
4. 文語体の文章である。

3.2.1 X 線 CT 所見作成における音節のマルコフ連鎖確率の収束率

X 線 CT 所見作成の文章は”mass effect”,”large magna”, などの外来語が数多く出現する。そのため音節の種類の数は、新たに “フェ”, “グゥ” などをくわえて 118 種類とした。図 3.6 に学習データ量に対する音節の unigram・bigram・trigram およびエントロピーの値の変化を示す。

図 3.6 から新聞記事と比較すると、X 線 CT 所見作成の文章は unigra,bigram,trigram いずれのエントロピーも低いことや、少ない学習データ量でカバー率が収束していることがわかる。

また、カバー率のデータを見ると、学習データが増加した場合、100%は収束しないが 98%はほぼ収束することがわかる。

3.2.2 X 線 CT 所見作成における漢字仮名のマルコフ連鎖確率の収束率

X 線 CT 所見作成の文章には外来語が多く出現する。ここでは、これらの外来語を全て 1 文字の全角文字として（例えば “mass effect” は MASS

EFFECT）漢字仮名のマルコフ連鎖確率の収束性を求めた。この結果を図 3.7 に示す。

図 3.7 から、新聞記事と比較すると X 線 CT 所見作成の文章はエントロピーは低いことや、少ない学習データ量でカバー率が収束していることがわかる。

これらから X 線 CT の所見作成の文章は新聞記事と比較して文章が単純であると言える。

3.2.3 X 線 CT 所見作成における単語のマルコフ連鎖確率の収束率

X 線 CT 所見作成の文章の語彙数は約 3000 語である。ただし、全体の認識性能を向上させるため文節出現率が高いものから上位 100 文節は単語として登録してあるため、通常、文節と考えられるものまで単語と見なしている（例えば” 脳実質を” は 1 単語）。X 線 CT 所見作成における単語のマルコフ連鎖確率の収束性を図 3.8 に示す。

図 3.8 からエントロピーは、単語のほうが漢字かな（図 3.7）と比較して高いことがわかる。また、カバー率も、単語は漢字かなと比較して、収束するために大量のデータが必要であると思われる。

X 線 CT 所見作成の文章のデータでは、単語の種類の数と漢字かな文字の種類の数は、ほぼ等しい。したがってこの結果は、日本語の単語の曖昧さを示している可能性がある。

3.3 ATR の国際会議のデータベース

3.3.1 ATR の国際会議における単語 trigram の値の収束率

現在ＡＴＲでは、各種言語現象を調査するために対話文を中心とする言語データベースの作成を進めている [10]。本来、対話音声の収録は話者に録音していることを気づかれずに録音することが好ましいが、通信の守秘義務などの問題の他に、話題が次々に移行するため会話の語彙が膨大な数になるという問題も生じる。このため、事前に話題のトピックやバックグラウンドを決め、会議の流れの不自然さを損なわないように打合せを行った後に収録をしている [74]。現在、発話内容で 5 種類、収録環境で 2 種類、話者で 2 種類、発話様式で 2 種類の日本語で種々の組合せを含むデータベースを収集してある [10]。

単語の trigram の値の信頼性を研究するために、この ATR の国際会議の申し込みにおけるテキストデータベースにおいて、データ量に対するエントロピーと “カバー率” の変化を調査した。ATR の国際会議のデータベースは、

約 7000 種類の単語でできている。発話の例文を表 3.1 に示す。

表 3.1: 文例

・［あっ、あえーっと］そちら第１回の通訳電話国際会議の事務局でしょうか。
・はいそうです。
・［えーっとちょっと］その会議のことでねあのー登録のことでお伺いしたいんですが。
・はい。
・どうぞ。
・［えーっと］今手元にあの登録用紙があるんですけれども［えーっと］その中でちょっとあの
・クレジットカードをね［あのー］クレジットカードの名前となんかナンバーを書くところ
・があるんですがはいそうです。［えーっと］それをちょっとクレジットカードを持ってい
・ない者がいるんですけれどもその場合はどうなんでしょうか。
・はい。

調査はカバー率 60%、カバー率 80%、カバー率 100%、およびエントロピーの合計 4 つの値で行なった。この結果を図 3.9 に示す。

図 3.9 から、データ量が増加するに伴いエントロピーは増加していて、安定な値になっていないことがわかる。また語彙の 58.8% (3486/5933)、単語 trigram の種類の数の 77.9% (60847/78138) は１回しか出現していなかった。このデータを X 線 CT 所見と比較すると単語のエントロピーの絶対値では差が少ないことがわかる。したがって、固有名詞など１度しか出現しない単語が多過ぎることを意味していて、データ収集に問題があると考えられる。

3.4 まとめ

ここでは、新聞記事および X 線 CT 所見作成および ATR の国際会議の申し込みの文章において、学習データ量に対する音声・漢字仮名・品詞・単語のマルコフ連鎖確率値の収束率を求めた。これらの結果から、以下のことが示される。

1. エントロピーとカバー率

   エントロピーとカバー率の収束性を比較すると、全てのデータにおいてエントロピーはカバー率よりも少ない学習データ量で収束することが示された。これは学習データ量に対するマルコフ連鎖確率値の変化について研究する場合、エントロピーだけでなく、カバー率も考察する必要があることを意味していると思われる。

2. カバー率

カバー率のデータを見ると、学習データが増加した場合、100%は収束しないが、98%は、ほぼ収束すると思われる。そして学習データが増加した場合、全体に占める割合は少ないが、たえず新しい種類の N-gram の組み合わせが出現していることがわかる。

これは、言語モデルとしてマルコフモデルを選択したときの妥当性に関して、滅多に出現しない言語現象は、あえてモデルに適合させる必要がないと判断すべきであると考えられる。

3. 新聞記事と X 線 CT 所見作成の比較

   X 線 CT 所見作成の文章と新聞記事を比較すると、音節・漢字仮名、いずれの場合もエントロピーが低く、かつ少ない学習データ量で収束している。これらから X 線 CT の所見作成の文章は新聞記事と比較して文章が単純であると言える。

4. 形態素解析プログラムの精度

   新聞記事におけるマルコフ連鎖確率の収束性を研究するために使用した形態素解析プログラムは単語認定率で約 95%の精度しかないため、人手によって文節単位に区切られた場合のマルコフ連鎖確率の値と、ここで得られた値に差がある可能性がある。特に品詞に関しては、trigram の有効性が見られなかった。これは、品詞の定義が人によって異なる（例えば形容動詞）などの問題点もあるが、形態素解析の精度の問題と関連している可能性があり、今後検討が必要である。

5. 日本語における単語の精度

   X 線 CT 所見作成の文章では、漢字かなと単語の種類の数がほぼ同じにも関わらず、単語のほうがエントロピーは高く、かつカバー率の収束性も低かった。また、ATR の国際会議における単語の trigram の収束性は非常に悪かった。これらの原因は、日本語では単語の境界が曖昧であることに起因している可能性がある。したがって、日本語において使用される言語モデルとしては、単語の trigram より漢字かなの trigram のほうが妥当であるかもしれない。また、日本語における単語の意味を今後考慮する必要がある。

6. ATR の国際会議における単語 trigram の値の信頼性

   図 3.9 から、データ量が増加するに伴いエントロピーは増加していて、安定な値になっていない。したがって信頼性のある N-gram の値を得るにはデータ量は少ないことがわかる。

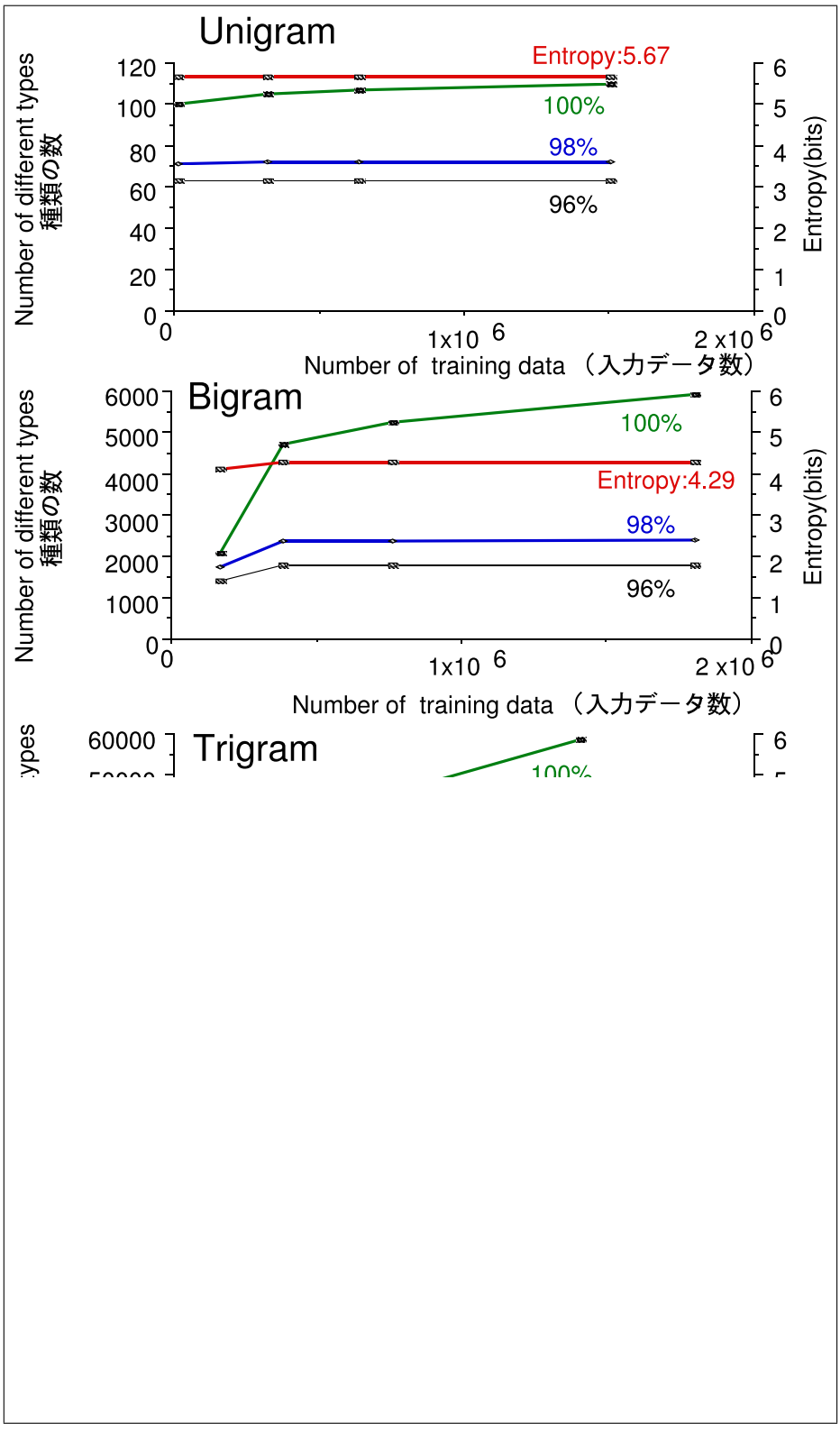


図 3.2: 新聞記事における学習データ数に対する音節のマルコフ連鎖確率値のカバー率およびエントロピー

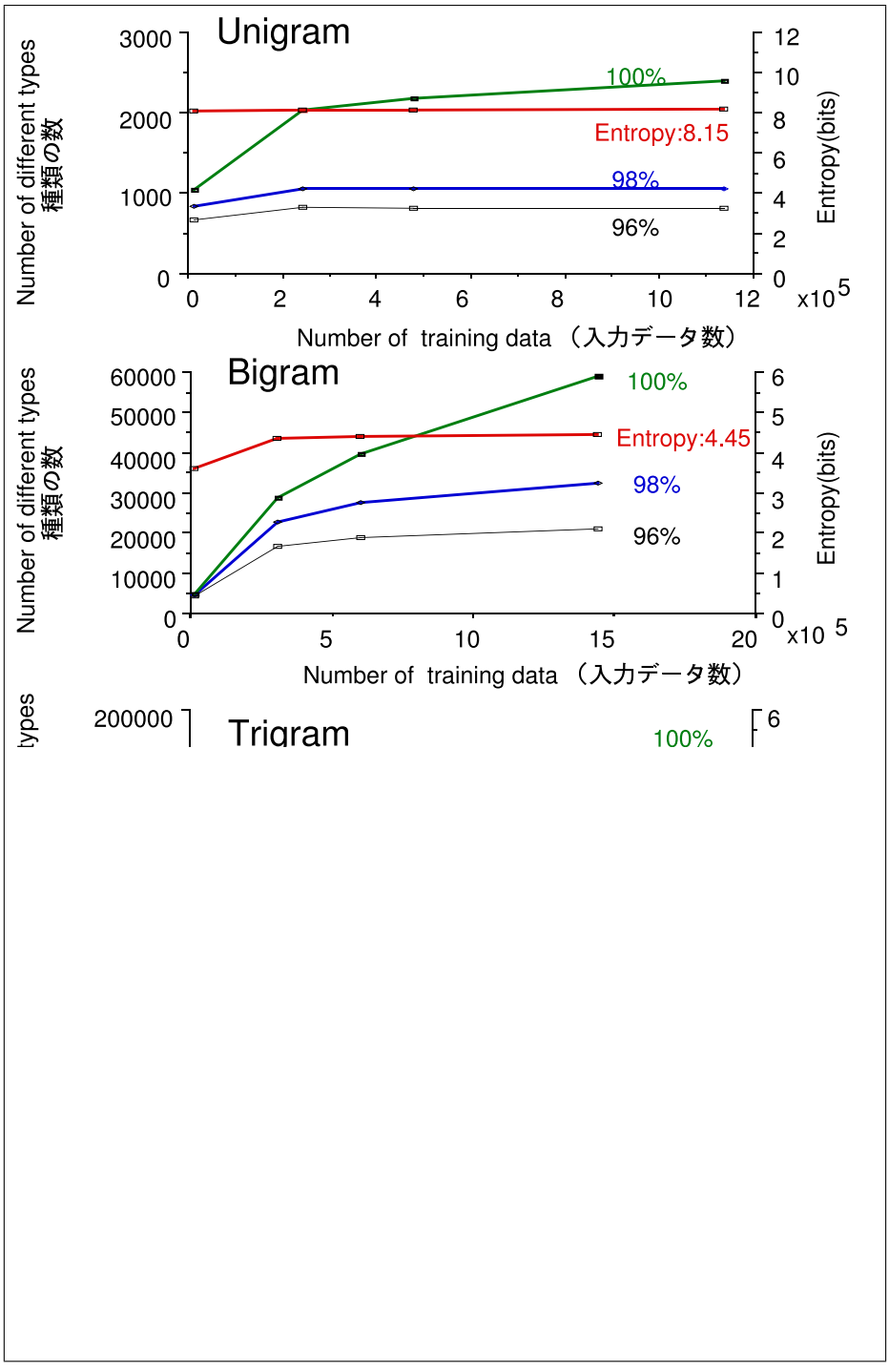


図 3.3: 新聞記事における学習データ数に対する漢字仮名のマルコフ連鎖確率値のカバー率およびエントロピー

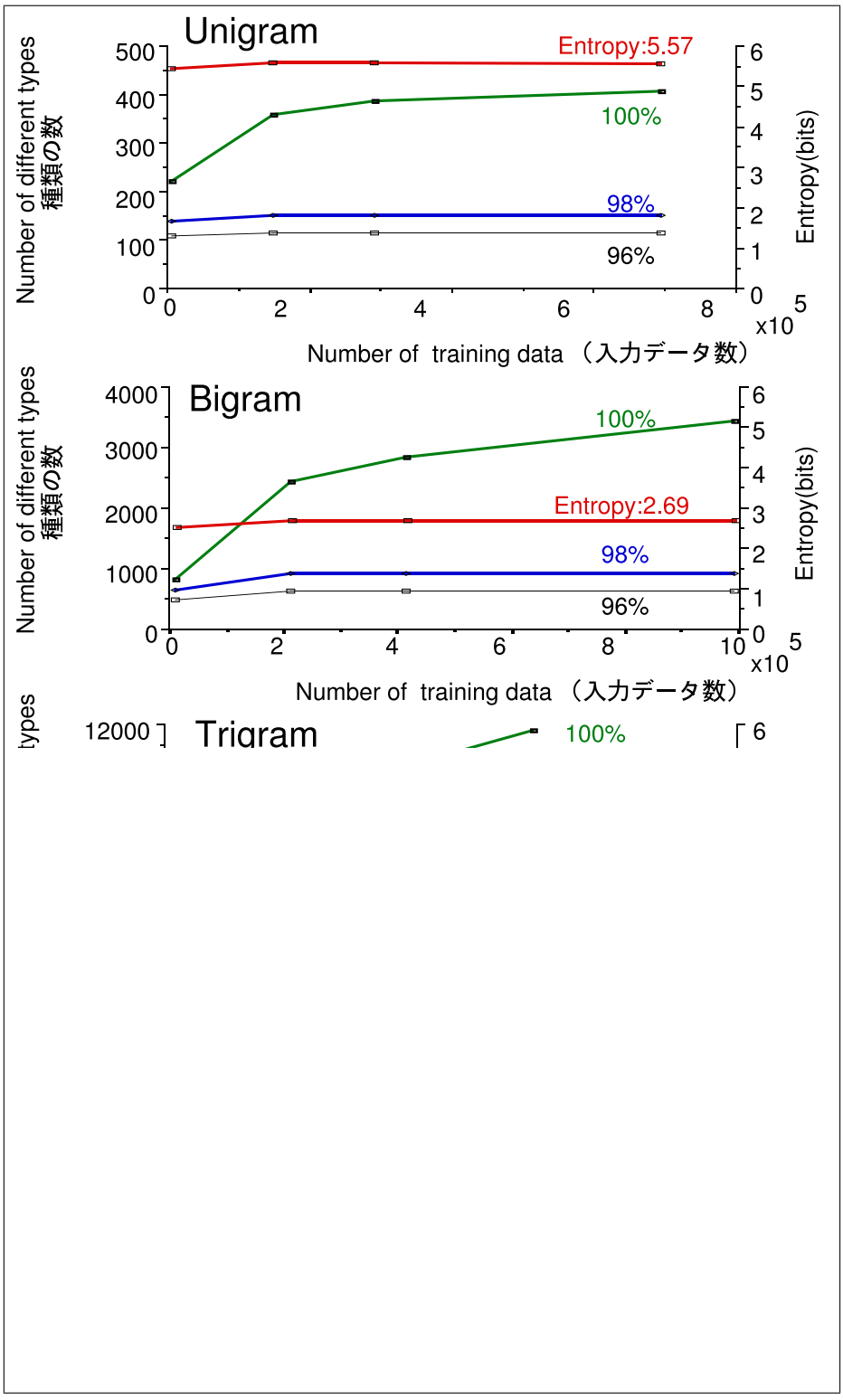


図 3.4: 新聞記事における学習データ数に対する品詞のマルコフ連鎖確率値のカバー率およびエントロピー

頭部ＣＴ 単純および造影
１、３月１３日のＣＴと比較した。
２、スライスのレベルが若干異なっているので正確な比較はできないが、鞍上槽の正中からやや右上方へ向かって進展している増強効果を示す腫瘤の大きさは本質的に変わっていない。ただし前回のＣＴでこの結節性腫瘤の右前方に見られた嚢胞性の成分については今回は描出されていない。３、側脳室の大きさ形も前回と同様である。

impression.....
鞍上槽の頭蓋咽頭腫の残存については明らかな変化はないが、右後方に見られた嚢胞性成分が消失しているかもしれない。

図 3.5: X 線 CT 所見作成の例

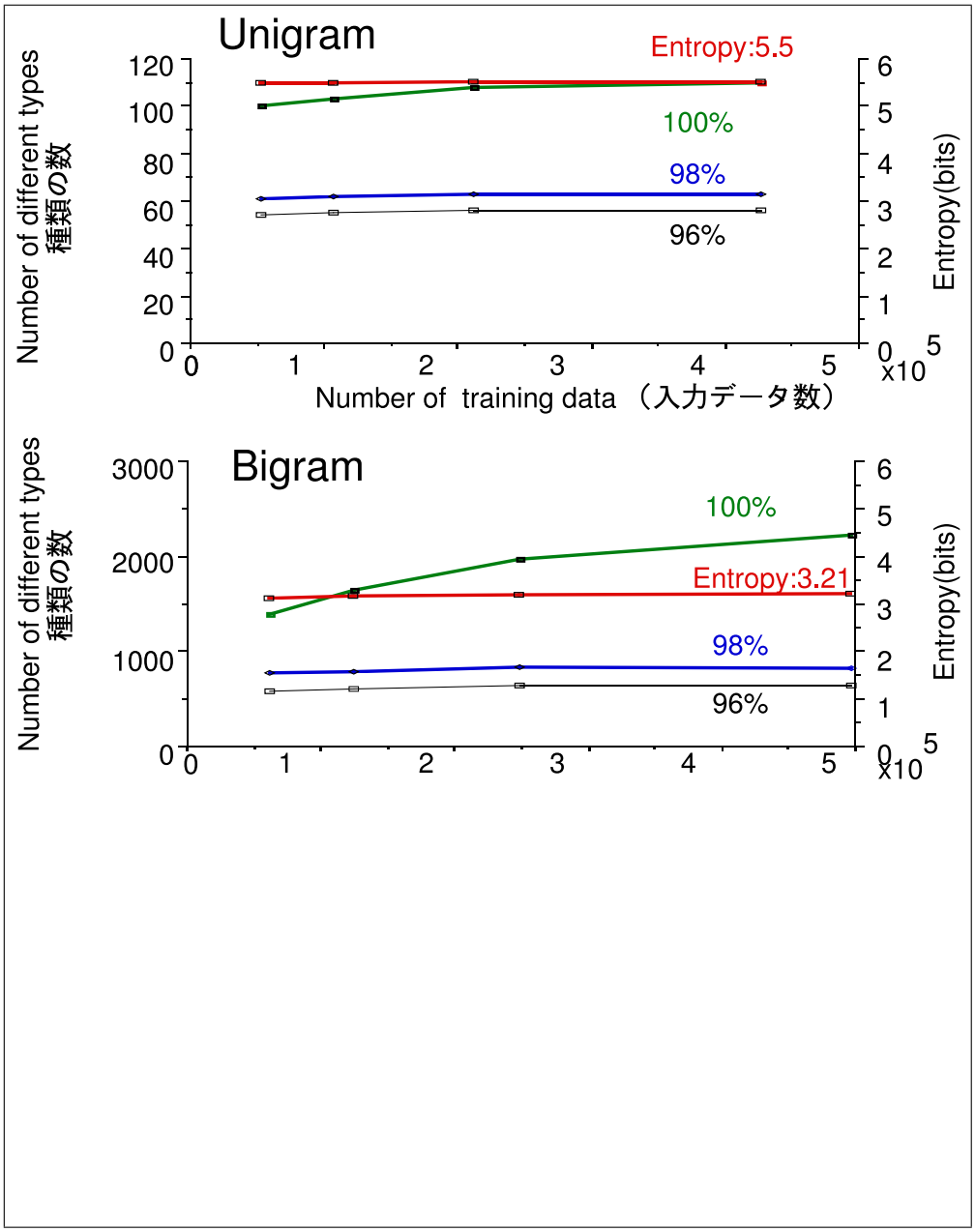


図 3.6: X 線 CT 所見における学習データ数に対する音節のマルコフ連鎖確率値のカバー率およびエントロピー

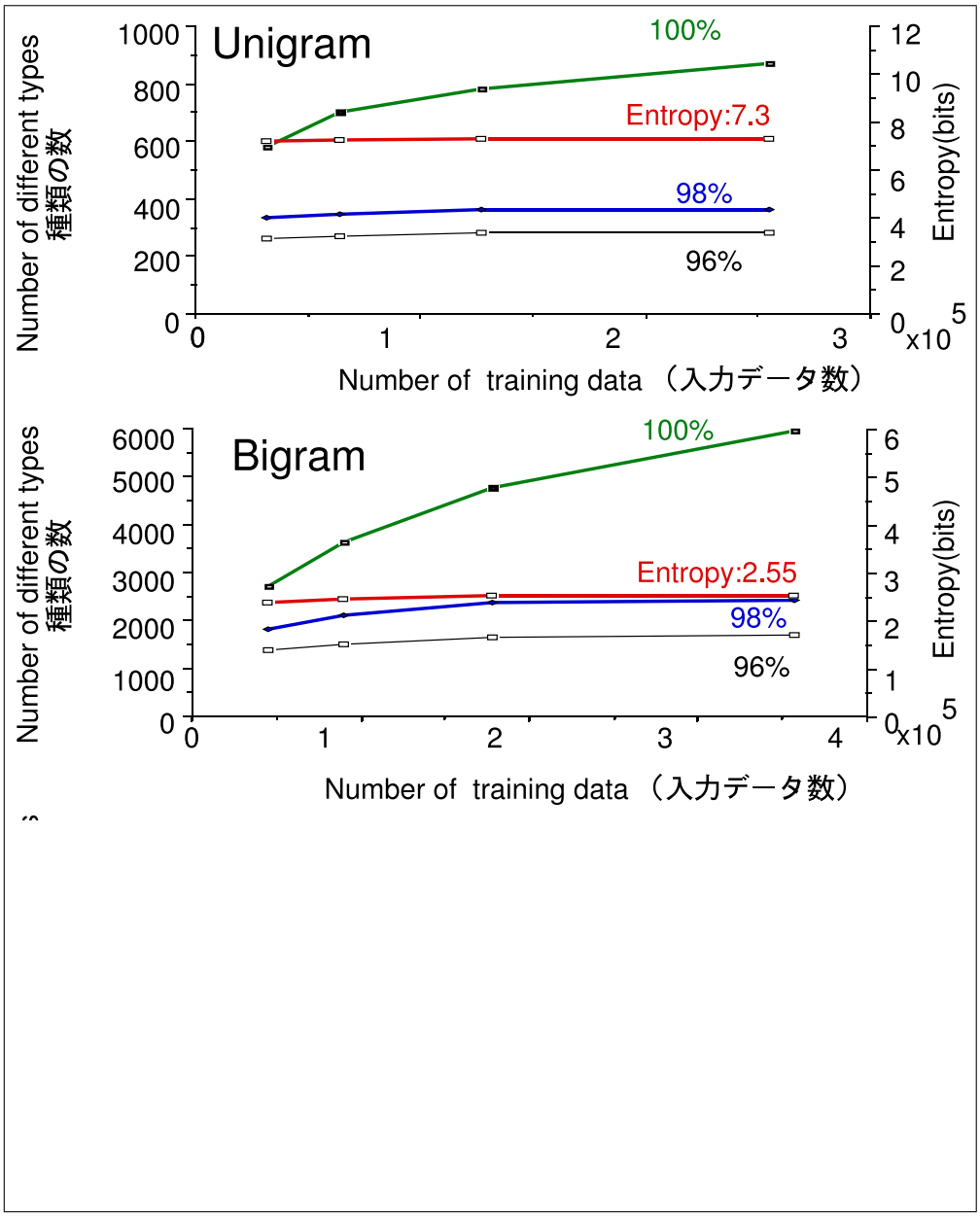


図 3.7: X 線 CT 所見における学習データ数に対する漢字仮名のマルコフ連鎖確率値のカバー率およびエントロピー

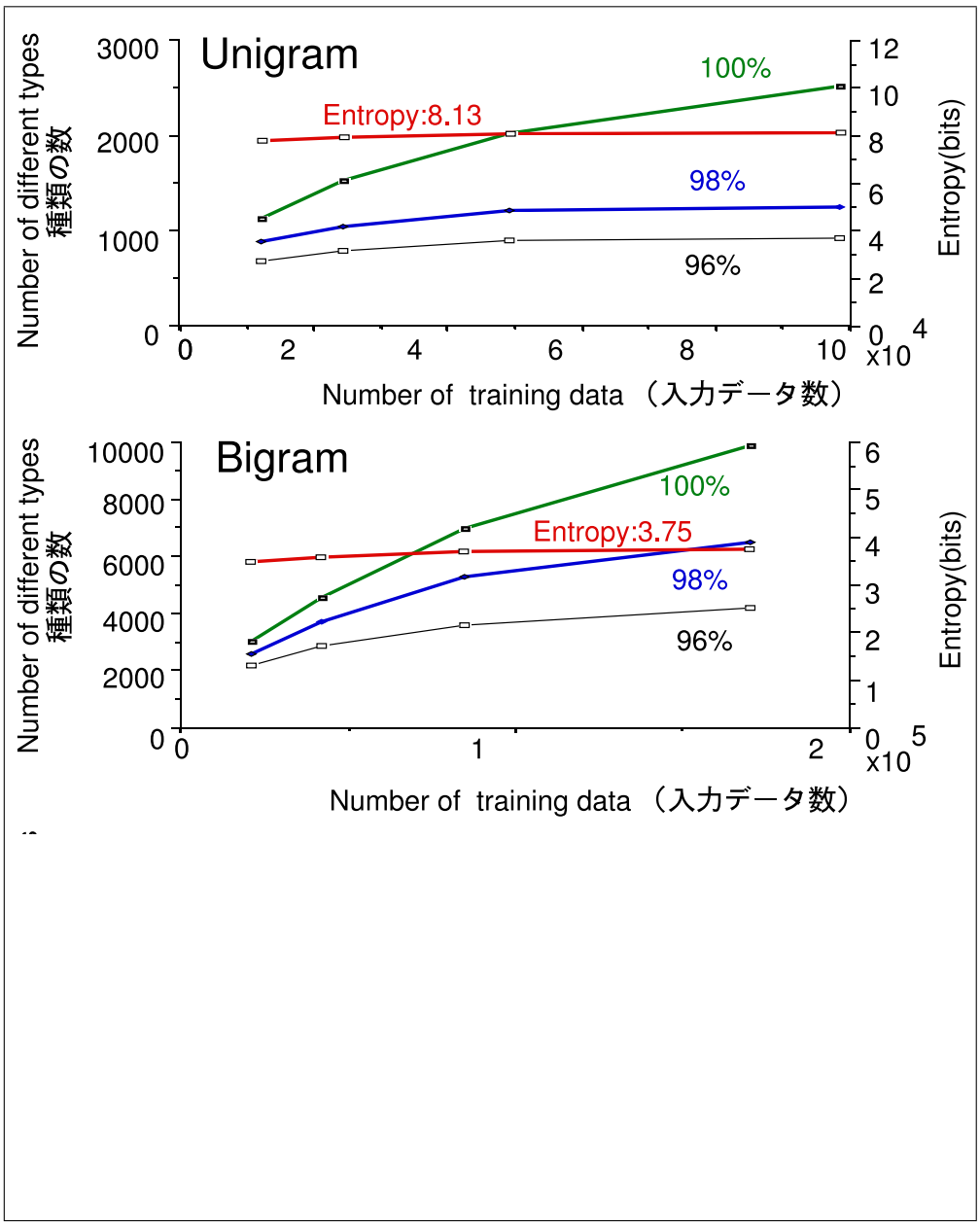


図 3.8: X 線 CT 所見における学習データ数に対する単語のマルコフ連鎖確率値のカバー率およびエントロピー

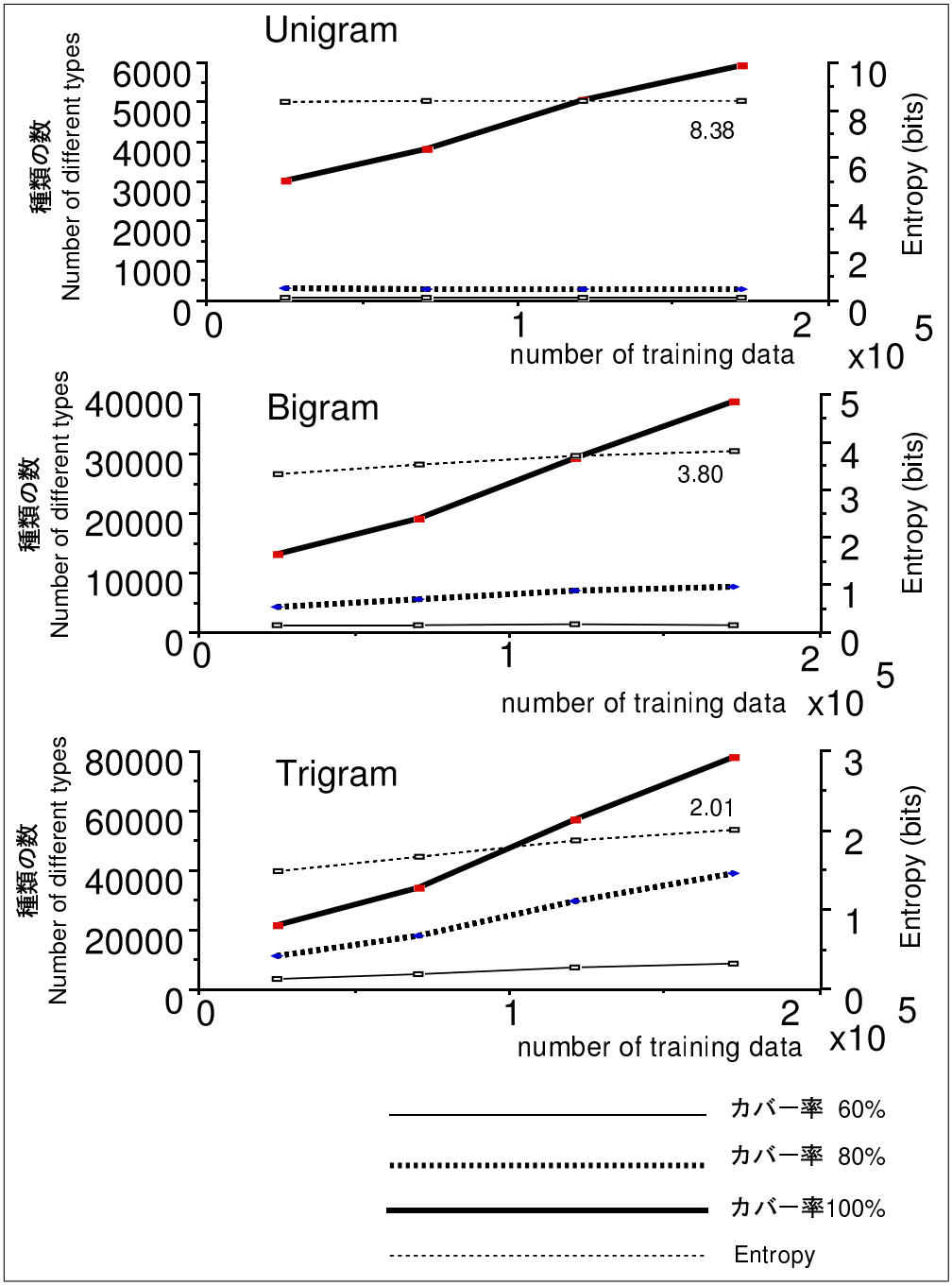


図 3.9: ＡＴＲの国際会議のデータベースにおける、学習データの入力データに対するエントロピーおよびカバー率の変化

第4章　N-gram を用いた音声認識

言語の N-gram モデルは、非常に簡単なモデルで、例えば bigram は直前の単語に対して現在の単語が接続する確率である。また trigram は、２つ前の単語と直前の単語に対して現在の単語が接続する確率である。しかし、音声認識の認識性能向上において非常に有効なモデルであることが知られている。N-gram モデルを音声認識の言語モデルとして使用し有効性を確かめた論文としては IBM の研究 [7] が有名である。

現在 N-gram モデルは、英文音声認識に使用する言語モデルの主流になっている。しかし、日本語において音声認識に N-gram を使用し有効性を確かめた論文は少ない [45]。この原因の１つに、日本語の大量のテキストデータベースの欠如にあると思われる。trigram の値を精度よく求めるためには、基本的には大量のテキストデータ量が必要である。英語ではデータベースの重要性が認識されていて古くから Brown corpus や AP corpus などがある。これらのデータベースは形態素解析などの研究のために使用されている。しかし日本語ではコンピュータに読み込める形式で利用できる大量のデータベースが最近まで存在していなかった。そのため、確率的な言語モデルの研究は最近まであまり報告されていなった。しかし、この状況も新聞記事が CD-ROM で提供されるようになり、国際電気通信基礎技術研究所（ATR）が各種対話データを販売する [10] など、状況が変化し始めている。

そこで本章では音声認識のための言語モデルとしての N-gram の有効性について研究した。

4.1　trigram の有効性について

日本文音声入力においては、音声の持つ物理的特性に着目した音声認識装置の限界を克服するため、日本語の文法や意味を用いた自然言語処理を併用することの必要性が指摘されている [91]。特に大語彙を対象とする音声には発音の個人差や曖昧さの他に、同音異義語なども多数含まれるため、その認識においては音声の物理的特性が完全に生かされたとしても、なお絞り切れない曖昧さが残り、元の文を推定するには、言語解析や意味理解の技術が必要と考えられる。

音響処理と自然言語処理を融合させた、日本文音声入力の一つの方法として、文節単位の音節マトリックスをインターフェースに用いて、音声認識装

置と自然言語処理を連携させる方法 [43] が考えられている。すなわち、音声認識装置が音声の物理的特性を解析して、文節単位に各音節候補をマトリックス形式で出力し、自然言語処理はそのマトリックスを入力として、正しい漢字かな混じりの文節候補を推定する方法である。 この場合の言語処理の方法としては、二つの方法が考えられる。その一つは、音節マトリックスに言語の文法情報や意味情報を直接適用して、正しい文節を推定しようとするもの [86] であり、もう一つは、音節や文字、単語の統計的な連鎖情報を適用して文節候補を絞り込む方法 [72] である。

前者は文法、意味情報を直接適用して文節を生成する点に特徴がある。しかし、単語ごとの文法情報と意味情報の付与ではなく、単語の代わりに単語の文法的カテゴリーや意味的カテゴリーが使用されるため、絞り込みの精度はこれらのカテゴリーの分解能に依存し、複数の単語候補が同一のカテゴリーに属するような大語彙の認識では、文節候補を絞り込むのは困難である [86]。一方、後者の方法で、筆者らは、大語彙の認識において、音節の trigram モデルが有効で、その適用により、文節単位の音節マトリックスから、第一位で約 70%、第 10 位までの累積正解率で約 95%の高い精度が得られることを報告した [3]。しかし、漢字かなの文節候補を生成するにはさらに膨大な曖昧性を絞り込むことが必要であった。ところで、漢字かな混じりの文の誤字、脱字等に漢字かなのマルコフモデル (N-gram モデル) が効果的であること [20] が知られている。

そこで本章では、音節マトリックスから文節候補を生成するための方法として、音節の trigram モデルのほかに漢字かなの trigram モデルおよび単語辞書を使用した。そして、これらを組み合わせた 2 種類の曖昧性絞り込みの方法を提案し、その効果を実験的に示した。

4.1.1 実験システムの構成

4.1.1.1 日本文音声認識の処理手順

日本文の音節や文字連鎖の持つ情報量を応用した「文節処理」の効果を研究するため、音声の持つ物理的特性に着目した音声認識処理と、それ以外の言語処理的な部分とを分け、図 4.1 に示すような日本文音声認識手順を考える。日本文の音声入力のマンマシーンインターフェースとしては、単音節単位、文節単位および連続音声の入力などが考えられるが、ここでは音声認識装置は一音節単位ごとに区切って発声したものを認識した複数の音節候補を文節の単位で出力、つまり文節単位の音節マトリックスの形態で出力するものとする（図 4.2 参照）。

「文節処理」では、音声認識装置から出力された音節マトリックスから単語候補を生成し、その中で適切と見られる漢字かな混じりの文節候補を、その数を限定して出力する。最後に「文処理」において文節候補を文単位に結合

して得られる最も適切な文節の組を入力文に対する認識結果として出力する。

以下では、以上の日本文認識手順の中の「文節処理」において、日本文の音節や漢字かなの trigram モデルを用いた認識候補絞り込みの方法を提案し、その効果を実験的に示す。

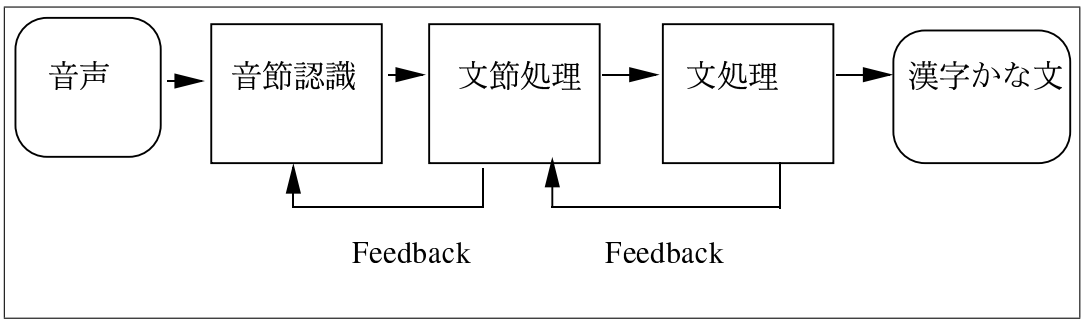


図 4.1: 仮想的な音声認識システムのフローチャート

4.1.1.2 文節処理の方法

「音声認識処理」と「文節処理」は図 4.2 に示すような文節単位の音節マトリックスで結合されるものとし、「文節処理」の結果としては、文節毎の漢字かな混じり文を出力する。

大語彙を対象とする漢字かな混じり文の生成では、同音異義語が多数存在し、同一のかな列に対して複数の漢字が対応するため、音節列の場合 [3] に比べて曖昧さが桁違いに大きく、通常、数億個以上の候補が出力される。従って、「文節処理」の課題は、このような膨大な文節候補の中から、正解を含む少数の文節候補を選択することである。 以下では、このような「文節処理」の方法として図 4.3 に示す二つの方法を考える。

1. 音節選出型文節処理方式

   入力された文節単位の音節マトリックスから、次節で述べる 3 段階の処理を経て、文節候補を生成する。すなわち、まず初めに、音節マトリックスに対して日本語の持つ音節の N-gram モデルを適用して音節の組み合わせ候補を絞り込む。次に、その結果に対して単語辞書を適用して文節を構成する単語候補を生成する。最後に、漢字かなの N-gram モデルを使用して、文節を出力する。

2. 直接選出型文節処理方式

   上記の方法が、はじめに音節連鎖情報を使用するのに対して、この方法は、音節マトリックスに直接単語辞書を適用するもので、2 段階で文節

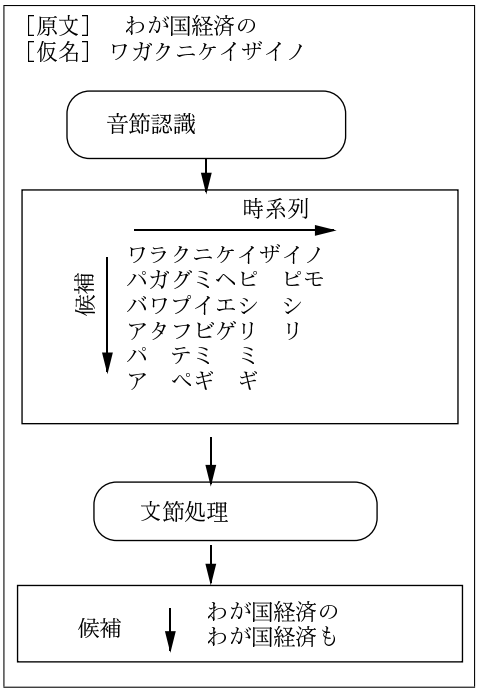


図 4.2: 文節処理の出力例

候補を生成する。はじめに単語認定においては音節マトリックス内の音節候補を組み合わせながら辞書引きを行い、単語として解釈可能な候補の組み合わせをすべて抽出する。次に漢字かなの N-gram モデルを使用して文節候補を生成する。

4.1.2 文節候補生成アルゴリズム

4.1.2.1 音節選出型文節処理のアルゴリズム

音節選出型文節処理方式における入出力データの流れを図 4.4 に示す。

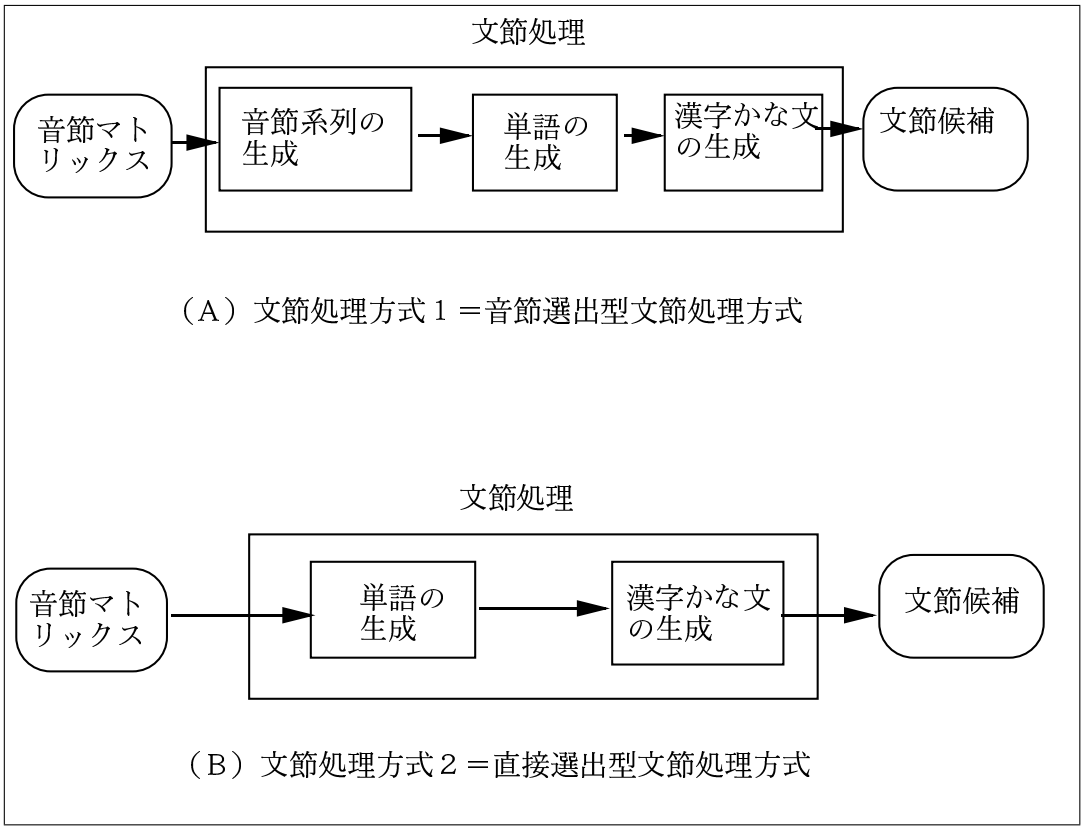


図 4.3: 文節処理の２つの方式

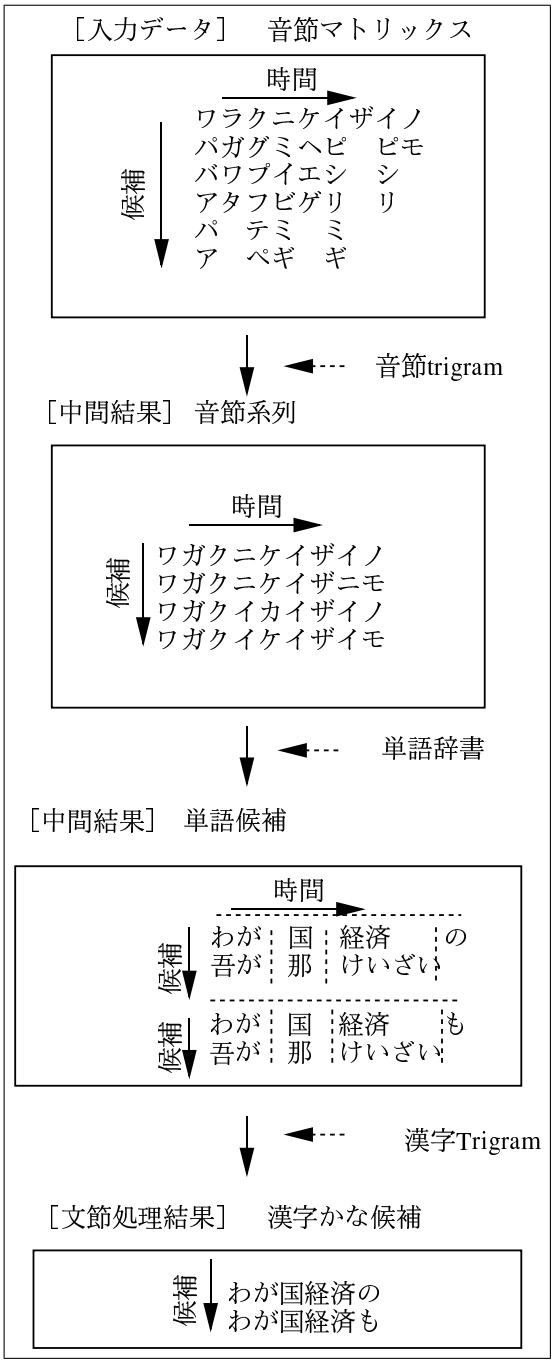


図 4.4: 音節選出型文節処理方式における入出力データ

1. 音節列候補の生成アルゴリズム

   音節マトリックスから音節文節候補を生成する方法として、trigram モデルを用いる。マルコフモデルによる候補絞り込みは、正しい文節候補は間違った候補よりもマルコフ連鎖値の積が大きいと仮定して、文節候補を評価する。例えば図 4.4 の例で、「ワラクニケイザイノ」の文節候補の尤度は

   $p($ ＿＿ワラクニケイザイノ＿＿ $) = p($ ワ／＿＿ $) \times p($ ラ／＿ワ $) \times p($ ク／ワラ $) \times p($ ニ／ラク $) \times p($ ケ／クニ $) \times p($ イ／ニケ $) \times p($ ザ／ケイ $) \times p($ イ／イザ $) \times p($ ノ／ザイ $) \times p($ ＿／イノ $) \times p($ ＿／ノ $)$.

   （ただし＿は空白を意味。）で与えられる。これを他の音節の組み合わせを含む 165,888 通りのすべてについて計算し、上位何候補かに絞り込む。この場合は第１位の候補として「ワガクニケイザイノ」が得られ、第２位としては「ワガクニケイザイニモ」が得られる。一般に、音節マトリックスを対象に直接この計算を行うのは計算量の点で困難であるが、Veterbi のアルゴリズムを使用することにより、少ない計算量で容易に評価することができる。

2. 単語認定アルゴリズム

   前項で得られた複数の音節列の上位 8 位までの音節列に対して、単語辞書を参照し、当てはまる単語候補を出力する。このプロセスはワードプロセッサのかな漢字変換と基本的に同じである。ここでは分割数最小法 [59] を基本とするが、正解候補のもれを防止するするため、最小分割数+1 までの単語候補を生成する。

3. 文節候補認定アルゴリズム

   最後に上記で得られた単語候補に対して漢字かなの trigram を使用して曖昧性を絞り込む。なお実験では同時に品詞の trigram を使用して、品詞における文節候補の絞り込みの効果も研究した。

4.1.2.2 直接選出型文節処理のアルゴリズム

直接選出型文節処理方式における入出力データの流れを図 4.5 に示す。

1. 単語認定アルゴリズム

   文節単位の音節マトリックスに以下の方法で直接単語辞書を適用し、可能な単語候補をすべて抽出する。まず音節マトリックスの音節候補をつなぎ合わせた音節列の中に文節を一単語として解釈できる単語候補があるかどうかを単語辞書を使って調べる。図 4.5 の例では、9 音節を一単語と考え、各音節を組み合わせた単語の有無を調べる。すなわち、

$4 \times 6 \times 4 \times 4 \times 6 \times 6 \times 1 \times 6 \times 2 = 27648$ 通りの音節の組み合わせに対して、9 音節の全てが一致するような単語が辞書に存在するか否かを調べ、存在すればすべて抽出する。もしそのような単語が存在しなければ音節マトリックスを二つに分割する。図 4.6 の例ではそのような単語候補はないので、長さを 1 音節短くして、下記（実線）のようにマトリックスを二つに分割する。すなわち 8 音節長と 1 音節の 2 つに分割する。

第 1 ブロック、第 2 ブロックの双方に対して前と同様の方法で単語辞書引きを行い、辞書上の単語の有無を調べる。何れかのブロックに対して単語が存在しないときは分割が不適切と考え、第 1、第 2 のブロックの分割の仕方を変える。すなわち長さをさらに 1 音節短くし、7 音節長と 2 音節長に分割する（破線）。

分割された二つのブロックの双方に一つ以上の単語候補が存在するような分割の仕方が無いときは、全体を三つのブロックに分割する。全てのブロックに対して一つ以上の単語候補が存在するようになるまで、この手順を繰り返し、辞書上で解釈可能な最小の分割数を求める。

また、このようにして求めた分割数最小の分割法の全てに対して、ブロック毎に辞書上解釈可能な全ての単語候補を出力する。

2. 文節候補認定アルゴリズム

   前項で抽出された単語候補を組み合わせて得られる漢字かな混じりの文節単語列に対して、同様の漢字かなの trigram を適用し、順位付けを行う。なお実験では同時に音節および品詞の trigram を用いて、それぞれの情報の効果を研究した。

4.1.2.3 両アルゴリズムの違いについて

音節選出型文節処理方式と直接選出型文節処理方式のアルゴリズムでは、使用される情報は同じであるが、その適用順序に違いがある。前者は音節の trigram を最初に使用するので、その後、評価対象となる候補数が大幅に減少する。そのため、全体としての計算量が少ないと言う利点があるが、逆に単語辞書の適用の段階で、正しい文節候補が失われている可能性がある。

これに対して、後者のアルゴリズムでは単語辞書を最初に適用するため、多数の単語候補が生成され、後の処理が重くなるが、正しい漢字かな混じり文の文節候補をもたらす可能性は、より高いと予想される。

4.1.3 実験条件

実験条件を以下に示す。

1. マルコフ連鎖値の計算

マルコフ連鎖値の計算には日経新聞記事 74 日分（82 年 1 月 4 日から 3 月 31 日）を使用した。これを日本文解析プログラムを使用して形態素に分割し、同時に音節変換を行った。そして、これを再合成して文節単位のデータを作成し、その後、音節、漢字かな、品詞について unigram,bigram,trigram のマルコフ連鎖値を計算した。

ただし実験を簡単にするため、この記事から、記号、外国語読み、数詞の文字のある文は文全体を削除した。その結果、マルコフ連鎖値の計算に使用した文字数は漢字かな混じり文字で数えて約 170 万文字である。(3.1 節におけるデータと同一)。

なお、新聞記事は、マルコフモデルに必要な、すべての組み合わせを持っていない。そのため、連鎖値が 0.0 となる組合せが出現する。そのような組み合わせに対しては、統計上の最小値を与える方法や次数の少ない連鎖値との補間で代用する方法 [27] などが考えられるが、ここではフロアリングをして、その確率値を exp(-1000.0) とした。

2. 音節マトリックス

文節処理の入力となる音節マトリックスは、従来の音声認識装置 [14] の認識率情報（コンフュージョン・マトリックス）に基づき、以下の条件でコンピュータ・シミュレーションにより生成した。実用的な観念からは、非現実的な仮定になっているが、言語情報の有効性を検証するには十分である。

(a) セグメンテーション誤りはないものと仮定する（音節単位発声を仮定)。

(b) 音節候補の数は最大 8 個とし、8 位までの候補の中に必ず正しい音節候補があるものとする。なお、平均の候補数は 4 個である。

(c) 音節の認識距離情報は使用しない。すなわち、音節マトリックスにおける候補順位は無視し、全て同一の重みと仮定する。

(d) 音節に長音「ー」、鼻音「ｶﾟ」行、促音「ッ」の存在を仮定する。これは音声出力用の形式で登録されて単語辞書とのインターフェースを合わせるためである。なお、これらの音節の 1 位正解率は 100%としている。

また、実験には以下の 2 種類の音節マトリックスを用意した。

(a) text-open data
マルコフ連鎖値の計算に使用した日本文以外の漢字かな混じり文から生成した文節単位の音節マトリックス。（日経新聞 82 年 1 月 1 日の記事文から抽出）

(b) text-closed data

マルコフ連鎖値の計算に使用した日本文の漢字かな混じり文から生成した文節単位の音節マトリックス。（日経新聞 82 年 1 月 5 日の記事文から抽出）

3. 単語辞書

単語辞書は一般語、使用頻度の高い人名地名などの固有名詞を含む 16 万語の日本文音声変換用の辞書を使用した。ただし、使用した情報は音節、漢字かな、品詞の三種類である。

4.1.4 実験結果

直接選出型文節処理方式において失敗した文節例と成功した文節例をそれぞれ図 4.7、図 4.8 に示す。実験の結果得られた音節、漢字かな、および品詞の文節候補の右端に示した数値は trigram の総積値の自然対数の逆数を文字数で割った値である。したがって値が小さいほど尤度が高いことを示している。

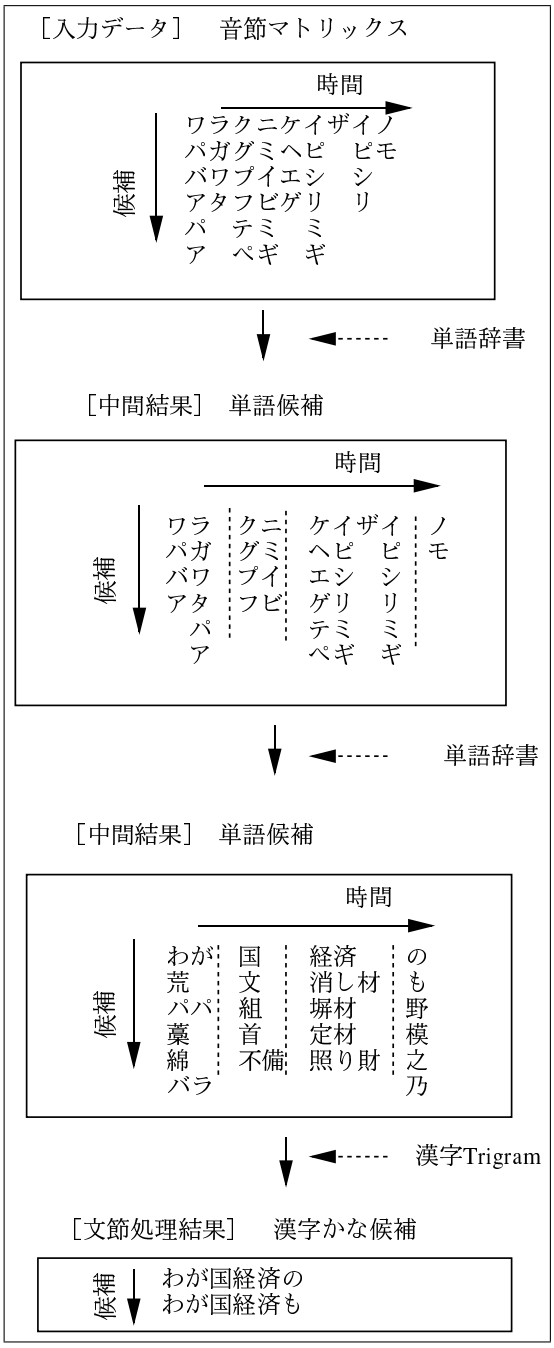


図 4.5: 直接選出型文節処理方式における入出力データ

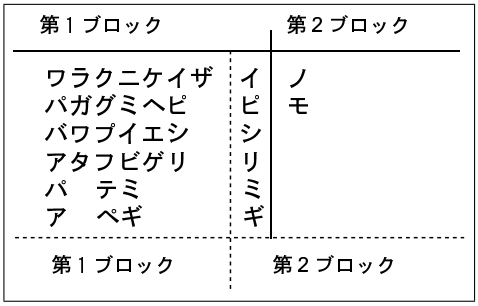


図 4.6: 入力データの一例

［原データ］

音節	ハンカク	シュウカイ	ハ
漢字かな	反核	集会	は
品詞	一般名詞	サ変名詞	副助詞

［入力データ］　（音節マトリックス）

ハ	ン	カ	ク	シュ	ー	ガ	イ	ワ
タ		タ	プ	チュ		カ	ピ	ア
カ		パ	フ	ヌ		ア	リ	バ
ア		チャ	グ	ツ		タ	シ	パ
パ		ア		チャ		ミ		
		ガ		ハ		ギ		
		ハ				パ		

［実験結果］

（1） 音節

順位	出力結果	確率値
1	カンガクシューカイワ	２．２４
2	ハンパクツーカイワ	２．２９
3	ハンタクシューカイワ	２．３１
4	カンカクシューカイワ	２．３５
5	ハンパクシューカイワ	２．３５
6	ハンガクシューカイワ	２．３９
7	ハンカクシューカイワ	２．３４
8	カンタクシューカイワ	２．４０

（2） 漢字かな

順位	出力結果	確率値
1	たんぱくちゅうたいわ	１６９．１７
2	タンパクちゅうたいわ	１６９．３３
3	たん白ちゅうたいわ	１８４．６８
4	たんぱく通貨市場	２０２．２６
5	タンパク通貨市場	２０２．４５
6	感覚ちゅうたいわ	２０２．６６
7	反核ちゅうたいわ	２０２．７２
8	間隔ちゅうたいわ	２０２．８５

（3）品詞

順位	出力結果				確率値
1	一般名詞	一般名詞	副助詞		1.27
2	サ変名詞	一般名詞	副助詞		1.41
3	一般名詞	サ変名詞	副助詞		1.42
4	一般名詞	サ変名詞	純体接尾	副助詞	1.51
5	一般名詞	サ変名詞	一般名詞	副助詞	1.55
6	一般名詞	一般名詞	一般名詞	副助詞	1.55
7	サ変名詞	サ変名詞	副助詞		1.58
8	一般名詞	一般名詞	純体接尾	副助詞	1.62

図 4.7: 直接選出型文節処理方式における誤りの例

［元データ］

音節	ガイコク	ギンコー	ハ
漢字かな	外国	銀行	は
品詞	一般名詞	一般名詞	副助詞

［入力データ］　（音節マトリックス）

ガ	イ	ホ	プ	ギ	ン	コ	ー	ワ
カ	ピ	コ	ク	キ		ホ		ア
タ	リ	オ	フ	リ		オ		バ
ア	ギ		グ	ピ				パ
パ	ミ							
ラ	シ							
ワ								

［実験結果］

（１） 音節

順位	出力結果	確率値	
1	ガイコクギンコーワ	２．１７	正解
2	ガイコクキンコーワ	２．１９	
3	ガイコクキンホーワ	２．２９	
4	タイコクギンコーワ	２．２９	
5	タイコクキンコーワ	２．３１	
6	カイコクギンコーワ	２．３６	
7	カイコクキンコーワ	２．３７	
8	タイコクキンホーワ	２．４１	

（２） 漢字かな

順位	出力結果	確率値	
1	外国銀行は	２．１５	正解
2	大国銀行は	１４４．８０	
3	開国銀行は	１４５．０５	
4	愛国銀行は	１４５．１５	
5	来国銀行は	１４５．１５	
6	愛国銀行は	１４５．１５	
7	買い越不均衡は	２２３．８１	
8	カシオ不均衡は	２２３．８５	

（３） 品詞

順位	出力結果				確率値	
1	一般名詞	一般名詞	副助詞		1.27	正解
2	サ変名詞	一般名詞	副助詞		1.41	
3	一般名詞	サ変名詞	副助詞		1.42	
4	一般名詞	サ変名詞	一般名詞	副助詞	1.55	
5	一般名詞	一般名詞	一般名詞	副助詞	1.55	
6	サ変名詞	サ変名詞	副助詞		1.58	
7	一般名詞	一般名詞	純体接尾	副助詞	1.60	
8	一般名詞	一般名詞	サ変名詞	副助詞	1.67	

図 4.8: 直接選出型文節処理方式における正解例

このような出力結果を入力文節 100 件について集計した結果を図 4.9 と図 4.10 に示す。

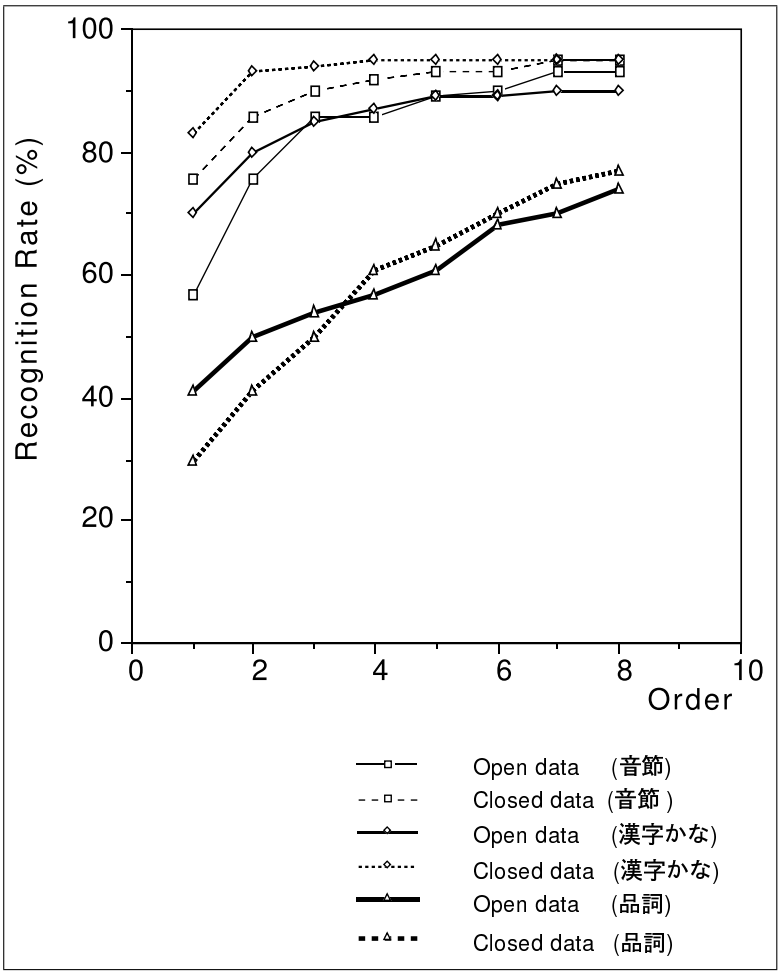


図 4.9: 音節選出型文節処理方式の実験結果

これらの図から以下のことがわかる。

1. 文節候補の音節 1 位正解率は最大値が直接選出型の text-closed data で 79%、最小値が text-open data で 56%であった。また、8 位までの累積正解率は音節選出型と直接選出型での差はなく、text-closed data で 94%、text-open data で 93%であった。

2. 漢字かなの文節候補の 1 位正解率は音節選出型で text-closed data では 83%、直接選出型では 84%であった。特に直接選出型では 4 位までの累積正解率は 99%を示した。また、text-open data では 1 位正解率が音節選出型は 70%、直接選出型では 65%であるが、8 位までの正解率は

共に 90%を越えた。この値は、音節の trigram を用いた文節候補で得られた正解率と同等以上である。

3. 品詞の文節候補の場合は音節や漢字かなの文節候補の場合より正解率が遥かに低く、両方式で見て、1 位正解率は 27 ～ 42%、8 位までの累積正解率は 72 ～ 77%にしか過ぎない。しかし、text-open data と text-closed data の正解率の差は殆どない。

4.1.5 考察

1. マルコフ連鎖値の収束性

   text-closed data と text-open data の正解率の差は音節、特に漢字かな文字の文節候補において顕著であるのにたいして、品詞の文節候補では差がほとんど認められない。これはマルコフ連鎖値の収束性の問題で、さらに多く日本文を収集することにより両者がお互いに接近する形で、その差は減少すると判断される。

2. 音節と漢字かなの情報量

   音節と漢字かなの特性を比較すると、unigram,bigram,4-gram の場合は音節の方がエントロピーが小さいが、trigram の場合は逆に漢字かなの方が小さくなっている点が特徴的である。これは、trigram においては、漢字かなの方が情報量が大きく、それ以上、次数を上げても効果は少ないのに対して、音節ではさらに次数を上げればそれだけ効果が得られることを意味していると思われる。

3. 誤りの原因

   漢字かなの文節候補の選出において、text-closed data の実験で、正解候補が最終的に 8 位以内に入らなかった文節を見ると、それらのすべてが、音節選出型の方式では音節の文節候補の失敗に起因し、直接選出型の方式では単語境界の分割数が足りないことに起因していることがわかった。

   前者の漏れを防ぐには、音節の trigram で抽出する文節候補の数を増やすことが考えられるが、計算量の増加を伴うので適当なトレードオフが必要となる。また、後者の漏れを防ぐには単なる分割数最小法ではなく、係り受け併用型の分割数最小法 [46] を採用した方が良いと考えられる。

4.1.6 まとめ

日本文音声認識において音声の物理的特性を使用した音声認識装置と自然言語処理の間を結ぶ処理として、trigram を用いた文節処理の 2 つの方法（音節選出型と直接選出型）を提案し、その効果を実験的に求めた。

その結果、両方の方式とも、漢字かな混じりの文節候補を従来の音節の trigram を用いた文節候補で得られた正解率と同じか、それ以上の精度で、生成できることが分かった。これは、漢字かなの trigram の効果は非常に効果的で、大語彙辞書を用いて、音節から漢字かな混じり文を生成する際に生じる膨大な曖昧性がほぼ完全に解消することを意味している。

音節選出型と直接選出型の文節処理を比べると、音節の文節候補の第 1 位正解率は、後者の精度が若干高いが両者に大きな差異は認められないことから、音節の trigram には単語内の音節の trigram の情報がかなり反映されており、音節における文節候補の推定の能力の点でみれば、音節間の trigram は単語辞書に代わり得る情報を持つことが推定される。また漢字かなの文節候補では直接選出型の方が精度は高い。これは漢字かなの trigram はかなり大きな情報量を持っているため、単語の候補が増加しても、これが文節候補の推定に影響を与えていないことがわかる。

text-closed data と text-open data の場合の比較では両者の差は音節、漢字かなに比べて品詞では差がなかった。このことから、品詞の trigram は前 2 者に比べて小量のデータで収束することが分かるが、これは同時に候補絞り込みに使用される情報量が少ないことも意味しており、実験では文節絞り込みの精度は最も小さくなっている。

本章では、大語彙の音声認識におけるマルコフモデルの効果を見る立場から、音声認識装置からの認識距離は使用せず、音節、漢字かな、品詞それぞれの trigram の効果について研究した。したがって今後、これらの情報を組合せた場合について検討する必要がある。

また、本章では対象外としたが、今後、音声認識部における脱落、挿入などを含むセグメンテーションの誤りの問題や、文節候補の曖昧性をさらに絞り込むための、文節間文法情報や意味、文脈等情報等の適用方法の検討、また、text-closed data において連鎖値が 0.0 である場合の値の定め方等の検討が必要である。

4.2 単語の HMM と bigram を利用した文節音声認識

ここでは、X 線 CT の所見作成入力用の音声ワードプロセッサーを目指して、認識単位として単語、言語情報として単語の bigram を使用した文節音声認識システムを作成した。語彙数は約 3000 である。このシステムの概要と

実験結果について報告する。

4.2.1 認識単位を単語とした文節音声認識

1. 音響モデル

   従来の多くの文（文節）音声認識システムでは認識単位として音節や音素を選択している [58],[40]。しかし、現実の音声データでは音素境界が曖昧な音素が多い。したがって、高い認識性能を目指す場合、長い認識単位が有利であると考えられる。したがって、ここでは認識単位として単語を選択した。ただし、このデータベースでは、文節出現率が高いものから上位 100 文節は単語として登録してある。しかし、単語を認識単位とした場合、単語の HMM の学習の時に、大量の単語発声の音声データが必要であること、また認識のときに、HMM のパラメータの記憶のために多くのメモリー空間が必要であることなどから、従来はあまり多く行なわれてきていない [45]。

   そこで、本節では学習データを減らすため、１つの単語の HMM の学習に１つの単語発声の音声データのみ使用することにした。つまり X 線 CT の所見作成入力用の音声ワードプロセッサーを使用する人に、事前に 3000 単語を１回発声してもらい、このデータで単語の HMM を学習した。そして、少ない音声データで精度の高い HMM のパラメータを推定するために Fuzzy-VQ HMM[4] を用いた。コードブックサイズは 256 である。また認識時において HMM のパラメータの記憶のためのメモリー空間を減らすために、単語の HMM のモデルは全て 4 状態 3 ループとした。

2. 言語モデル

   言語モデルには単語の bigram のみをもちいた。bigram の連鎖確率値の計算には、今まで入手できた X 線 CT の所見作成の全文章、71198 単語から計算した。また、連鎖確率値が 0.0 である場合は deleted-interpolation[7] などの平滑化はおこなわず、$\exp(-1000.0)$ に置き換えた。

3. 単語の bigram を用いた文節音声認識アルゴリズム

   実験に用いた認識アルゴリズムの基本は、単語の HMM に Viterbi サーチ (One-pass DP) に単語の bigram とした。また実験では HMM の累積尤度 $G(l, w, i)$ を複数 (N 個) 持たせることによって複数の候補を出力する N-best サーチを行なった。

4.2.2 文節音声認識実験

1. 実験条件

認識実験では duration control と N-best のサーチ幅を変化させて行なった。また、単語の HMM の学習のデータを増加させた場合の実験も行なった。これらの実験の条件を表 4.1 に示す。その他の実験条件は表 4.2 にまとめた。なお duration control は同一話者の単語発声の 3 回分のデータの平均発声時間と分散を測定し、この値からガウス分布を計算し、duration control に使用した。（単語のマッチングが終了してから duration control の尤度を乗じた。）

表 4.1: 実験条件

実験番号	duration control	N-best	学習データの個数
実験 1	なし	2	1
実験 2	あり	2	1
実験 3	あり	8	1
実験 4	あり	2	3

2. テストデータ

   X 線 CT 所見作成の文章は大きくわけて正常所見と異常所見に分類される。そして異常所見は正常所見と比較すると文章が複雑なため、認識率が低くなることが知られている [86]。そこで実験は、bigram の連鎖確率を計算するのに使用したテキストを発声した音声データ (text-closed data) と bigram の連鎖確率を計算するのに使用しなかったテキストを発声した音声データ（text-open data) について、各々異常所見と正常所見について合計 4 つの条件で行なった。実験は平均 100 文節で行なった。例文は図 3.5 参照。

4.2.3 実験結果

文節認識の実験結果を表 4.4 に示す。この結果から得られたことを以下に示す。

1. 実験 1 から test-closed の正常所見で 96.8%、異常所見では 78.1%、text-open data の正常所見でも 86.5%、異常所見では 72.1%の高い文節認識率が得られた。したがって HMM の学習データが 1 つでも Fuzzy-VQ を使用することにより高い文節認識性能が得られることがわかった。

2. 実験 1 と実験 2 の比較から、duration control を行なうと認識性能が低下した。この原因として duration control に使用した平均・分散の値の不正確さが考えられる。これらの値は同一話者が発声した 3 つの単語発声の音声データから計算したため値の信頼度はかなり低い。

表 4.2: 文節音声認識の実験条件

使用アルゴリズム	word HMM + Viterbi search + word bigram 特定話者認識
話者数	1
発話様式	文節発声
認識単位	word
語彙数	約 3000
学習データ	単語発声
言語情報	単語 bigram
音響パラメータ	log power + 16 次 LPCcepstrum + Δlog power
距離尺度	簡易マハラノビス
VQ コード数	256
単語モデル	4-state 3-loop Fuzzy-VQ HMM
フレーム窓長	18ms
フレーム周期	9ms
ファジネス	1.5
近傍数	5
サンプリング周波数	12kHz
HMM と bigram の 結合値 α	32

3. 実験結果 2 と実験結果 3 の比較から、N-best の幅を広げた方が高い認識率を出すことが示された。

4. 実験結果 2 と実験結果 4 の比較から、音声データを増加させることによって認識性能が向上することが示された。これは HMM のパラメータを推定するための学習データが１つでは、不十分であることを示している。しかし不特定話者認識の場合、一人の発話データが１つしかなくても、複数の話者が発話することによって、多くの音声データが利用できるため、認識単位が単語でも問題はないと思われる。

4.2.4 考察

1. HMM の種類について

   本実験では、HMM の学習に使用する音声データを１つとしたため Fuzzy-VQ HMM を使用した。しかし不特定話者認識のためには連続分布型

表 4.3: テストデータの実験

(a) text-closed data の正常所見

(b) text-closed data の異常所見

(c) text-open data の正常所見

(d) text-open data の異常所見

表 4.4: 実験結果

実験番号	1	2	3	4
duration control	なし	あり	あり	あり
N-best	2	2	8	2
学習データ数	1	1	1	3
text-closed data の正常所見	96.8%	82.6%	100.0%	100.0%
text-closed data の異常所見	78.1%	76.3%	78.9%	84.2%
text-open data の正常所見	86.5%	86.5%	89.2%	94.6%
text-open data の異常所見	72.1%	68.9%	72.1%	77.0%

HMM（2.1.9 節参照）のほうが相応しいと考えられる。しかし、連続分布型 HMM を使用した場合、mixture 数にも依存するが学習に大量の音声データが必要である。そこで、単語を認識単位とするばあいは、学習データがある程度少なくてすむ semi-continuous HMM[4] が有望ではないかと考えている。

2. 認識単位・単語

認識単位として音素を選択したとき、HMM の学習のために、音素ラベルが付与された音声データが必要になる。ラベリング作業は自動化がある程度可能であるが、最終的には人手に頼らざるを得ないため、音声データベースの作成のコストはかなり高い。一方認識単位を単語にしたばあい、ラベリング作業は不用になる。そのかわり、数個の単語発声が必要があるため、発話者の負荷が大きくなる。認識システムの仕様や目的にも依存するが、連結学習も考慮にいれて、認識単位を考えるべきであろう。

3. リアルタイムにむけて

音声認識のリアルタイム化には２つの方法がある。１つにはアルゴリズムによる計算量の削減であり、もう１つはハードウエアによる計算コストの分散化である。フレーム同期型の認識アルゴリズムにおいて計算量を削減する方法としてビームサーチが知られている [68]。しかし、超並列コンピュータなどを考えた場合、ビームサーチを採用しないほうが早くなる可能性がある。今後、リアルタイム化はハードウエアも考慮して最適なアルゴリズムを考えていく必要があると思われる。

4.2.5 まとめ

本章では、言語モデルとして単語の bigram を用いて特定話者の文節認識実験を行なった。この実験の結果、認識単位を単語とした場合、HMM の学習用の音声データが１つでも、かなり高い認識率が得られること、そして単語の bigram の情報と組み合わせることにより、text-open data の正常所見でも 86.5%、異常所見では 72.1%の文節認識率が得られることが示された。

4.3 tree-trellis サーチと単語の trigram モデルを用いた文音声認識

現在、音声認識に用いられる言語モデルとしては、簡潔さ・有効性などの点から単語の bigram モデルが主流である。しかし、単語の trigram は一般的に bigram より小さな perplexity を示す。だが、trigram は、2 つ前の単語と直前の単語が存在したときに現在の単語に遷移する確率であるため、認識アルゴリズムに trigram を組み込んだ場合、大量のメモリ量と計算量が必要になる。本節では、2.2 節で述べた tree-trellis サーチを基本に朗読発話において単語の trigram を利用したときの認識実験結果について報告する。

ところでポーズは音声データのあらゆる場所に出現する可能性がある。しかし言語モデルではこれに対応しきれないため、ポーズを含む音声データは誤認識が起きやすい。ここで利用した tree-trellis サーチでは、各時刻・各状態において最尤の単語列を知ることができる。この特徴を生かして、音響モデルではポーズを認識しなから言語モデルではポーズをスキップすることにより、ポーズがある音声でも誤認識が起こりにくくなる。最後にこのアルゴリズムの有効性について述べる。

4.3.1 単語の trigram モデルを用いた文音声認識実験

4.3.1.1 認識アルゴリズム

本章の実験では、認識アルゴリズムとして tree-trellis サーチを用い、trellis でグリッドを選択した。また、計算量を削減するためにフレーム毎にビームサーチをかけている。また、音素の HMM を連結させて単語の HMM を作成した。言語モデルとしては単語の trigram を使用している。

4.3.1.2 実験条件

実験は特定話者認識および不特定話者認識の 2 つの様式で行なった。単語の HMM は音素の HMM を連結して作成した。また音素の HMM の学習データには、特定話者認識の場合はテストデータと同一話者の 2620 単語発声を使用し、不特定話者認識の場合は評価話者とは別の男性話者 12 名の 736 単語発声を利用した。単語の perplexity は trigram で 4.0、 bigram で 13.9 である。テストデータは、国際会議の問い合わせのタスクの 261 文で、話者はナレータ 1 名である。実験条件を表 4.5 にまとめる。なお、テストデータの先頭と最後には約 200ms のポーズ区間がある。また、 trigram の連鎖確率値は、 ATR の対話データベース [10] のなかから国際会議の予約に関するデータ約 1 万 2 千文章、約 17 万単語（3.3.1 節参照）にテストデータのテキストを加えて計算した。したがって認識実験は text-closed の実験になる。ただしテキストデータ中の「あのー」、「えーと」などの間投詞は削除している。

4.3.1.3 実験結果

ここで提案したアルゴリズムは HP735、語彙数 1567、ビーム幅 4096 において、メモリ量 15Mbyte 平均文認識時間 平均 1 分 30 秒（リアルタイムの約 50 倍) で動作した。実験結果は文認識率と単語正解率 (word correct) と単語認識精度 (word accuracy)[15] で評価した。なお、単語正解率は以下の式で計算される

$$word\ \ correct \quad = \quad H/(H+D+S) \times 100\% \tag{4.1}$$

$$word\ \ accuracy \quad = \quad (H-I)/(H+D+S) \times 100\% \tag{4.2}$$

ただし

H ... 正解の単語数
D ... 脱落誤りした単語数
S ... 置換誤りをした単語数
I ... 挿入誤りをした単語数

また比較のために単語の bigram を使用したときの実験も行なった。実験結果を表 4.6 に示す。実験の結果、特定話者認識において trigram を用いたとき、文認識率で 66.7%、8 位までの累積認識率で 75.1%が得られた。しかし、不特定話者認識では、テストデータ全てにおいて、データの先頭のポーズ区間に 1 音節の単語が挿入されたため、文認識率は 0.0%になった。(例えば「はい」を「と、はい」と認識。) したがって、認識精度が正解率と比較して大きく低下している (31.1% ← 74.2%)。

表 4.7 に、特定話者で単語 trigram を使用したときの誤認識の例を示す。例文においてアンダーラインは誤認識を示す。誤認識された文の中には、意味的には正しい文が多い。意味的に正しい文を正解に含めた時、1 位文理解率は約 80%であった。

4.3.2 ポーズの処理

表 4.7 において、入力された文と大きく異なる文が出力された音声データを調査すると、ポーズの区間から誤りが始まっていることがわかった。そこで言語モデルにおいてポーズのスキップ、音響モデルにおいてポーズの HMM の学習をすることで認識性能の向上を試みた。

4.3.2.1 ポーズのスキップ (言語モデルにおける処理)

ポーズは、文節間に出現することが多いが、音声データのあらゆる場所に出現する可能性がある [34]。そこで単語と単語の境界にポーズがあっても、誤認識が起き難いようにアルゴリズムを改良した。ここで使用したアルゴリズムでは、各時刻・各状態において累積尤度が最大の単語列を知ることができる。そこでポーズを 1 単語と考えて、ポーズに接続されたときの連鎖確率値は 1.0 にする。そしてポーズ以外の単語に接続されるときポーズをスキップして trigram の連鎖確率値を計算する。例えば

「“東京都” “港区” “新橋” /$pause$/ “1 丁目”」

と発声されたとき、単語 trigram の値を

P(“ 新橋”|“東京都”,“港区”) ×1.0×P(“1 丁目”|“ 港区”,“新橋”)

と計算する。なお、ポーズの HMM の尤度の学習には、学習データの前後にある無音区間を利用した。

この改良したアルゴリズムを用いて認識実験を行なった。実験条件は表 4.5 と同一である。この結果を表 4.8 に載せる。このポーズのスキップにより、特

定話者認識では、認識性能が向上した (66.7% → 71.6%)。また、不特定話者認識では、認識性能が顕著に向上した (0.0% → 61.7%)。

4.3.2.2 ポーズの HMM の学習（音響モデルにおける処理）

不特定話者認識の実験において誤認識された文を調査すると、テストデータの先頭のポーズ区間から誤認識している例が多いことがわかった。そこでテストデータの先頭の無音区間を利用して、Baum-Welch アルゴリズムでポーズの HMM を再学習した。学習にはテストデータ 100 文の先頭の 100ms を使用した。

4.3.2.3 ポーズ処理をしたときの実験の結果

4.3.2.1 節および節に示すような改良をして、文認識実験を行なった。この実験結果を表 4.9 に載せる。これからわかるように認識性能が向上する。特に不特定話者認識においては効果が著しい。特定話者認識における誤認識の例を表 4.10 に載せる。これからわかるように、誤認識された文には意味的に合っている文が多い。意味的に正しい文を正解に含めたとき 1 位理解率は 99%に達した。

これらの実験から、誤認識の原因になっているポーズの対策として、言語モデルではポーズのスキップ、音響モデルではポーズの HMM を学習することが有効であることが示された。

4.3.3 各種パラメータの検討

4.3.3.1 ビーム幅

ビームサーチは、各フレームごとの尤度計算において、累積尤度の低い単語列は以後の探索から除外できる可能性が高いことを仮定している。そこでビーム幅を変えた時の文認識率の変化を研究した。ビーム幅以外の実験条件は、表 4.5 と同一である。また、 4.3.2.1 節および 4.3.2.3 節で述べたポーズ処理はおこなっている。この実験結果を図 4.11 に示す。

この実験結果からビーム幅を広げるに従い認識性能は向上するが、ビーム幅が 1024 を越えると、認識性能はあまり変化しないことがわかる。ここでは認識語彙数を変化させた実験を行っていないため明確にはいえないが、このビーム幅 1024 は語彙数 1567 に近いことから、朗読発話においてビーム幅は語彙数程度、必要であると考えている。

4.3.3.2　音響尤度と言語の連鎖確率の結合値 α

ここでは音響尤度と言語の連鎖確率の結合値 α を変化させたときの文認識率の変化を研究した。他の実験条件は表 4.5　と同一である。この結果を図 4.12 に示す。この図において横軸は結合値 α で、この値が大きいほど言語尤度の重みが音響尤度と比較して増加することを意味している。縦軸は文認識率である。

この実験から音響尤度と言語の連鎖確率の結合値 α が 16 のとき最も高い文認識率が得られた。

4.3.3.3　text-open data における認識率

trigram の連鎖確率の計算に使用するテキストデータの学習量に対する文認識率の変化を研究するために、認識実験を行なった。実験は、言語モデルとして bigram と trigram、特定話者認識と不特定話者認識、さらに text-close data （ATR の対話データベースにテストデータを加えて連鎖確率を計算した場合）と text-open data（ATR の対話データベースにテストデータを加えずに連鎖確率を計算した場合）の合計 8 種類の実験を行なった。実験条件は、表 4.5 と同一である。また 4.3.2.1 節および 4.3.2.3 節で述べたポーズ処理はおこなった。

この実験結果を図 4.13 に示す。この図では横軸は trigram の連鎖確率値を計算するのに使用した学習データの単語数で縦軸は文認識率である。この実験では、text-closed data では trigram のほうが bigram と比較してかなり高い認識性能が得られるが、text-open における実験では、bigram のほうが trigram よりも認識性能は高いことがわかり、trigram の算出のためのデータ量の不足を示している。

4.3.3.4　単語の trigram の値を平滑化した場合の認識率

単語の trigram の値に deleted-interpolation を利用して平滑化した場合の認識率の変化を表 4.11 に示す。

なお、平滑化の値は、次式の trigram,bigram,unigram, フロアリングに対して各々

$\lambda_3 = 0.35, \lambda_2 = 0.48, \lambda_1 = 0.11, \lambda_0 = 0.06$

である。

$$\hat{P}(w_i|w_{i-2}, w_{i-1}) = \lambda_3 P(w_i|w_{i-2}, w_{i-1}) + \lambda_2 P(w_i|w_{i-2}, w_{i-1}) + \lambda_1 P(w_i) \tag{4.3}$$

これから単語の trigram を平滑化することで text-open data において認識性能が向上することがわかる。

4.3.4 考察

1. ポーズの HMM の学習に関して

   本実験では、ポーズの HMM は Baum-Welch アルゴリズムを用いて再学習をおこなった。しかし、データ量が少ない場合のことを考えると、混合分布の平均値を移動させる話者適応化技術 [64] が好ましいと考えている。

2. ポーズ処理

   今回の実験から、誤認識の原因になっている音声に含まれるポーズの対策として、言語モデルではポーズのスキップ、音響モデルではポーズの HMM を学習することで文認識性能が向上することが示された。今後、ポーズは促音やクロージャとも併せて考慮する必要があろう。

3. ビーム幅

   ビーム幅は語彙数と正の相関を持つと考えられる。しかし実験ではビーム幅が 1024 を越えると、認識性能はあまり向上しないことが示された。認識語彙数を変化させた実験を行っていないため明確ではないが、このビーム幅 1024 は語彙数 1567 に近いことから、ビーム幅は語彙数程度で十分であると思われる。ただし、ここで実験に用いた話者はナレータであるため、音声は非常に丁寧にに発話されている。したがって、通常の話者の音声ではこのビーム幅では不足する可能性もある。

4. 音響尤度と言語の連鎖確率の結合値

   音響尤度と言語の連鎖確率の結合値を変化させた時の文認識率の変化を調査した実験から α が 16 のとき最も高い文認識性能が得られた。

   しかし、単語の HMM と単語の bigram を考えて、これらを組み合わせたモデルは Ergodic HMM に似たモデルになる。そして単語の bigram の値は１つの単語の HMM の最終状態の遷移確率を別の単語に接続されたときの値の分配率になる（図 4.14）。 この時の音響尤度と言語の連鎖確率の結合値 α は 1 になる。この値は trigram でも同様であると考えられる。したがって理論的には音響尤度と言語の連鎖確率の結合値 α は 1 であると考えている。

   ただし、連続型 HMM では遷移確率は離散値であるのに対しシンボル出力確率は確率密度関数であるためダイナミックレンジが大きく異なる。そのため、α は 1 より大きい方が好ましいと思われる。

4.3.5 まとめ

本章では、単語 trigram を利用した実験結果を報告した。実験の結果、朗読発話の text-closed data において特定話者認識では 66.7%の文認識率が得られた。

この論文では tree-trellis サーチを利用している。したがって、各時刻・各状態において累積尤度が最大の単語列を知ることができる。この特徴を生かして、音響モデルではポーズを認識しながら言語モデルではポーズをスキップすることにより、ポーズによる誤認識を削減できる。また、テストデータの先頭の無音区間を利用して、ポーズの HMM を再学習した。このようなポーズの処理をすることにより不特定話者認識の text-closed data において 83.9%の文認識率が得られた。

これらの実験の結果、このアルゴリズムの有効性が示された。

4.4 まとめ

本章では、音声認識システムに言語モデルとして N-gram を利用した有効性を定量的に研究した。

4.1 節では、日本文音声認識において音声の物理的特性を使用した音声認識装置と自然言語処理の間を結ぶ処理として、trigram モデルを用いた文節処理の二つの方法を提案し、その効果をシミュレーションで実験的に求めた。入力データは新聞記事である。その結果、両方の方式とも、漢字かな混じりの文節候補を音節の trigram を用いた文節候補で得られた正解率と同じか、それ以上の精度で生成できることが分かった。これは、漢字かなの trigram モデルの効果は非常に効果的で、大語彙辞書を用いて、音節から漢字かな混じり文を生成する際に生じる膨大な曖昧性がほぼ完全に解消することを意味している。

4.2 節では、X 線 CT 作成の文章において、言語モデルとして単語の bigram を用いて特定話者の文節認識実験を行なった。この実験の結果、単語の bigram の有効性が示された。また認識単位を単語とした場合、HMM の学習用の音声データが１つでも、Fuzzey-VQ を使用すれば、高い認識率が得られることが示された。

4.3 節では、単語 trigram を利用した文音声認識の実験結果を報告した。実験の結果、朗読発話の text-closed data において特定話者認識では 66.7%の文認識率が得られ、単語の trigram の有効性が示された。また音声中にあるポーズの対策として、言語モデルではポーズのスキップ、音響モデルではポーズの学習をすることによって認識性能が向上することが示された。

これらの結果、言語モデルとしての N-gram の有効が示された。

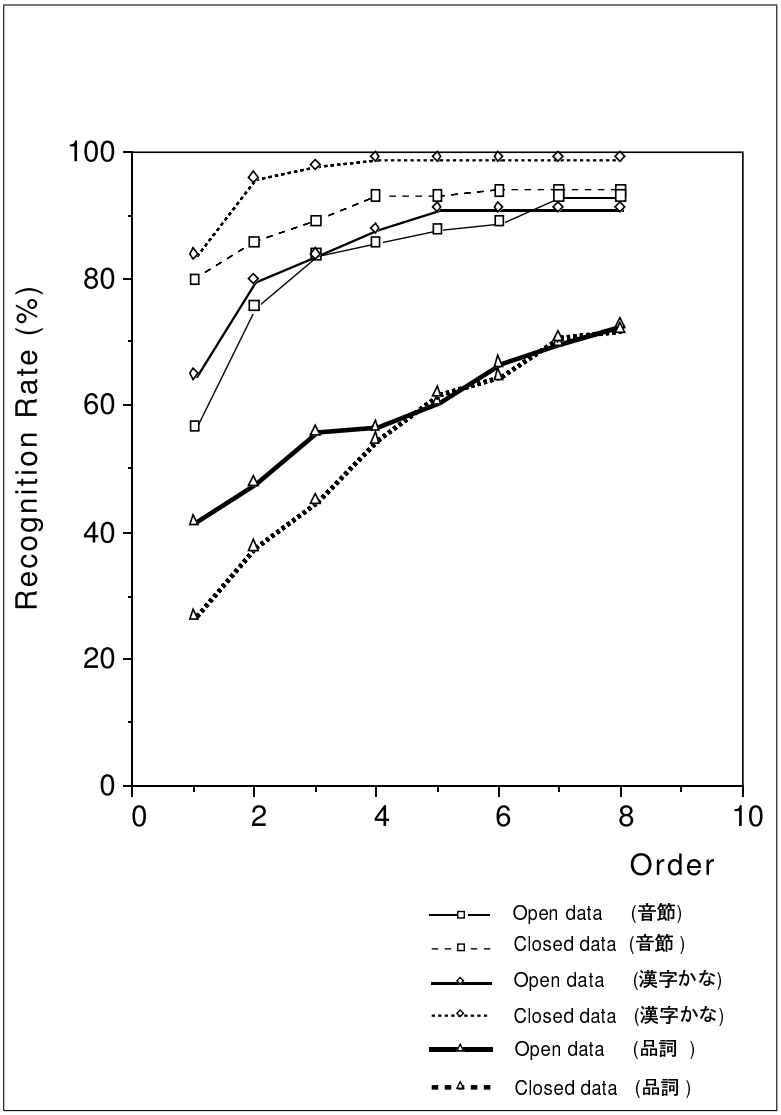


図 4.10: 直接選出型文節処理方式の実験結果

表 4.5: 文音声認識の実験条件

音素モデル	Continuous mixture HMM, diagonal
Mixture 数	最大 14（各音素によって変化）
1 音素あたりの状態数	4-state 3-loop left-right model
使用パラメータ	LPC ケプストラム 16 次 + パワー + Δ パワー+Δ ケプストラム 16 次
ウインド幅	20ms
フレーム周期	5ms
HMM の学習音声	
（特定話者認識）	テストデータと同一話者の 2,620 単語発声
（不特定話者認識）	男性話者 12 名の 736 単語発声
音素カテゴリ数	52 音素
認識単語数	1,567
ビーム幅	4,096
継続時間制御	なし
実験文数	261 文, 話者 1 名
発声様式	朗読発話（連続発声）
発声内容	国際会議の問い合わせ
単語 trigram の値の 推定に使用した テキストデータ量	約 1 万 2 千文章 171,978 単語 テストデータのテキストを含む （間投詞は削除）
単語 trigram の perplexity	4.0
単語 bigram の perplexity	13.9
フロアリングの値	exp(-1000.0)
言語尤度と音響尤度の 結合値 α	1

表 4.6: 認識実験の結果　文認識率 (%)

言語 model		bigram		trigram	
		特定話者	不特定話者	特定話者	不特定話者
累積文認識率	1	42.5%	0.0%	66.7%	0.0%
	∼ 2	47.9%	0.0%	74.2%	0.0%
	∼ 8	51.3%	0.0%	75.1%	0.0%
word correct		80.7%	55.8%	88.8%	74.2%
word accuracy		63.0%	1.2%	81.1%	31.1%

text-closed; ビーム幅:4,096; α:1

表 4.7: 実験において誤りが出力された文の例

text-closed; ビーム幅:4,096; α:1

正解文 → 1位出力
京都プリンスホテルが会議場には近いのですが → 京都プリンスホテルが会議場には近い ん ですが ホテルの手配もしていただけるのですか → ホテルの手配もしていただける ん ですか
どのようなご用件でしょうか → どのような _ 用件でしょうか ご住所とお名前をお願いします → ご住所とお名前 _ お願いします
住所は東京都港区新橋１丁目１番３号です → 住所は東京都 になったのを送っしかし去年一番可能 です 電話番号は３３１の２５２１です → 論文を発表 ３３１の２２日です

表 4.8: 認識実験の結果（ポーズのスキップ）　文認識率 (%)

言語 model		bigram		trigram	
		特定話者	不特定話者	特定話者	不特定話者
累積文認識率	1	49.4%	31.4%	71.6%	61.7%
	∼ 2	56.3%	41.0%	77.0%	72.0%
	∼ 8	60.2%	44.4%	79.7%	76.7%
word correct		81.3%	62.5%	89.4%	85.1%
word accuracy		66.8%	43.0%	85.0%	77.9%

text-closed; ビーム幅:4,096; α:1

表 4.9: 認識実験の結果（ポーズのスキップ、ポーズ学習）　文認識率 (%)

言語 model		bigram		trigram	
		特定話者	不特定話者	特定話者	不特定話者
累積文認識率	1	60.5%	44.8%	90.4%	83.9%
	∼ 2	68.2%	51.0%	95.4%	92.7%
	∼ 8	76.2%	55.6%	97.7%	96.6%
word correct		87.2%	72.4%	97.6%	96.2%
word accuracy		79.6%	58.3%	97.1%	95.7%

text-closed; ビーム幅:4,096; α:1

表 4.10: 実験において誤りが出力された文 （ポーズのスキップ、ポーズ学習）

text-closed; ビーム幅:4,096; α:1

正解文 → 1 位出力
京都プリンスホテルが会議場には近いのですが → 京都プリンスホテルが会議場には近い ん ですが ご住所とお名前をお願いします → ご住所とお名前 _ お願いします
ではお名前とご住所をお願いします → ではお名前と お 住所をお願いします どのようなご用件でしょうか → どのような _ 用件でしょうか
失礼します → そうします 言語学や心理学を専攻する方にも参加していただく予定です → 言語学や心理学を専攻する方にご参加して あるというん です

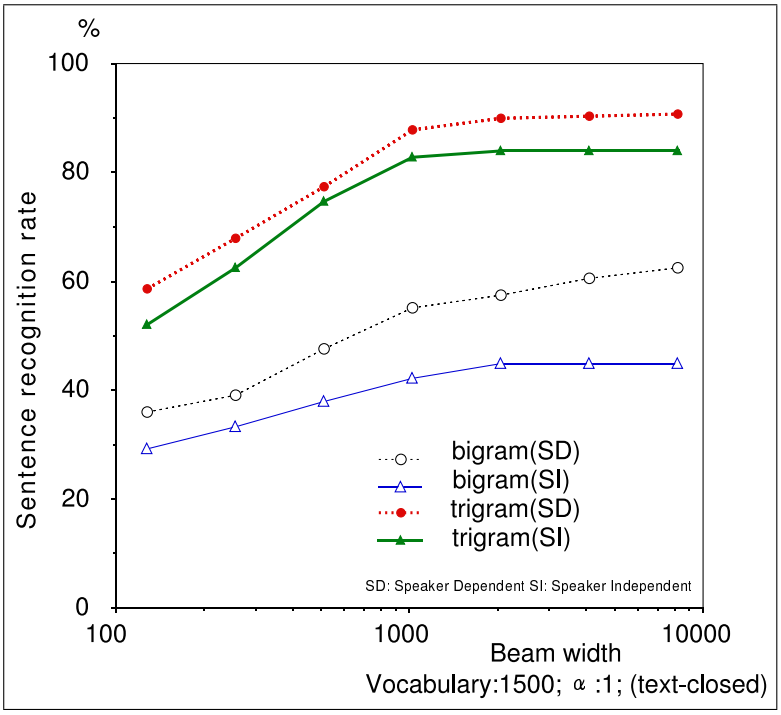


図 4.11: ビーム幅を変化させたときの変化　文認識率 (%)

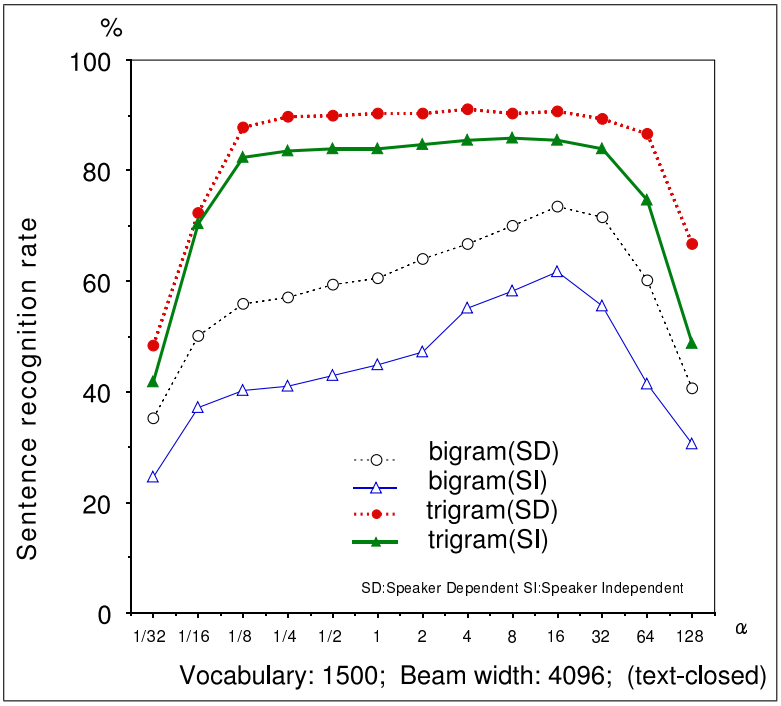


図 4.12: 音響尤度と言語の連鎖確率の結合値を変えたときの認識性能の変化

文認識率

(%)

表 4.11: 認識実験の結果（単語の trigram の値を平滑化したとき）　文認識率 (%)

言語 model		text open data		text closed data	
		特定話者	不特定話者	特定話者	不特定話者
base line	1	35.6%	33.7%	90.8%	85.1%
	~ 2	37.5%	36.8%	96.6%	93.5%
	~ 8	38.3%	37.9%	98.8%	97.7%
interpolation	1	51.7%	43.3%	79.3%	78.2%
	~ 2	58.6%	47.9%	88.5%	86.2%
	~ 8	62.4%	53.6%	91.9%	90.0%

text-closed; ビーム幅:4,096; α:1

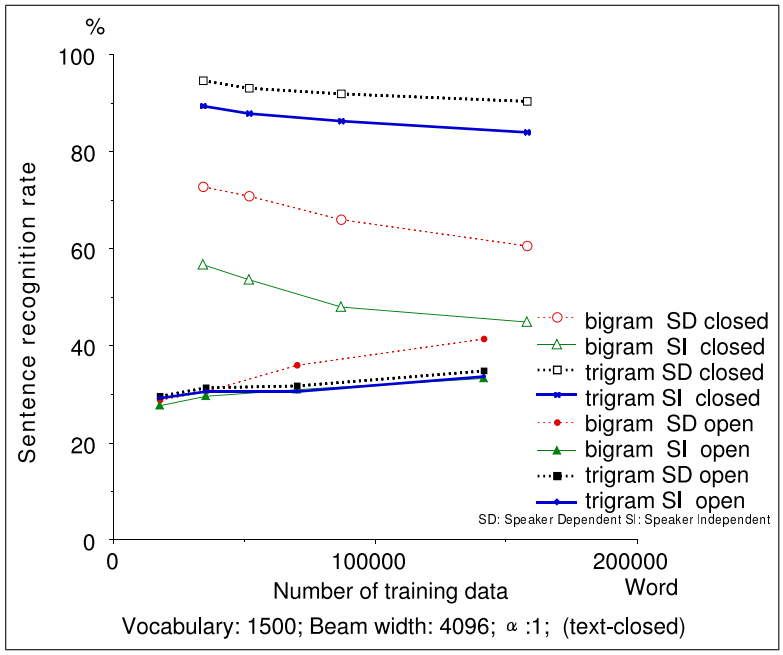


図 4.13: 学習データ量における認識結果の変化　認識率 (%)

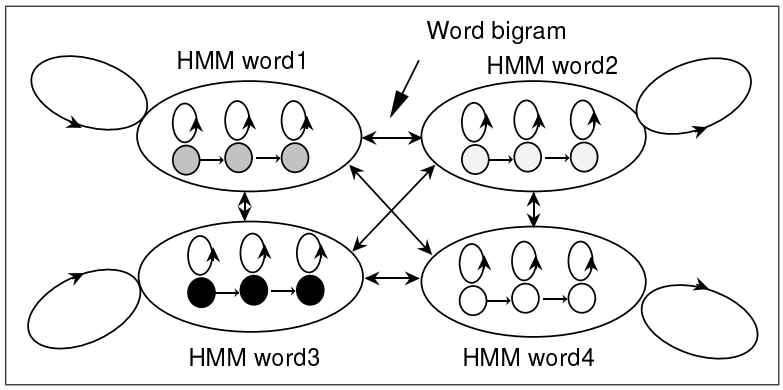


図 4.14: bigram と HMM を組み合わせた Ergodic HMM

第5章　自由発話の音声認識

従来の音声認識システムの多くは丁寧に発声された音声を入力対象にしている。しかし、人間同士のコミュニケーションでは、「あのー」「えーと」などに代表される間投詞や、言い淀みや言い誤りおよび言い直しや倒置などが頻繁に出現する。このような音声でも認識できる、いわゆる自由発話の音声認識が、今後の重要な研究課題になると思われる。しかし、このような発話様式では、認識精度の高い音響モデルの作成は困難であると考えられる。そこで認識性能を向上させるため、perplexity の低い言語モデルが必要になる。

現在、音声認識に用いられている言語モデルは、簡潔さ・有効などの点から単語の bigram モデルが主流である [40]。しかし、単語の trigram モデルの perplexity は bigram より一般的に低いことが知られている。そこで、ここでは 2.3 節で報告したアルゴリズムを基本に言語モデルとして単語の trigram を用いて自由発話の認識を試みた。

5.1　間投詞や言い直しの対策

自由発話においては、「あのー」「えーと」などに代表される間投詞や、言い淀みや言い誤りおよび言い直しなどが頻繁に出現するが [49]、この対策方法の１つに garbage モデルを使用する方法がある。 garbage モデルは、キーワードスポッテングにおいて使用されていたモデルで、キーワード以外の音素を数個の HMM でモデル化しようとするものである [18] [92]。これを Viterbi サーチに組み込み、間投詞や言い直しなどの不要語を garbage モデルで対応する方法を井上らは提案している [23]。

この方法は、間投詞や言い直しを音響モデルで解決した方法と言える。しかし、言語モデルにおいて、間投詞や言い直しを音素の系列とみなし、この音素をスキップすることで同様なことが実現できる。本節では言語モデルに単語の trigram を用いて、この２つの方法で自由発話の認識を検討した。

5.1.1　garbage モデル（音響モデルによる対策）

garbage モデルは、間投詞や言い直しを１個ないし数個の garbage HMM で処理しようとする方法である。 garbage HMM は、あらかじめ全ての音素を学習しておく。そして認識アルゴリズムにおいて 4.3.2.1 節のポーズの HMM

と同様に扱う [22]。この結果、音声データ中の間投詞や言い直しは garbage モデルで認識しながら、言語モデルではこれらの言語現象をスキップすることで自由発話の音声が認識できる [23]。

5.1.2 音素スキップ （言語モデルによる対策）

間投詞や言い直しは、文の全ての場所に出現する可能性があるという点でポーズと似た性質がある。そこで、間投詞や言い直しを音素系列として認識しながら、言語モデルでは音素系列をスキップすることにより自由発話の音声が認識できる。ただし、このようなアルゴリズムでは、音声データ全てが音素系列と認識される可能性があるため、本節ではペナルティとして音素の trigram を使用する。

例えば「“東京都” “港区” “新橋” “あのう (anou)” “1 丁目”」と発声されたとする。また「あのう」は間投詞とする。

このときの 言語モデルの連鎖確率値は $P($ “新橋” | “東京都”, “港区”) $\times P(/a/|/sh/,/i/) \times P(/n/|/i/,/a/) \times P(/o/|/a/,/n/) \times P(/u/|/n/,/o/) \times P($“1 丁目” | “港区” “新橋”) と計算する。

ここで、$P(/a/|/sh/,/i/)$ は音素の trigram でペナルティ、 $P($“1 丁目” | “港区” “新橋”) は単語の trigram で、” あのう” を音素系列と見てスキップしたことを意味する。

この方法は、garbage モデルを言語モデルで実現する方法であるとも言える。また、既に提案されている未知語検出のアルゴリズムと基本的には同一の思想である [6], [31], [26], [29], [44], [17]。ただし、これらの論文では未知語検出を目的にしている。また、使用している言語モデルも異なる。

5.2 自由発話の文認識実験条件

認識実験は、音響モデルには不特定話者の HMM、言語モデルには単語の trigram を使用して行なった。実験条件は表 4.5 とほぼ同じであるが、語彙数やビーム幅などは異る。garbage モデルは、4 状態 3 ループの 10 混合のモデルで、男性話者 12 名の音素バランス 216 単語から作成した。音素の trigram の連鎖確率値は「あのー」、「えーと」などの間投詞を含めて国際会議の予約に関するデータ約 1 万 2 千文章、約 17 万単語から作成した。実験条件を表 5.1 に示す。また全ての実験において 4.3.2.1 節および 4.3.2.3 節で報告したポーズの処理を行なっている。

表 5.1: 文音声認識の実験条件

HMM の学習音声	男性話者 12 名の 736 単語発声
garbage モデルの学習音声	男性話者 12 名の音韻バランス 216 単語
garbage モデル	4-state 3-loop 10 mixture left-right model
音素の数	26
認識単語数	435
ビーム幅	16,384
単語 trigram の値の推定に使用したテキストデータ量	約 1 万 2 千文章 171,978 単語 テストデータのテキストを含む （間投詞は削除）
音素 trigram の値の推定に使用したテキストデータ量	約 1 万 2 千文章 171,978 単語 テストデータのテキストを含む （間投詞を含む）
言語尤度と音響尤度の結合値 α	16
テスト文	261 文

5.2.1 自由発話の音声データ

音声データは以下に示すような方法で収録した。ただし、評価用の話者は 2 名で一般人である。

1. 朗読発話

   テキストを読みあげた音声データ。テキストの内容は 4.3.1.2 節において使用されたテストデータと同一。間投詞や言い淀み・言い直しは無い。このデータは text-closed の実験になる。評価した文数は 261 文である。

2. 疑似自由発話

   間投詞を含むテキストを読みあげた音声データ。間投詞を除いて、「1 朗読発話」と発話内容は同一。言い淀み・言い直しは無い。

3. 自由発話

   話者はテキストを覚えて、その意図を理解し、自由に発話した音声データ。発話内容は「1 朗読発話」と異なる。間投詞や言い直しや未知語を含む。このデータは text-closed data に近いが text-open data の実験といえる。

5.2.2 単語の trigram の平滑化

単語 trigram は語彙数の 3 乗のパラメータの数をもつ。したがって全ての trigram の値を直接推定できるだけの大量のテキストデータを収集することは困難である。そのため、text-open の音声データを認識させる場合、通常 trigram の連鎖確率値は平滑化して使用される。ここでは deleted-interpolation[27]（4.3.3.4 参照）を使用した。また、単語の trigram の値を平滑化した場合としなかった場合の両方で実験を行った。

5.3 自由発話の文認識実験結果

表 5.2 に単語の trigram の連鎖確率値を平滑化しないで認識実験を行なった結果を示す。また、表 5.3 に単語の trigram の連鎖確率値を deleted-interpolation で平滑化して認識実験を行なった結果を示す。なお、平滑化の値は、trigram,bigram,unigram, フロアリングに対して各々$\lambda_3 = 0.35, \lambda_2 = 0.48, \lambda_1 = 0.11, \lambda_0 = 0.06$ となった。

表 5.2: 自由発話の文認識実験結果 （平滑化無し）

		文認識率 (%)		
		base line	garbage	skip-phone
朗	累積文認識率 1	89.7%	83.2%	88.5%
読	～ 2	97.3%	90.5%	96.2%
発	～ 8	100.0%	97.3%	99.2%
話	Word correct	97.5%	93.4%	96.4%
	Word accuracy	96.9%	93.2%	96.0%
疑	累積文認識率 1	41.6%	64.5%	73.3%
似自	～ 2	43.1%	70.2%	79.0%
由	～ 8	44.3%	78.2%	82.8%
発	Word correct	70.6%	81.5%	89.5%
話	Word accuracy	34.2%	76.6%	82.3%
自	累積文認識率 1	10.7%	37.8%	47.7%
由	～ 2	15.3%	46.9%	57.2%
発	～ 8	19.5%	56.1%	66.8%
話	Word correct	44.7%	65.7%	80.9%
	Word accuracy	9.1%	58.9%	73.3%

不特定話者認識; 語彙数:435; ビーム幅:16,384; α:16
trigram の連鎖確率を直接使用

これらの実験から以下のことがわかる。

表 5.3: 自由発話の文認識実験結果 （平滑化有り）

		文認識率 (%)		
		base line	garbage	skip-phone
朗	累積文認識率 1	47.3%	49.2%	46.9%
読	～ 2	52.2%	53.8%	53.4%
発	～ 8	61.5%	61.8%	64.1%
話	Word correct	77.4%	70.1%	77.1%
	Word accuracy	71.9%	68.0%	72.2%
疑	累積文認識率 1	29.0%	36.3%	28.6%
似自	～ 2	30.1%	37.8%	30.9%
由	～ 8	33.2%	42.0%	36.3%
発	Word correct	63.1%	59.6%	63.2%
話	Word accuracy	28.8%	44.3%	29.9%
自	累積文認識率 1	10.3%	16.4%	10.7%
由	～ 2	14.1%	18.3%	13.0%
発	～ 8	17.5%	22.1%	16.8%
話	Word correct	51.0%	41.5%	46.7%
	Word accuracy	27.9%	26.5%	19.2%

trigram; 不特定話者認識; 語彙数:435; ビーム幅:16,384; α:16
trigram の連鎖確率を deleted-interpolation して使用

1. 自由発話において、音素スキップの方法を使用した場合も、garbage モデルを使用した場合も、認識性能は向上する。

2. garbage モデルを使用したときと音素スキップの方法を利用したときの認識率を比較すると、音素スキップの方法を利用したときの方が高い認識性能を得ている。

3. trigram の値を平滑化した場合と平滑化しない場合の認識率を比較すると、平滑化をしないほうが高い認識性能を得ている。これに関しては 5.4 節において考察する。

4. trigram の連鎖確率値を平滑化をしないで、音素スキップの方法を利用することで、自由発話では 47.7%の文認識率が得られた。また朗読発話でも、この処理を加えることの認識性能の低下は少なかった (89.7% → 88.5%)。

自由発話において音素スキップをしたときの誤認識の例を、表 5.4 に載せる。表中の括弧内は、実際の発話内容である。

自由発話では発話内容が朗読発話と異なっている。しかし、文認識率の計算においては、朗読発話の単語と一致した場合に正解とした。そのため、発

表 5.4: 実験において誤りが出力された文（自由発話認識）
（ビーム幅:16,384; α:16）

正解文 （発声内容）→ 1 位出力
会議の宿泊施設についてお尋ねしたいのですが
（会議の宿泊施設についてお尋ねしたいんですけれど）
→ 会議の宿泊施設についてお尋ねしたい んですよ
私共でご紹介できるホテルは京都ホテルと京都プリンスホテルです
（えーと、私共でご紹介できるホテルは京都ホテルと京都プリンスホテルです）
→ 登録を ご紹介できるホテルは京都ホテルと京都プリンスホテルです
ではお名前と住所お願いします
（ではお名前と住所お願いします）
→ ではお名前と ご 住所をお願いします
会議の参加料について教えていただきたいのですが
（えー、会議の参加料について教えていただきたいのですけれども）
→ 会議の参加料について教えて いただけますか
失礼します
（う、失礼します）
→ そうします
京都プリンスホテルに８月４日から８日まで一人部屋をお取りしました
（えーっと、京都プリンスホテルに８月４日から８日まで
えーっと一人部屋をお取りしました ）
→ 国際会議が８月に行われているんでしょうか

話内容と認識結果が合っていても、誤認識とした。（つまり ” おたずねしたいんですが ” が認識されたとしても、朗読発話の文が “おたずねしたいのですが” であった場合、誤認識とした。）したがって実際の認識性能は 47.7%より高い。意味的に正しい文を正解とすると、1 位文理解率で約 75%、8 位までの累積文理解率は 90%になった。したがって、音素スキップの方法は、自由発話の認識において有効であると考えられる。

5.4 考察

1. 自由発話認識における trigram の平滑化に関して

   今回の自由発話の実験では、単語の trigram の値を平滑化をしない方が、deleted-interpolation で平滑化をした場合より高い認識性能が得られた。これは、1 つの原因としてテストデータが text-closed data に近い text-open data であったためと考えられるが、その他にも以下の理

由が考えられる。

音声認識において利用される言語モデルは、通常エントロピー（もしくは perplxity）が低くかつカバー率が広いことが要求される。一般に単語の trigram はエントロピーは低いがカバー率も低い。そこでカバー率を上げるために、deleted-interpolation などの平滑化の方法が利用されている。しかし言語モデルのエントロピーは増加する。一方 garbage モデルや音素スキップは、言語モデルで対応出来ない音声を音素で対応するアルゴリズムである。したがって、この方法を利用したばあい間接的に言語モデルのエントロピーは増加する。したがってこれらのアルゴリズムと deleted-interpolation を組合せると、テストデータにおける perplexity は増加する可能性がある。そのため、認識性能が低下する。

自由発話では、文字化した文章と発話した音素列の差は朗読発話より大きくなる。例えば「会議にー（い）」と発声している音声を「会議に」と文字化している。また、「あのー」「えーと」などの間投詞や言い直しは対話文の 50%に出現する（6.1.4 参照）。したがって、自由発話の認識では、全ての音素を完全に認識する必要はなくて、意味的に合っている文章を出力すれば十分であると思われる。そして、自由発話の認識において使用される言語モデルには低い perplexity が求められ)、言語モデルがカバーできない範囲は garbage モデルや音素スキップで対処するのが妥当であると考えている。

2. 音素スキップと garbage モデルの比較

   今回の実験では、音素スキップの方法が garbage モデルより高い認識性能が得られた。これは、言語モデルが適応できない音声区間は garbage モデルよりも音素モデルで認識したほうが認識性能は高くなることを意味している。しかし、この方法は garbage モデルより一般的に広いビーム幅が必要になると考えている。したがって、語彙数が多い場合やビーム幅が小さい場合、garbage モデルのほうが認識性能は高くなる可能性があると思われる。

3. 間投詞の音素に関して

   間投詞には従来の音素では表現できない音素がある [49]。例えば「んー」（考え込むとき発声している音）は /N/あるいは/uN/の 両者に解釈できる。したがって間投詞に関しては認識単位を単語にするべきであると思われる。

4. 自由発話の認識に関して

   現在自由発話の認識アルゴリズムとしては、garbage モデルなどを使用する方法の他に、キーワードスポッティングを利用する方法や、始めに音素ラティスを作成し次にキーワードを選択する手法 [87][89] などが試

みられている。今後自由発話の認識において、これらの方法も考慮する必要があると思われる。

5.5 まとめ

本章では、自由発話の認識実験について報告した。このような発話に特有な間投詞や言い誤りは、音声のあらゆる場所に出現する可能性があるという点でポーズと似た性質がある。そこで、間投詞や言い誤りを音素の系列と捉え、この音素をスキップをすることにより、文認識率が 10.7% から 47.7%に向上した。意味的に正しい文まで正解とすると、1 位文理解率で約 75%、8 位までの累積文理解率は 90% になった。ただし、この実験は text-closed data に近い text-open data の認識実験である。これらの実験の結果、このアルゴリズムの有効性が示された。

第6章　自由発話音声における音響的・言語的な特徴

近年、連続音声認識の研究が盛んに行なわれ、多くの研究機関で文音声音声システムが構築されている [40],[58],[93],[69]。これらのシステムの多くは、朗読発話のような丁寧に発声された音声を入力対象にしている。しかし、人間同士のコミュニケーションでは、「あのー」「えーと」などの間投詞や、助詞落ち・言い淀み・言い誤り・言い直し・倒置などが頻繁に見受けられる。このような音声の認識が今後の重要な研究課題になると思われる。

この研究の第一歩として、本章では視察によるラベリングをして自由発話の音声データを研究した結果について報告する。自由発話の定義は研究者によって異なるが、ここでは話者がテキストを見ないで対話した音声を自由発話と見なした。そして自由発話と朗読発話の差を見るために、間投詞と言い直しの出現頻度、発話速度、融合ラベルの付与率、HMM による認識精度などを研究した。だたし、音響的な傾向に関して調査した話者は 4 名のみである。

なお、自由発話の視察によるラベリング（音素のセグメンテーションと音素ラベルの付与）には多くの人手が必要であるため、自由発話の音響的な特徴を報告した研究は少ない。文献 [34] において、小林らは日本音響学会のデータベース [24] を利用して、自由発話の文の中に出現するポーズの長さを報告している。

一方、音声を文字化（発声内容を書きおこしたもの）するコストはラベリングのコストよりも少なくてすむため、間投詞や言い直しなどの言語現象を調べた論文は比較的多い。日本語では、間投詞や言い直しの出現頻度を調べた報告 [50],[19],[5] や、助詞落ち・倒置の分析を行なった報告 [97] などがある。英語では自由発話のデータベースとして Air Travel Information Service (ATIS) がよく知られている。このデータベースを利用して自由発話の特徴（ポーズの長さなど）が報告され [99],[100]、従来の音声認識で使用されたアルゴリズムを用いて、認識率を報告した論文が多く見られる [89],[101]。

6.1　自由発話の言語的な特徴

自由発話における言語の特徴については、自然言語処理の立場から 出現する言語表現を調べた報告が既にある [5]。また、山本ら [97] は実際の対話文約

1800 文を名詞文節の助詞落ちや倒置の点から解析している。ここでは、自由発話と朗読発声の言語現象の差を研究する立場から、朗読発話では見られない言語（発話）現象、特に言い直しと間投詞に焦点をあてて、それぞれの出現頻度を調査した。今回調査した会話文は、国際会議の問い合わせの対話文 11054 文である。

6.1.1 研究に用いたデータベース

自由発話の言語的な特徴を把握するための言語データベースとして、ＡＴＲの言語データベースのなかから、申し込み者と事務局員が通訳を通して国際会議の問い合わせをしているデータを利用した（ 3.3.1 節参照)。このデータの収録条件を表 6.1 に示す。申し込み者役にはナレータ（アナウンサや声優など音声を使うのを職業としている人）の他に一般の話者も含まれているが、対応する事務局員役は、この分野の専門家が演じている。また収録は、遮音室の他に通常の部屋でも行なっている。

表 6.1: 研究に用いた自由発話の言語データ

発話内容	国際会議の申し込みに関する参加者と事務局の対話
データ量	3178 対話、11054 文
発話様式	自由発話 「トピック」（質問項目と、その背景に関する情報）や「バックグラウンド」（会話の前提になる背景）を詳細に設定して対話したもの。
発話環境	
1 通常の部屋	大部分が家庭用のカセットテープレコーダで録音。外来雑音も混在。
2 スタジオ録音（遮音室）	ＤＡＴで録音。明瞭。
話者	
1 事務局員役	当該分野の専門家
2 申し込み者役	ナレータ + 一般話者（複数話者）

6.1.2 自由発話の文例

自由発話では文の定義が曖昧になる。ここでは、発話権が相手から渡されて相手に返すまでの話者の発話を「文」と定義した。したがって、通常２文と考えられる文が１文になる。発話の例文を表 6.2 に示す。この例文において“『 』”で括られた範囲は１文を示している。また“『”の前の番号は、文番号である。

表 6.2: 文例

１『[あっ、あの］わたくし、
武蔵野電機システム開発研究所の小金沢と申します。』
２『[あのー］参加の申込みをしていたんですけれども、
［ちょっと］出られなくなりましたので、
［あの］キャンセルしたいんですが。』
３『はい。［あのー］キャンセルは書面にてっていうふうに
書いてあるんですけども、
どういうふうにしたらよろしいんでしょうか。』
４『[ああー]。』
５『[あっ］そうですか。』
６『[あっ］そうですか。』
７『それでですね、
［えーと］８０パーセント返していただけるのは
９月３０日までというふうに書いてあるんですけども、
［あのー］３０日までにそれが届けばですか。
そ れとも、こちらが出した［その］消印が
３０日でも（かまい）かまわないんでしょうか。』
８『[ああ］そうですか。』
９『[あっ］そうですか。』
１０『わかりました。じゃ、登録ナンバーと名前、住所、所属なんかを
（かき）書いて、書いたものをお送りすればよろしいですね。』
１１『はい、どうも、ありがとうございました。』

6.1.3 自由発話の文の長さ

図 6.1 に、自由発話における文の長さと出現回数を示す。

漢字仮名混じり表記に書き下したテキストデータを調査した結果、自由発話の１文の平均文字数は 25.4 文字、もっとも短い文は「あ」の１文字、もっ

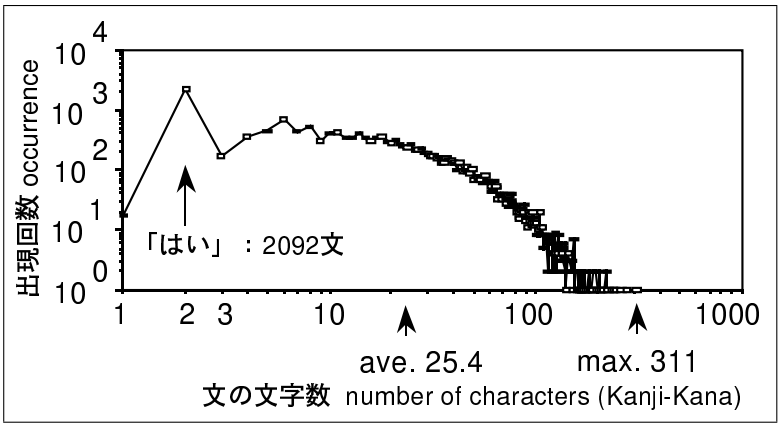


図 6.1: 自由発話における文の長さの分布

とも長い文は 311 文字であった。最も出現頻度の高い文は、「はい」の 2092 文であった。また、自由発話の文の 73%は 32 文字以下であった。長い文の例を図 6.2 に示す。

> えー、松下の場合にはですね、もうすでに、まー、あのー、見学コースっていうのが設定されておりまして、えー、会議の参加者のみならず、いろんな興味のある方々、これは日本人の方も外国人の方も見れる訳ですが、そういった、松下電器が、今までどの様な製品を作り、現在どの様なシステムで、えー、いろんな製品を作っておるか、そして、今後将来、松下がどういう方向性を目指してるか、という過去現在未来といった様な、製品の製作展開等のコースを見て頂くことになります。

図 6.2: 長い文の例

6.1.4 自由発話における間投詞および言い直しの出現頻度

自由発話の音声には、「あのー」や「えーと」などの間投詞（冗長語）や、言葉の言い直しおよび言い誤りなどがある。これらの言語現象は、朗読発声では通常出現しないため、従来の文法の枠組では、あまり考慮されていない。そこで、自由発話における間投詞や言い直しの出現頻度を研究した。

なお、本章では間投詞を『活用しない自立語。主語・述語によらない。言い淀む場合などに、文の中に挿入されて用いられる。間投詞を取り除いても文の文法性および意味には影響しない [98]。』と定義した。また言い直しを『前にいった事の誤りを訂正してもう一度言い換える。』もしくは『他の適当なや

さしい言葉で言う。』と定義した。また言い淀みを『言おうとしてためらったり、話しの途中でちょっと言葉につまったりする。』[73] と定義した。なお表 6.2 の例文において、“[ ]” で括られた個所は間投詞、“( )” で括られた個所は言い直しを意味している。

結果を図 6.3 に示す。この結果において、間投詞も言い直しも共にない文は、全体の約 5 割であった。これらの多くは「はい」(23%) 「もしもし」(5%) 「はい、わかりました」 (3%) 「どうも、ありがとうございました」(1.5%) などの定型文で、この種類の 8 割の文は 14 文字以下の短い文であった。

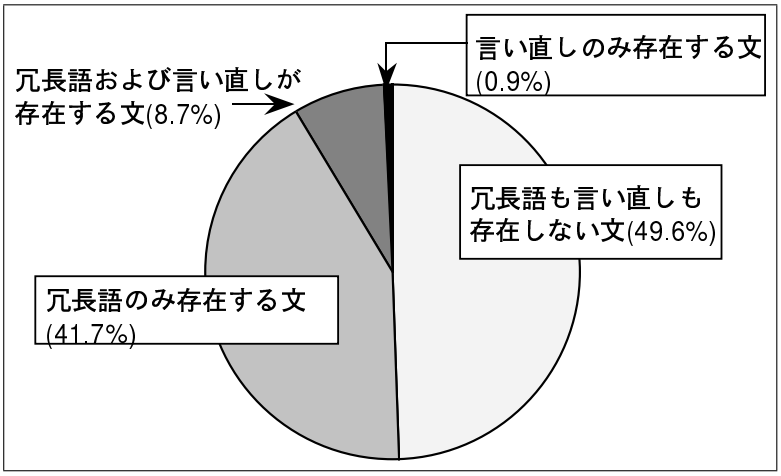


図 6.3: 自由発話における間投詞や言い直しの出現頻度

自由発話の文の約 5 割は間投詞を含み、多くの単語が続く文の多くは間投詞を含んでいた。ただし、間投詞には個人差が多く、間投詞を多く話す話者とあまり話さない話者がいた。また、一人の話者が話す間投詞の種類は限られていた (”あのー ”、”えーと ”)。言い直しがある文は自由発話全体の約１割であった。そして、「はい」「もしもし」などの独立語も間投詞に含めた場合、全体の文の 83%(9121 文) は間投詞があった。この中で文頭に間投詞があるものは、全体の文の 65.8%(7303 文) であった。また、言い直しがある文の 46.1%は、言い直しの前もしくは後に間投詞が付加されていた。(言い直しの前に間投詞が付加されているのが 14%、言い直しの後に間投詞が付加されているのが 24%、言い直しの前後ともに間投詞が付加されているのが 8%であった。)

なお、今回調査した自由発話のデータは、ナレータや実際の事務局員など、言葉の対応に慣れた話者が発話した音声である。したがって、言葉の応対に慣れていない一般の話者では、間投詞や言い直しの出現頻度が増加する可能性がある。

6.1.5 自由発話における間投詞の種類と出現確率

自由発話において観測された間投詞の種類のなかで、出現頻度の高いものを図 6.4 に示す。また観測された間投詞を表 6.4 に示す。この図から、間投詞の種類はかなり多いが、上位４種類で間投詞全体の出現頻度の約７割を占めていることがわかる。

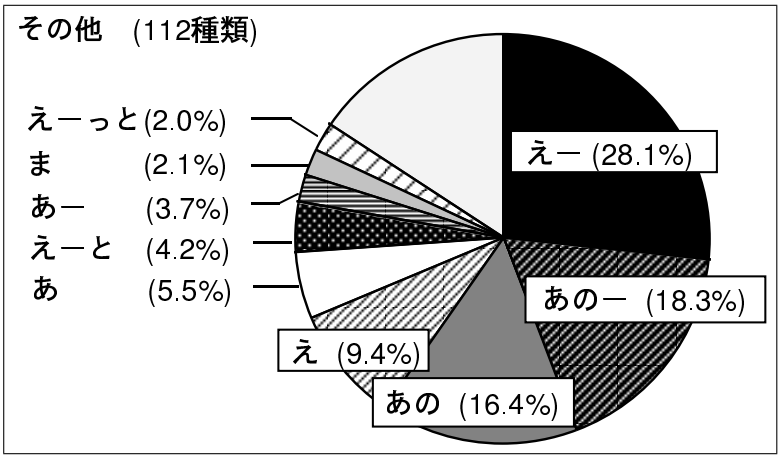


図 6.4: 自由発話における間投詞の種類と出現頻度

自由発話中では、しばしば間投詞と言い淀みの区分が不明確になる。例えば「100 パーセント、え 日本語と英語で行われます。」における「え」は、間投詞とも「英語」の言い淀みとも解釈できる。また、語尾音の継続時間には、話者間に大きなバラツキがあるため、語尾の伸びる語と伸びない語（例えば「えー」と「え」）の決定は、文字化した人の判断に依存している。ここで示した間投詞の出現頻度のデータには、このような意味で曖昧さがある。

なお、話し相手と対面して話す自由発話に対して、電話のような音声のみによる対話では、間投詞は相手の注意を促す役割を持つ場合がある [19]。このため、今回調査した間投詞の出現頻度は、高めに評価されている可能性がある。

また、自由発話における間投詞の出現頻度は多くの研究期間で報告されている。文献 [34] や文献 [80] では日本音響学会連続音声データベースの書き起こしテキストを調査して報告している。また、文献 [19] では、本章で使用したデータベースの小量のときの開始符合の種類を示している。この場合、「えー」、「えーっ」などの単語が多いことを報告している。また、文献 [66] では NHK ラジオ第一放送の電話相談番組を書き起こしている。これらの報告と比較すると比率に多少の違いがあるが、高頻度で生じる間投詞に関してはほぼ同じ割合といえる。

6.1.6 自由発話における言い直しの種類と出現頻度

自由発話において特有な言い誤りは、文法的、意味的な前後関係を考慮して決定する必要がある。また、言い淀みは音声を注意深く聞いて決定する必要がある。したがって言い誤りや言い淀みの言語現象は話者が言い直さないかぎり検出するのは困難である。したがって、ここでは言い直しの出現頻度のみを研究した。調査は言い直しを含む 200 文に対して行なった。この言い直しの分類と出現頻度を、図 6.5 に示す。また例文を以下に示めす。例文中においてアンダーラインは言い直しを意味する。

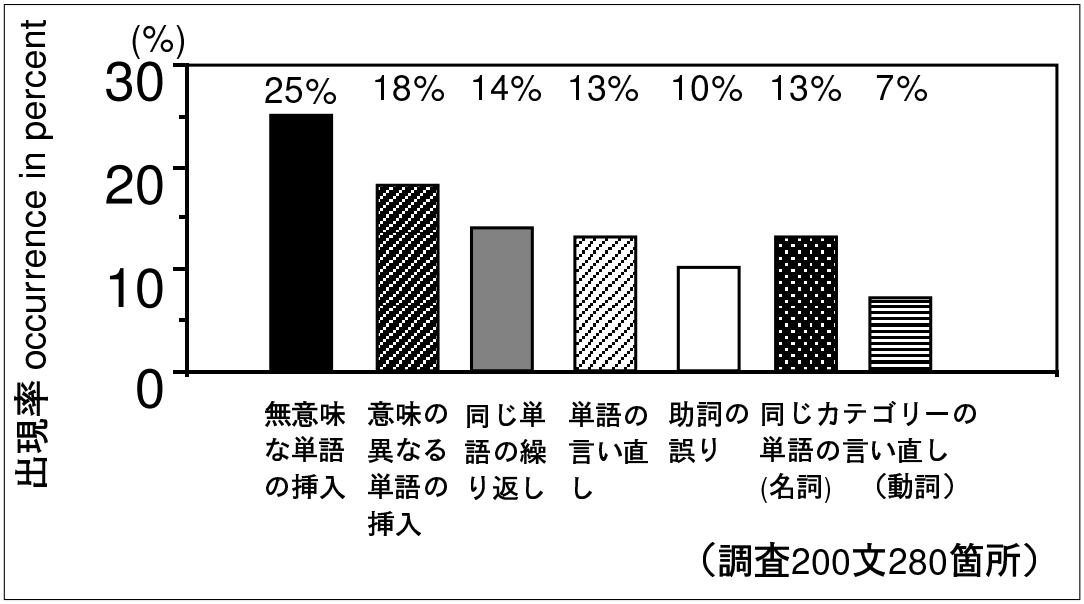


図 6.5: 言い直しの出現頻度

自由発話における言い直しの例文

1. 無意味な単語の挿入　25%

   - 日本語から英語へというように
     、と、翻訳を、す、あ、通訳をするコンピュータを開発している
     （「通訳」と言おうとして「翻訳」と言い間違いをし、これに気がついて直そうとして言い淀んでいる。）
   - えーっと、あのー、こ、会議期間中は特にあのー、バスを運行しておりまして、土曜ダイヤでバスが、あのー、運行するようになっております。

（原因が不明、「こ」は、無意味な音の発声であるため、間投詞と判断される可能性がある。）

- 最終的な、えーっと、草稿、原、えーとスピーチ原稿を提出していただきたいと思います。
（「原稿」と言おうとして「草稿」と言い間違いをし、これに気がついて直そうとして言い淀んでいる。）
- パンフレットの方を 拝、見ていただきましたら
（「拝見」と言おうとして敬語の間違いをして言い淀んでいる。）

2. 意味の異なる単語の挿入　18%

- あの、そのようなことが、あの、そちらの方に お教え、お知らせできないんです。
（「知らせる」を「教える」に言い間違えている。）
- タクシーに、あのー、京都駅からお乗りになれば、大体３５分か４０分位で着きますし、旅費、料金としては、大体１５００円位になります。
（「旅費」と「料金」は、意味的にはほとんど同じであるため、『丁寧な言葉への言い直し』とも分類できる。）
- この件に関しましては、えーっと、大阪まで、あのー、新幹線で来られますと、飛行機で来られますと４５分間位で参ります。
（「飛行機」を「新幹線」と言い間違えている。文全体の挿入の誤り。）

3. 同じ単語の繰り返し　14%

- えーっと、その、その中でちょっと、あの、クレジットカードをね書類の方は、
（「その中」を１つの単語と捉えたならば『単語の言い直し』とも解釈できる。）
- 会議の内容なんかをかいつまんで お話、お話し下さればと思うんですが。

4. 単語の言い直し　13%

- あの、この、クレ、クレジットカードというのは本来外国人のゲストの方
- 従いまして、２、あ、２時間半位で東京から国際会議の行なわれる場所まで行けるわけですから、

5. 助詞の誤り　10%

- まだ割引を私の方で、あのー、すること に、はできないんですが
- はい、それ で、はそうですね。
- コンピュータによる同時通訳 を、に関する、あのー会議を開こうということです。
- オーバーヘッドープロジェクタと２インチ×２インチのスライドを、と使えるようになっています。

6. 丁寧な単語を用いた言い直し （名詞） 13%

- えーっと、郵送でＶＬＤＢ８６の、えーと、会議事務局、国際会議事務局宛にお送りいただきたいと思います。
（意味的には『 同じ単語の繰り返し』ともみなせる。）
- それで、えーっと、受領の通知は、受け取りの通知は１２月３１日までに出させていただきます。
- これは現在の為替 でいきますと、レートでいきますと、大体 16,000 円程になりますので
- その次に日本の総理大臣中曽根首相から 挨拶を、スピーチをすることになってます。

7. 丁寧な単語を用いた言い直し （動詞） 7%

- はがきででも 来られない、参加できないという風に、御通知いただければ、
- ええ、外国人の申し込みの方は、現在までで１３名で あり、ございます
- そうですか、という、といいますと、それは英語でしなければいけないわけでしょうか。

この図における言い直しの分類は、かなり主観的である。例えば、単語の意味の違いは明確でないため『意味の異なる単語の挿入』と『丁寧な単語での言い直し』の区別の差は明確でない。また、日本語では単語の概念が曖昧なため、『同じ単語の繰り返し』と『単語の言い直し』の区別の差も明確でない。

なお文献 [75] では言い直した単語に着目して、言い直した単語の長さを報告している。これを見ると言い直しの 59% は、言い誤った単語を直ちに言い直している。この傾向はほぼ同じである。また文献 [62] においてもほぼ同様な結果が見られる。この論文では単語にならない syllable が 39%、直後に言い直しているのが 52%であることが示されている。これらの結果は、今回の結果に類似している。

6.2　話者ごとの自由発話の言語的な特徴

自由発話における個人差を見るために 4 名の話者を個別に研究した。この音声データは、ナレータ（アナウンサや声優など音声を使うことを職業としている人）が申し込み者の役になって発話しているため、舌打ちの音などはほとんどない。事務局員側とは完全に分離されて録音されているため音声区間の重畳はない。 収録は遮音室で行なわれたためドアの開閉音などの日常雑音はない。したがって、この音声データは自由発話としてはかなり クリーンな音声であると言ってよい。このデータの収録条件を表 6.4 に示す。ただし、各々の話者の発話内容は異なっている。なお、通常の話者のデータはラベリングされていないため、研究対象から除外した。

また、自由発話の音声データを文字に書き起こした後に、間投詞や言い直しを削除して作成したテキストを自由発話の発声者と同一の話者が発声し、朗読発話の音声データとして使用した。したがって、同一話者における自由発話と朗読発話の発話内容は、間投詞および言い直しを除いてほぼ同一である。

6.2.1　話者ごとの間投詞や言い直しの出現頻度

自由発話には、「あのー」や「えーと」などの間投詞（冗長語）や、言い直しがある。これらの１文あたりの出現頻度を各話者ごとに調べた。なお表 6.2 の例文において、“[　]” で括られた個所は間投詞、“（　）” で括られた個所は言い直しを意味している。

各話者ごとの間投詞の出現頻度を図 6.6 に、言い直しの出現頻度を図 6.7 に示す。

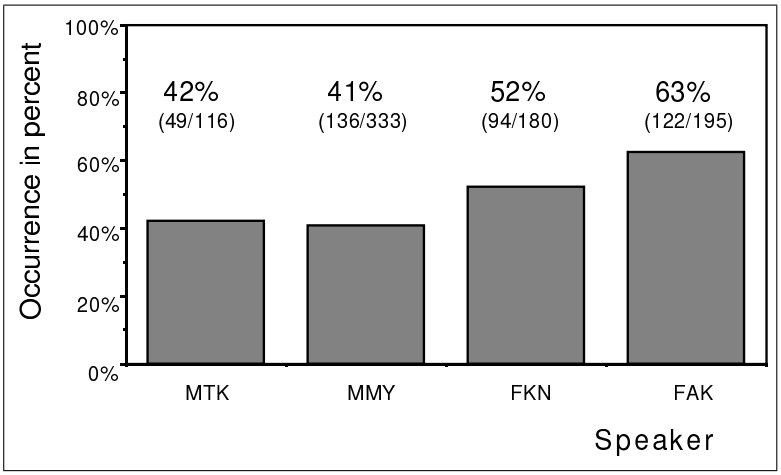


図 6.6: 話者ごとの間投詞の出現頻度

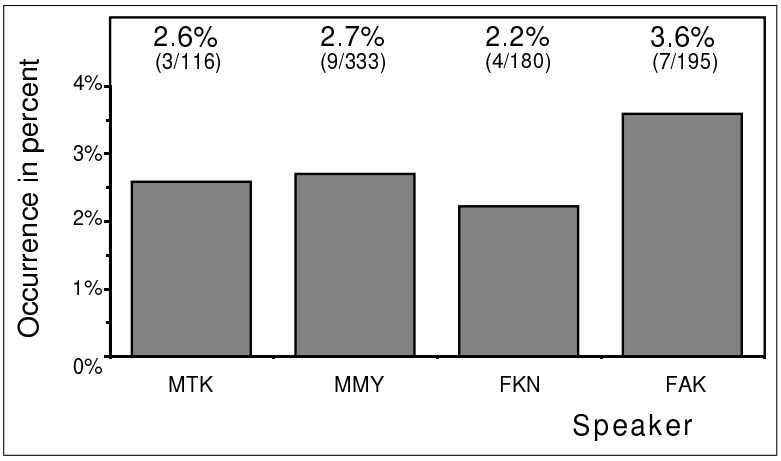


図 6.7: 話者ごとの言い直しの出現頻度

これらから次のようなことが判る。

1. 間投詞は、話者によって相違が見られるが、文章全体の 40%から 60%の文に出現する。
2. 言い直しは、話者によって相違が見られるが、文章全体の 2%から 4%の文に出現する。

6.2.2 間投詞の種類と出現頻度

間投詞には多くの種類があるが、出現頻度の高い間投詞は限られていることが既に報告されている [50],[33],[66]。ここでは各話者ごとに、出現頻度の高い上位４つの間投詞の種類と、その出現頻度を調査した。この結果を図 6.8 に示す。

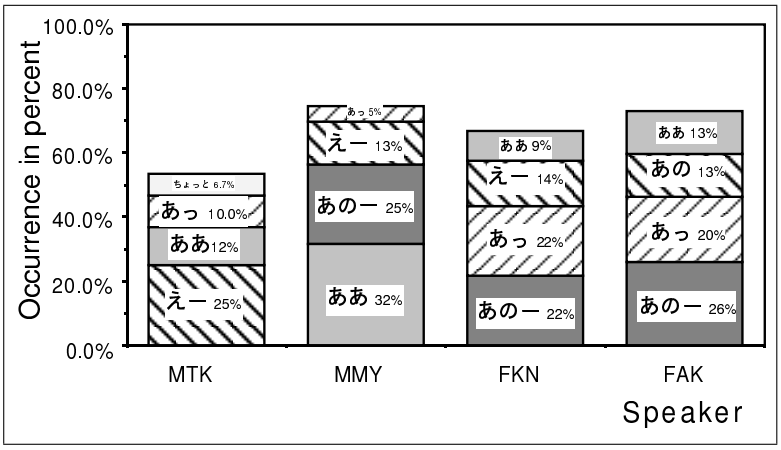


図 6.8: 間投詞の種類別の出現頻度

表 6.3: 自由発話における間投詞の一覧

間投詞	出現回数	間投詞	出現回数	間投詞	出現回数
「あ」	604	「えーっとお」	1	「その」	115
「あー」	268	「えーっとですね」	8	「そのー」	48
「あーっと」	2	「えーと」	466	「だか」	1
「あーと」	1	「えーとー」	4	「ちょっと」	8
「あーん」	5	「えーとですね」	3	「つ」	2
「ああ」	7	「えーまあ」	3	「で」	61
「あっ」	151	「えーん」	2	「でー」	13
「あっと」	1	「ええ」	13	「でぃ」	1
「あと」	1	「ええー」	1	「と」	77
「あなー」	1	「えええっと」	1	「とー」	11
「あの」	1809	「えっ」	22	「ねー」	1
「あのー」	2025	「えっーと」	4	「のー」	1
「あのーえー」	1	「えっと」	62	「は」	4
「あのう」	77	「えっとー」	11	「はあ」	1
「あのうー」	3	「えっとおー」	1	「はあー」	2
「あのと」	1	「えと」	47	「ははぁーん」	1
「あれ」	1	「えとー」	13	「ひ」	1
「あん」	1	「えへへっ」	1	「ふーん」	2
「い」	26	「えん」	1	「ま」	263
「いー」	58	「お」	59	「まー」	8
「いゃー」	1	「おー」	196	「まぁ」	186
「いやー」	2	「おーえー」	1	「まぁね」	1
「う」	23	「おっ」	2	「まあ」	176
「うー」	71	「ぐっ」	1	「まああのう」	1
「うーん」	26	「こう」	9	「まあまあ」	1
「うーんと」	2	「この」	9	「まっ」	5
「うっ」	1	「このー」	4	「も」	2
「うん」	7	「じゃ」	4	「もう」	1
「え」	1040	「じゃー」	1	「よ」	1
「えー」	3105	「じゃあ」	1	「りー」	1
「えーえ」	1	「す」	8	「わあ」	1
「えーちょっと	1	「すー」	2	「わっ」	1
「えーっ」	1	「すい」	1	「ん」	27
「えーって」	1	「すっ」	2	「んー」	19
「えーっと」	256	「せ」	1	「んっ」	1
「えーっとー」	2	「そ」	2	「んっと」	1
「えーっとえー」	1	「そう」	1	「んで」	1
				「んと」	2

表 6.4: 研究に用いた自由発話の音声データの収録条件

話者	ナレータ４名
収録環境	遮音室
発話内容	国際会議の申し込みに関する参加者と事務局の対話
発話様式	自由発話 「トピック」（質問項目と、その背景に関する情報）や「バックグラウンド」（会話の前提になる背景）を詳細に設定して対話したもの。
入力系	マイクロフォン、ＤＡＴ録音
データ量	13 対話 116 文 3943 音素 (MTK) 18 対話 333 文 11520 音素 (MMY) 13 対話 180 文 6918 音素 (FKN) 14 対話 195 文 7588 音素 (FAK)

これらから次のようなことが判る。

1. 間投詞全体の 50%から 75%は、使用頻度の高い４種類の間投詞で占める。
2. 使用頻度の高い間投詞の種類は、話者によって相違がある。ただし、「あー」、「あのー」、「ああ」など似ている。

6.3 話者ごとの自由発話の音響的な特徴

6.3.1 ラベリング作業からみた自由発話

音声データのラベリングの作業において見受けられた自由発話の音素の定性的な特徴を表 6.5 に示す。なお、ラベリングの基準は文献 [82] に従った。これらから自由発話では音素境界がかなり曖昧になっていることや、従来の朗読発話には見られない音素が現れていることがわかる。

表 6.5: 自由発話の音素の定性的特徴

1	文の語尾の音素の発音が弱くなることがある。 （例: 「なんですか」の「か」がほとんど聞こえない。） （ただし、発声が弱くでも音素境界が明瞭な場合がある。）
2	母音/a,i,u,e,o/全てが無声化することもある。 （朗読発話では/a,e,o/は、あまり無声化しない。） （例：「それで」の/o/と/e/が無声化する場合がある。）
3	判断できない不明瞭な音素がある。 （例:「んー」（考え込むとき発声している音）は /N/あるいは/uN/の 両者に知覚できる。）
4	子音/r/をともなう音節の発音が全体的に弱い。 （例:「そうすると」の「る」がほとんど聞こえない。）
5	母音（特に、文末の母音『ａ』）の第１フォルマントが あらわれないことがある。

6.3.2 融合ラベルの付与率から見た自由発話

1. 話者ごとの融合ラベルの付与率

   ＡＴＲでは、発話テキストを参照しながら人手で音素境界を決定するラベリング作業において、音素境界が不明瞭な音素区間に対して付与するラベルのことを融合ラベルと呼んでいる。この融合ラベルの付与率を 4 名の話者の自由発話と朗読発話において調べた。この結果を図 6.9 に示す。この図から以下のことがわかる。

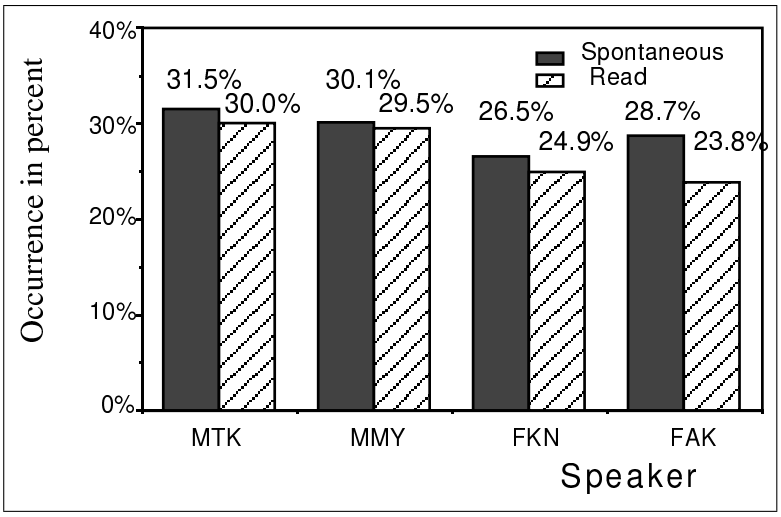


図 6.9: 話者ごとの融合ラベルの付与率の変化

(a) 自由発話の融合ラベルの付与率は朗読発話より若干高く、自由発話では、全音素の 25%から 32%が融合ラベルに、朗読発話では、全音素の 24%から 30%が融合ラベルになる。

(b) 自由発話、朗読発話共に、融合ラベルの付与率に話者間の相違が見られる。

(c) 自由発話と朗読発話を比較すると、融合ラベルの付与率の増加の割合に話者の相違が見られる。話者 MMY では 2%(29.5% → 30.1%) しか増加しないのに対し、話者 FAK では 21%(23.8% → 28.7%) 増加する。

なお、 /i,y/, /u,g/, /N,g/, /N,j/, /f,u/, /i,m/, /k,u/, /u,h/, /u,w/, /u,y/, /u,n/, /N,b/, /N,d/, /N,n/, /a,a/, /e,e/, /e,i/, /i,i/, /k,i/, /o,o/, /o,u/, /s,u/, /sh,i/, /u,u/ などの音素環境は朗読発話、自由発話ともに、融合ラベルになりがちであった。また、文末の 2 音素を調査したところ、 母音では/e/と/u/、子音では/g/と/n/が融合ラベルになりやすかった。

2. 発話様式の違いによる融合ラベルの付与率の変化

話者 2 名において単語発話、文節の朗読発話、文の朗読発話、自由発話における融合ラベルの付与率、発話速度、および音素認識誤り率を調査した。これらのデータは、文節の朗読発話と文の朗読発話の発話内容は同一であるが、単語発話および朗読発話および自由発話の発話内容は異なる。また、単語発話、文節の朗読発話、文の朗読発話の発話内容は話者間に相違はないが、自由発話では、各話者の発話内容は異

なっている。なお単語発話のデータは ATR のデータベースにおいて通称 (D0-D5)、文節の朗読発話は通称 DSA、文の朗読発話には通称 DSC と呼ばれているものを使用した。

融合ラベルの付与率の結果を図 6.10 に示す。この図から読みとれることを以下に示す。

(a) 自由発話と文の朗読発声を比較すると、融合ラベルの付与率は話者 MTK では 33%(23.9% → 31.7%)、話者 FKN では 15% (23.0% → 26.4%)、増加する。

(b) 音素別に自由発話と文の朗読発声を比較すると、母音では/a/の増加が顕著である (MTK:4.0% → 13.3%, FKN:3.9% → 8.1%)。子音では、/m/の増加が著しい (MTK:1.0% → 18.1%, FKN:1.6% → 10.8%)。

(c) 自由発話では、全音素の約 1/4 以上が融合ラベルになる。

(d) 単語発声・文節単位の朗読発声・文単位の朗読発声では融合ラベルの付与率に話者の相違は見られない。しかし、自由発話では話者の相違が見られる (MTK:31.7%, FKN:26.4%)。

(e) 単語発声・文節単位の朗読発声・文単位の朗読発声・自由発話の順に融合ラベルの付与率が増加する。

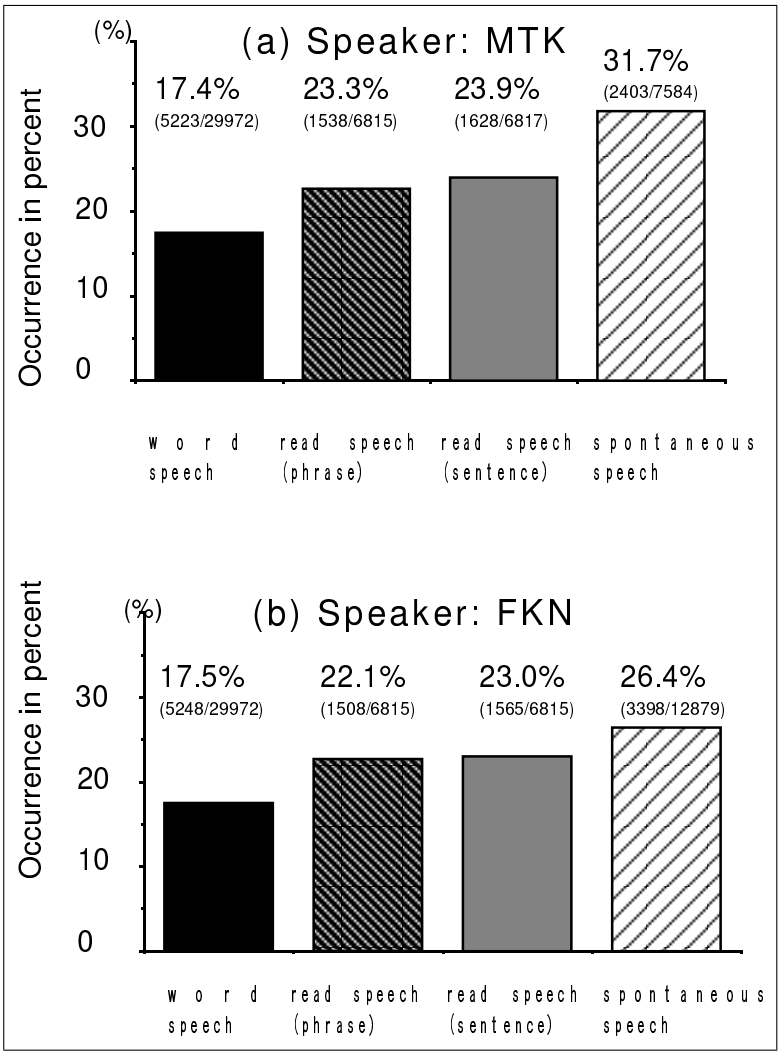


図 6.10: 発話様式の違いによる融合ラベルの付与率の変化

6.3.3 発話速度からみた自由発話

1. 話者ごとの発話速度の変化

   ここでは、自由発話と朗読発話の発話速度の差をモーラ速度で調査した。ただし、息つぎなどの長いポーズ区間および間投詞および言い直しの音声区間は除去した。この結果を図 6.11 に示す。

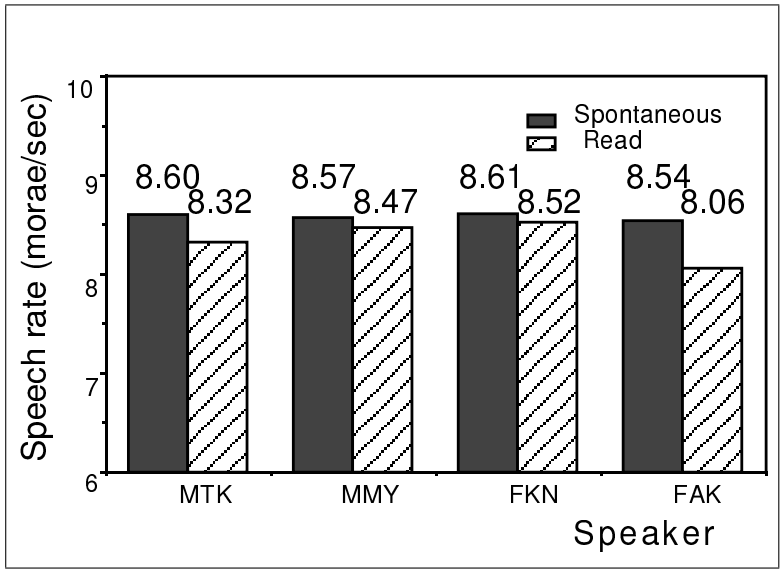


図 6.11: 話者ごとの発話様式の違いによる発話速度の変化

また、自由発話および朗読発話における各音素の平均音素継続時間を表 6.6 に示す。ただし融合ラベルが付与された音素は評価対象から削除した。

表 6.6: 各音素ごとの平均音素継続時間

(a) 自由発話

話者	MTK		MMY		FKN		FAK	
音素	平均 (ms)	標準偏差	平均 (ms)	標準偏差	平均 (ms)	標準偏差	平均 (ms)	標準偏差
a	92.6	49.1	87.6	41.0	86.1	38.2	93.4	63.4
i	72.0	36.1	65.1	39.1	64.2	36.8	70.5	45.5
u	85.0	62.0	77.0	53.8	69.0	43.4	77.7	45.5
e	93.5	64.3	92.7	72.2	80.6	44.3	85.0	53.7
o	91.8	61.5	97.4	71.6	92.5	59.6	101.3	74.5
p	61.4	7.1	17.4	6.9	20.7	7.8	12.8	4.5
t	41.4	21.0	20.4	7.1	17.2	5.7	14.8	5.8
k	49.3	19.9	36.3	19.3	30.2	10.5	26.6	12.1
b	48.3	13.4	11.1	5.0	11.8	4.4	11.7	2.4
d	43.5	16.9	12.0	4.7	11.6	4.0	10.8	3.3
g	45.9	25.3	13.8	4.7	13.9	5.5	14.3	5.2
m	53.1	17.3	47.1	19.6	52.6	15.6	51.8	17.2
n	48.2	18.3	41.6	20.0	47.9	19.1	53.4	24.6
N	75.7	24.0	66.6	37.5	55.2	28.2	70.7	32.4

(b) 朗読発話

話者	MTK		MMY		FKN		FAK	
音素	平均 (ms)	標準偏差	平均 (ms)	標準偏差	平均 (ms)	標準偏差	平均 (ms)	標準偏差
a	100.2	57.5	88.7	36.3	88.0	33.3	91.5	43.1
i	81.3	46.9	70.8	36.3	63.6	30.9	70.7	35.1
u	76.1	48.6	71.2	42.0	66.8	39.4	73.3	38.5
e	92.2	67.3	85.6	58.6	73.6	28.6	82.3	41.8
o	86.9	50.2	91.7	53.5	86.5	38.5	94.5	46.7
p	61.3	14.1	17.6	7.4	16.8	4.5	17.9	8.9
t	40.1	22.9	18.6	6.7	15.3	4.6	13.8	4.3
k	42.5	18.7	35.0	18.1	28.1	10.5	28.1	12.8
b	52.9	38.9	9.7	2.2	11.6	5.1	10.0	0.0
d	43.1	18.7	11.7	4.1	11.3	3.1	10.0	2.5
g	45.6	26.7	19.7	9.8	13.6	5.7	13.5	4.0
m	53.4	18.1	48.1	19.0	47.2	14.2	50.1	14.6
n	47.3	15.5	44.5	21.7	44.1	17.6	48.4	16.1
N	103.6	57.4	70.2	33.4	63.3	37.7	65.3	24.1

これらの結果から以下のことがわかる。

(a) 自由発話の発話速度は朗読発話より、やや早く、自由発話の発話速度は 8.5(morae/sec) から 8.6(morae/sec) である。

(b) 朗読発話の発話速度は自由発話より話者の相違が大きく 8.1(morae/sec) から 8.5(morae/sec) である。

(c) 話者 FAK を除くと、自由発話における母音/a/の平均音素継続時間は朗読発話より短い。しかし母音/u/,/e/,/o/は朗読発話より長い。

(d) 話者 MTK を除くと、自由発話の音素継続時間の分散は朗読発話より大きい音素が多い。

2. 音素/a/と/k/の継続時間毎の分布図

なお、話者 MMY における音素/a/と/k/の継続時間毎の分布図を図 6.12 に示す。この図では、横軸は音素の継続時間で、縦軸は出現回数である。この図から、音素/a/では自由発話の音素継続時間の分散は朗読発話より大きいことがわかる。しかし音素/k/では継続時間の分散の差が小さいことがわかる。

6.3.4 認識精度 (phone accuracy) から見た自由発話

ここでは自由発話と朗読発話の差を、連続音素認識実験を行ない音素正解率 (phone correct) および音素認識精度 (phone accuracy)[60],[15] （4.3.1.3 節参照）で評価した。

1. 実験条件

特定話者の同一発話様式の認識実験を行なうために、同一話者の同一発話様式の音声データの、文番号の奇数番目を学習データに偶数番目を評価データにした。学習プログラムには主に HTK Software Tools[15] を使用した。特徴パラメータには LPC ケプストラムを使用し、HMM には対角共分散の混合連続分布型を用いた。表 6.7 に実験条件を示す。

認識実験は以下のようにしておこなった。

(a) 学習データにおいて、融合ラベルが付与されなかった音素のみを切り出して Baum-Welch アルゴリズム [4] を用いてパラメータの再推定をする。学習回数は 10 回。

(b) 学習データを文単位で連結学習する。学習データは間投詞や言い直しを含む。学習回数は 3 回。

(c) 学習データと同一話者・同一発話様式の評価データを文単位で連続音素認識 (one-pass DP) する。なお評価データは間投詞や言い直しを含む。

表 6.7: 音素認識の実験条件

認識対象	26 音素
サンプリング周波数	12kHz
話者	男性 2 名、女性 2 名のナレータ
学習データ	約 50 文
音響パラメータ	log power + 16 次 LPCcepstrum
	+ Δlog power + 16 次 Δcepstrum
フレーム窓長	20ms
フレーム周期	5ms
LPC 分析	16 次
打ち切り次数	16 次
音素モデル	4-state 3-loop 3 mixture
	Gaussian continuous HMM (diagoal)

(d) 評価データの音素ラベルを正解として、音素正解率 (phone correct) と音素認識精度 (phone accuracy) を計算する。

2. 話者ごとの自由発話における音素認識率

図 6.13 に、認識実験の結果得られた音素正解率 (phone correct) と音素認識精度 (phone accuracy) を示す。また母音の音素認識誤り傾向を表 6.8 に示す。

これから次のような結果が示される。

(a) 自由発話は朗読発話と比較して、音素正解率も音素認識率も低下する。

(b) 自由発話の正解率 (phone correct) は、65% ~ 72%である。

(c) 自由発話の認識精度 (phone accuracy) は、58% ~ 63%である。

(d) 自由発話は朗読発話と比較すると認識精度は 7% ~ 10%程度低下する。

(e) 各音素の認識率をみると、母音の/u/の認識精度が他の音素と比較して低い。

3. 発話様式の違いによる音素認識率

ここでは各発話様式の差を音素認識誤り率で評価した。音素モデルとして混合連続分布型 HMM を用い、認識アルゴリズムには Viterbi サーチを用いた。ただし、融合ラベルを付与された音素は実験では用いなかった。また学習データとして単語発声から視察によって切り出した音素を

使用した場合と、同一発話様式の音声データから視察によって切り出した音素を使用した場合の、2 種類の実験を行なった。

実験は表 6.7 とほぼ同一である。たたし、学習データに単語発声を使用した場合、HMM の混合数は 10 mixtures で、その他は 3 mixtures である。学習データに単語発声を使用した場合の、各発声様式における音素認識誤り率を、図 6.14 に示す。また、同一発話様式の音声データを２つにわけ、一方を学習データとし、一方をテストデータとして実験した場合の音素認識誤り率を、図 6.15 に示す。これから次のような結果が示される。

(a) 学習データが単語発話のとき、自由発話の音素認識誤り率は高い。朗読発声の音素認識誤り率と比較すると、ナレータ MTK は約 160% 程度増加し (21.6% → 37.6%)、ナレータ FKN では約 240%も増加している (18.8% → 44.4%)。

(b) 学習データに自由発話の音声を利用することにより、音素認識誤り率は大きく低下する (MTK:37.6% → 16.0%, FKN:44.4% → 15.0%)。学習データが単語発声のときの文の朗読発声の音素認識誤り率 (MTK:21.6%, FKN:18.8%) より低くなる。

(c) 自由発話を学習データとした場合、母音の中では/u/の認識誤り率が高い (MTK:43.9%, FKN:27.9%)。また、調査音素の数が少ないため明確ではないが、子音では/w/の認識誤り率が高い。(MTK:78.9%, FKN:66.7%)、

(d) 単語発声、文節単位の朗読発声、文単位の朗読発声、自由発話の順に音素認識誤り率が増加する。

(e) 学習データが同一発話様式の場合、各発話様式において話者の相違はあまり見られないが、学習データが単語発話のとき、話者の相違が見られる。

6.4 まとめ

1. 間投詞の出現頻度と種類に関して

   今回の研究では、話者によって相違があるが、間投詞が出現する文は文章全体の 40%から 65%を占めることが示された。しかし、電話のような音声のみによる対話では、間投詞は相手の注意を促す役割を持つ場合がある [19]。したがって話し相手と対面して話す自由発話では、この出現頻度より低くなる可能性がある。

   なお、自由発話における間投詞（冗長語、不要語）の出現頻度は多くの研究期間で報告されている。文献 [34] や文献 [80] では日本音響学会

連続音声データベースの書き起こしテキストを研究して報告している。また、文献 [19] では、開始符合としての間投詞の種類と出現頻度を報告している。また、文献 [66] では NHK ラジオ第一放送の電話相談番組を書き起こして報告している。これらの論文と比較すると、間投詞の出現頻度はほぼ同じ割合と言える。また、間投詞の種類も、これらの報告と比較すると比率に違いがあるが、代表的な間投詞に関してはほぼ同じ割合といえる。

2. 自由発話における言い直しに関して

   今回の研究では、話者によって相違が見られるが、言い直しを含む文は文章全体の 2% から 4%を占めることが示された。しかし、今回研究対象とした話者はナレータ（アナウンサや声優など音声を職業としている人）であるため、一般の人の言い直しの出現頻度は、これよりも高いと思われる [50]。

   なお、文献 [75] では言い直した単語に着目して、言い直しを分析している。これを見ると言い直しの 59%は、言い誤った単語を直ちに言い直している。また文献 [62] においてもほぼ同様な結果が見られる。今回の自由発話データの言い誤りを分析すると、単語にならない音節となっているものが 39%、直後に言い直しているのが 52%であり、傾向はほぼ同じであった。

3. 自由発話と朗読発話の音響的な差

   本章では、自由発話の音響的な特徴を研究するために、主に融合ラベルの付与率、発話速度、HMM における音素認識誤り率で朗読発話と比較した。その結果、自由発話は朗読発話と比較すると、発話速度は最も差がある話者でも 6%しか増加しないが、融合ラベルの出現頻度は約 20%も増加する話者がいることが示された。しかし、自由発話と朗読発話の認識精度 (phone accuracy) の差は 7%から 10%程度であることが示された。したがって、少なくとも同一話者（特定話者）、同一発話様式で HMM を学習をする限り、音響モデルに関しては自由発話と朗読発話に大きな差はないように思われる。

   ただし、本章で調査した話者は音声による対話に慣れた人である。したがって、一般の話者が雑音下で制約の少ない状態で話した音声では、この論文で研究した結果と異なる可能性がある。

4. 自由発話の可能性について

   自由発話において特徴的な言語現象に、間投詞や言い直し・言い誤り・言い淀みなどがある。そして、今回の研究の結果、間投詞は発話全体の 40%から 65%の文に、言い直しは約 2%から 4%の文に出現することが

示された。自由発話の認識には、これらの言語現象の処理方法が大きな問題になると考えられる。

現在自由発話の認識アルゴリズムとしては、これらの現象に対応するため、1) キーワードスポッティングを利用する方法 [87]、2) 音素モデルにガーベージモデルなどを使用して認識する方法 [23][22]、3) 言語モデルの一部に音素系列として認識する方法 [44],[34] もしくはこれらの組合せの手法 [89] などが試みられている。しかし、これらのアルゴリズムには挿入誤りが増加することや、広いビーム幅が要求されるなどの問題点が残っている。

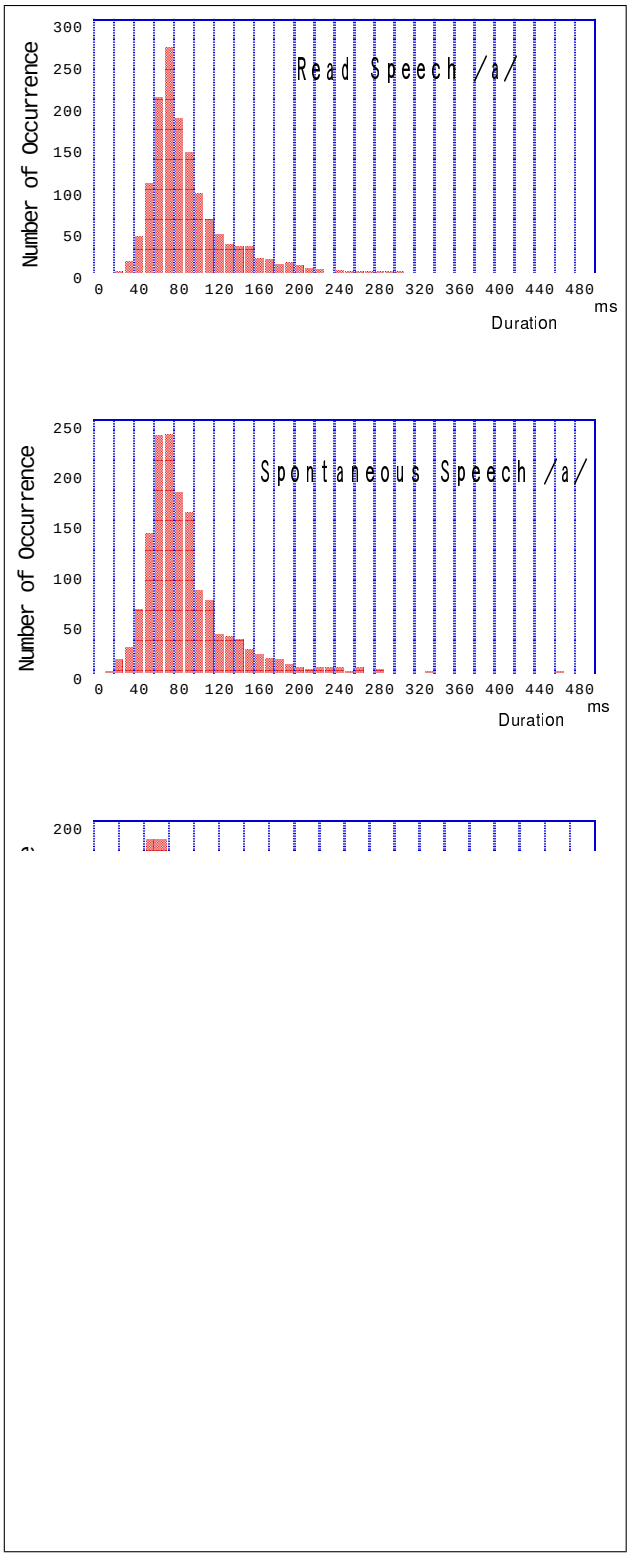


図 6.12: 音素/a/および/k/の継続時間の分布 (話者 MMY)

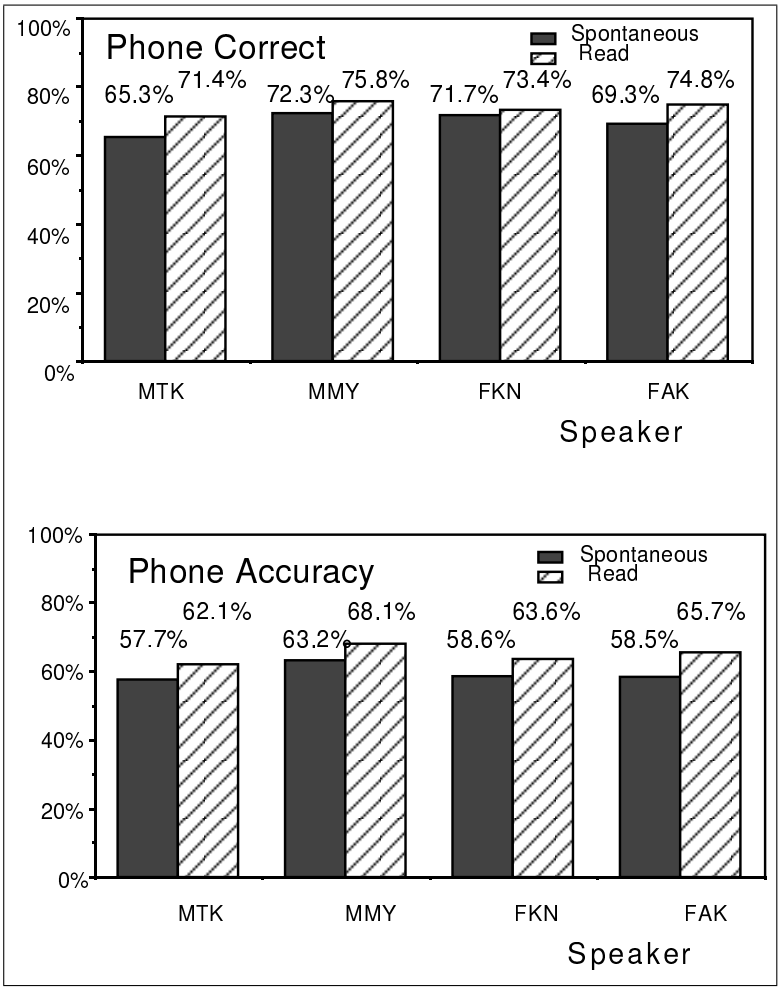


図 6.13: 音素認識実験結果 音素認識率 (%)

表 6.8: 音素認識誤り傾向

(a) 話者 MTK (認識音素数/対象音素数)

		出力				
		a	i	u	e	o
入力	a	83.1% (167/201)	0.0% (0/201)	1.5% (3/201)	3% (6/201)	7.5% (15/201)
	i	0.7% (1/128)	85.1% (109/128)	3.9% (5/128)	3.9% (5/128)	0.7% (1/128)
	u	7.3% (6/ 82)	4.8% (4/ 82)	48.7% (40/ 82)	2.4% (2/ 82)	9.7% (8/ 82)
	e	3.0% (4/131)	13.7% (18/131)	1.5% (2/131)	76.3% (100/131)	2.2% (3/131)
	o	3.5% (5/140)	0.7% (1/140)	5.7% (8/140)	2.8% (4/140)	80.0% (112/140)

(b) 話者 MMY (認識音素数/対象音素数)

		出力				
		a	i	u	e	o
入力	a	93.2% (633/679)	0.1% (1/679)	0.7% (5/679)	3.0% (21/679)	1.1% (8/679)
	i	0.0% (0/426)	81.4% (347/426)	3.2% (14/426)	4.9% (21/426)	0.0% (0/426)
	u	1.2% (4/320)	4.0% (13/320)	45.6% (146/320)	3.4% (11/320)	7.1% (23/320)
	e	1.4% (6/405)	3.4% (14/405)	2.2% (9/405)	83.4% (338/405)	0.7% (3/405)
	o	1.5% (8/522)	0.0% (0/522)	1.7% (9/522)	3.4% (18/522)	88.5% (462/522)

(c) 話者 FKN (認識音素数/対象音素数)

		出力				
		a	i	u	e	o
入力	a	83.7% (381/455)	0.4% (2/455)	1.9% (9/455)	4.6% (21/455)	1.5% (7/455)
	i	0.0% (0/289)	76.4% (221/289)	2.0% (6/289)	3.8% (11/289)	0.3% (1/289)
	u	1.4% (3/205)	0.9% (2/205)	52.6% (108/205)	9.7% (20/205)	4.3% (9/205)
	e	0.4% (1/227)	4.8% (11/227)	3.0% (7/227)	84.1% (191/227)	0.0% (0/227)
	o	1.2% (4/318)	0.0% (0/318)	4.4% (14/318)	0.3% (1/318)	88.6% (282/318)

(d) 話者 FAK (認識音素数/対象音素数)

		出力				
		a	i	u	e	o
入力	a	80.6% (393/487)	0.0% (0/487)	4.1% (20/487)	4.7% (23/487)	2.0% (10/487)
	i	0.0% (0/265)	73.9% (196/265)	1.1% (3/265)	7.9% (21/265)	0.3% (1/265)
	u	6.0% (12/199)	3.5% (7/199)	43.2% (86/199)	6.0% (12/199)	4.0% (8/199)
	e	0.8% (2/244)	9.0% (22/244)	2.8% (7/244)	78.6% (192/244)	0.4% (1/244)
	o	2.6% (10/381)	0.0% (0/381)	3.9% (15/381)	1.5% (6/381)	83.7% (319/381)

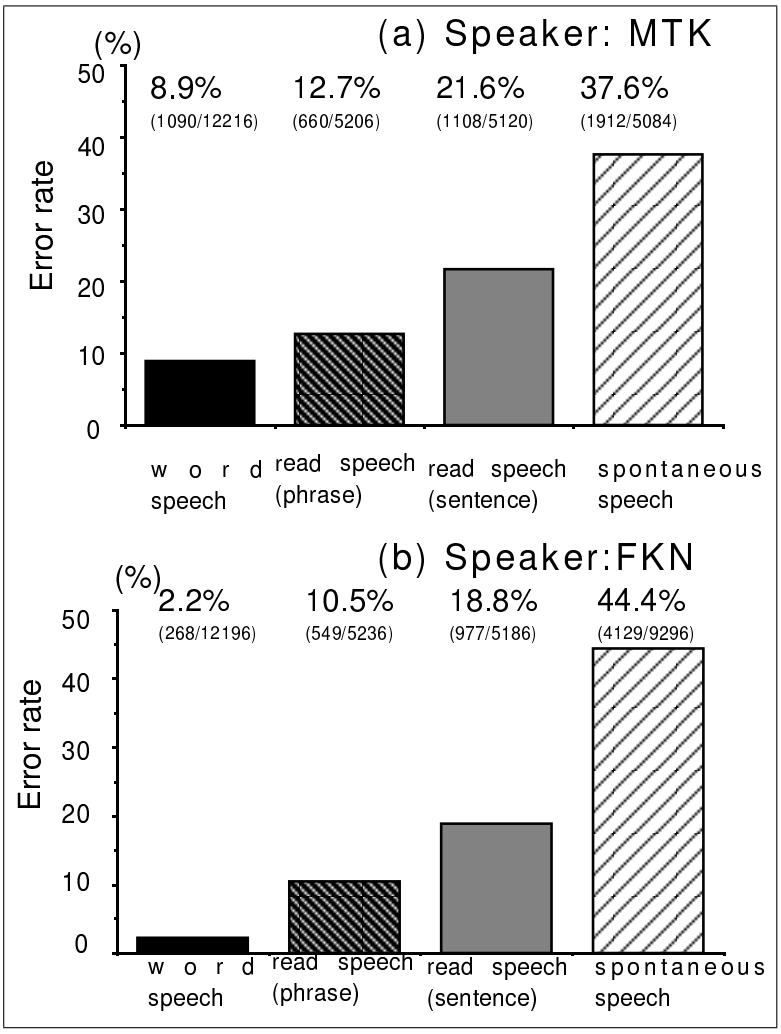


図 6.14: 発話様式の違いによる音素認識誤り率（学習データ:単語発声）

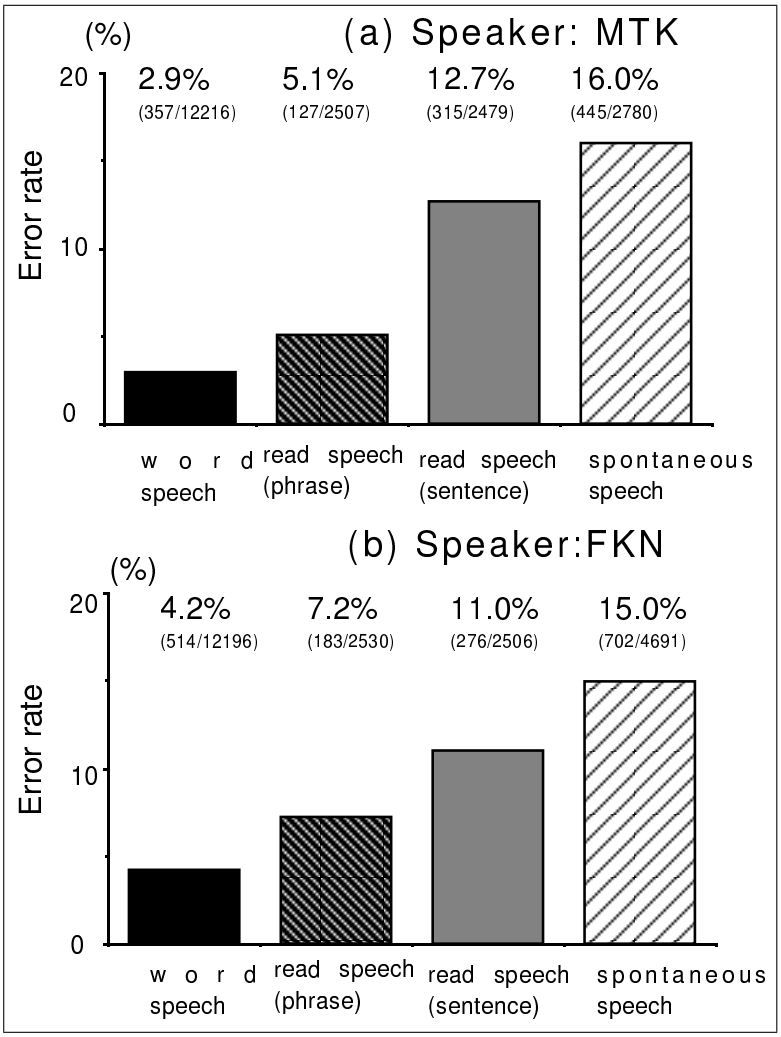


図 6.15: 発話様式の違いによる音素認識誤り率（学習データ:同一発話様式）

第7章　音声におけるアクセント情報の持つ情報量の考察

音声信号は、音素情報の他にも韻律情報を含んでいる。したがって音声合成の研究では、自然な音声を合成するためにアクセント・ポーズ・イントネーションなどの韻律情報を自動的に付与する方法について研究が行なわれてきた [47][21]。これらの論文では、一般名詞・固有名詞・数量表現などにおける複数の読みや特殊な読みの存在や、名詞連続複合語や文節におけるアクセント核の移動などの問題に対する解決方法が提案されていて、一般の日本文（漢字仮名交じり文）に対し高い精度で韻律情報を付与できる。

しかし、音声認識の研究では主に音素情報に着目されていて、韻律情報を利用した研究は全体としては少ない。高橋ら [81] や Singer ら [77] は、韻律情報を使用することにより単語認識性能や音素認識性能が向上することを報告している。だが、これらの研究結果を見ると認識性能が大きく改善されたとは言えない。これは、現段階ではピッチの抽出が必ずしも容易でないことや、韻律情報は個人差が多く、さらに地域差が存在することなどが理由にあると思われる。しかし、韻律情報は音声認識における１つの有効な情報源であり、特に日本語においては同音異義語が多いことも考えると、今後も韻律情報を利用した音声認識の手法の研究は重要であろう。

ところで、韻律情報の有効性を他の種類の情報 (音響情報、言語情報) と同一の尺度で比較することも重要な課題であると考えられる。柴田ら [71] は辞書から同音異義語の数とアクセント型を研究し、この結果日本語において、ある語が他の同音異義語からアクセントによって弁別される確率は 13.57%と計算している。しかし、このように韻律の持つ情報量を定量的に測定した研究は少ない [54][71]。

本章では、仮名漢字変換において出力される漢字仮名交じり文の候補の数の減少度という点に着目して、韻律の持つ情報量を研究した。情報量は、その情報が存在する場合と存在しない場合の曖昧さの差として捉えることができる。したがって、韻律情報の持つ情報量は、音素情報から生成される漢字仮名交じり文の数と、音素および韻律情報から生成される漢字仮名交じり文の数を比較することによって測定することが可能と思われる。このとき、音素および韻律情報から生成される漢字仮名交じり文の数の推定方法が問題になるが、これは、音声合成において研究された漢字仮名交じり文から韻律情報への変換の機能を用いることにより、ある程度解決できる。

韻律情報は F_0, power, duration などの多くの要素から構成されているが、本章では、この中から特にアクセント句境界の位置およびアクセント核の位置の持つ情報量に焦点を当てて情報量を測定した。

7.1 アクセント情報の持つ情報量の基本的な測定方法

音声中における韻律情報には F_0, power, duration, アクセント句、アクセント核、メジャーフレーズ、マイナーフレーズなどが考えられる。本章では、この中からアクセント句境界の位置およびアクセント核の位置の情報量を研究した。

なお、本章では以後、アクセント句境界の位置の情報とアクセント核の位置の情報を合わせてアクセント情報と呼ぶことにする。

7.1.1 情報量の定義

情報は確率事象に対して定義され、確率事象 I が生起したことを知ったとき、$E = -\log_2 P(I)$ の情報を受けとったと言う。ただし $P(I)$ は事象 I の生起確率を示す。

このことは一般に事象 I が生起したことを知ることによって対象とする対象系に関する解釈の曖昧さが E ビットだけ減少することを意味する。この解釈の曖昧さの減少の割合が事象 I の情報量 (相互情報量）に相当する。この考え方によってアクセント情報の持つ情報量を求めることができる。つまり、アクセント情報の情報を知ることによって曖昧さが減少するような事象系を１つ設定し、アクセント情報が存在しないときに得られる解釈の数とそれらの情報が存在するときに得られる解釈の数を比較すれば良いことになる。

7.1.2 アクセント情報の持つ情報量の基本的な測定方法の考え方

本章では、仮名漢字変換において出力される漢字仮名交じり文の候補の数の減少度という点に着目した。つまり、アクセント情報（アクセント句とアクセント核）の持つ情報量は、音素情報から生成される漢字仮名交じり文の数と、音素情報およびアクセント情報から生成される漢字仮名交じり文の数を比較することによって測定できると思われる。具体的なアルゴリズムを次に示す。

1. 初めに、正しい音素情報およびアクセント情報が与えらたと仮定する。

2. 音素情報から漢字仮名交じり文を生成する。この音素ー漢字変換は一般的に使用されている仮名漢字変換とほぼ同一と考えられ、複数の漢字仮名交じり文が生成される。この候補の数を $c1$ とする。

3. 音素情報およびアクセント情報から漢字仮名交じり文を生成する。この場合音素情報の他にアクセント情報も加わるため、生成される漢字仮名交じり文の数は $c1$ と比較すると減少すると思われる。この候補の数を $c2$ とする。

4. アクセント情報の持つ情報量は、音素情報から生成される漢字仮名交じり文の数と、音素情報およびアクセント情報から生成される漢字仮名交じり文の数を比較することによって測定できる。アクセント情報の持つ情報量 E の計算式は $-\log_2(c2/c1)$ と計算できる。

7.1.3 基本的な測定方法のフローチャート

図 7.1 に例をあげてアクセント情報の持つ情報量の基本的な測定方法を示す。例文として “私は牡蛎を投げた。” を用いた。

まず初めに、音素情報から漢字仮名交じり文を生成する（音素ー漢字変換)。この例文の音素情報は “ワタシワカキオナゲタ” である。音素情報から漢字仮名交じり文に変換するとき、日本語では同一の読みに対し複数の漢字が存在するため、複数の漢字仮名交じり文が生成される。この音素情報からは “私は牡蛎を投げた。”、“私は柿を投げた。”、“私は火器を投げた。”、“私、若木を投げた。” などが生成される。

次に、音素情報およびアクセント情報が与えられたとして、これらの情報から漢字仮名交じり文を生成する（音素・アクセントー漢字変換)。この例文の音素情報およびアクセント情報は “ワタシワ_カキオ_ナゲタ” である。ここで “^” はアクセント核の位置を示し、“_” はアクセント句境界の位置を示す。アクセント情報が加わっても、漢字仮名交じり文は複数生成されるが、この数は音素情報から生成された漢字仮名交じり文の数よりも少なくなる。この例文では “私は牡蛎を投げた。”、“私は火器を投げた。” などが生成される。

最後に、この両者の漢字仮名交じり文の数の比を計算する。音素情報から生成される漢字仮名交じり文の数と、音素情報およびアクセント情報から生成される漢字仮名交じり文の数の差は、アクセント情報の持つ情報に起因する。したがって、この比がアクセント情報の持つ情報量となる。

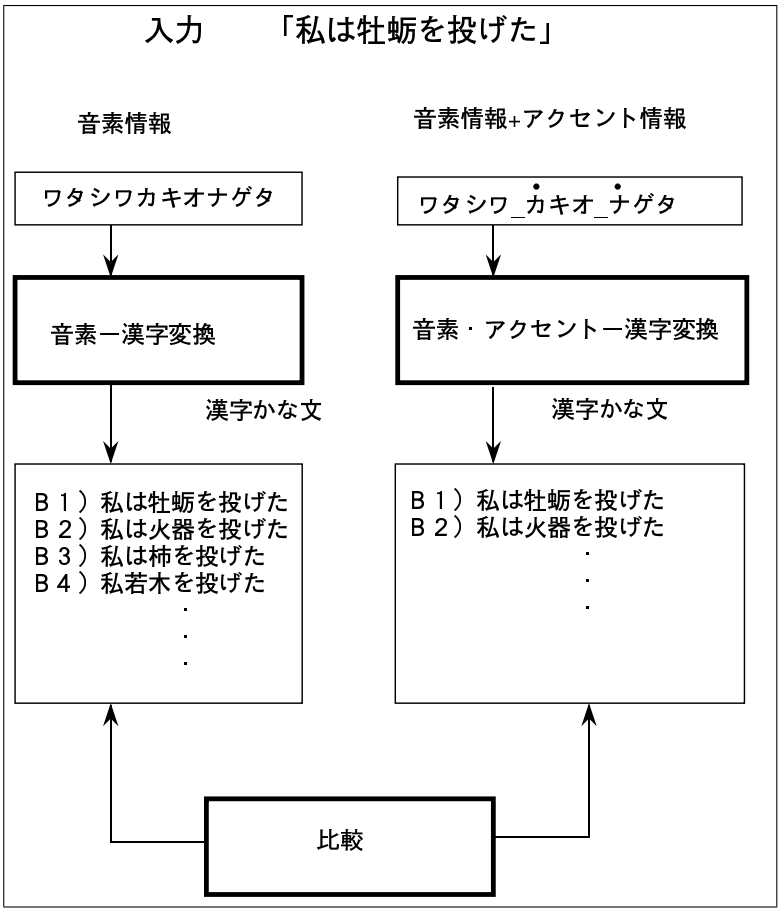


図 7.1: アクセント情報の持つ情報量の基本的な測定方法

7.2　漢字ー音素・アクセント変換を利用したアクセント情報の持つ情報量の測定方法

7.2.1　基本的な方法の問題点

上記、7.1.2 節においてアクセント情報の情報量を測定する基本的な方法を提案した。しかし、現時点では正しいアクセント情報を予め知ることはかなり困難である。また、音素情報およびアクセント情報から漢字仮名交じり文に変換する音素・アクセントー漢字変換が、まだ確立していないこと、およびアクセント情報は個人差が大きいことなどを考えると、この方法で精度の良い実験を行なうことは困難である。

7.2.2　漢字ー音素・アクセント変換を利用した測定方法

ところで漢字仮名交じり文を音素情報およびアクセント情報に変換する漢字ー音素・アクセント変換は、規則音声合成の分野で研究され続けている。この結果アルゴリズムも、かなり確立していて高い変換精度が得られている[16][46]。そして、音素情報から漢字仮名交じり文に変換する音素ー漢字変換は表記と発音が異なる単語（例えば鼻音や長音を含む単語および助詞の “は” と “を”）が存在していることをのぞけば、従来の仮名漢字変換と技術的に同等である。

以上のことを考慮して、本章では漢字ー音素・アクセント変換を利用することによって 7.1.2 節で示したアクセント情報の持つ情報量の基本的な測定方法と等価な方法を考案した。この方法を次に示す。

1. 初めに任意の漢字仮名交じり文を音素情報とアクセント情報に変換する。この変換は１つの漢字仮名交じり文に対して１つの正しい音素情報とアクセント情報が出力されるとする。

2. 次に、この情報から音素情報のみを選択して、この情報から複数の漢字仮名まじり文を生成する。この音素ー漢字変換では１つの音素情報に対し複数の漢字仮名交じり文が出力される。

3. 次に、これらの漢字仮名交じり文を再び音素情報とアクセント情報に変換する。この変換は１つの漢字仮名交じり文に対して１つの正しい音素情報およびアクセント情報が出力されるとする。

4. 最後に、これらの出力された音素情報とアクセント情報を、元の音素情報およびアクセント情報と比較して、音素情報およびアクセント情報の一致する漢字仮名交じり文の数と音素情報の一致する漢字仮名交じり文の数を比較する。

この方法は、漢字仮名交じり文を音素情報およびアクセント情報に正しく変換できると仮定すれば、7.1.2 節で示したアクセント情報の持つ情報量の基本的な測定方法と同一になる。この仮定は正しくないが、漢字仮名交じり文を音素情報およびアクセント情報に変換する漢字ー音素・アクセント変換の変換精度は、かなり高いため、この方法はアクセント情報のもつ情報量を、近似ではあるが、得られると予測される。

7.2.3　漢字ー音素･アクセント変換を利用した測定方法のフローチャート

図 7.2 に、本章で用いた、アクセント情報の情報量の測定方法のフローチャートを示す。例文として “私は牡蛎を投げた” を用いた。

まず初めに、漢字仮名交じり文を音素情報およびアクセント情報に変換する。この例文では “ワタシワ_ガキオ_ナゲタ” が出力される。

次に、この出力結果からアクセント情報を除いて音素情報のみを選択する。例文では “ワタシワカキオナゲタ” になる。この音素情報から漢字仮名交じり文を生成する。この結果 “私は牡蛎を投げた。”、“私は火器を投げた。”、“私は柿を投げた。”、“私若木を投げた。”、“私は牡蛎を和げた。” の漢字仮名交じり文が出力される。

次に、これらの漢字仮名交じり文を再び音素情報およびアクセント情報に変換する。この音素・アクセント－漢字変換は１つの漢字仮名交じり文に対して１つの正しい音素情報およびアクセント情報が出力される。

B1 “私は牡蛎を投げた。” →
“ワタシワ_ガキオ_ナゲタ”

B2 “私は火器を投げた。” →
“ワタシワ_ガキオ_ナゲタ”

B3 “私は柿を投げた。” →
“ワタシワ__カキオ_ナゲタ”

B4 “私若木を投げた。” →
“ワタシ__ワカキオ_ナゲタ”

B5 “私は牡蛎を和げた。” →
“ワタシワ_ガキオ__ヤワラゲタ”

最後に、これらを元の音素情報およびアクセント情報 “ワタシワ_ガキオ__ナゲタ” と比較する。B1) は元の漢字仮名交じり文である。 B2) は音素とアクセント句境界およびアクセント核が一致する。B3) は音素情報とアクセント句境界が一致するがアクセント核が異なる。B4) は音素情報が一致するがアクセント句境界およびアクセント核は異なる。 B5) はすべての情報が異なっている。 この例では、元の音素情報およびアクセント情報に一致する漢字仮名交じり文は２文である。また、音素情報が一致する漢字仮名交じり文は４文である。したがってアクセント情報の持つ情報量 E は $-\log_2(2/4) = 1(bits)$ となる。

7.3 実験結果

7.3.1 実験条件

アクセント情報の情報量を測定するために、7.2.2 節で示した方法で実験を試みた。実験条件を次に示す。

1. 入力データ

   入力文には 1982 年の日本経済新聞を使用した。また入力単位を文にしたとき、音素情報から大量の漢字仮名交じり文が生成されたため、入力単位を文節にした。実験は 50 文節おこなった。この入力文節を表 7.1 に示す。

表 7.1: 実験に使用した文節

大蔵省は　これによって　在日外銀に　関する　法的 根拠が　明確になるほか　在日外銀の　国内活動が　しやすくなり　欧米諸国の　間に　出始めている　わが国の　金融制度に　対する　不満を 和らげるのに 役立つと　みている　国際化に　伴い　日本に進出する　外国銀行は　急増している　大蔵省は　邦銀に　対しては　銀行の　経営基盤を　安定させる　目的で　同　準備金の積立てを　義務　づけて　おり　大きな　損失などが　生じる　時だけに　取り崩しを　認めている　大蔵省は　この　規定を　在日外銀に　適用することで　取引先などの

2. 漢字ー音素・アクセント変換

   実験に用いた漢字仮名交じり文から音素情報およびアクセント情報に変換する漢字ー音素・アクセント変換の変換精度は、音節正解率で 99.8%、アクセント情報の正解率で 95%である [47]。このシステムではアクセント句境界の位置の他に、3 種類の境界の長さが出力されるが、今回の実験では 1 種類にまとめた。またアクセント核も、第 1 アクセント核と第 2 アクセント核が出力されるが、同様に第 1 アクセント核だけを利用した。したがって、実験ではアクセント句境界の位置の情報と第 1 アクセント核の位置の情報量が測定される。

3. 音素ー漢字変換

   音素情報から漢字仮名交じり文を生成する音素ー漢字変換に、文節数最小法を使用した。ただし変換精度を上げるため分割数を最小分割数+1 まで分割した [59]。ただし生成された漢字仮名まじり文に対する単語接続情報、頻度情報などの言語情報による選択はしなかった。したがって音素情報から連想される、すべて漢字仮名交じり文が出力される。ただし分割の違いから生成される、同じ表記の重複する漢字仮名交じり文は、1 つの候補とした。例えば “大蔵省” は固有名詞の “大蔵省” と一般名詞の “大蔵” と “省” の 2 種類が出力される。このように重複する漢字仮名交じり文は “大蔵省” の 1 つと数えた。音素ー漢字変換において使用した単語辞書は、約 16 万語である。

4. 情報量の計算

   情報量の計算は以下の 3 種類について行った。

   (a) アクセント句境界の位置の持つ情報量

   計算式は P_1 を（音素情報およびアクセント句境界の一致する漢字仮名交じり文の数／音素情報の一致する漢字仮名交じり文の数）として、アクセント句境界が持つ情報量 E_1 を $-\log_2(P_1)$ とする。

   (b) アクセント核の位置の持つ情報量

   計算式は P_2 を（音素情報およびアクセント句境界およびアクセント核の一致する漢字仮名交じり文の数／音素情報およびアクセント句境界の一致する漢字仮名交じり文の数）として、アクセント核の持つ情報量 E_2 を $-\log_2(P_2)$ とする。

   (c) アクセント情報の持つ情報量

   計算式は P_3 を（音素情報およびアクセント句境界およびアクセント核の一致する漢字仮名交じり文の数／音素情報の一致する漢字仮名交じり文の数）として、アクセント情報が持つ情報量 E_3 を $-\log_2(P_3)$ とする。

7.3.2 実験結果

1. アクセント情報が一致する漢字仮名交じり文の数

   各文節に対する実験結果を図 7.3 に示した。この図では横軸は文節番号で、縦軸は各情報で一致した漢字仮名交じり文の数を log スケールで書いた。×は音素ー漢字変換によって生成された漢字仮名交じり文の数、○は音素情報が一致した漢字仮名交じり文の数、△は音素情報とアクセント句境界が一致した漢字仮名交じり文の数、□は音節情報とアクセント句境界およびアクセント核が一致した漢字仮名交じり文の数を示している。

   この結果から、文節番号によって、各情報で一致した漢字仮名まじり文の数に大きな差があることがわかる。なおグラフ中空白になっている文節は、音素ー漢字変換の出力の漢字仮名交じり文の数が多すぎるため、処理を中止したことを示している。

2. アクセント情報の持つ情報量 50 文節を実験して得られた各情報量の平均値を、表 7.2 に示した。実験の結果、アクセント句境界の位置が持つ情報量は 3.21bit、アクセント核の位置の持つ情報量は 1.97bit、アクセント情報が持つ情報量は 5.16bit であることが示された。

表 7.2: アクセント情報の持つ情報量 (bit)

情報	情報量	分散
アクセント句境界の位置	3.21	3.37
アクセント核の位置	1.97	1.62
アクセント情報	5.16	3.20
（アクセント句境界＋アクセント核）		

なお文献 [60] では、日本語における音節のエントロピーは 5.55bit であることが報告されている。今回、78 日分の日経新聞の記事を文節に区切って、音節のエントロピーを計算したところ 5.67bit であった。これらの値と比較すると、アクセント情報の持つ情報量は絶対量としては高い情報量を持っていると評価できる。

7.4 アクセント情報の情報量の値の信頼性

今回の実験には、多くの仮定を含んでいるため、得られたアクセント情報の情報量の値に大きな誤差がある可能性がある。そこで他の方法によってアクセント情報の情報量を計算した。

7.4.1 生成確率によるアクセント句境界の位置の持つ情報量の値

本章では、情報量とはその情報が存在したときに減らせる曖昧さの値であるとした。しかし、この値は出現率によっても測定が可能であると思われる。以下に、出現率を使用したアクセント句境界の位置の持つ情報量の測定方法を示す。

1. 大量の日本語を用意する。
2. これを漢字ー音素・アクセント変換して音素情報およびアクセント情報に変換する。
3. アクセント句境界および音素の出現回数をもとめる
4. 出現回数からアクセント句境界の情報量を計算する。

日経新聞 1982 年の 1 月 5 日の 1 日分の記事、約 10 万文字を漢字ー音素変換してアクセント句境界および音素の出現回数を求めたところ、それぞれ 28677 回と 130068 回出現した。したがってアクセント句境界の持つ情報量は

$E = -\log_2[(28677/(130068 + 28677)] = 2.47(bit)$
となる。

7.4.2 他の論文におけるアクセント核の位置の持つ情報量の値

文献 [71] では、日本語において、ある語が他の同音異義語からアクセントによって弁別される確率は 13.57%と計算している。これを本章の情報量の計算式に当てはめると $-\log_2(0.1357) = 2.88(bits)$ となる。

7.4.3 アクセント情報の情報量の値の信頼性について

本章で使用したアクセント情報の持つ情報量の計算方法は、３種類の仮定を含んでいる。

- 文と文節

  この実験では、入力単位を文節にした。したがって文と文節においてアクセント情報の情報量の差はないと仮定している。

- 漢字ー音素・アクセント変換

  漢字ー音素・アクセント変換は、漢字仮名交じり文を、必ず正しい音素情報およびアクセント情報に変換すると仮定している。

- 音素ー漢字変換

  音素ー漢字変換において変換方法として文節数最小法を用いた。しかし、文節数最小法が元の漢字仮名交じり文を生成するとは限らない。また、単語接続情報などの文法による候補の数の絞り込みを行なっていない。そのため大量の非文が生成される可能性がある。

これらの仮定により 7.2.2 の実験から求めたアクセント情報の情報量は大きな誤差を含んでいる可能性がある。しかし別の方法で求めたアクセント句境界の位置の持つ情報量の値およびアクセント核の位置の持つ情報量と、ほぼ一致することから、本報告で求めた情報量は、日本語の音声におけるアクセント情報の持つ情報量を、ほぼ正しく表していると考えている。

7.5 考察

7.5.1 漢字の読みの知識の情報量とアクセント情報の情報量の比較

日本語の漢字には複数の読みかたが存在する。そして前後の漢字や意味によって、この読みかたが変化する。したがって音素情報を漢字仮名交じり文に

変換し、これを再び音素情報に変換したとき、元の音素情報には戻らない漢字仮名交じり文が存在する。図 7.2 において “私は牡蛎を和げた。” が良い例文である。 ここに漢字の読みの知識がはいると考えられる。この情報量を今回の実験から計算した。計算式は P を（音素情報の一致する漢字仮名まじり文の数／音素ー漢字変換が出力する漢字仮名交じり文の数）として、漢字の読みの知識の情報量 E は $-\log_2(P)$ となる。実験の結果、情報量は 2.26bit、分散は 0.88 となった。これはアクセント情報の持つ情報量より小さい。つまりアクセント情報の情報量は、漢字の読みの知識の情報量より大きいと言える。

7.5.2 文法規則の情報とアクセント情報の情報量の比較

日本文校正支援システム (REVISE) は、人間が書いた単語接続規則などの文法規則から、日本語の誤字脱字などの誤りを検出する機能を持っている。そして人間が侵す誤りの中の 90%を検出する性能を持っている [21]。このプログラムを用いて、図 7.2 の漢字仮名交じり文の候補から誤りが検出された数を研究した。結果を以下に示す。

表 7.3: 音素情報およびアクセント情報の一致した漢字仮名交じり文　入力「大蔵省は」

大蔵省は	大蔵商は	大蔵将は	大蔵小は
大蔵称は	大蔵抄は	大蔵賞は	大蔵衝は
大蔵賞は	大倉商は	大倉賞は	大倉小は
大倉抄は	大倉省は	大倉ショウは	

この結果から単語接続規則などの文法規則の持つ情報量は $-\log_2(0.855) = 0.23(bits)$ となり、かなり小さいことがわかる。これは同音異義語の曖昧さは単語接続規則などの文法規則では絞れないことを示している。また、この実験から音声認識においては文法情報よりアクセント情報の方が情報量が大きく、その扱いが重要であることが分かる。

7.5.3 アクセント情報と文法の関係

元の音素情報およびアクセント情報が一致した漢字仮名交じり文の候補を調べると元の漢字仮名交じり文と同じ文法構造を持つものが多い。表 7.3 に “大蔵省は” を入力したときの音素情報およびアクセント情報が一致する漢字仮名交じり文を例として示す。この例文では、音素ー漢字変換において生成

される漢字仮名交じり文の候補の数は 2174 文であり、元の音素情報と一致した漢字仮名交じり文は 361 文、音素情報およびアクセント句境界と一致した漢字仮名交じり文は 72 文、音素情報およびアクセント情報と一致した漢字仮名交じり文は 15 文であった。この結果からわかるように、音素情報およびアクセント情報と一致した全ての漢字仮名交じり文は名詞＋助詞の構造を持ち、元の漢字仮名交じり文と等しい品詞列であった。これから、アクセント情報と文法には、かなり強い相関があるように思われる。

7.5.4　音声認識におけるアクセント情報の持つ情報量

実際の音声認識 (例えば [81]) においては、音素系列を決定する目的のために、アクセント情報を利用する場合が多いと考えられる。このために用いられるアクセント情報の情報量を定量的に求めることは困難であろう。

今回の実験は、音素系列が正しく与えられた上で、仮名漢字変換したときに生成される漢字仮名交じり文をどれくらい減らせるかを示すことによって、アクセント情報の情報量を定量的に把握したものである。アクセント情報は正しく与えられるものと仮定しているから、これは仮名漢字変換の視点から見た、利用できるアクセント情報の情報量の上限を与えるものと考えられる。

この意味では、今回情報量を把握したアクセント情報は、上記のような一般の音声認識でのアクセント情報とは若干異なる性格のものであるが、一応の目安にはなるであろう。今回の実験によって、アクセントのもつ情報量は、かなり大きいことが示されたことから、アクセント情報は音声認識において認識性能を向上させる有効なパラメータであると思われる。

7.6　まとめ

本章では、日本語の音声の韻律情報の中から、アクセント句境界の位置および第１アクセント核の位置の持つ情報量を定量的に測定した。これらの情報量は、音素情報を漢字変換したときに生成される漢字仮名交じり文の数と、音素情報とアクセント情報を漢字変換したときに生成される漢字仮名交じり文の数を比較することによって得た。ただし実験では、漢字仮名交じり文を音素情報およびアクセント情報に変換する漢字ー音素・アクセント変換を利用して、近似的に同等と考えられる方法で、実験をおこなった。

この実験によれば、アクセント句境界の位置と第１アクセント核の位置を合わせた情報量は 5.16bit であった。一方、日本語における音素情報の平均情報量は音節あたり 5.67bit 程度である。すなわち、アクセント情報の情報量は、かなり大きい情報量であると思われる。

本章では、韻律情報の中からアクセント句境界の位置および第１アクセント核の位置に焦点をあてたが、韻律には、その他アクセント境界の長さの情

報や第２アクセント核などの多くの情報がある。したがって、全韻律情報の情報量は、上で示した値より大きいことが予想される。これらの情報は、方言などの個人差が大きいことや、信号処理により検出することは容易でないため、断片的な情報しか得られない可能性はあるが、それでもなお音声認識に有効な情報を含む可能性があると思われる。

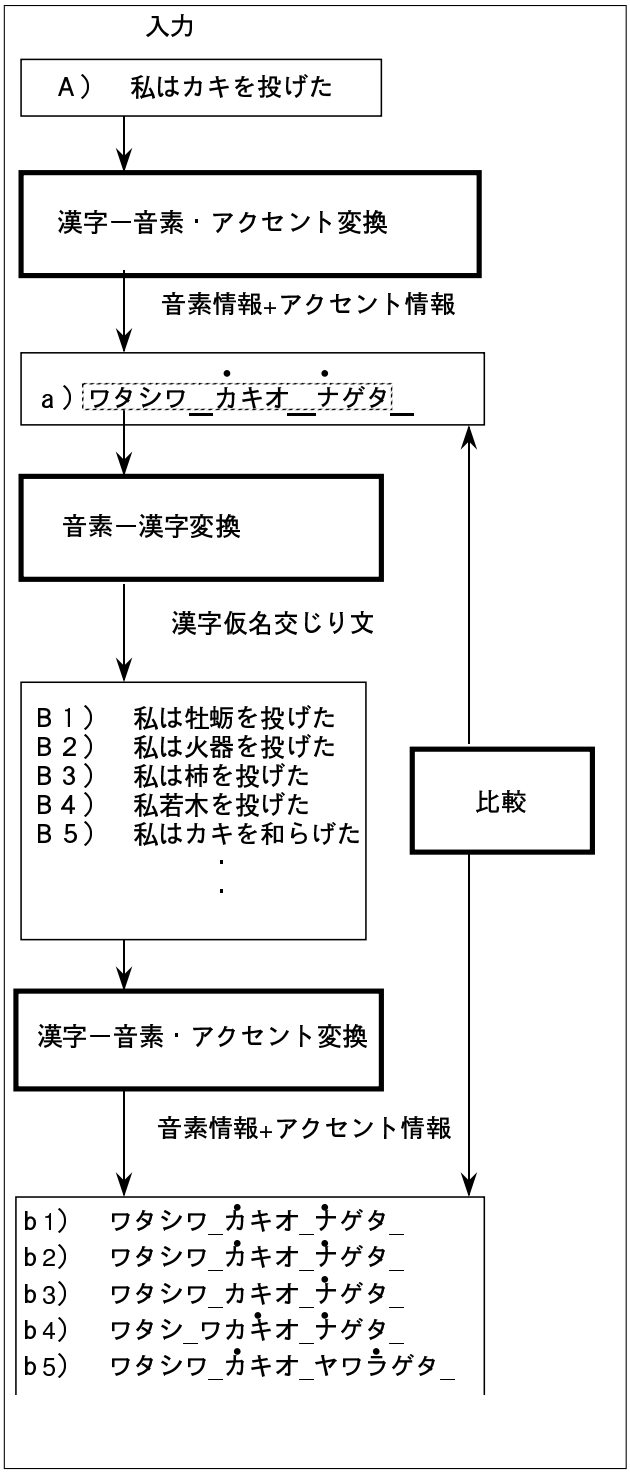


図 7.2: 漢字ー音素・アクセント変換を利用したアクセント情報の持つ情報量の評価方法

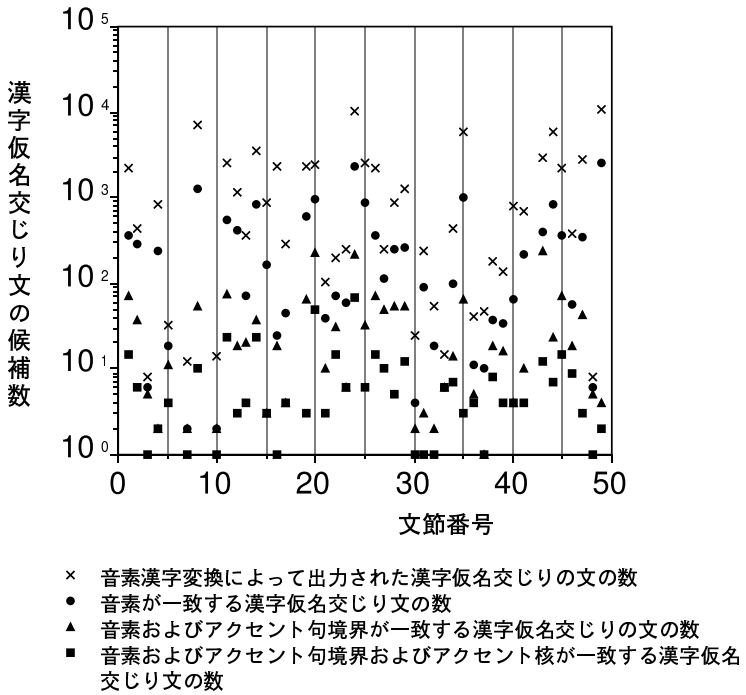


図 7.3: 一致した漢字仮名交じり文の数

第8章　Ergodic HMMを用いた未知・複数信号源クラスタリング問題の検討

異なる N 個の信号源より生成された信号系列が、どの信号源から生成されたのかを分割・識別する問題を、未知・複数信号源クラスタリング問題とする。この問題は、音声処理分野に限らず言語処理などの分野でも重要なテーマである。たとえば音響単位の自動決定問題 [35] や、混在する話者や言語を識別する問題、発話様式（単語・文発声・連続発声等）の識別問題、音声・非音声の識別問題、さらに言語における品詞ラベルの自動的な作成および形態素解析 [56] などがこの問題に相当する。

この未知・複数信号源クラスタリング問題は大きくわけて以下の 4 つの部分問題から構成される。

1. カテゴリを特徴付ける特徴量の問題
2. カテゴリ遷移の セグメンテーション位置を決める問題
3. セグメンテーションされた各ブロックを N 個のカテゴリに識別する問題
4. カテゴリ数 N を推定する問題

本論文では、カテゴリ数 N が既知の場合に、観測された信号系列に対し、自動的にセグメンテーションを行ない、セグメンテーションされた各区間のカテゴリを識別する問題を扱った。

ところで left-to-right HMM は、非定常信号源の一つのモデルとして、特に音声認識の分野で広く用いられている [4]。このモデルはオートマトン制御の下で確率的定常信号源を次々に切替えることにより、非定常信号源を表現している。音声認識では、音声の特性を考慮して、left-to-right 型の HMM が用いられる。しかし、話者認識や言語のモデリングにおいては、全ての状態が全ての状態に接続している Ergodic HMM が使用されている [42]。この Ergodic HMM を未知・複数信号源クラスタリング問題に利用した時、カテゴリーが状態に相当し、信号系列は状態から出力されるシンボル系列と考えることができる。

実験では未知・複数信号源クラスタリング問題の応用として複数話者発話の識別問題を検討した。実験から、男性話者 4 名の場合、長時間窓分析を用

いた LPC ケプストラムを用いることにより、音声資料により異なるが、フレーム単位で約 67.5%の平均識別率が得られることを示した。次に異なる初期モデルから尤度の高いモデルを選択することにより、約 78%の平均識別率が得られることを示した。最後に残された課題と、その解決の展望について報告した。

なお過去に行なわれた類似した研究としては、筆者らは文献 [79] においてセグメンテーション位置およびカテゴリ数が既知の場合に、universal コードブックおよびその出現頻度による Kullback 情報量を使用して話者を識別する方法を報告した。文献 [8, 9] では本論文と同様な複数話者発話の識別問題を扱っている。しかし 1 話者の音響パラメータを 1 つのガウス分布であると仮定し、全音声データに対し VQ clustering を続けることによって問題の解決を図っている。また、通常の話者識別問題 [42] は事前に多量の音声を用いて話者モデルを作成し、それとは異なる認識用の入力音声に対して逐次識別を行なう問題であるのに対し、本研究は連続した音声中に含まれる複数の信号源を、その音声のみを用いて事前学習なしで分割・分類する問題を扱っている。

8.1 未知・複数信号源クラスタリング問題

8.1.1 問題の定式化

図 8.1 に示すように、n 次元ベクトルで与えられる信号の系列を $\boldsymbol{X} = (\boldsymbol{x}_t)(t = 1, 2, ..., T)$ とする。この系列は、K 個のブロック $\boldsymbol{X}_k$ $(k = 1, 2, ..., K)$ からなり、各ブロックが $N(\leq K)$ 個のカテゴリ $\boldsymbol{C}_j$ $(j = 1, 2, ..., N)$ のいずれかから生じた系列であるとする。ここで、ブロック $\boldsymbol{X}_k$ の構成要素を $(\boldsymbol{x}_{t_{k-1}+1} ... \boldsymbol{x}_{t_k})$ (ただし、$t_0 = 0, t_K = T$) 、継続長を M_k $(= t_k - t_{k-1})$ とする。

未知・複数信号源クラスタリング問題は、与えられたベクトル信号系列に対し、信号源の特性の違いに着目してブロックの切れ目の位置 $t_k (k = 1, 2, ..., K-1)$ を探し、この K 個のブロックを $N(\leq K)$ 個のカテゴリに識別しカテゴリ数 N を推定することである。

8.2 Ergodic HMM を用いた解法

8.2.1 Ergodic HMM

カテゴリ数 N が既知で、セグメンテーション未知のカテゴリ識別を考える時、Ergodic HMM の適用が考えられる。この場合、カテゴリが状態に対応し、信号系列は状態から出力されるシンボル系列と考えることができる。そ

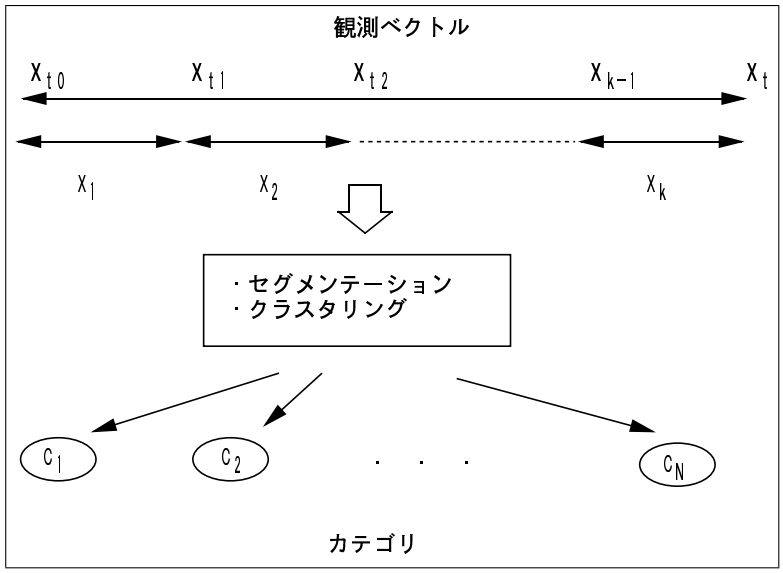


図 8.1: 未知・複数信号源クラスタリング問題の定義

して問題点は次の 2 つになる。

1. 信号系列 $\boldsymbol{X}$ の尤度を最大にする HMM のパラメータ $\boldsymbol{M}$ を推定する問題（モデルの推定、または学習)。なお、パラメータ $\boldsymbol{M}$ は初期状態確率 $\boldsymbol{\pi} = (\pi_i)$、状態遷移確率 $\boldsymbol{A} = (a_{ij})$ 及びシンボル出力確率 $\boldsymbol{B} = (b_j(l))$ で構成されるとする。

2. HMM のパラメータ $\boldsymbol{M}$ が信号系列 $\boldsymbol{X}$ を出力する可能性の高い状態遷移系列を推定する問題（最適状態遷移系列の推定)。

Ergodic HMM による解法の概略を、図 8.2 に示す。

8.2.2 HMM のパラメータ推定

HMM のパラメータ $\boldsymbol{M}$ の推定手法として、Baum-Welch アルゴリズムが知られている。これは EM アルゴリズム [4] を $\boldsymbol{M}$ の推定に適用したもので、学習データの尤度が最大になるように HMM のパラメータを繰り返し演算で計算する。ただし、この計算は局所的最大点の方向に進むため [4]、初期パラメータの設定が重要になる。

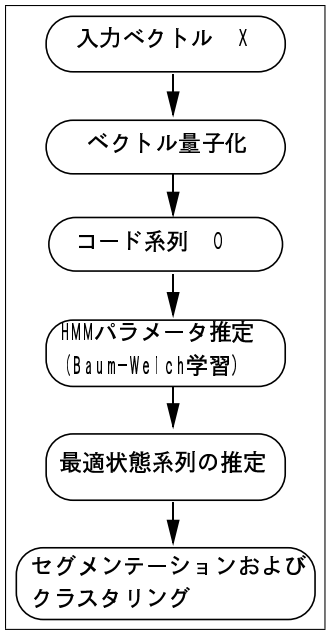


図 8.2: Ergodic HMM を利用した未知・複数信号源クラスタリングの手順

8.2.3　最適状態遷移系列の推定について

ここでは、Baum-Welch アルゴリズムで求まったパラメータ推定値から、信号系列を生み出す最適状態遷移系列を推定する問題を考える。この推定手法として、Viterbi アルゴリズムと forward アルゴリズムが考えられる。この復号法（サーチアルゴリズム）を次に示す。この復号法から計算された最適状態遷移系列 $\boldsymbol{S}^* = \{s_1^*, ..., s_T^*\}$ から、セグメンテーション位置とカテゴリ識別が直接得られる。

1. Viterbi アルゴリズム

   推定された HMM のパラメータ $\boldsymbol{M}$ がコード系列 $\boldsymbol{O}$ を出力する可能性の高い最適状態遷移系列は、Viterbi アルゴリズムにより効率的に求まる（2.1.7 節参照)。Viterbi アルゴリズムを再度次に示す。

(a) 全ての $i \in \{1, ..., N\}$ に対し、

$$\delta_1(i) = \log \pi_i + \log b_i(\boldsymbol{o}_1), \phi_1(i) = 0 \tag{8.1}$$

とおく。

(b) 時間軸 $t = 2, ..., T$ に沿って、全ての $j \in \{1, ..., N\}$ に対し

$$\delta_t(j) = \max_i [\delta_{t-1}(i) + \log a_{ij} + \log b_j(\boldsymbol{o}_t)], \tag{8.2}$$

$$\phi_t(j) = arg \max_i [\delta_{t-1}(i) + \log a_{ij}] \tag{8.3}$$

(c) 最適状態遷移系列に対する対数尤度及び T 時刻目の最適状態を次式で求める。

$$\max_{\boldsymbol{S}} P(\boldsymbol{O}, \boldsymbol{S} \mid \boldsymbol{M}) = \max_j \delta_T(j) \tag{8.4}$$

$$s_T^* = arg \max_j \delta_T(j) \tag{8.5}$$

(d) 時間軸 $t = T - 1, ..., 1$ に沿って、次式により最適状態遷移系列を得る。

$$s_t^* = \phi_{t+1}(s_{t+1}^*) \tag{8.6}$$

2. forward アルゴリズム

最適状態遷移系列の推定には Viterbi アルゴリズムの他に forward アルゴリズムが考えられる。このアルゴリズムは、始めに HMM のパラメータ $\boldsymbol{M}$ がコード系列 $\boldsymbol{O}$ を出力する時の尤度を、各状態からの総和で計算する。次に最適状態遷移系列は各時刻における最大の尤度を持つ状態とする。この forward アルゴリズムを次に示す。

(a) 全ての $i \in \{1, ..., N\}$ に対し、

$$\delta_1(i) = \pi_i \times b_i(\boldsymbol{o}_1) \tag{8.7}$$

$$s_1^* = arg \max_i \delta_1(i) \tag{8.8}$$

(b) 時間軸 $t = 2, ..., T$ に沿って、全ての $j \in \{1, ..., N\}$ に対し

$$\delta_t(j) = \sum_i [\delta_{t-1}(i) \times a_{ij} \times b_j(\boldsymbol{o}_t)] \tag{8.9}$$

$$s_t^* = arg \max_j \delta_t(j) \tag{8.10}$$

(c) 全ての可能な状態遷移系列に対する尤度及び T 時刻目の最適状態を次式で求める。

$$\sum_{\boldsymbol{S}} P(\boldsymbol{O}, \boldsymbol{S} \mid \boldsymbol{M}) = \sum_j \delta_T(j) \tag{8.11}$$

$$s_T^* = arg \max_j \delta_T(j) \tag{8.12}$$

8.3 複数話者発話の識別実験

本論文では未知・複数信号源クラスタリング問題の応用例として、複数話者発話の識別問題を取り上げる。この場合、信号系列は LPC ケプストラムの音響パラメータに、各カテゴリは各話者に、セグメンテーション位置は話者の遷移に対応する。

8.3.1 音声資料

実験には、単語音声 (ATR 5240 単語データ) の波形データを用いて疑似的な音声資料を作成した。話者は 4 名とし、同一話者の異なる単語音声を 8 個つないで 1 ブロック（1 発話）とした。なお、各単語の前後の無音区間は削除した。1 セットの音声資料は、ブロックごとに話者間でランダムに連接して、1 話者につき 8 ブロックで作成した。従って 1 セットの音声資料は 32 ブロックで構成され、話者は 31 回遷移する。音声資料の例を図 8.3 に示す。

この音声資料を 8 セット作成し実験に用いた。音声資料の 1 セットの平均発話時間は約 150 秒である。

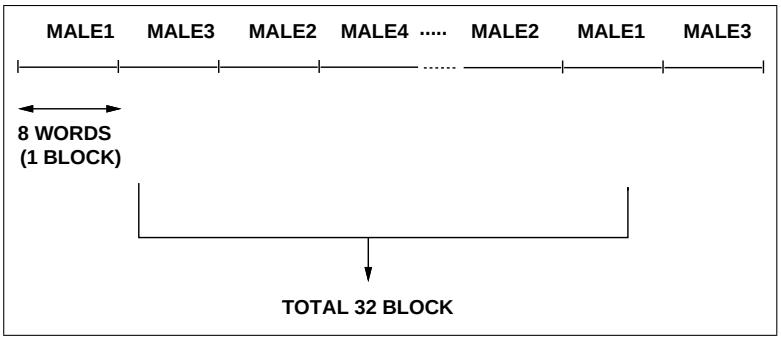


図 8.3: 実験に使用した音声資料の例

8.3.2 音響パラメータ

本実験では、話者特徴量を表す音響パラメータとして LPC ケプストラム (1 次〜16 次) を用いる。音声分析条件を表 8.1 に示す。 Universal コードブックはテストデータ（音声資料）から Euclid 距離を用いて作成した。

表 8.1: 音声分析条件

音声特徴量	LPC ケプストラム係数
標本化周波数	12 kHz
LPC 分析	14 次
打ち切り次数	16 次
分析窓長	21.3 ms (256 点)
フレーム更新周期	10.7 ms (128 点)
高域強調	$(1 - 0.97z^{-1})$
コードブックサイズ	256

8.3.3 HMM の初期パラメータ

HMM には多くの種類があるが、本論文では各状態がシンボルを出力する状態出力タイプ (Moore 型) の離散型 (discrete) HMM を考えた。 また状態数（カテゴリ数）は、話者数と同数の 4 状態とした。

なお、Baum-Welch アルゴリズムは局所的最大点の方向に進むため、初期パラメータによって識別率が大きく変動することが知られている [39]。そこで初期パラメータを以下の 3 種類の方法で計算し、識別実験を行なった。

1. 実験１：（全パラメータに真値を与えた場合）

   HMM パラメータ $\boldsymbol{M}$ の初期パラメータとして、初期状態確率 $\boldsymbol{\pi}^{(0)}$ と

状態遷移確率 $\boldsymbol{A}^{(0)}$、とシンボル出力確率 $\boldsymbol{B}^{(0)}$ 全てに真値を与えた。なお話者および発話時間を決めて音声資料を作成しているため、真値はこれから直接計算できる。

2. 実験２：（シンボル出力確率に真値を与えた場合）

   初期パラメータとして、シンボル出力確率 $\boldsymbol{B}^{(0)}$ は真値を与えるが、初期状態確率 $\boldsymbol{\pi}^{(0)}$ と状態遷移確 率 $\boldsymbol{A}^{(0)}$ は等確率にした。

3. 実験３：（シンボル出力確率にランダムな値を与えた場合）

   初期パラメータとして、初期状態確率 $\boldsymbol{\pi}^{(0)}$ と状態遷移確率 $\boldsymbol{A}^{(0)}$ を等確率に、シンボル出力確率 $\boldsymbol{B}^{(0)}$ をランダムに与えた。なお、 $\boldsymbol{B}^{(0)}$ を等確率に与えた場合、Baum-Welch アルゴリズムは動かない。

8.3.4 識別率の評価方法

Viterbi アルゴリズムや forward アルゴリズムから得られる最適状態遷移系列はカテゴリの番号であって、カテゴリと話者の関係は未知である。そのため以下の式で識別率を算出した。

$$R = \frac{1}{T} \max_{\sigma} \sum_{t=1}^{T} d(\tau(\boldsymbol{x}_t), \sigma(S_t)) \tag{8.13}$$

ここで、 τ は最適状態遷移系列、σ は $(1, 2, \ldots, N)$ の任意の置換、S_t は各発話の正解カテゴリ番号、d は値が一致した時のみ 1 それ以外は 0 である関数である。

本節では、フレーム毎に、各カテゴリ番号を σ で置換し、$N!$ 通りの置換について正解率を算出しその中の最大値を識別率としている。従って話者が４名の場合 $24(= 4!)$ 通りの組み合わせを調べることになる。

なお、複数話者発話の音声データでは、LPC 分析のフレーム更新周期の間に話者が遷移する。このフレームでは話者を一意に決めることができない。そこで話者が遷移したフレームは、どちらの話者が選択されても正解にする。なお、実際の応用のときは、例えば１秒ごとにブロックにわけ、この間は、同一話者が話していると仮定して HMM を学習しても近似的には問題ない。

8.4 Ergodic HMM による複数話者発話の識別の実験結果

8.4.1 基本手法の実験結果

LPC ケプストラムを分析窓長 21.3ms で計算した時の実験結果を図 8.4 に示す。平均識別率は、実験１、２に関しては 8 セットの音声資料の平均値で、

実験３に関しては、 8 セットの音声資料それぞれに対し 16 回の異なる初期モデルで実験した、合計 128 回の平均値である。この図において、縦軸が平均識別率で横軸が HMM の学習回数である。□は実験１の、△は実験２の、○は実験３の結果である。また、実線で■▲●は Viterbi 復号法による平均識別率で、破線で□△○は forward アルゴリズムによる平均識別率である。この図からわかることを以下に示す。

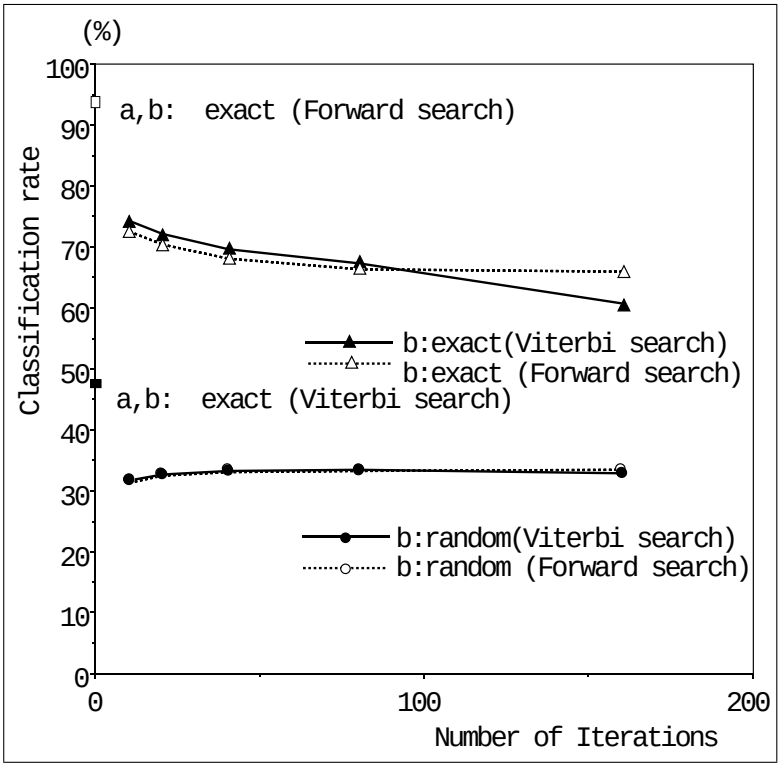


図 8.4: 学習回数と平均識別率の関係

1. Viterbi アルゴリズムと forward アルゴリズムの平均識別率の差は小さい。しかし、パラメータの全てを真値にした実験（実験１）では、forward アルゴリズムでは 94.0% であったのに対し、Viterbi アルゴリズムでは 48.3%しか得られな かった。この原因として Viterbi アルゴリズムは誤った経路を選択した場合、最後まで経路を間違えてしまうのに対し、 forward アルゴリズムは間違った経路を選択しても、その後正しい経路を選択する可能性があるためと考えている。

2. シンボル出力確率 $\boldsymbol{B}^{(0)}$ のみ真値にした実験（実験 2）では平均識別率約 75%が得られた。しかし、学習回数を繰り返すに従い、平均識別率は低下した。また $\boldsymbol{B}^{(0)}$ をランダムにした実験（実験３）では平均識別

率で 30%から 35%と低い値になった。この値は学習回数を増加してもあまり向上しない。この原因として、今回の実験は音素認識で使用される分析条件で実験を行なったため、パラメータは主に音素のカテゴリを特徴づけるパラメータになっていて、話者を特徴づけるパラメータになっていないためと考えている。

8.4.2 話者特徴量と長時間窓分析

話者識別の研究から話者識別には長時間平均スペクトルが有効であることが知られている [14]。従って分析窓長を長くすることによって話者の識別性能が向上することが予想される。そこで分析窓長を変化させたときの識別率の変化を研究した。Universal コードブックサイズは 256 と 64 で行なった。また、いずれの実験でもフレーム更新周期は分析窓長の半分とした。平均識別率は、各音声資料に対し、乱数により 16 個のランダムな初期モデルを作成して、その各々について試行を行なった音声資料 8 セットの平均値、すなわち計 128 回の試行に対する平均値で求めた。HMM の学習は 160 回繰り返して終了する。その他の実験条件は実験 3 と同一である。この実験結果を図 8.5 に示した。この図において縦軸は平均識別率、横軸は分析窓長である。

この図から以下のことが示される。

1. 分析窓を長くするに従い、平均識別率は向上するが、ある値を越えると低下する。

2. 分析窓長が 128ms 以下ではコードブックサイズ 256 の方が 64 よりも識別性能が高い。しかし分析窓長が 128ms 以上ではコードブックサイズ 64 の方が 256 よりも識別性能が高い。

3. コードブック 64、分析窓長 341ms において最も高い識別率が得られる。

これらの結果は、次のような原因によると考えている。

分析窓長を広げるとケプストラムのパラメータに含まれる話者特徴量は増加する。 従って、分析窓長を広げると識別性能が向上する。しかし、分析窓長を広げると得られるデータ量が減少する。例えば分析窓長 683ms ではデータは平均約 440 個である。そのため Ergodic HMM のパラメータの推定精度が低下する。よって分析窓幅がある閾値を越えると、識別性能は低下する。

なお、データ量の減少を防ぐために、フレーム更新周期を短くすることが考えられる。しかし、この場合フレーム更新周期の間に話者が遷移するデータが増加する。このデータは 2 人の話者の特徴量が入るため、不安定な特徴量を含む。したがってフレーム更新周期を短くしてデータ量を増加させても、識別率は向上しないことが予想される。実際に、フレーム更新周期を変化させて行なった実験で、この予想が確かめられた。

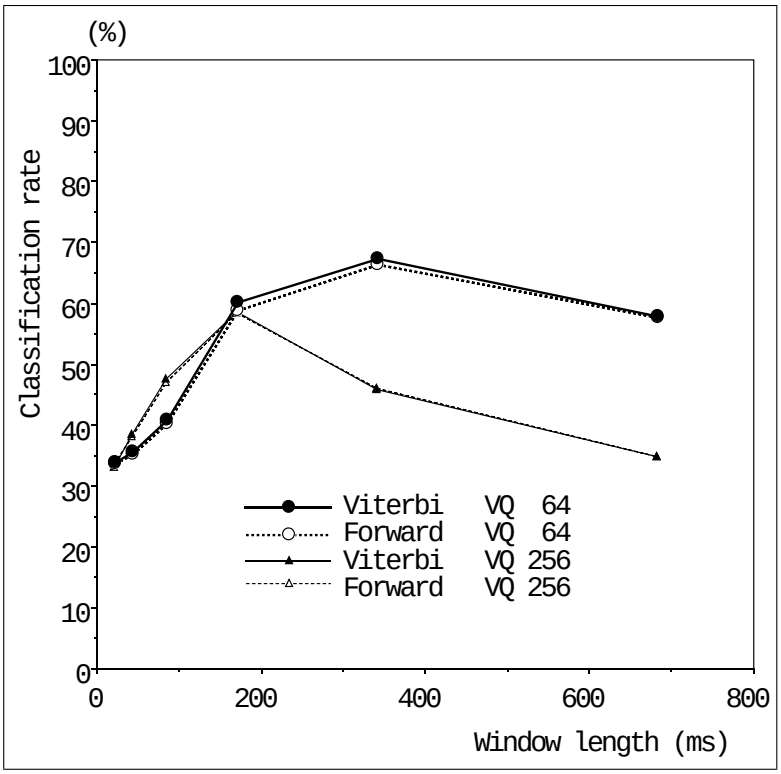


図 8.5: 分析窓長と平均識別率の関係

8.4.3 コードブックサイズ

ここでは Universal コードブックサイズの数を変化させたときの認識性能の変化を示す。コードブックサイズの数は推定するパラメータとパラメータ推定に使用されるデータ量との関係で決定される。複数話者発話の識別は不特定話者であるためコードブックサイズが 256 では小さい可能性がある。一方、サイズを大きくした場合、出現するコードの個数が少なくなるためシンボル出力確率の信頼性が低下すると思われる。実験は分析窓長 341.3ms と 170.7ms で行なった。その他の実験条件は前の実験と同一である。この実験結果を図 8.6 に示す。これからコードブックサイズの数が 64、分析窓長 341ms のとき認識性能が最大になることがわかる。

8.4.4 対数尤度に対する識別率の変化

Baum-Welch アルゴリズムは局所的最大点の方向に進むため、初期パラメータによって学習後のパラメータは大きく変化する。そして学習データに対する尤度も変化する。そこで尤度に対する識別率の関係を研究した。実験は分

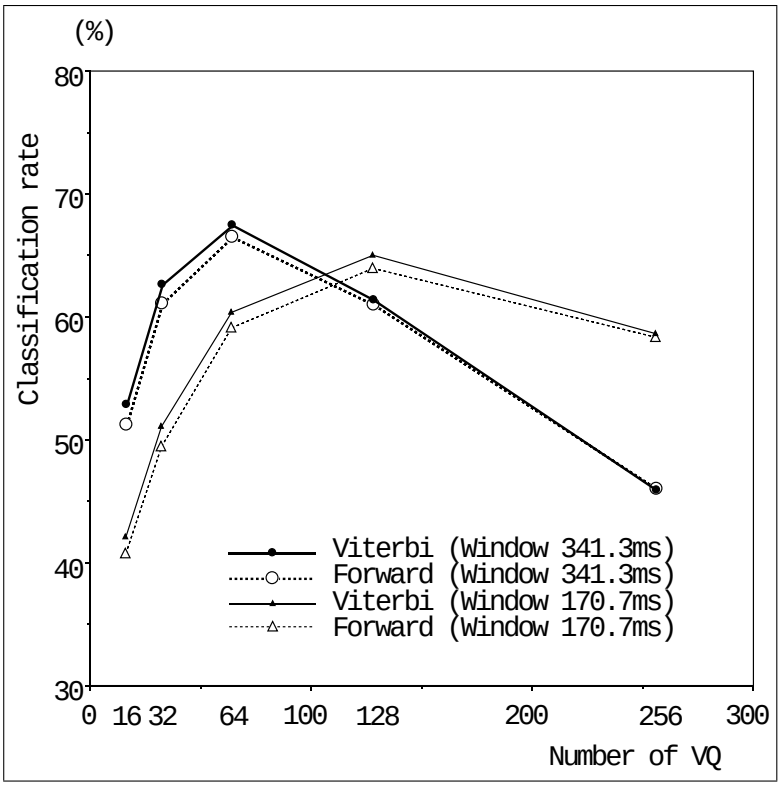


図 8.6: コードブックサイズと平均識別率の関係

析窓長 341.3ms、コードブックサイズを 64 とし、 HMM の学習は 160 回繰り返した。他の実験条件は実験 3 と同様である。1 セットの音声資料に対し、異なる 16 個の初期モデルを実験した 8 セットの音声資料の、合計 128 回の実験結果を図 8.7 に示す。縦軸は識別率を示し、横軸は対数尤度を示している。この図から HMM の尤度と識別率には相関があることがわかる。

8.4.5 初期モデルの選択

前の実験から、尤度の高いモデルは識別率が高いことが示された。そこで異なる初期モデルをランダムに 16 個作成し、 Baum-Welch 学習をした後、尤度の最も高いモデルを選択したときの、音声資料 8 セットに対する平均識別率を図 8.8 に示した。この図では、横軸がコードブックのサイズで縦軸は平均識別率である。 また分析窓長は 341.3ms である。その他の実験条件は実験 3 と同様である。この結果から、尤度の高い HMM を選択することにより平均識別率が約 10%程度向上し、最適な条件で 78.8%が得られることがわかった。

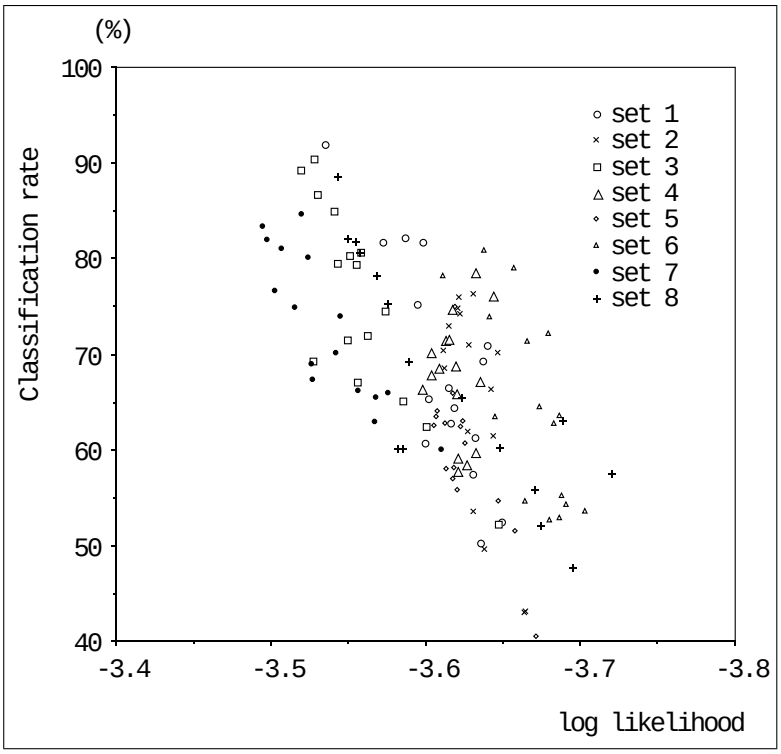


図 8.7: 尤度と識別率の関係

8.5 考察

本論文では、未知・複数信号源クラスタリング問題の中で複数話者発話の問題を取り上げた。しかし、多くの問題が未解決である。これらについて述べる。

1. Baum-Welch アルゴリズムにおける初期パラメータの設定

   HMM の尤度関数は様々な局所的最大点が存在する。従って初期パラメータの設定は重要である。今回の実験では尤度の高い HMM のモデルを選択することにより 78.8%の平均識別率を得た。しかし、分析窓長 341.3ms でコードブックサイズの数が 64 のときシンボル出力確率に真値をいれて Baum-Welch 学習をおこない、平均識別率を計算したところ 96.2%が得られた。したがって、別の初期パラメータの計算方法を考案することにより、より高い識別性能が得られる可能性がある。

2. 識別率の評価方法

   話者数が多い場合の識別率の評価方法を考える必要がある。今回の実験では全ての可能性を探索して、最も高い値を識別率としたが (式 (8.13))、

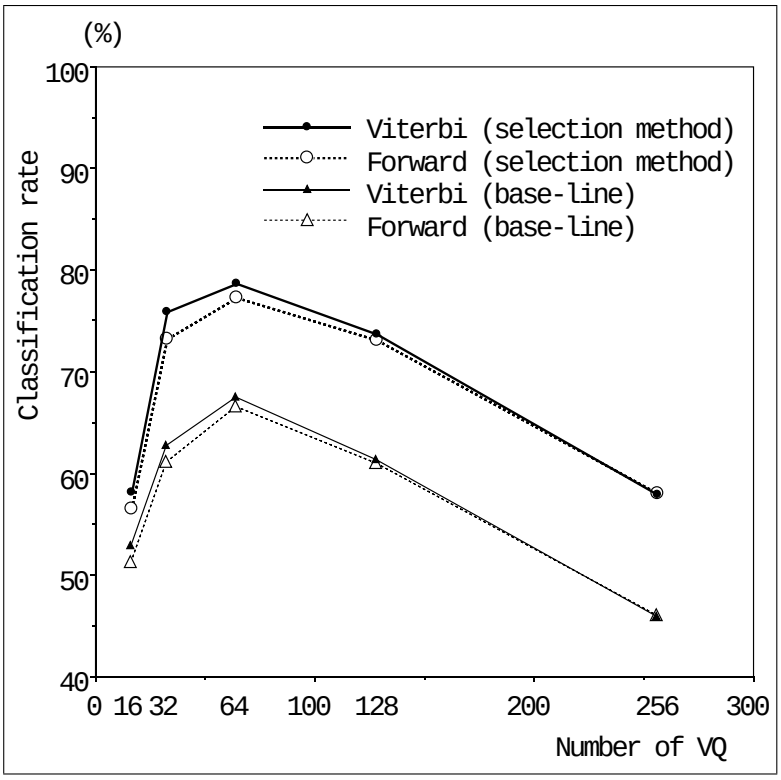


図 8.8: 初期モデルを選択したときの平均識別率の変化

この方法では、話者数が多くなるに従い、組み合せの数は階乗で多くなる。そこで評価方法の高速化が必要である。これに対しては、分岐限定法などの組み合せ最適化の適用が考えられる。

3. 話者識別の時間分解能

   複数話者発話の音声データでは、LPC 分析のフレーム更新周期の間に話者が遷移する。このフレームは話者を一意に決めることができない。従って話者識別の時間分解能はフレーム更新周期に依存する。この時間分解能を向上させる方法を考える必要がある。

4. Ergodic HMM の状態数

   一般に音声データには無音区間がある。この区間は、物理的に話者を特定できない。この解決方法として、Ergodic HMM の状態の数を話者の数より１つ多くして、無音区間のみを生成する状態を入れる方法が考えられる。

5. semi 連続分布型 HMM および連続分布型 HMM

HMM には多くの種類がある。言語のような離散型のデータを扱う場合は離散型 HMM が好ましいが、話者識別などで連続量を扱うには連続型 HMM が好ましいと考えられる。この場合コードブックサイズの問題や VQ 歪などの問題点がなくなるため、識別率は向上すると考えられる。たたし、今回のような場合は学習データが少ない場合が考えられる。したがって semi 連続分布型 HMM[4] も考慮する必要がある。

6. カテゴリ数 N の推定

   今回の実験では、話者数（カテゴリ数）を 4 として実験を行なった。そして、この話者数は事前にわかっていると仮定した。この話者数を推定する方法を考える必要がある。

8.6 まとめ

本論文では、信号系列を複数個の信号源に分割する問題を取り上げ、Ergodic HMM を用いた解法を示した。応用例として複数話者発話の識別をあげ、実験により識別性能を示した。この結果、以下の事柄が示された。

1. セグメンテーション位置とカテゴリ識別を同時に推定する、Ergodic HMM による解法では、パラメータ推定の際の収束計算の初期パラメータの設定が重要である。

2. HMM のパラメータの初期状態確率、状態遷移確率、シンボル出力確率のうち、シンボル出力確率が最も重要である。

3. 複数話者発話の識別においては 341ms 程度の長時間窓分析した LPC ケプストラムを用いることにより、より良好な識別性能が得られる。

4. 尤度の高いモデルを選択することにより平均識別率は向上する。

第9章　Ergodic HMMを用いた確率付きネットワーク文法の自動獲得

音声認識に利用される言語モデルには、ネットワーク文法や文脈自由文法に代表される構文モデル [31] や、bigram・trigram に代表される統計モデル [40][49] がある。

ネットワーク文法や文脈自由文法などの構文的な言語モデルは、自然言語処理の分野で実績があるが、言語に関する知識に基づいて構文規則を人間が記述するため多大な労力を要する。一方 bigram や trigram などの統計モデルは、簡単なモデルであるため音声認識の分野で言語モデルとして広く利用されている [40] が、このモデルは言語を表現するにはあまりにも単純である。そこで両モデルの問題点を補完するために、構文モデルに確率を加えた確率つきネットワーク文法や確率つき文脈自由文法などの研究がある [28][39]。

ところで音声認識の分野では隠れマルコフモデル (HMM) が広く利用されている [28]。HMM の種類の中で全状態間の遷移の許された離散型 Ergodic HMM の構造と確率つきネットワーク文法の構造は類似している。離散型 Ergodic HMM は状態遷移確率、シンボル出力確率、初期状態確率で特徴づけられ、確率つきネットワーク文法は、状態遷移確率と単語（品詞）出力確率を用いて記述した言語モデルである。Ergodic HMM の出力シンボルを単語（品詞）とすれば、両者は等価となる。また HMM は Baum-Welch アルゴリズムを用いることによって、学習データの生成尤度が最大になるように各パラメータを推定することができる。そこで言語モデルとして Ergodic HMM を考え、テキストデータを学習データとして Baum-Welch アルゴリズムを利用することにより、確率つきネットワーク文法を自動的に獲得できる可能性がある。

なお、村瀬等 [57] はカテゴリーを学習データとして学習後のモデルのエントロピーを研究し、bigram や trigram と比較し、Ergodic HMM による言語のモデル化の可能性を報告している。また、英語では Ergodic HMM は確率つきネットワーク文法の獲得手段としてでなく [28][39]、形態素解析として研究されることが多かった [38]。この場合、品詞ラベルが付与された大量のテキストデータがあれば HMM のパラメータは直接計算できるため、品詞ラベルがないテキストデータから Baum-Welch アルゴリズムを用いた大規模な実

験はまだ行なわれていないようである。

本論文では ATR 対話データベース (ADD)[10] における日本語会話文を全遷移型 (Ergodic)HMM でモデル化することを試みた。なお、入力データとして単語を選択した場合、Ergodic HMM はネットワーク文法と同時に、単語に対する新しい品詞体系を得ることができる可能性がある。この観点から学習後の Ergodic HMM のパラメータを研究した。また文音声認識における言語情報として用いたときの有効性なども研究した [94][95]。

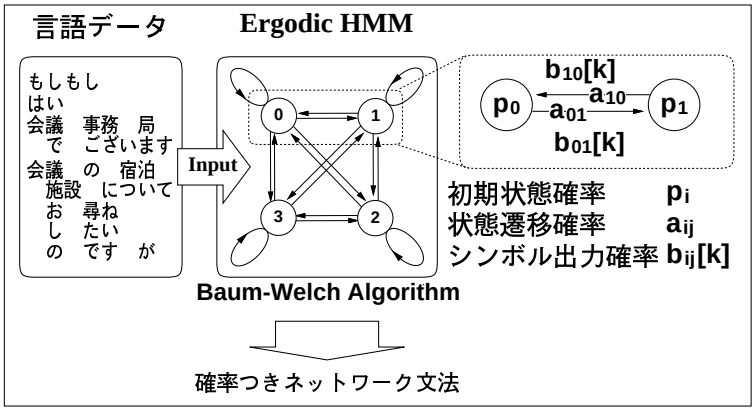


図 9.1: Ergodic HMM を用いた確率つきネットワーク文法文法の自動獲得

9.1 品詞列を入力とした文節内文法の獲得

9.1.1 文節データについて

9.1.1.1 文節の定義

ここでは、最初に HMM による日本語対話文の文節内における形態素の品詞連鎖のモデル化を行なった。本節では、(接頭辞、接尾辞の付加を含む) 自立語で構成される自立部と後続する付属語の連鎖で構成される意味的なまとまりを文節と定義した。

⟨ 文節 ⟩	$\longrightarrow$	⟨ 自立部 ⟩
⟨ 文節 ⟩	$\longrightarrow$	⟨ 文節 ⟩⟨ 付属語 ⟩
⟨ 自立部 ⟩	$\longrightarrow$	（接頭辞）⟨ 自立語 ⟩（接尾辞）.

実際の対話文では、明確な意味的解析を行なうことができない文節が多い。しかし、これらは、例えば、慣用的表現を一つの付属語と解釈して、より明確な意味的解釈を持つ文節に包括することで解消している。(例えば「参加 料

について な ん です けど」の「について」を格助詞とすることで全体を一つの文節とする。) 本研究では、このような拡張文節に対してモデル化を行ない、文節内文法を抽出することを試みた。

文節内の形態素には以下の 24 品詞を用いた。

形容詞	名詞	サ変名詞	代名詞	数詞	副詞
連体詞	接続詞	感動詞	助動詞	副助詞	接続助詞
格助詞	終助詞	接尾辞	接頭辞	補助動詞	固有名詞
形容名詞	本動詞	間投詞	準体助詞	並列助詞	係助詞

9.1.1.2 文法の複雑さ

モデル化されるデータのタスクの複雑度の指標として perplexity（2.1.10 節参照）を用いた。perplexity は、情報理論的な意味での平均分岐数である。例えば、言語 L の 1 単語あたりのエントロピーが $H(L)$ ならば、次の単語を決定するのに $H(L)$ 回の二者択一の選択が必要になる。言い換えれば $2^{H(L)}$ 個の単語から 1 単語を選び出すことになる。モデル化対象の集合を L とし、L のエントロピーを $H_0(L)$ とする。また、1 単語あたりのエントロピー $H(L)$ とエントロピーをもとに算出したパープレキシティ $F_p(L)$ は次のように求められる [60]。

$$L = \{w_k^i \mid w_k^i = w_{i1}w_{i2}\ldots w_{ik}\}, \quad \ldots \text{言語 } L \text{ の文集合} \tag{9.1}$$

$$H_0(L) = -\sum_{w_k^i} P(w_k^i)\log_2 P(w_k^i), \quad \ldots \text{言語 } L \text{ のエントロピー} \tag{9.2}$$

$$H(L) = -\sum_{w_k^i} \frac{1}{k} P(w_k^i)\log_2 P(w_k^i), \quad \ldots \text{言語 } L \text{ の 1 文あたりのエントロピー} \tag{9.3}$$

$$F_p(L) = 2^{H(L)}. \quad \ldots \text{言語 } L \text{ のパープレキシティ} \tag{9.4}$$

9.1.2 対話データ

文節内文法を獲得するためのデータとして、ATR 対話データベース (ADD) (3.3.1 節参照) の中から電話による 2 つの対話記録、国際会議及び旅行代理店に対する問い合わせ (以下 SET1・SET2) を使用した。なお、これらのデータベースには、形態素レベルから文・会話レベルまでのオブジェクトに関する情報、及びオブジェクト間の関係を表す情報が付加されているが、これらの情報から会話に関係しない記号や言い誤りの訂正などの部分を取り除いて、

文節区切りと文節内の形態素並びの情報を抽出した。それぞれのデータについての統計情報を**表 9.1** に示す。

表 9.1: 対話データ

(a) SET1

文節数	28500
品詞列の種類の数	1008
平均連鎖長	2.09 形態素 / 文節
パープレキシティ	9.02

文節長の分布	
1	12469
2	7835
3	4438
4〜	3758

品詞列の種類	
間投詞	15%
名詞+格助詞	12%
感動詞	11%
副詞	5%

(b) SET2

文節数	30419
品詞列の種類の数	1078
平均連鎖長	2.15 形態素 / 文節
パープレキシティ	8.82

文節長の分布	
1	12326
2	9282
3	4500
4〜	4311

品詞列の種類	
間投詞	18%
名詞+格助詞	13%
副詞	6%
名詞	4%

なお、**表 9.1** から、2 つのタスクとも 1 文節内が短い形態素の連鎖が多いことが分かる。これは、間投詞・感動詞のような 1 形態素 1 文節のパタンが多く存在するためと考えられる [102]。なお、キーボード会話では、間投詞・感動詞があまり現れないため分布が連鎖の長い方に移動する [11]。また、2 つのタスクの特徴的な違いは、SET1 に見られる感動詞の瀕出が SET2 には見られない。

9.1.2.1 モデル化実験

2 つの文節集合 (SET1・SET2) の文節内の形態素連鎖の品詞を状態数の異なる (2 状態, 5 状態, 8 状態, 10 状態)Ergodic HMM でモデル化した。

SET1・SET2 の各文節データは、形態素連鎖の品詞列のみをモデル化のデータとした。(文節の開始記号及び終了記号は付加していない。) また、任意の状態で遷移が開始・終了できるようなモデルとした。(HMM に開始状態及び終了状態を指定していない。)

HMM によるモデルの抽出には、Baum-Welch アルゴリズムを用いた。Baum-Welch アルゴリズムを用いた場合、再推定の回数の判定が問題となる。再推定回数の基準として、尤度 $P(\ \boldsymbol{O}\ |\ \boldsymbol{\lambda}\ )$ がある一定値に収束するまで再推定を繰り返す方法をとった。

Ergodic モデル HMM は、自由度が大きいためにモデル化によって得られたパラメータが初期状態によって大きく左右される。この初期状態の揺らぎによるパラメータの変化を考慮して初期状態の異なる HMM で実験を複数回行なった。

1. データの規模とモデル化の関係

   データの規模とモデル化の関係を調べるため、文節集合の大きさ (データの先頭から 100, 1000, 10000) を変えながら状態数の異なる (2 状態, 5 状態, 8 状態, 10 状態) HMM でのモデル化実験を行なった。

   実験は SET1 のデータをモデル化の対象とした。テストデータは、学習データに対してクローズなセットになっている。

   尤度の変化が 10^{-5} に収束するのを再推定の打ち切り条件として 10 回ずつモデル化を繰りかえし、各実験ごとにエントロピーを算出、平均した。

   表 9.2 に各データサイズにおけるモデルのエントロピーを示す。この結果エントロピーは、状態数の増加によって単調に減少するのではなく、ある状態数 (100 文節、1000 文節では 8 状態、10000 文節では 10 状態) で最小になることがわかる。また全般的に文節データ数の増加に従ってデータのバリエーションが増加するためモデルのエントロピーが上昇するが、品詞パターンの生成確率などを個別に分析するとデータ数の増加によって尤度が改善される結果が示された。

2. タスクの変化とエントロピーの関係

   次にタスクの変化とモデル化の関係を調べるため、2 つのタスク (SET1・SET2) をそれぞれ状態数の異なる (2 状態, 5 状態, 8 状態, 10 状態) HMM でモデル化した。実験は SET1・SET2 の全てのデータをモデル化の対象とした。実験は、HMM の各状態数について、尤度の変化が 10^{-5} に

表 9.2: データの規模とエントロピーの関係

SET1

状態数	2	5	8	10
100 文節	3.06	2.33	2.05	2.08
1000 文節	3.17	2.39	2.05	2.30
10000 文節	3.29	2.50	2.47	2.38

収束するのを再推定の打ち切り条件として初期状態を変えて 10 回ずつモデル化を繰り返した。

表 9.3: タスクの変化とモデル化の関係

(a) SET1

状態数	2	5	8	10
entropy	3.27	2.43	2.23	2.38

(b) SET2

状態数	2	5	8	10
entropy	3.18	2.41	2.27	2.14

このようにして算出した結果の平均値を**表 9.3** に示す。エントロピーは、状態数の増加によって単調に減少せず、タスクの雑さや種類に応じてある状態数で最小となっていることがわかる。

9.1.2.2 Ergodic HMM の解析結果

図 9.2 に 5 状態 HMM のモデルの概略を、**図 9.3** に 8 状態 HMM のモデルの概略を、**図 9.4** に 10 状態 HMM のモデルの概略を示す。

これらの図は、各 HMM における状態遷移確率分布行列 $\boldsymbol{A}$ 及びシンボル生成確率分布行列 $\boldsymbol{B}$ において、遷移確率および品詞の生成確率が 0.1 以上のものを抽出し、それ未満のものは省略することで、各 HMM の状態遷移と各遷移における品詞の生成確率を遷移ネットワークの形で表現している。

その結果、次の様なネットワークの特徴が観察された。

1. ネットワークの形態

   - 2 状態 HMM を除く各ネットワークに共通する部分として、3 個のノードで構成される自己遷移ループを持つ循環グラフが存在す

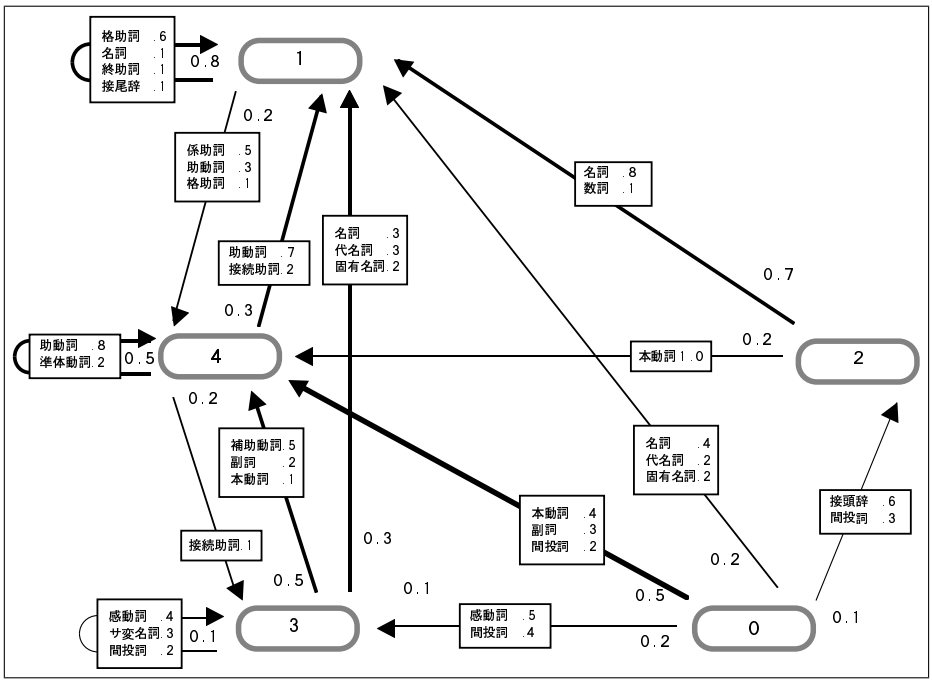


図 9.2: 5 状態 HMM による遷移ネットワーク

る。その他のノードは、この循環グラフに接続する非循環グラフを構成している。この循環グラフ部を分析した結果を以下に示す。

(a) 自己遷移を持つノードのうち、ひとつは体言に付属する部分に、もうひとつは用言に付属する部分に相当する。(8 状態 HMM **図 9.3** ノード ○0, ○5)

(b) 残りのノードは、感動詞、間投詞の自己遷移ループで、用言や副詞などを生成した後、用言に続く部分を生成するノードに遷移する。

(c) 2 状態 HMM によるモデルは、他の状態数のモデルとは異なり、体言と用言にそれぞれに付属する語の部分の自己遷移ループを自立語を生成するリンクが結合するモデルと、体言、用言のような自立語の分化が縮退し自立語を生成するリンクと付属語を生成する自己遷移ループのみで構成されるモデルの形をとる。

- 非循環グラフの部分は、1 回から 3 回の遷移で循環グラフ部に到達する。状態数の少ない HMM では遷移は短く、状態数の増加に従って長い遷移が現れる。
- すべてのグラフに共通して見られる傾向として、体言に続く部分

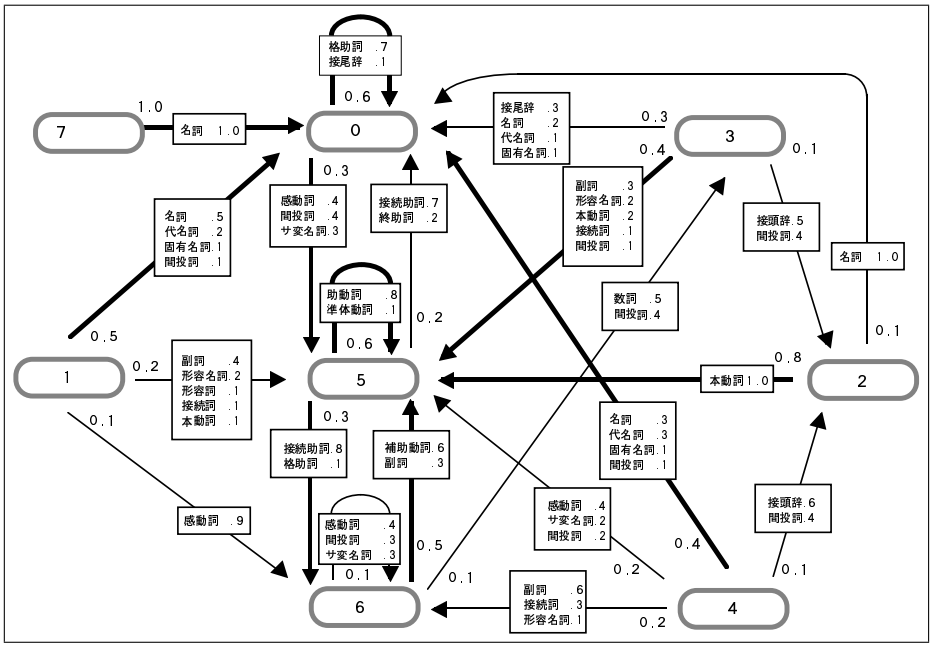


図 9.3: 8 状態 HMM による遷移ネットワーク

に接続するリンクは、体言が支配的に生成されるのに対し、用言に続く部分に接続するリンクは、用言や副詞などの語が支配的である。

(a) これらのノードは、5 状態、8 状態の HMM では体言あるいは用言を生成するリンクのいずれか一方のみを持ち、ノードがはっきりと分化している。

(b) 10 状態の HMM では、[接頭辞 → 名詞 | 数詞 → 接尾辞] といった複雑な遷移が観察できる一方、ノード数が増え、リンクが増加しているにも関わらず、支配的な遷移を行なうリンクがさほど増えていない。状態遷移確率が平均化され、あるノードが体言・用言のどちらに分化しているのか明確に識別できない。

2. 同一タスクにおけるモデル化の揺らぎ

ネットワークを構成する各遷移は、様々な構文規則 (名詞+格助詞、動詞 助動詞 準体助詞 補助動詞 終助詞など) が重ね合わさって構成されていると考えられる。

初期状態の違いによるモデル化の揺らぎは、これらの構文規則の重なり方の違いによって生じると考えられる。そこで、まずネットワークの形態と情報量（エントロピー）の関係を調査した。

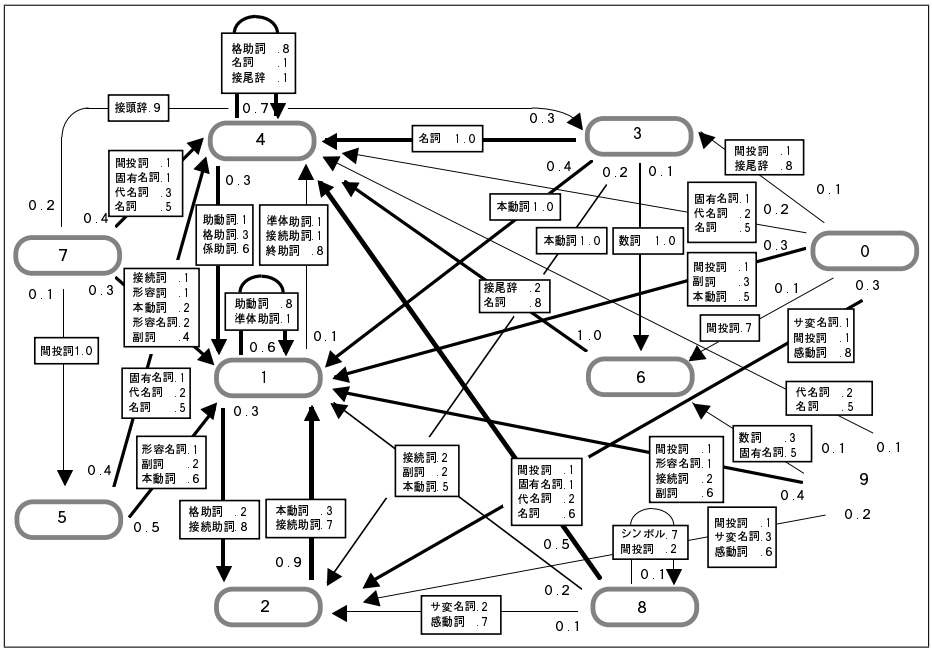


図 9.4: 10 状態 HMM による遷移ネットワーク

HMM のモデルの形態的な特徴を把握するため、次のようにしてモデルからネットワークを抽出した。

(a) モデル内の全ての $a_{ij}, b_{ij}(k)$ についてシンボル (品詞) 生成確率が最小になるリンクを削除する。

(b) 削除後、残されたリンクの確率を正規化し、モデルの情報量（エントロピー）がある一定値になるまで 1. を繰り返す。

重ね合わさっていると考えられる個々の構文規則を抽出・解析することは困難なので、特定の構文規則について各 HMM におけるモデル化の状態を調べた。

調査の対象となる構文規則として副詞を含む文節を取り上げた。

- 副詞を含む文節は、そのほとんど全てが副詞を先頭に始まる。唯一の例外は、副詞の前に接頭辞が先行するもので (「お　いくら」など) その連鎖確率は 10^{-3} 以下である。そのため、副詞以前の文脈を考慮する必要がなく解析が比較的容易である。
- 文節内文法の構造が比較的単純で明確に分化していること。(副詞が単独で文節になるもの「例えば」、体言が後続するもの「もう一　つ」、用言が後続するもの「どう　でしょ　う」など、それぞれの割合は、69.5%, 1.9%, 28.5% (SET1)、である。)

副詞を含む文節に着目したモデルの解析によって次の結果が得られた。

- 5 状態 HMM では、副詞を生成する遷移は、モデルにつき 1 ないし 2 箇所に存在する。その全てが用言に付属する語を生成する部分に結合する。解析を行なった 10 モデルのうち、副詞を生成する遷移が 1 箇所だったものは 6 モデル、残りは 2 箇所の遷移で副詞が生成されていた。この 2 つのグループの間には、エントロピーの有意な差は見い出せなかった。
- 8 状態 HMM では、同様の遷移は、モデルにつき 2 から 5 箇所に存在する。副詞を生成する遷移の少ないモデルでは、5 状態 HMM に同じく用言に付属する語を生成する部分に結合する。これらのモデルのエントロピーの平均は 2.51 であった。10 モデルのうち 2 モデルに副詞を生成するリンクが体言に付属する語を生成する部分にも結合していた。これらのモデルのエントロピーは 2.20 で若干の減少がみられた。
- 10 状態 HMM では、エントロピーの小さいモデルが複数存在したが、どのモデルも副詞を生成するリンクが体言に付属する語を生成する部分に結合していなかった。

3. タスクの違いによるモデル化の変化

   データベースの説明で報告したように、モデル化に使用したデータは、品詞の生成確率が若干異なっている。(感動詞の生成確率が 11% (SET1)、3% (SET2)) 先に示した副詞を含む文節の生成確率も同様に異なっている。(副詞単独、体言が後続、用言が後続の 3 グループの割合は、69.5%, 1.9%, 28.5% (SET1)、86.1%, 3.2%, 10.6% (SET2) である。)

   - 感動詞は、単独で文節となるので SET1 と SET2 で出現確率が異なってもモデルの遷移ネットワークの構造に大きな差異は見られない。ただし、ある遷移が感動詞だけを生成する割合は SET1(16.8%) より SET221.1% が大きく、タスクがモデルに反映されている。
   - 体言が後続する遷移の割合が SET1 (1.9%) より SET2 (3.2%) の方が大きい。このため、SET2 をモデル化した場合、副詞が生成されるリンクは、状態数 5 の HMM から体言と用言の二つのグループに分化する。この場合もタスクが反映されている。

9.1.2.3 モデルからの文法抽出

モデル化実験で得られた結果のうち、もっともエントロピーの小さかったモデル (10 状態 HMM、SET2、entropy = 1.98) から抽出した文法の概略を

図 9.5 に示す。抽出のアルゴリズムは、同一タスクにおけるモデル化の揺らぎを調査した時と同じものを用いた。

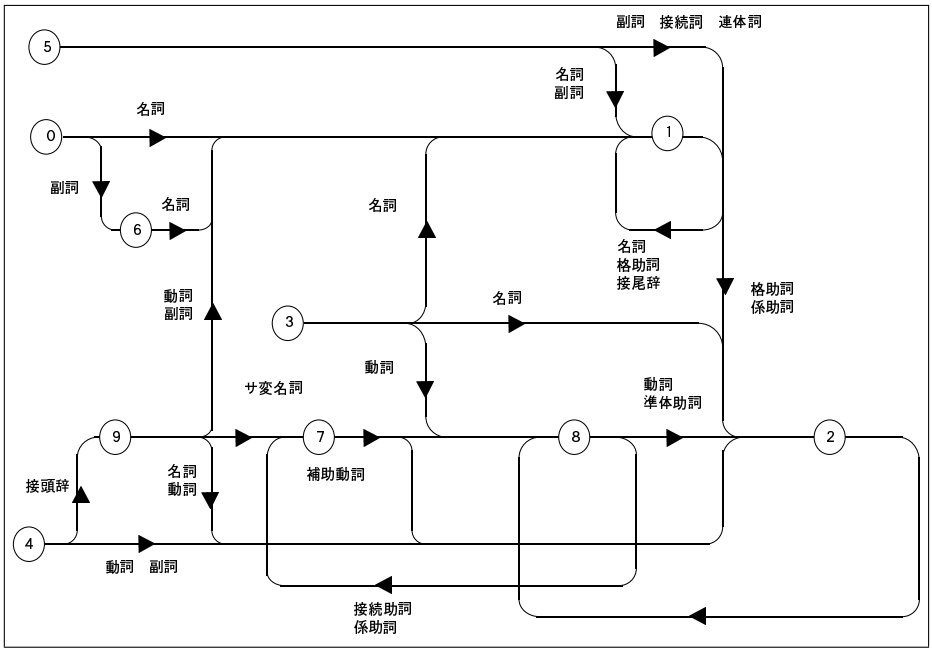


図 9.5: 10 状態 HMM から抽出した文節内文法

図 9.5 に示される番号は、HMM の状態番号である。 先に述べた副詞を含む文節を生成する遷移は、図の左半分、 ○0→ ○6 、 ○4→ ○9の部分に相当する。図の上半分、 ○5→ ○1 、○3→ ○1の部分は、接尾辞を伴う名詞ないし複合名詞を生成する。図の下半分は、用言を生成する遷移に相当する。○4 → ○9 が自立部を形成し、○7, ○8, ○2が付属語を生成する。なお、間投詞は状態 ○1, ○2, ○8, ○9 を除く全ての状態から遷移する時に生成されるが図では省略している。また、このモデルでは、任意の状態から遷移を開始することができるが状態 ○5, ○0, ○3, ○4 は、他の状態からの遷移がないので、事実上イニシャル・ノードとなっている。

図中の遷移には、○0→ ○1 、○5→ ○1 、○3→ ○1 、のように異なる経路で同じ品詞列を生成するものがあり、やや冗長な構造になっている。ただし、この他のモデルでは、エントロピーの増加とともに冗長なパスや生成する品詞が曖昧な遷移を多数生じるようになる。

文法抽出に対する考察

HMM による文法抽出は、エントロピーを指標としてモデルを評価することができるが、最適なモデルを得るためには、生成・評価のサイクルを繰り返しながらエントロピーの小さなモデルを探す必要がある。これを避けるには、例えば、エントロピーの小さなモデルを生成した HMM の初期状態を解

析し、得られた知見によってよりエントロピーの小さいモデルを生成するように初期状態を設定する手法が考えられる。

初期状態とともに考慮すべきこととしてモデルの状態数の最適化がある。状態数の多い HMM はより複雑なタスクを扱うことができるが状態数が多過ぎるとかえって冗長・曖昧性が生じ、モデルの最適性が失われてしまう。この問題は、タスクのパープレキシティを考慮して状態数を選択することにより、ある程度解消できる。

さらに、同じ品詞でも連体修飾・連用修飾を行なうものは、遷移が分化しているものがあるように、品詞のサブカテゴリを扱うことでより詳細なモデルを生成することや、逆に HMM によって品詞のサブカテゴリ化を図ることができる。

9.1.3 まとめ

HMM はランダムプロセスのモデルであるが状態数や初期状態を考慮することにより言語の持つ非ランダム性を遷移ネットワークの形で抽出・獲得する可能性がある。

考慮すべき HMM のパラメータのひとつとして状態数がある。状態数の増加は、HMM のモデル化能力の増大させるが現象の非ランダム性を抽出するには、現象のタスクの複雑さに応じた値を選択する必要がある。もう一つのパラメータとして HMM の初期状態がある。本研究ではランダムな分布確率を用いたが、初期状態分布確率によってイニシャル・ノードを指定することができ、また状態遷移確率分布とシンボル生成確率分布を操作することでモデルの構造をあらかじめ決めておくことができる。

このようにパラメータを考慮することにより経験的に得られている生成文法に似た形態の確率つきネットワーク文法を自動的に獲得することができた。

9.2 単語を入力単位とした日本文文法の自動獲得

9.2.1 HMM による言語のモデル化

前節では、Ergodic HMM の学習に品詞などのカテゴリーに分類した言語データを用いた。本節では、実際の会話から作成した文単位に、単語列（品詞系列ではない）を Ergodic HMM に学習させて、確率つきネットワーク文法を自動的に抽出することを試みた。品詞情報を持たない単語列を学習させることにより、文法だけでなく、単語のカテゴリーも出力の偏りとして同時に学習されることが期待できる。

また得られた言語モデルを文音声認識システムの言語モデルとして利用することで、この言語モデルの有効性を研究した。

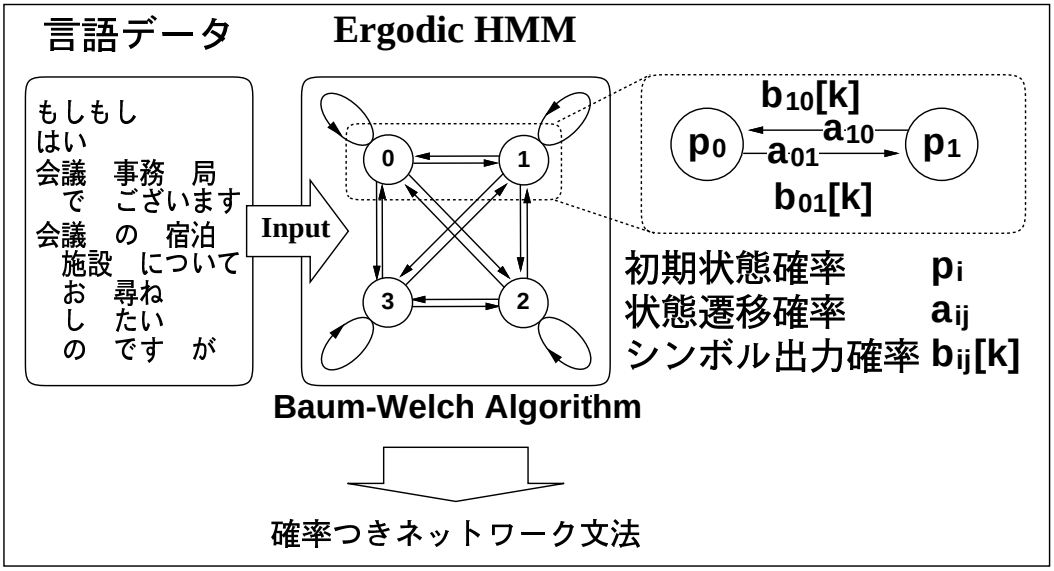


図 9.6: HMM を用いた文法の自動獲得

9.2.2 言語データ

HMM の学習に用いる言語データベースとして、ADD(ATR 対話データベース) の中から “国際会議に関する問い合わせ” の 8000 文を用いた。文の例を表 9.4 に示す。

表 9.4 のように、このデータベースはあらかじめ単語（形態素）に区切られており、同じ表記でも読み方の異なる場合や、品詞、活用形、活用型の異なる場合は、別単語として扱っている（表 9.5 参照)。また、日本語において、単語の概念はあいまいであるが、単語の単位はデータベースの形態素解析 [10] に依存した。

実験に使用したデータベース中の単語の種類は全部で 6418 種類である。品詞は 25 種類に分類され、活用を持つものは、さらに活用形および活用型の違いで分類している。これらを表 9.6 および表 9.7 および表 9.8 に示す。

また、これらの単語には数字のラベルが付いており、このデータでは 0 から 6417 までの数字が付けられている。

実験では、データベースの 8000 文を奇数番目の文の set と偶数番目の文の set とに分け、さらにそれぞれ先頭から 1000 文の set、先頭から 2000 文の set、4000 文の set に分けている。奇数番目の set3 種類をそれぞれ odd1000, odd2000, odd4000、偶数番目の set を even1000, even2000, even4000 と名づけ使用した。それぞれのデータ (set) における品詞の出現頻度などは、表 9.9 および付録の表 A.1 〜表 A.8 に示す。

これらの表から、以下のことが示される。

表 9.4: 文の例

はい もしもし
えーっと そちら 第 1 回 の 通訳 電話 国際 会議 の 事務 局で しょ う か
はい そう です
えーっと ちょっと その 会議 の こと でね
はい どうぞ
えーっと 今 手元 に あの 登録 用紙 が ある ん です けれども
えーっと その 中 で ちょっと あの クレジットカード を ね
あのー クレジットカード の 名前 と なん か ナンバー を 書くところ が ある ん です が
はい そう です
えーっと それ を ちょっと クレジットカード を 持っ て い ない 者 が いる ん です けれども
その 場合 は どう な ん でしょ う か

表 9.5: 同一表記の単語の扱い

車を持って ない 人もいる。	助動詞連体形
がまんするほか ない。	形容詞終止形
これしか ない のですか。	形容詞連体形

1. データベース中の文は、主に、普通名詞、格助詞、本動詞、助動詞で構成される。
2. 一文章の平均単語数が 13 前後である（表 9.9 ）。しかし、実際の分布は単語数の少ない方に偏っている。
3. 「はい。」「もしもし。」の様な感動詞 1 単語のみからなる文や「わかりました。」「そうですか。」などの受け答えの会話が多く存在する。
4. 間投詞（あのー、えー など）や感動詞（もしもし、はい など）が多く含まれている。

これらの特徴は、電話対話という特殊な環境を反映した言語データを意味していると考えている。

表 9.6: 品詞分類

形容詞	副詞	副助詞	接頭語	間投詞
普通名詞	連体詞	接続助詞	補助動詞	準体助詞
サ変名詞	接続詞	格助詞	固有名詞	並立助詞
代名詞	感動詞	終助詞	形容名詞	係助詞
数詞	助動詞	接尾語	本動詞	慣用句

表 9.7: 活用形分類

変則型	五段	上一	下一	サ変	カ変	特殊
文語四段	文語上二	文語下二	文語ラ変	文語ナ変	形容詞ク変	

9.2.3 Ergodic HMM を用いた確率つきネットワーク文法の自動獲得の実験

前節で作成した対話データ set odd1000, odd2000, odd4000 を Ergodic HMM の学習データとして使用して、HMM の状態数を変えて、Ergodic HMM を用いた確率つきネットワーク文法の自動獲得の実験を行なった。

なお、実験に用いた Ergodic HMM は任意の状態から状態遷移を開始し、任意の状態で終了できるモデルである。学習を始める際のモデルのパラメータ A, B, Π の初期値を以下に示す。

$$\Pi : \pi_i = 1/N, A : a_{ij} = random, B : b_{ij}(v_k) = random \qquad (9.5)$$

ただし、 $1 \leq i, j \leq N$, $1 \leq k \leq L$, $\sum_{j=1}^{N} a_{ij} = 1.0$, $\sum_{k=1}^{L} b_{ij}(v_k) = 1.0$
N : 状態数, L : 語彙数（単語の種類）

また、モデルがシンボル系列を生成する確率の上昇率

$$\frac{\text{学習後の尤度} - \text{前回の学習後の尤度}}{\text{学習後の尤度}}$$

がある一定値以下になったとき、HMM の学習を終了させた。

表 9.8: 活用型分類

未然	連用	終止	連体	仮定	命令	語幹

表 9.9: 構成単語数

set	odd1000	odd2000	odd4000	even1000	even2000	even40000
文数	1000	2000	4000	1000	2000	4000
単語数	13299	20730	57354	13824	21114	56826
文平均単語数	13.30	10.37	14.34	13.82	10.56	14.21
最大単語数	99	99	128	81	81	118

実験は、状態数の異なる 4 種類の Ergodic HMM(2 状態、4 状態、8 状態、16 状態) の場合について行なった。学習データは odd1000, odd2000, odd4000 を用いた (16 状態は odd4000 のみ)。また、初期パラメータを変えた場合のモデル化の変化を研究するため、初期状態の異なる 8 種の 8 状態の Ergodic HMM について、再推定回数 20 回を学習終了条件として odd4000 を学習させた。その他の実験の条件を表 9.10 に示す。

表 9.10: 言語モデル生成実験の条件

HMM の構造	状態遷移出力型 Ergodic HMM
HMM の状態数	2 状態,4 状態,8 状態,16 状態
HMM の出力シンボル	単語
開始 · 終了状態	任意
初期状態遷移確率	ランダム
初期シンボル出力確率	ランダム
初期状態確率	均等
語彙数	6418
学習データ set	odd4000,odd2000,odd1000
学習データ数	4000 文,2000 文,1000 文
単語総数	57354 単語,20730 単語,13299 単語
学習終了条件	尤度上昇率 1%未満

9.2.4 実験結果

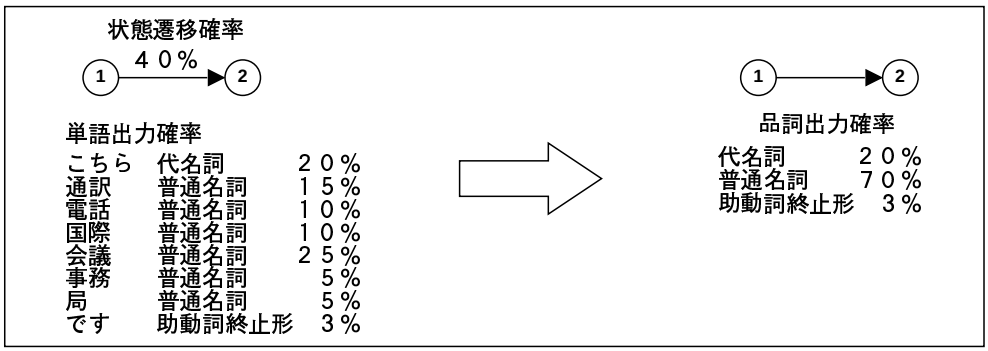
学習された Ergodic HMM を評価するために、HMM を解析して獲得されたモデルの特徴を研究した。

9.2.4.1 Ergodic HMM の解析

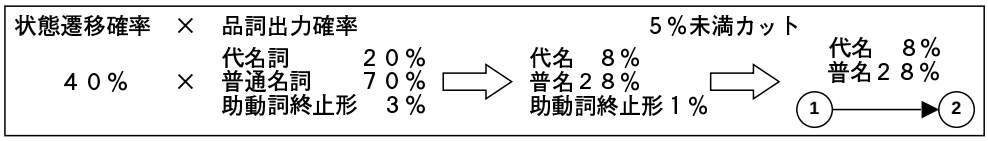
Ergodic HMM の解析方法を以下に示す。

1. 単語に品詞 (活用するものは活用形の区別をしている) のラベルをつける。

2. 各遷移について、同一品詞の単語のシンボル出力確率の和を品詞ごとに求める。



3. (状態遷移確率 × 出力確率) が 5%未満の品詞はカットしてネットワークを表示する。



ネットワークの表示に用いた略号を表 9.11 に示す。

以後示すネットワークの図では、遷移の太細は遷移確率の大小を示し、品詞名、活用形の略号の右の数字は (状態遷移確率 × 出力確率) の値を示している。また、初期確率が最大になっている状態 (イニシャルノード) は太丸で示した。

1. 2 状態 Ergodic HMM の解析結果

   2 状態の Ergodic HMM について解析した結果を図 9.7 に、初期状態確率、状態遷移確率を表 9.12 に示す。2 状態の Ergodic HMM で見られる特徴を以下に示す。

表 9.11: 品詞・活用形の略号

品詞名	略号	品詞名	略号	品詞名	略号	品詞名	略号	品詞名	略号
普通名詞	普名	助動詞	助動	格助詞	格助	準体助詞	準助	未然形	未
代名詞	代名	間投詞	間投	係助詞	係助	接頭辞	接頭	終始形	終
本動詞	本動	感動詞	感動	接続助詞	接助	接尾辞	接尾	連体形	体
補助動詞	補動	副詞	副詞	終助詞	終助	連用形	用		

(a) 主として、○1⇒ ○0の遷移で普通名詞を出力し、これに続く状態○0からの遷移 (○0⇒ ○1, ○0⇒ ○0) で普通名詞に接続する格助詞を出力している。

(b) 実験に用いた言語データは電話での対話であるため、間投詞 (あのー、えー、など) や感動詞 (もしもし、はい、など) が文頭や文の切れ目で用いられた文が多く含まれている。これらが Ergodic HMM では、太丸で示したイニシャルノード (=遷移を開始する状態=文頭) ○1のループで出力されており、学習データの特徴を示している。

(c) 状態数が少ないために、全体的に表現力が乏しい。

表 9.12: 2 状態 Ergodic HMM のパラメータ

初期状態確率	$\pi_0 = 0.002$	$\pi_1 = 0.998$
状態遷移確率	$a_{00} = 0.36$	$a_{01} = 0.64$
	$a_{10} = 0.54$	$a_{11} = 0.46$

2. 4 状態 Ergodic HMM の解析結果

4 状態の Ergodic HMM について解析した結果を図 9.8 に、初期状態確率、状態遷移確率を表 9.13 に示す。

2 状態の Ergodic HMM に比べ、遷移の数が増加していて文法的な特徴が見られる。また、単語の分離が生じ、ネットワーク上では品詞ごとに集まって出力されているのがわかる。そして、体言と用言（活用する品詞）の分離が特徴的である。以下に 4 状態 Ergodic HMM に見られる特徴を述べる。

(a) 文頭の間投詞、感動詞はイニシャルノード○1のループで出力されている。

(b) 名詞は主として、状態○0に集まる遷移 (○2⇒ ○0, ○3⇒ ○0) で出力される。

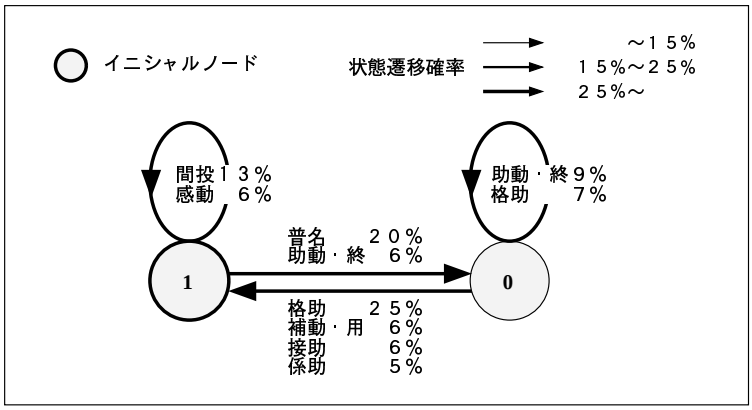


図 9.7: 2 状態 Ergodic HMM の解析結果

(c) 活用する品詞（本動詞、補助動詞、助動詞、形容詞）は状態○2○3 に集まる遷移 (○2○3⇒ ○2, ○0○1○2⇒ ○3) で出力されている。そして、連体形のものは状態○2に集まる遷移 (○2○3⇒ ○2) で出力され、連用形のものは主に状態○3に集まる遷移 (○1○2○3⇒ ○3) で出力されている。

表 9.13: 4 状態 Ergodic HMM のパラメータ

初期状態確率	$\pi_0 = 0.00$	$\pi_1 = 1.00$	$\pi_2 = 0.00$	$\pi_3 = 0.00$
状態遷移確率	$a_{00} = 0.38$	$a_{01} = 0.41$	$a_{02} = 0.07$	$a_{03} = 0.14$
	$a_{10} = 0.27$	$a_{11} = 0.43$	$a_{12} = 0.17$	$a_{13} = 0.13$
	$a_{20} = 0.35$	$a_{21} = 0.11$	$a_{22} = 0.22$	$a_{23} = 0.32$
	$a_{30} = 0.04$	$a_{31} = 0.48$	$a_{32} = 0.23$	$a_{33} = 0.25$

3. 8 状態 Ergodic HMM の解析結果

8 状態の Ergodic HMM について解析した結果を図 9.9 に、初期状態確率、状態遷移確率を表 9.14 に示す。

図 9.9 から、品詞、活用形が同じ単語が、4 状態の場合よりも同じ状態遷移に収束して出力されていることがわかる。体言や活用する品詞など同一品詞が複数の遷移で出力されている。また、2 状態、4 状態の場合に比べ、より細かな文法的特徴が見られる。以下に、8 状態 Ergodic HMM から抽出される特徴を述べる。

(a) 間投詞、感動詞について

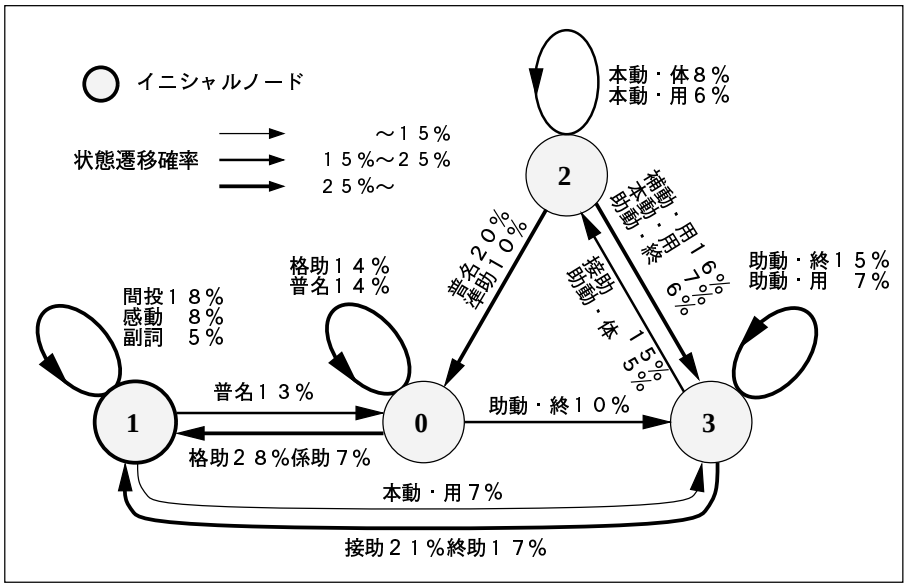


図 9.8: 4 状態 Ergodic HMM の解析結果

間投詞 (あのー、えー、など) や感動詞 (もしもし、はい、など) がイニシャルノード (遷移を開始する状態) ○0からの遷移で出力されている。

(b) 品詞の基本形と活用形について

本動詞、補助動詞、助動詞、形容詞のような活用する単語は品詞の基本形ではなく、活用形でグループ化されている。連体形は状態○3への遷移 (○4, ○1⇒ ○3) で、連用形は状態○4への遷移 (○4, ○5, ○6⇒ ○4) で、助動詞の終止形は状態○6への遷移 (○1, ○4, ○5⇒ ○6) で、それぞれ出力されている。

名詞は主として状態○0○3○7から状態○1○2に遷移する際に出力されている。しかし、各々の遷移には特徴がある。○3からの遷移では、形式名詞が他の名詞類よりもやや高い。また、○3⇒ ○1の遷移では準体助詞「の」が出力されている。状態○7からの遷移では、個々の単語出力の上位は代名詞が多い。状態○0からの遷移では、主として形式名詞「方」(ほう) が出力されている。また、名詞以外の単語では副詞の出力が多く見られた。

(c) 格助詞について

格助詞は複数の遷移 (○2⇒ ○0, ○4, ○5, ○7), (○1⇒ ○5, ○7) で出力されているが、各遷移は別々のそれぞれ異なる単語を出力している。例を以下に示す。

表 9.14: 8 状態 Ergodic HMM のパラメータ

	j=0	j=1	j=2	j=3	j=4	j=5	j=6	j=7
初期状態確率 π_j	0.999	0.000	0.001	0.000	0.000	0.000	0.000	0.000
状態遷移確率 a_{ij}								
i=0	0.20	0.15	0.18	0.05	0.05	0.03	0.02	0.33
i=1	0.02	0.04	0.05	0.05	0.04	0.35	0.28	0.17
i=2	0.29	0.05	0.12	0.03	0.15	0.24	0.02	0.10
i=3	0.02	0.57	0.23	0.03	0.01	0.05	0.02	0.06
i=4	0.02	0.01	0.08	0.19	0.18	0.26	0.24	0.03
i=5	0.04	0.04	0.10	0.10	0.31	0.06	0.17	0.19
i=6	0.27	0.02	0.19	0.03	0.16	0.04	0.01	0.28
i=7	0.05	0.14	0.24	0.09	0.05	0.04	0.09	0.30

i. 状態◯2⇒ 状態◯0 「の」 (94%)
ii. 状態◯2⇒ 状態◯7 「から」 (28%) 「で」 (25%)
iii. 状態◯2⇒ 状態◯4 「と」 (94%)
iv. 状態◯2⇒ 状態◯5 「に」 (73%) 「で」 (18%)
v. 状態◯1⇒ 状態◯5 「を」 (39%) 「が」 (27%)
vi. 状態◯1⇒ 状態◯7 「は」 (80%) 「も」 (7%)

なお、付録 B に 8 状態 Ergodic HMM から抽出される細かい特徴を述べた。

4. 16 状態 Ergodic HMM の解析結果

16 状態の Ergodic HMM について解析した結果を図 9.10 に示す。ネットワークが複雑なため、より簡略化した略号を用い (表 9.15 参照)、出力確率値の%は省略して表示した。また、初期状態確率を表 9.16 に示した。

16 状態の Ergodic HMM では、状態数の少ないモデルに比べ、多くの状態遷移において一つの品詞のみが出力され、状態遷移ごとの単語の出力の偏りが顕著になっている。

8 状態に比べ、活用する品詞の記述がより細かくなり、また、2 つの状態 (◯2, ◯3) で初期状態確率の値を持つなど、より複雑なネットワークを形成している。

８状態の場合と同様に、同一品詞が複数の遷移で出力される場合は、遷移の起点か、または終点が同じ状態であることが多い。主に、ネットワー

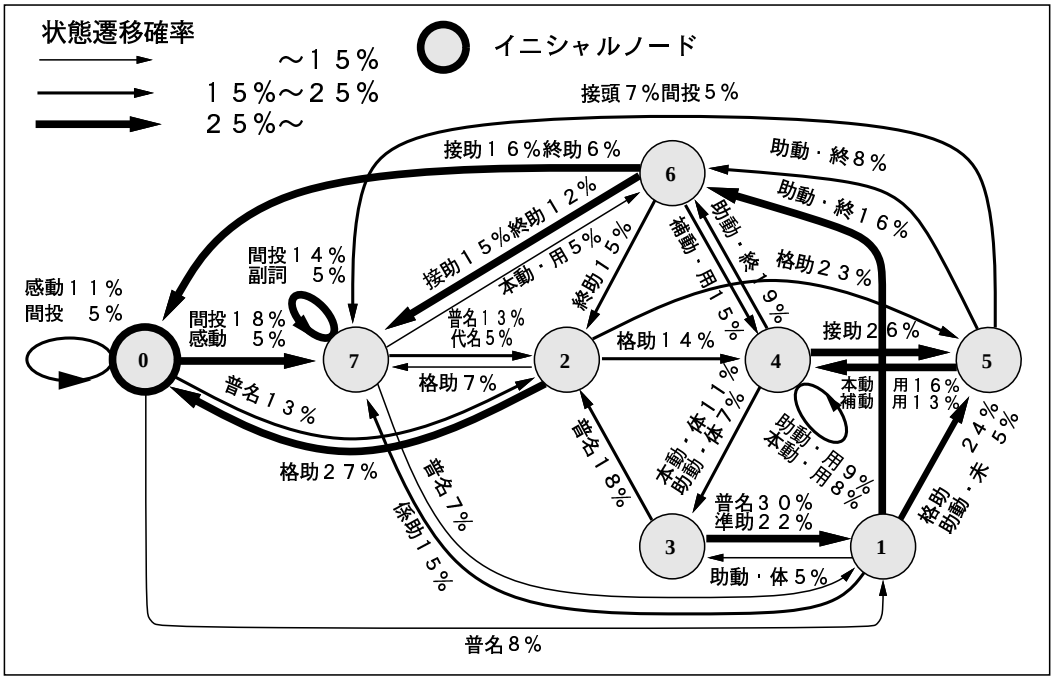


図 9.9: 8 状態 Ergodic HMM の解析結果

クの左の部分で「名詞＋格助詞」などの主部が記述され、右の部分で「動詞＋助動詞」などの述部が記述されている。以下に 16 状態 Ergodic HMM から抽出される特徴を品詞ごとに述べる。

(a) 間投詞、感動詞について

間投詞、感動詞は、4、8 状態の場合と同様に初期状態確率の高い状態 (○2, ○3) からの遷移で出力される。ただし、初期状態確率は状態○2○3の 2 状態に値を持ち、初期状態確率が最も高い状態○2での自己ループは見られない。

(b) 名詞について

名詞は主として、○5, ○6, ○7から○1, ○9, ○12への遷移で出力されている。

状態○6○7からの遷移では主として形式名詞が出力され、状態○5からの遷移では、形式名詞以外の名詞が出力される。このように普通名詞と形式名詞の分化がおきている。なお、状態○5からの遷移では、出力確率が特に高い単語はなく、多種類の普通名詞の単語がこの遷移で出力されている。

(c) 活用する品詞について

表 9.15: 品詞・活用形の略号 2

品詞名	略号	品詞名	略号	品詞名	略号	品詞名	略号	品詞名	略号
普通名詞	普	助動詞	助	格助詞	格	準体助詞	準	未然形	未
代名詞	代	間投詞	間	係助詞	係	接頭辞	頭	終始形	終
本動詞	本	感動詞	感	接続助詞	接助	接尾辞	尾	連体形	体
補助動詞	補	副詞	副	終助詞	終	連用形	用		

表 9.16: 16 状態 Ergodic HMM の初期状態確率

π_0	0.00	π_4	0.00	π_8	0.00	π_{12}	0.00
π_1	0.00	π_5	0.00	π_9	0.00	π_{13}	0.00
π_2	0.89	π_6	0.00	π_{10}	0.00	π_{14}	0.00
π_3	0.11	π_7	0.00	π_{11}	0.00	π_{15}	0.00

8 状態の Ergodic HMM では活用形でグループ化されていた。これに対し 16 状態では品詞の基本形でもグループ化されている。本動詞は状態○8,○5からの遷移で、補助動詞は状態○0,○4からの遷移で、助動詞は状態○0,○1,○13からの遷移でそれぞれ出力されている。

本動詞は複数の遷移において出力されているが、主に○8, ○15からの遷移で出力される。しかし、各遷移によって出力される単語に違いが見られる。また、本動詞の中でも連用形が他の活用形に比べ非常に高い。補助動詞は、○0, ○4からの遷移で出力される。本動詞と同様に連用形の出力が多く、助動詞をともなうことが多い。また。状態○0から出力される単語は表記が異なるものの意味的には「する」と同じものが多い。助動詞は、遷移の起点となる状態によって全く異なるを出力しており、8 状態と同様に「です」「ます」がグループ化され、さらに 16 状態では「た」「たい」が他の単語からグループ化して出力されている。

(d) 格助詞について

格助詞は、主に名詞を出力した遷移の集まる状態○8,○9,○12からの遷移で出力される。例を次に示す。8 状態では同じ遷移で出力されていた「で」と「に」、「が」と「を」が別々の遷移に分かれて出力されている。

i. 状態○0⇒ 状態○8 「で」 (86%)

ii. 状態○1⇒ 状態○8 「に」 (87%)

iii. 状態○8⇒ 状態○15 「と」 (63%) 「お」 (28%)

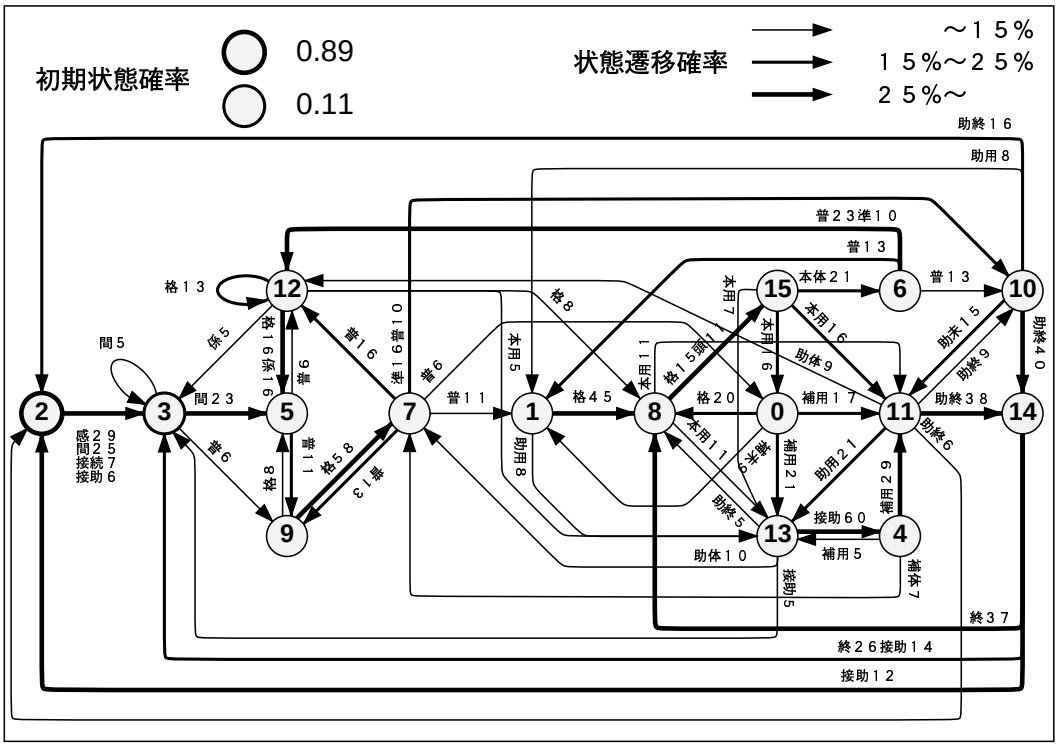


図 9.10: 16 状態 Ergodic HMM の解析結果

iv. 状態○9⇒ 状態○5 「から」 (57%) 「で」 (23%)

v. 状態○9⇒ 状態○7 「の」 (93%)

vi. 状態○12⇒ 状態○3 「は」(46%) 「ので」 (35%)

vii. 状態○12⇒ 状態○5 「が」 (55%) 「は」 (24%)

viii. 状態○12⇒ 状態○8 「を」 (75%)

ix. 状態○12⇒ 状態○12 「を」 (74%) 「について」 (24%)

なお、付録 C において、16 状態 Ergodic HMM から抽出される特徴を品詞ごとに詳細に報告した。

9.2.4.2 文の平均尤度およびモデルのエントロピー

1. １文あたりの平均尤度

   本節では学習終了後の Ergodic HMM に文（単語列）を入力し、HMM が入力された文を出力する確率（尤度）を計算した。HMM の学習に用いた Baum-Welch アルゴリズムは、尤度（入力データの生成確率）を最大にするようにパラメータを調整する。そこで、学習していないデー

タ (text-open data) の尤度と学習データ (text-closed data) の尤度と比較することにより、Ergodic HMM が獲得した文法の一般性を調べた。

2. 尤度の計算方法

言語モデル生成実験で得られた Ergodic HMM が text-closed data および text-open data を生成する一文あたりの平均対数確率を、forward probability （2.1.4 節参照）を用いて計算した。なお、単語の出力確率が 0.0 の場合は 10^{-5} でフロアリングした。

計算方法を簡単な例で図 9.11 に示す。

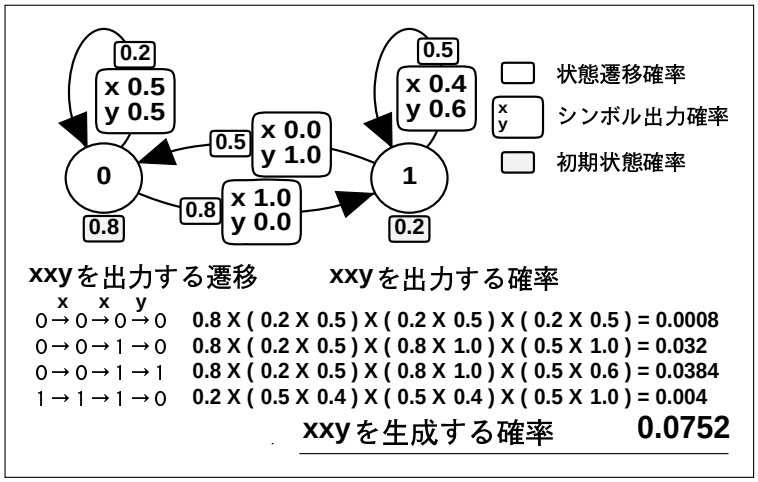


図 9.11: 文の尤度の計算方法

例えば、出力シンボルが x と y の 2 状態 Ergodic HMM の学習終了後の各パラメータが図 9.11 のようになってるとする。この HMM が “xxy” を生成する確率について以下に述べる。

- HMM が “xxy” を出力する全ての経路を列挙する (例では４経路)。
- 各経路を通る確率を求める。
- 全ての経路の確率の和を求める (この和が “xxy” を生成する確率)。

以上の手順で求められた尤度の対数値の和を求め、一文当たりの平均を求めたものを平均尤度とした。

3. エントロピー

本節では、言語モデルの評価基準としてエントロピーを用いた（2.1.10 節参照)。エントロピーはモデルの複雑さを表す指標である。あるモデル λ のエントロピーが $H(\lambda)$ ならば次のシンボルを決定するのに、平均

$H(\lambda)$ 回の yes/no の質問を繰り返す必要がある。いい換えれば、$2^{H(\lambda)}$ 個の等出現確率のシンボルの中から一つのシンボルを決定することになる。すなわち、エントロピーが大きいほど、モデルは複雑であるといえる。

4. 計算結果

odd4000 を学習させた各状態数の Ergodic HMM について text-open data、text-closed data それぞれ 4000 文の尤度を求め、一文当たりの平均と HMM のエントロピーを計算した結果を表 9.17 , 図 9.12 , 図 9.13 に示す。

表 9.17: 平均尤度・エントロピー

Ergodic HMM の状態数	エントロピー	平均尤度	
		text-closed data	text-open data
2 状態	7.53	-76.48	-77.37
4 状態	6.70	-69.34	-71.30
8 状態	5.99	-62.93	-67.48
16 状態	5.29	-56.81	-64.50

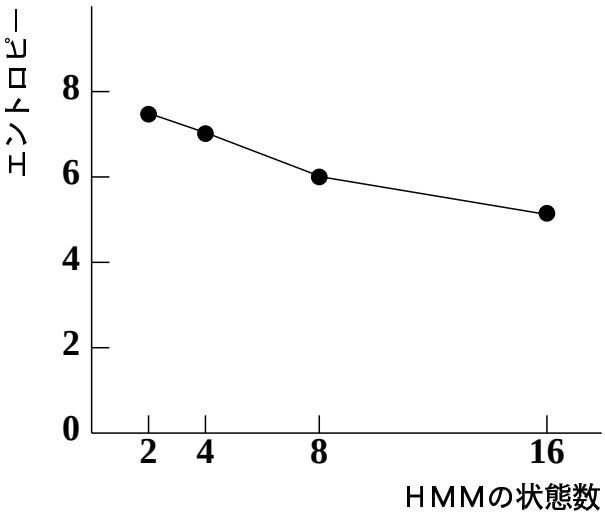


図 9.12: モデルのエントロピー

表 9.17 , 図 9.12 , 図 9.13 から、HMM の状態数が多くなるにしたがいエントロピーが減少しているのがわかる。

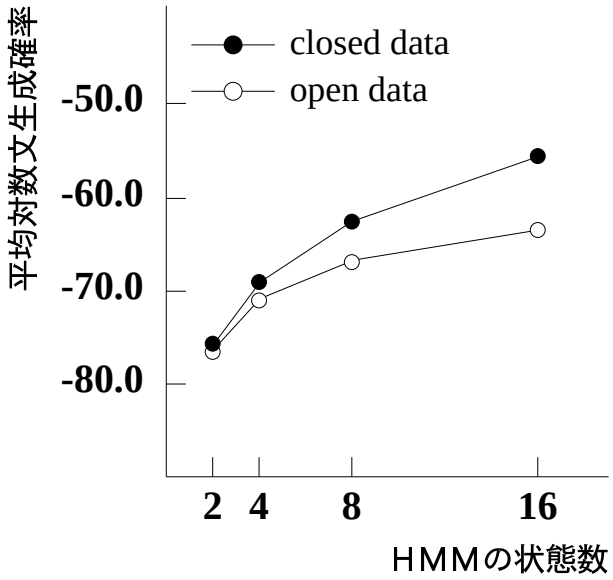


図 9.13: 平均尤度

一方、Ergodic HMM を解析した結果、状態数が増えることによって、シンボル出力確率の分布の偏りが大きくなることが観測された。つまり、状態数が増えることによって、一つの遷移で出力される単語の分布の偏りが大きくなり、そのためエントロピーが下がると思われる。

また、状態数が増すにつれて、text-closed data と text-open data の平均尤度の差が開くことがわかる。この原因として、text-open data に text-closed data に存在しない単語（未知語）が多数含まれていることが考えられる。実際の調査でも even4000 には未知語を含む文が 990 文あった。

5. 学習データ量とモデル化の関係

ここでは、学習データ量を変化させたときの、平均尤度およびエントロピーの変化を研究した。2 状態、4 状態、8 状態の Ergodic HMM に odd1000、odd2000、odd4000 を学習させた結果について、text-open data の平均尤度を計算した。なお、テストデータとして even4000 を用いて平均尤度を求めた結果を表 9.18 、図 9.14 に示す。また、合わせて各学習データ量におけるモデルのエントロピーを表 9.18 に示す。

表 9.18 、図 9.14 から、全ての HMM で学習データを増加すると、text-open data の平均尤度が高くなることが示された。また、データ数の増加にともないエントロピーが増加することも示された。

表 9.18: 学習データ量と平均尤度の関係

状態数	学習データ	テストデータ	平均尤度	エントロピー
2	odd1000	even4000	-82.60	7.20
	odd2000	even4000	-80.43	7.22
	odd4000	even4000	-77.37	7.53
4	odd1000	even4000	-77.33	6.35
	odd2000	even4000	-75.22	6.34
	odd4000	even4000	-71.30	6.72
8	odd1000	even4000	-76.31	5.58
	odd2000	even4000	-73.35	5.59
	odd4000	even4000	-67.48	6.00

9.2.5 連続音声認識への適用

ここでは、Ergodic HMM を用いて学習した確率つきネットワーク文法を連続音声認識の言語モデルとして利用した場合の有効性について研究した。その概要を次に報告する。

9.2.5.1 実験条件

連続音声認識の基本アルゴリズムとして one-pass DP (2.2.2 節参照) を用い、音響尤度の計算には音素モデル HMM を用いた。これに言語モデルとして odd4000 で学習した状態数 2,4,8 の Ergodic HMM を使用した。その他の実験条件を表 9.19 に示す。テストデータとして、学習データ odd4000 とタスクが同じ 38 文（学習データに含まれない文）を用いた。テストデータを図 9.15 に示す。

同一のテストデータを使って text-open data,text-closed data 両方の文認識を行なうために、odd4000 で学習したパラメータを初期値として、odd4000 にテストデータ 38 文を加えた 4038 文を学習させた Ergodic HMM を新規に作成し、text-closed data における文認識のための言語モデルとして使用した。4038 文で学習させた時の学習条件は、パラメータの初期値以外は odd4000 で学習させた時と同じである。

単語間の接続部分で、対数音響尤度の値に Ergodic HMM から Viterbi アルゴリズムで計算された単語間接続確率の対数値を加えた。この際、Ergodic HMM から得られる値に

$$\text{対数音響尤度} : \text{対数単語間接続確率} = 1 : 16$$

(言語尤度と音響尤度の結合値 $\alpha = 16$) の比率で重みをつけた。また、認識時の計算量を削減するためビーム幅 4096 でビームサーチを行なった。

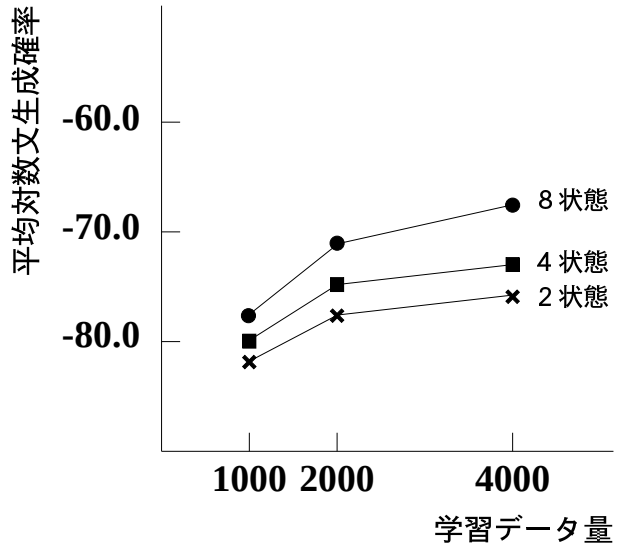


図 9.14: 学習データ量と平均尤度との関係

9.2.5.2 実験結果

表 9.20 および図 9.16 に実験結果を示す。比較のため、言語モデルを用いない場合 (音響モデルのみで認識した場合) と言語モデルとして単語 bigram (言語尤度と音響尤度の結合値 $\alpha = 1$) を用いた場合を合わせて示す。表中に示した () 内の値は (正解を出力した文数/認識に用いた文数) である。

text-open の実験において Ergodic HMM は単語 bigram を超えた認識性能が得られたことから、Ergodic HMM は単語 bigram よりもロバスト性があることがわかる。また、状態数が増えるにつれ、text-closed データに対する認識率が上がることがわかる。

9.2.5.3 初期パラメータの違いによるモデルの変化

Ergodic HMM のパラメータの初期値の違いによる生成されるモデルの変化を研究するために、初期値の異なる 8 状態 Ergodic HMM を 8 個用意し、odd4000 を学習させた。初期値が異なること以外はすべて同一条件で学習を行ない、パラメータの再推定回数 20 回を学習の終了条件とした。初期パラメータは 9.2.3 節で示した方法で決定した。

表 9.21 に学習終了後の各 Ergodic HMM のエントロピー、学習データ odd4000 の平均尤度、音声認識に用いた場合の text-closed data に対する認識率などを示す。表中、一番左は各 Ergodic HMM を区別するために番号をつけた。なお表中の 1 は、 3 節で得られた、学習終了条件を尤度の上昇率 1%未満にした場合の結果である。

表 9.19: 連続音声認識実験の条件

音素モデル数	52 音素
音素音響モデル	4 状態 3 ループ対角混合分布型 HMM 混合数は音素ごとに異なる。 継続時間長制御なし。
話者	男性アナウンサー 1 名 (MAU)
音響パラメータ	log パワー + 16 次 LPC ケプストラム +Δlog パワー + 16 次 LPCΔ ケプストラム
音響分析条件	サンプリング周期 12kHz フレーム窓長 20ms フレーム周期 5ms
ビーム幅	4096
認識語彙数	435 単語
テストデータ	同一話者発声 (MAU) 38 文

表 9.21 から、パラメータの初期値によって、生成されたモデルのエントロピーや平均尤度が異なることがわかる。これを音声認識の言語モデルに用いた場合、学習回数の異なる 1 を除いた 8 種の Ergodic HMM 間で、認識率の最高値 (4 の Ergodic HMM) と最低値 (9 の Ergodic HMM) の差は約 13%(5 文) と、初期モデルの違いでかなりの違いが見られる。

なお、学習後の平均尤度と認識率の関係を図 9.17 に示す。

学習させた回数が 20 回で十分でなかったことやデータが 8 種類しかないことなどから明確ではないが、図 9.17 から、エントロピーと Ergodic HMM には相関があることがわかる。

表 9.20: 認識実験の結果

言語モデル	文認識率	
なし	29.0% (11/38)	
	text-open data	text-closed data
2 状態 Ergodic HMM	31.5% (12/38)	34.2% (13/38)
4 状態 Ergodic HMM	36.8% (14/38)	39.5% (15/38)
8 状態 Ergodic HMM	39.5% (15/38)	47.3% (18/38)
16 状態 Ergodic HMM	36.8% (14/38)	——-
単語 bigram	34.3% (13/38)	52.7% (20/38)

表 9.21: 初期値の違いによる生成モデルの変化

	学習回数	エントロピー		平均尤度		認識率
		初期状態	学習後	初期状態	学習後	
1	749	12.61	5.99	-126.39	-62.93	47.3%
2	20		6.21	-126.53	-64.48	42.1%
3	20		6.14	-126.36	-64.36	39.5%
4	20		6.03	-126.43	-63.68	44.7%
5	20		6.27	-126.10	-64.71	34.2%
6	20		6.09	-126.41	-64.95	36.8%
7	20		6.10	-126.24	-64.06	36.8%
8	20		6.16	-126.47	-64.62	39.5%
9	20		6.13	-126.32	-65.07	31.6%

9.2.6 考察

1. HMM による文法の自動獲得の可能性

   学習後の Ergodic HMM の解析結果から Ergodic HMM が文法的な特徴を獲得していることや、text-open data と text-closed data の尤度に大差がないことから、Ergodic HMM を用いて、言語データを利用して Baum-Welch アルゴリズムでパラメータを学習することにより、一般性のあるネットワーク文法が自動的に獲得できる可能性があると考えられる。また、体言、活用する品詞、格助詞などの分離に見られることから、Ergodic HMM は、従来の品詞より詳細な単語のカテゴリーを獲得する能力もあると思われる。

   また、Ergodic HMM の状態数を増加させることにより、カテゴリーごとに、精密に分類されたのをはじめ、他の品詞でも状態数を増やすことで、より精密に分離して出力された。モデルのエントロピーも状態数を増やすことで改善されることも実験結果から明らかになった。したがって、今後さらに状態数を多くすることにより、エントロピーが小さくなり、モデルの表現能力も高くなることが予想され、より詳細な文法・単語のカテゴリーを獲得できると思われる。

2. 認識実験の結果

   学習された Ergodic-HMM を言語モデルとして、連続音声認識実験を行った結果、text-open data、text-closed data ともに、言語モデルを用いない場合より高い認識率が得られた。これは、Ergodic HMM によって得られた文法は音声認識のための言語モデルとして有効であることを示している。

しかし、認識率は HMM の初期モデルによって異なる値を示す。したがって初期状態を変えたモデルについて認識実験を行なう必要がある（9.2.5.3 節参照）。また、認識時に言語モデルから計算される接続確率につける重みの最適値の決め方も問題が残っている。

3. 学習データ量

   本研究では、Ergodic HMM を用いて、大量のテキストデータを使用し Baum-Welch アルゴリズムでパラメータを学習をすることで、文法および単語カテゴリーを獲得し表現する能力があることが示された。しかし、本研究で使用した学習データ量はまだ不十分であると思われる。

   8 状態の Ergodic HMM に odd4000 を学習させた場合を例にとると、8 状態 Ergodic HMM が持つパラメータ数は、初期状態確率が 8 個、状態遷移確率が 64 個 (＝ 8 状態× 8 状態)、シンボル出力確率が 410752 個 (＝ 8 状態× 8 状態× 6418 単語) で合計 410752 個である。これに対し、odd4000 の総単語数は 57354 個であった。

   HMM のパラメータの推定に必要な学習データ量の明確な基準はないが、推定すべきパラメータの数に対し、推定に用いられる学習データが十分でないと考えられる。したがって、パラメータを精度良く推定するには、さらに学習データ量を増やす必要があると思われる。

4. 初期モデル

   本研究で行なった実験では、状態数の異なる 4 種の状態遷移出力型の Ergodic HMM を用いたが、状態数が多いほどエントロピーが改善され、また Ergodic HMM の構造解析の結果からも、より良いモデルを得ていていることがわかった。しかし、最適な状態数の推定方法が未解決の問題として残る。

   また、状態数が同じ場合でも、Ergodic HMM のパラメータの初期値を変えた学習の実験結果から、初期値によって異なるモデルを生成することがわかった。これは Baum-Welch アルゴリズムがローカルミニマムに収束するのが原因である。また、パラメータ数、学習データ量が多くなるとパラメータ推定に膨大な計算量がかかり、初期モデルによって収束するまでにかかる学習回数が異なることが予想される。

   効率の良い学習法、最適な初期値の決定法は今のところ知られていないが、考えられる有効な一つの方法として、いくつかの初期モデルに適当に学習を繰り返した後に、文の尤度やエントロピーを計算し、結果の良好なものについて学習を続行する方法などが考えられる。

5. 学習データの品詞に関する問題点

   本研究で用いた言語データは単語の表記が同一なものでも品詞や活用形の異なる場合、異なるラベルを付与した（9.2.2 節参照）。これによ

り単語に品詞情報が加わると考えられる。したがって “品詞情報のない単語列からの文法および単語カテゴリーの自動獲得” の可能性を正確に検証するためには、同一表記の単語には同じラベルをつけたデータを HMM に学習させる実験を行なう必要があると思われる。

6. 学習データの分布

本研究では、 9.2.2 節で説明したように、言語データベースを奇数偶数の 2 set に分け、それぞれ先頭から 1000 文, 2000 文, 4000 文を data set とした。表 9.9 から、even と odd では構成単語数に大差はない。しかし、データの前半部分と後半部分で対話文の長さに違いがあることがわかる。表 9.22 に odd4000 の前半と後半の 2000 文、odd1000, odd4000 のデータを示す。(前半 2000 文は odd2000 と同じ。)

表 9.22: 構成単語数 2

set	odd1000	odd2000	odd4000 の後半 2000 文	odd4000
文数	1000	2000	2000	4000
単語数	13299	20730	36624	57354
文平均単語数	13.30	10.37	18.31	14.34
最大単語数	99	99	128	128

表 9.22 から、前半後半で構成単語数がかなり異なり、後半部分の一文あたりの単語長が前半部分の約 2 倍になっている。

odd1000,odd2000 は文平均単語数が odd4000 に比べ少ない。表 A.7 を見ると、odd2000 では「はい」「もしもし」などの一単語のみの文の割合が 19.95%で odd1000(12.70%), odd4000 (15.25%) に比べ多く含まれ、odd2000 で (odd1000 に) 新たに加わった文の中に 30 単語以上の文が 5 文で非常に少ない。odd2000 では 30 単語を越える長い文の比率が odd1000,odd4000 に比べ小さくなっている。また、odd1000 では一単語の文はそれほどないが、3 単語、4 単語の短い文の比率が高く、逆に odd4000 では長い文が多く含まれている。

以下に 8 状態の Ergodic HMM に odd1000, odd2000 を学習させた場合の解析結果を図 9.18 、図 9.19 に示す。(odd4000 を学習させたものは図 9.9 参照。)

これらの図から、一単語の文が多く含まれる odd2000 で学習した HMM は、状態○1での自己ループでの感動詞の出力が他の HMM に比べて高く、odd4000 で学習した HMM では感動詞の出力とともに自己ループの遷移確率も低くなっている。学習データ量が異なるので学習データの

性質の違いによるものかどうかは不明であるが、ネットワークの形態そのものもかなり異なっている。

1000 文、2000 文の set を作成する際に、odd、even それぞれの 4000 文の全体から 4 の倍数番目、2 の倍数番目のように全体から均一に抽出すれば、このような data set による文の性質の違いを緩和できたと思われる。

9.2.7 まとめ

本節では、Ergodic HMM と確率つきネットワーク文法が類似した構造を持ち、同種のパラメータで表現されること、さらに大量のテキストデータを学習データとし、Baum-Welch アルゴリズム （2.1.6 節参照）で HMM のパラメータを学習することで、Ergodic HMM による言語のモデル化の検討を行なった。

9.2.1 節で、Ergodic HMM に品詞情報を持たない単語列を学習させ、学習後の HMM を解析した結果、Ergodic HMM の構造は学習データの特徴をとらえた文法的な特徴を示しており、単語を文中での機能によって分類して出力していることがわかった。また、Ergodic HMM の状態数が増えるほど詳細な表現が可能となり、より精密な単語の分類を行なっていることがわかった。

9.2.4.2 節で平均尤度を研究した結果、text-open data も text-closed data と大差なく生成されることから、Ergodic HMM が学習によって一般性のある文法を生成していることがわかった。また、状態数が増えることにより、言語モデルの複雑さの指標となるエントロピーが改善されることがわかった。

以上のことから、Ergodic HMM を用いてテキストデータから一般性のある確率つきネットワーク文法を自動生成し、単語を分類できる可能性が示された。また、一般性のあるよりよいモデルを得るためには、Ergodic HMM の状態数を増やし大量の言語データで学習する必要があることが示された。

9.2.5 節で、得られた Ergodic HMM を言語モデルとして用いて、連続音声認識実験を行なった結果、text-open data、text-closed data ともに言語モデルのない場合に比べ認識率が向上することが得られ、HMM が獲得した文法は連続音声認識に有効であることが示された。

9.2.5.3 節では、HMM の学習に問題になる初期パラメータの変化によるモデル化の違いについて研究した。その結果、生成されるモデルは初期モデルによって異なり、それらのモデルを使った文認識実験では、認識率の差は最高値と最低値で約 13%の違いが見られることが示された。この結果、初期パラメータの設定法が重要であることが明らかになった。また文認識率とモデルの尤度には相関が見られ、尤度の高いモデルほど文認識率は向上する傾向が見られた。

9.3 メモリ量および計算量を削減したBaum-Welchアルゴリズムの提案と言語モデルへの適用

全状態間の遷移が許されている (Ergodic) 離散型 HMM において単語を出力シンボルとした場合、その構造はネットワーク文法記述と形式的に類似する。したがって大量の単語列データから、Baum-Welch アルゴリズムを使用して、確率つきネットワーク文法を自動的に獲得できる可能性がある [28]。しかし状態数を大きくするとメモリ量および計算量は増加するため、現実的に計算が不可能になる。そのため従来の研究では状態数が少なく、認識性能や perplexity は単語の bigram と比較して良くない [57] [94] [37]。そこで本節では状態数が多い Ergodic HMM を学習するために、メモリ量および計算量を削減した Baum-Welch アルゴリズムを提案した。さらに、得られた Ergodic HMM を言語モデルとして連続音声認識に用いた実験結果についても述べた。

9.3.1 メモリ量および計算量を削減した Baum-Welch アルゴリズム

遷移出力型で状態数 N の Ergodic HMM のパラメータ $\lambda = (\Pi, A, B)$ を次のように定義した。

$\Pi = \pi_N(i); i = 1, .., N$	初期状態確率
$A = a_N(i,j); i = 1, .., N, j = 1, .., N$	状態遷移確率
$B = b_N(i,j,k); i = 1, .., N, j = 1, .., N, k = 1, .., V$	シンボル出力確率

ただし V は語彙数。

本稿で提案するアルゴリズムは次の 2 つで構成される。

○1 小さいシンボル出力確率の削除

Baum-Welch アルゴリズムを使用してパラメータを推定するとき、シンボル出力確率 $b_N(i,j,k)$ が閾値より小さいとき 0 にして、再推定およびメモリから削除する。

一般的には forward probability $\alpha_t(j)$ は以下の式で計算される。

$$\alpha_t(j) = [\sum_i \alpha_{t-1}(i) a_N(i,j)] b_N(i,j,O_t) \tag{9.6}$$

この式の代わりに、本節では以下の式を使用する。

$$\alpha_t(j) = \begin{cases} \sum_i [\alpha_{t-1}(i) a_N(i,j)] b_N(i,j,O_t) & if \quad b_N(i,j,O_t) > C \\ 0.0 & if \quad b_N(i,j,O_t) \leq C \end{cases} \tag{9.7}$$

これによりメモリ量および計算量が削減できる。実験では閾値 C を 10.0^{-300} にした。なお類似したアルゴリズムが文献 [37] において提案されている。

◯2状態数の逐次増加

状態数が大きな Ergodic HMM のパラメータを再学習する場合、大量のメモリが必要になる。そこで状態数を逐次的に増加させる。N 状態の Ergodic HMM のパラメータが既に推定されたとして、$2N$ 状態の Ergodic HMM の初期状態確率および状態遷移確率の初期パラメータを次のように計算する。

$$\pi_{2N}(i) = 0.5 \times \pi_N(i/2) \qquad i = 1, .., 2N \tag{9.8}$$

$$a_{2N}(i,j) = 0.5 \times a_N(i/2, j/2) \qquad i = 1, .., 2N, j = 1, .., 2N \tag{9.9}$$

ここで / は小数点以下切り上げを意味。

シンボル出力確率の初期パラメータは乱数を利用して次のように計算する。

$$b_{2N}(i,j,k) = b_N(i/2, j/2, k) \times random(i,j,k) \tag{9.10}$$
$$i = 1, .., 2N, j = 1, .., 2N, k = 1, .., V$$

ただし $\sum_k b_{2N}(i,j,k) = 1.0$ となるように正規化する。

状態数が大きな Ergodic HMM のパラメータは以下のフローを繰り返すことで学習できる。

1. 1 状態の Ergodic HMM を作成する。初期状態確率および状態遷移確率は 1.0、シンボル出力確率は乱数とする。
2. 学習アルゴリズム◯1を用いて HMM のパラメータの再推定をする。
3. 学習アルゴリズム◯2を利用して、再推定された N 状態の HMM のパラメータから $2N$ 状態の HMM の初期パラメータを計算する。
4. 学習アルゴリズム◯1を用いて HMM のパラメータの再推定をする。

9.3.2 Ergodic HMM を用いた確率つきネットワーク文法の獲得

ここで提案した Baum-Welch アルゴリズムを用いて、大量の単語列データから確率つきネットワーク文法を獲得する実験を行った。

9.3.2.1 実験条件

学習データとして、ATR 対話データベース (ADD)[10] の中の “国際会議に関する問い合わせの電話での対話” を用いた。実験の条件を表 9.23 に示した。

表 9.23: 言語モデル生成実験の条件

HMM の構造	状態遷移出力型
学習語彙数	6420 単語
学習データ数	8475 文
総単語数	5,7354
HMM パラメータの再学習における Baum-Welch アルゴリズムの終了条件	40 回の繰り返し

9.3.2.2 状態数と値を持つシンボル出力の数

状態数の増加にともなう、値を持つ（確率値が 0.0 ではない）シンボル出力確率のパラメータ数の変化を基本的な Baum-Welch アルゴリズムを使用した場合と、本節で提案した学習アルゴリズムを使用した場合の両者を図 9.20 に示した。

横軸は Ergodic HMM の状態数で縦軸は、値を持つシンボル出力確率のパラメータ数である。

この図から、本手法で学習した場合のパラメータ数は、通常の学習のパラメータ数より大幅に減少することがわかる。状態数 512 の Ergodic-HMM において、シンボル出力確率に値があるものは 2106967 個であった。したがって本稿で提案したアルゴリズムは基本的な Baum-Welch アルゴリズムと比較すると、メモリ量および計算量を 0.125% に削減していることがわかる。

ただし、同一の言語データにおける単語の bigram のパラメータの数は 37,752、単語の trigram では 78,138 であった。この値と比較すると、Ergodic-HMM のパラメータ数はかなり多いといえる。

9.3.2.3 エントロピー

Ergodic HMM の状態数の増加にともなうエントロピーの変化を、本手法で提案したアルゴリズムを用いた場合と通常の Baum-Welch アルゴリズムを用いた場合の両者を図 9.21 に示した。比較のために単語 bigram および単語 trigram のエントロピーも載せた。

通常の Baum-Welch アルゴリズムを用いた実験では、コンピュータのメモリーの制限から状態数 32 までしか計算ができなかった。一方、本手法で提案したアルゴリズムは、状態数 512 でも学習可能であった。また、状態数 32 まで、通常の Baum-Welch アルゴリズムを用いて学習したときのエントロピーと本手法において学習したときのエントロピーがほぼ同一であることから、本手法において学習したパラメータと通常の Baum-Welch アルゴリズムを用いて学習したパラメータには、大きな差がないことが示された。

また、状態数 128 以上で Ergodic HMM のエントロピーが単語 bigram より低くなることがわかる。

9.3.3 連続音声認識実験

学習された Ergodic HMM を言語モデルとして、連続音声認識実験を行なった。この結果を次に述べる。

9.3.3.1 実験条件

連続音声認識実験には、音素モデルに連続分布型 HMM、サーチアルゴリズムにビームサーチ、言語モデルに Ergodic HMM を使用した。テストデータには Ergodic HMM の学習に使用したテキストデータと同一タスクの会話 38 文（学習データに含まれない。図 9.15 参照）を用いた。評価は文認識率で行なった。また、学習データにテストデータのテキストを加えた text-closed data の実験も行なった。これらは 9.2.5 節の実験と同一条件である。

その他の実験条件を表 9.24 に示した。

表 9.24: 連続音声認識実験の条件

音素音響モデル	4 状態 3 ループ対角混合分布型 HMM
音響パラメータ	log power + 16 次 LPCcepstrum
	+ Δlog power + 16 次 Δcepstrum
学習データ	男性アナウンサー 1 名、2620 単語発声
認識語彙数	435 単語
ビーム幅	4096
テストデータ	同一話者発声 38 文
発話様式	朗読

9.3.3.2 実験結果

図 9.22 に text-closed data の実験結果を示した。また比較のために 単語 bigram および trigram の実験結果も示した。この図からわかるように、状態数 128 において単語 bigram より高い文認識率が得られた。

図 9.23 に text-open の実験結果を示した。この図には文認識率と共に単語の正解率（4.3.1.3 節参照）も載せた。また比較のために単語 trigram の実験結果も示した。この実験では状態数 128 の Ergodic HMM において文認識率は 44.7% が得られた。一方単語 trigram は 44.7 提案したアルゴリズムの有効性が示された。

9.3.4 考察

1. HMM の初期パラメータ

   Baum-Welch アルゴリズムを用いてパラメータを学習した場合、パラメータはローカルミニマムの方向に学習されるため、初期パラメータが重要になる。本節では状態数を逐次増加するとき、シンボル出力確率に乱数の重みをつけたが、シソーラスなどの辞書を使用して、同一のカテゴリに属する単語に同一の重みをつけることにより、より良い確率つきネットワーク文法が得られる可能性がある。

2. 2nd order HMM

   ここで提案したアルゴリズムを容易に 2nd order HMM に拡張できる。2nd order HMM にすることにより、より高い性能が得られることが期待される。

3. 他の分野での応用

   本節では、ここで提案したメモリ量および計算量を削減した Baum-Welch アルゴリズムを、確率つきネットワーク文法の獲得に利用したが、Ergodic HMM は形態素解析などの他の多くの分野において応用が考えられる。したがって提案したアルゴリズムが、これらの分野においても利用できると考えている。

9.3.5 まとめ

本節ではメモリ量および計算量を削減した Baum-Welch アルゴリズムを提案した。そして確率つきネットワーク文法の獲得に、このアルゴリズムを利用した。実験の結果、通常の Baum-Welch アルゴリズムを用いて学習するときよりパラメータ数が大幅に減少すること、それに伴いメモリー量や計算量が大幅に減少することが示された。またエントロピーの結果から、ここで提

案した学習アルゴリズムを用いて学習したパラメータと通常の Baum-Welch アルゴリズムを用いて学習したパラメータには大きな差がないことが示された。さらに、得られた Ergodic HMM を言語モデルとして連続音声認識に用いた。この実験の結果、状態数が大きい場合、単語 bigram よりも高い性能が得られ、提案したアルゴリズムの有効性が示された。今後の課題として、特に初期パラメータが挙げられる。Baum-Welch アルゴリズムでパラメータを学習した場合、パラメータは local maxnum の方向に学習される。そのため、初期パラメータによって、学習結果が大きく異なる可能性がある。

9.4 まとめ

この章では、Ergodic HMM と確率つきネットワーク文法が類似した構造を持ち、同種のパラメータで表現されること、さらにデータを入力すると HMM のパラメータは 2.1.6 節で紹介した Baum-Welch アルゴリズムで学習できることに着目し、Ergodic HMM による言語のモデル化の検討を行なった。

9.1 節では、品詞を入力として、HMM による日本語対話文の文節内における形態素の品詞連鎖のモデル化を行なった。この結果、経験的に得られている生成文法に似た形態の確率つきネットワーク文法を自動的に獲得することができた。

9.2 節では、実際の会話から作成した単語列を Ergodic HMM に学習させて、確率つきネットワーク文法を自動的に抽出することを試みた。その結果、Ergodic HMM の構造は学習データの特徴をとらえた文法的な特徴を示しており、単語を文中での機能によって分類して出力していることがわかった。また、Ergodic HMM の状態数が増えるほど詳細な表現が可能となり、より精密な単語の分類を行なっていることがわかった。

9.3 節では、メモリ量および計算量を削減した Baum-Welch アルゴリズムを提案した。そして確率つきネットワーク文法の獲得に、このアルゴリズムを利用した。さらに、得られた Ergodic HMM を言語モデルとして連続音声認識に用いた。これらの実験の結果、単語 bigram よりも高い性能が得られ、提案したアルゴリズムの有効性が示された。

これらの結果から、Ergodic HMM を考え、大量のテキストデータを用意し、Baum-Welch アルゴリズムで学習することで、言語のモデル化が可能であることが示された。

もしもし。
はい。
会議事務局でございます。
会議の宿泊施設についてお尋ねしたいのですが。
そちらでどこか紹介していただけますか。
はい。
私共でご紹介できるホテルは京都ホテルと京都プリンスホテルです。
一人部屋の値段は一晩７千円から１万円です。
二人部屋の値段は９千５百円から６万円です。
そうですか。
どちらのホテルが会議場に近いですか。
京都プリンスホテルが会議場には近いんですが。
それでは京都プリンスホテルを予約したいのですが。
ホテルの手配もしていただけるのですか。
はい。
京都ホテルと京都プリンスホテルは予約できます。
そうですか。
では京都プリンスホテルの７千円の一人部屋をお願いします。
はい。
京都プリンスホテルの７千円の一人部屋ですね。
はい。
そうです。
いつからお泊まりになりますか。
８月４日の夜からです。
８日の朝までお願いします。
わかりました。
少々お待ちください。
お部屋が取れるかどうか調べます。
お部屋をお取りできます。
ではお名前とご住所をお願いします。
中村一雄です。
住所は東京都港区新橋１丁目１番３号です。
電話番号もお願いします。
電話番号は３３１の２５２１です。
わかりました。
京都プリンスホテルに８月４日から８日まで一人部屋をお取りしました。
どうもありがとうございました。
失礼します。

図 9.15: テスト文

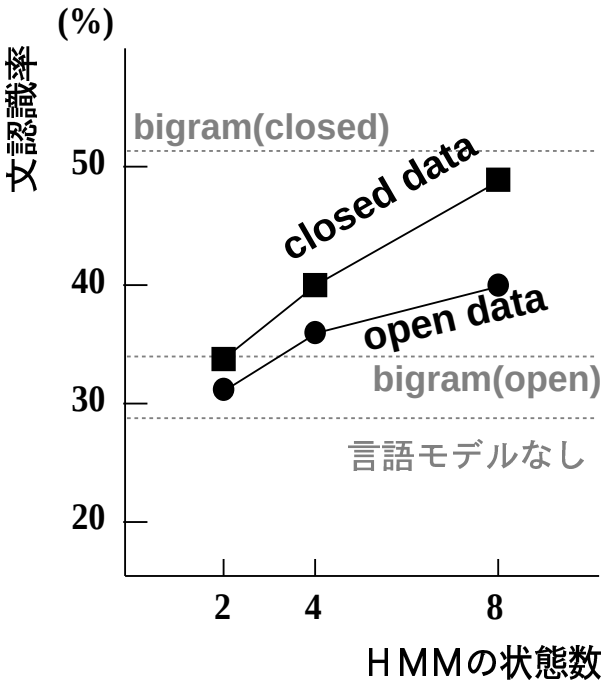


図 9.16: 文認識実験結果

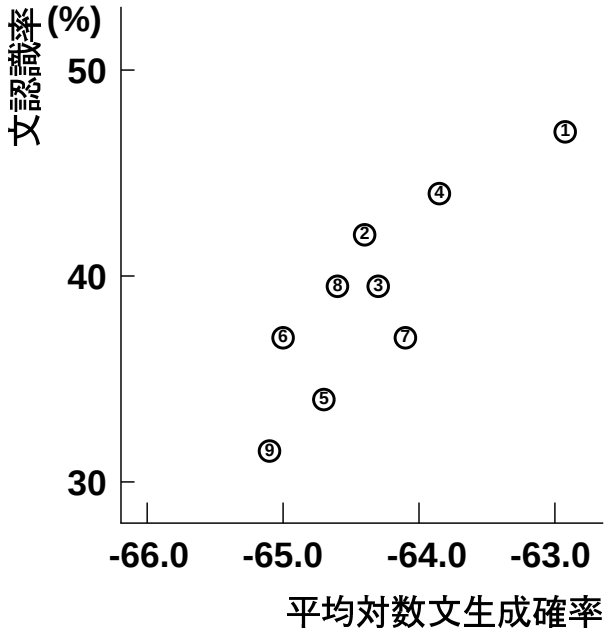


図 9.17: 文認識結果

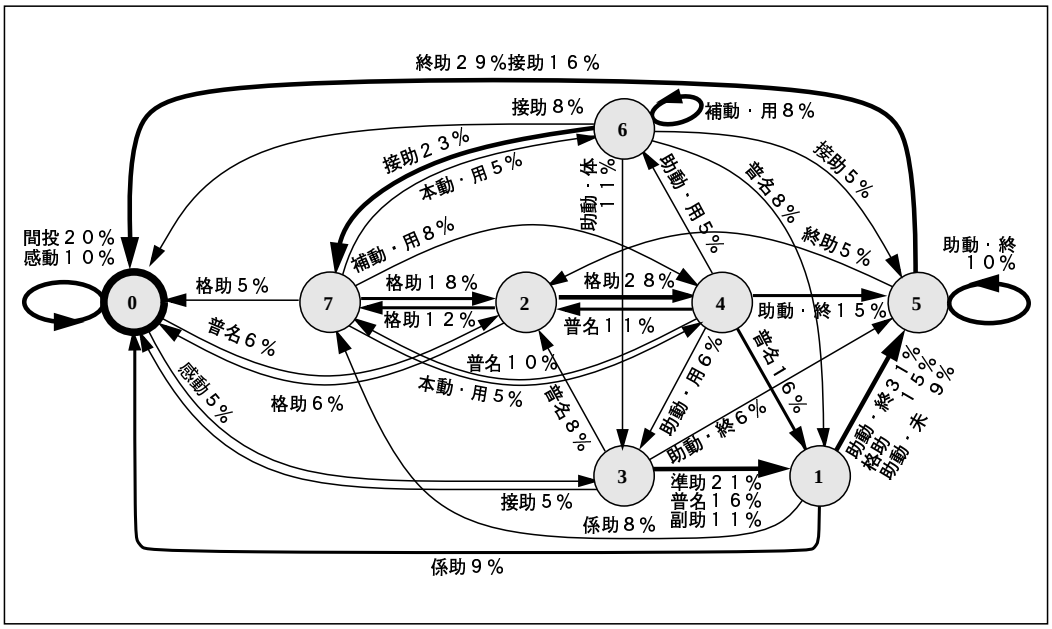


図 9.18: odd1000 を学習させた 8 状態 Ergodic HMM

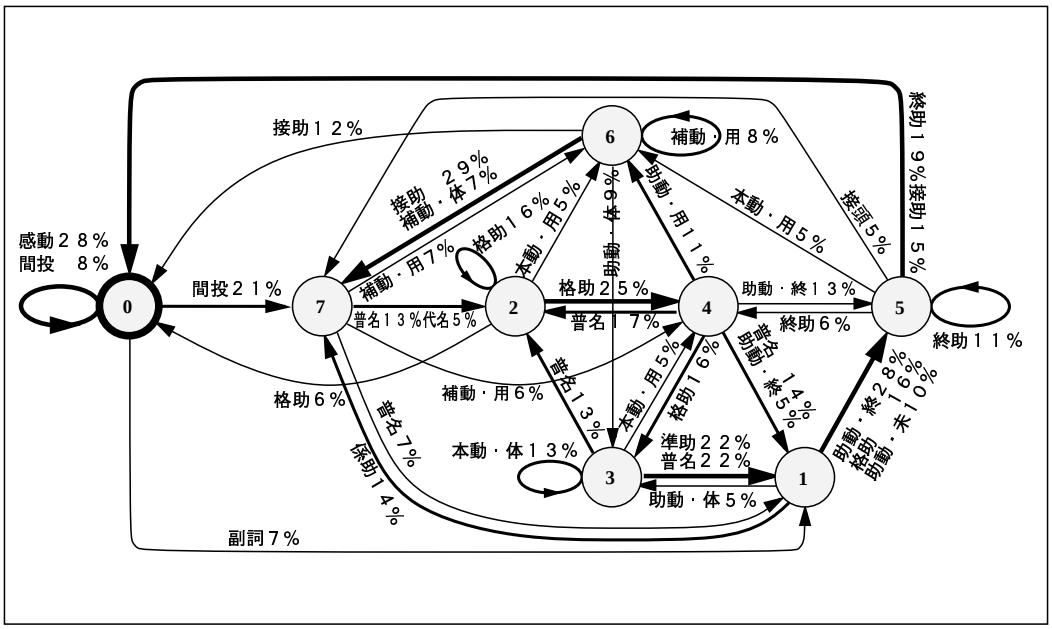


図 9.19: odd2000 を学習させた 8 状態 Ergodic HMM

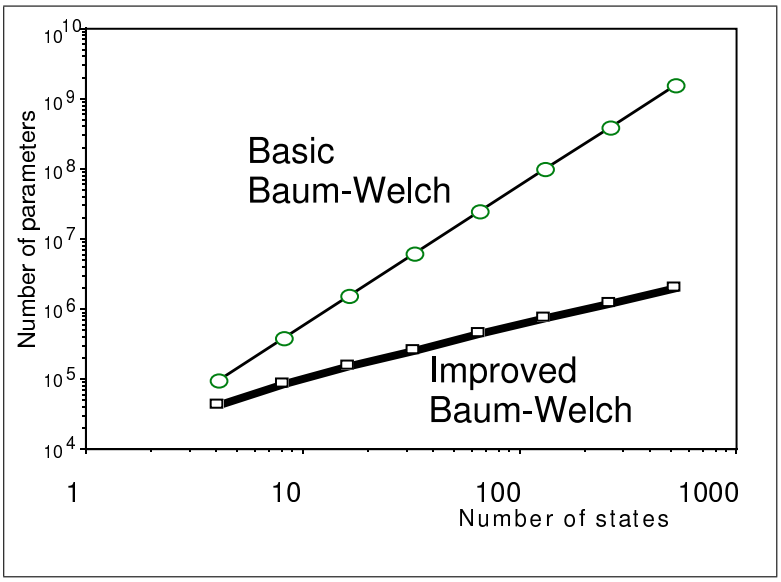


図 9.20: HMM の状態数に対するシンボル出力確率の数の変化

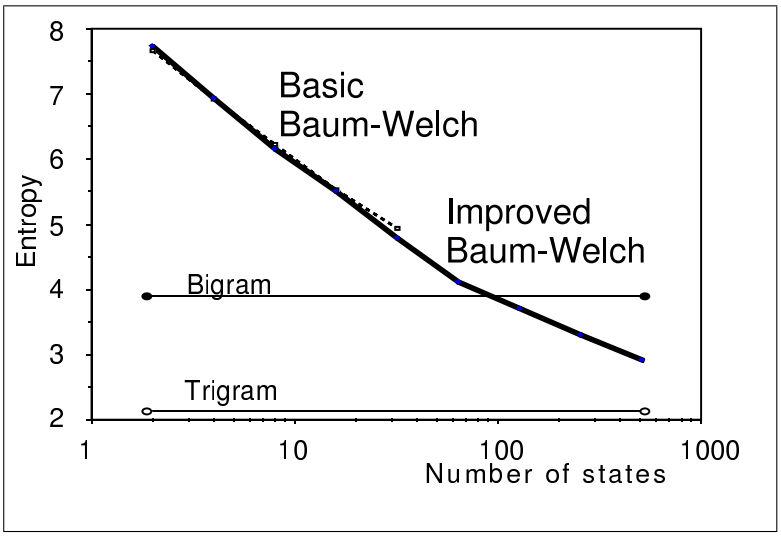


図 9.21: 状態数に対するエントロピーの変化

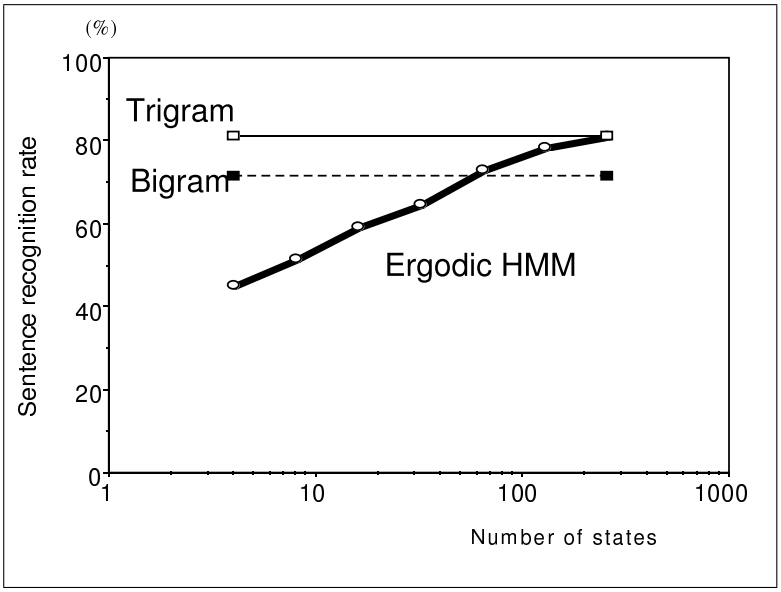


図 9.22: text-closed data における認識率

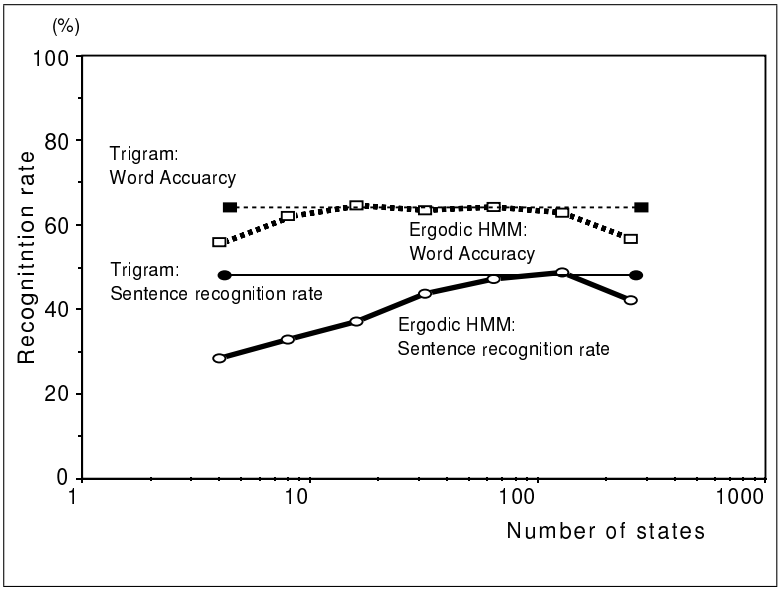


図 9.23: text-open data における認識率

第10章　結論

本論文では N-gram モデルを使用した連続音声認識システムの概要と自由発話認識のための認識アルゴリズムと自由発話のための言語モデルについて述べた。各章の内容は以下の通りである。

第 2 章では、第 4 章以降の研究内容の理解を用意にするために、音声認識システムを実現するために必要な要素技術について述べた。2.1 節では、HMM の学習方法 (Baum-Welch アルゴリズム) や Viterbi サーチについて述べた。2.2 節では、連続音声認識システムのアルゴリズムについて述べた。認識アルゴリズムには多くの種類があるが、ここでは tree-trellis サーチと Viterbi サーチ (one-pass DP) について述べた。また、2.3 節では、音声認識アルゴリズムにおいて計算量およびメモリー量を削減する方法について述べた。

第 3 章では言語をマルコフモデルで表現したときのデータ量と収束性について述べた。調査項目としては、主にエンロトピーとカバー率である。3.1 節では新聞記事について、3.2 節では X 線 CT 所見作成の文章について、3.3 節では ATR の国際会議のデータベースについて述べた。これらの研究の結果、全テキストデータの 98%はマルコフモデルで近似できるが、残り 2%が収束しないことが示された。これは、言語モデルとしてマルコフモデルを選択したときの妥当性に関して、滅多に出現しない言語現象は、あえてモデルに適合させる必要がないと判断すべきであると考えられる。

第 4 章では、日本語における N-gram の有効性について述べた。4.1 節では、かなや漢字や品詞の bigram および trigram の有効性を、新聞記事を入力にシュミレーションで効果を確かめた。そして、この結果、データが大量にあれば text-open data でも、高い認識性能が得られることが示された。4.2 節では、単語の bigram の有効性を実際の音声認識実験を行なって調べた。入力文は医療用 X 線 CT の所見作成である。この実験では、貧弱な音響モデルでも、言語モデルに単語 bigram を使用することにより、高い文節認識率が得られることを示した。4.3 節では、ATR の国際会議の予約のタスクにおいて、連続分布型 HMM と単語 trigram を使用した文認識結果について報告した。これらの実験から、言語の N-gram モデルは有効であることを示した。

第 5 章では、自由発話認識のアルゴリズムとその実験結果について述べた。自由発話では間投詞や、言い淀みや言い誤りおよび言い直しなどが頻繁に出現する。このような発話様式では、認識精度の高い音響モデルの作成は困難であると考えた。そこで認識性能を向上させるため、perplexity の低い言語

モデルと、そのサーチ問題について研究した。5.1 節においてる間投詞や、言い淀みや言い誤りおよび言い直しなどの対応方法について述べた。これらの間投詞や言い直しは文の全ての場所に出現する可能性がある。そこでこれらの単語を、音響モデルでは音素系列として認識しながら、言語モデルではスキップすることで、自由発話の認識が可能になる。5.3 節では、実際に自由発話の認識実験を行い、その結果について述べた。この実験の結果、このアルゴリズムの有効性が確かめられた。

第 6 章では、自由発話の特徴について言語的な面と音響的な面から研究した。6.1 節では、大量の対話データから言語の特徴について述べた。この結果、対話文の 50% は「あのー」、「えーと」などの間投詞を含み、言い直しは約 10%に出現する [49] ことが示された。6.2 節では、４人の話者における自由発話の言語的な違いについて述べた。この結果、間投詞として話す言葉に個人差があるが、出現頻度はほぼ同等であることが示された。6.3 節では、4 人の話者について朗読発話と自由発話の音響的な違いについて述べた。そして、音素認識率で両発話の違いを調査したところ、あまり大きな差は無いことが示された。

第 7 章では、音声情報に含まれている韻律情報の情報量について述べた。韻律情報は F_0, パワー, 継続時間などの多くの要素から構成されているが、本章では、この中から特にアクセント句境界の位置およびアクセント核の位置の持つ情報量に焦点を当てて情報量を測定した。7.1 節では、その基本的な測定方法について述べた。韻律情報を測定する方法としていくつか考えられるが、ここでは、仮名漢字変換において出力される漢字仮名交じり文の候補の数の減少度という点に着目して、韻律の持つ情報量を研究した。実験の結果、アクセント句境界の位置が持つ情報量は 3.21bit、アクセント核の位置の持つ情報量は 1.97bit、アクセント情報が持つ情報量は 5.16bit であることが示された。この量はかなり大きいと思われる。

第 8 章では、異なる N 個の信号源より生成された信号系列が、どの信号源から生成されたのかを分割・識別する問題について述べた。8.2 節では、Ergodic HMM を用いた問題の解決方法を提示した。8.3 節では、応用例として複数話者発話の識別をあげ、実験により識別性能を示した。8.4 節では、この実験結果について述べた。この実験の結果、複数話者発話の識別においては 341ms 程度の長時間窓分析した LPC ケプストラムを用いることにより、より良好な識別性能が得られること、および尤度の高いモデルを選択することにより平均識別率は向上することが得られた。

第 9 章では、Ergodic HMM を利用した確率付ネットワーク文法の自動学習について述べた。Ergodic HMM と確率つきネットワーク文法が類似した構造を持ち、同種のパラメータで表現される。したがって、大量のテキストデータから Baum-Welch アルゴリズムを用いて HMM のパラメータを推定することによって確率付ネットワーク文法の自動獲得が可能になる。9.1 節で

は、品詞を入力として、HMM による日本語対話文の文節内における形態素の品詞連鎖のモデル化を行なった。この実験の結果、経験的に得られている生成文法に似た形態の確率つきネットワーク文法を自動的に獲得することが示された。9.2 節では、実際の会話から作成した単語列を Ergodic HMM に学習させて、確率つきネットワーク文法を自動的に抽出することを試みた。その結果、Ergodic HMM の構造は学習データの特徴をとらえた文法的な特徴を示しており、単語を文中での機能によって分類して出力していることがわかった。また、Ergodic HMM の状態数が増えるほど詳細な表現が可能となり、より精密な単語の分類を行なっていることがわかった。9.3 節では、メモリ量および計算量を削減した Baum-Welch アルゴリズムを提案した。このアルゴリズムを用いることにより状態数が多い Ergodic HMM の学習が可能になった。そして、得られた Ergodic HMM を言語モデルとして連続音声認識の実験を行なった。この認識実験の結果、単語 bigram よりも高い性能が得られ、提案したアルゴリズムの有効性が示された。

今後の課題としてまずあげられるのは、より大量のテキストデータを収集したとき、text-open data と text-closed data の認識率の差がどこまで接近するかを調査することである。新聞記事の研究から、約 200 万文字のテキストデータを収集すれば、text-open data と text-closed data において認識性能の差は小さいことが示された。しかし、まだ差がある。

また、大量のテキストが入手できないときの対応策も必要である。そのために、言語モデルの分野依存性を抽出し、適合させる研究が必要と思われる。そしてルールベースと確率ベースの言語モデルの結合の研究も必要になる。また、１つの解決策としては、本論文で述べた確率付ネットワーク文法に人間によって修正を加える方法も考えられる。

最後に、自由発話認識における音響モデルの問題がある。この論文では、garbage モデルは、良い結果を示さなかった。しかし garbage モデルには、多くの作成方法がある。これらを研究する必要がある。

謝辞

この研究にあたって多くの人の協力を得ました。

新聞記事の解析には日本文訂正支援システムの辞書を使用しました。これらの辞書は宮崎 正弘氏（当時 NTT、現在新潟大教授）、安田 恒雄氏（NTT）、高木 伸一郎氏（NTT）、島崎 勝美氏（NTT）の方々と池原 悟氏（当時 NTT、現在鳥取大学）が開発したものを使用させていただきました。また、認識実験において用いた HMM の特定話者モデルは山口耕一氏（当時 ATR、現在シャープ株式会社）から、不特定話者モデルは小坂哲夫氏（当時 ATR、現在キヤノン株式会社）から頂きました。また、磯谷 亮輔氏（当時 ATR、現在 NEC）には deleted-interpolation の値や単語の trigram を用いた Viterbi サーチのアルゴリズムに関してコメントを頂きました。また、Baum-Welch の学習アルゴリズムに関して NTT ヒューマンインターフェース研究所の今村 明弘氏から協力を頂きました。また確率付きネットワーク文法の研究には田本 真詞氏（当時 東京工業大学、現在 NTT） や 山本 寛樹氏（当時 東京工業大学、現在 キヤノン） の協力を得ました。自由発話の言語のデータベースは、ATR 音声翻訳通信研究所第一研究室長の匂坂 芳典氏や江原 暉将氏（当時 ATR、現 NHK）の指示のもとに作成されたものを使用いたしました。また X 線 CT 所見作成のデータベースに関し坪井 俊明（NTT ヒューマンインターフェース研究所）の協力を得ました。 また、ATR 音声翻訳通信研究所の森元 逞室長や飯田 仁室長の他、各研究員に多くの協力をいただきました。そして、音声翻訳通信研究所 山崎 泰弘社長および第一研究室匂坂 芳典室長には研究の機会を与えて頂きました。さらに音声翻訳通信研究所の第一研究室の方々には熱心な御討論と有益な御助言をいただきました。また、荒木 哲朗氏（当時 NTT、現在福井大学）や、杉山 雅英氏（当時 ATR、現在会津大学）や 嵯峨山 茂樹氏（当時 ATR、現在 NTT）には、この研究に際し多くの助言を頂きました。

そして、本論文をまとめるに当たり、種々のご指導、ご教示を頂きました豊橋技術科学大学情報工学系の中川 聖一教授に心から感謝致します。同教授には、本研究の遂行にあたっても種々のご相談を頂きました。また、本論文について多くの御意見、御助言を頂きました、金子 豊久教授および増山 繁助教授に深く感謝いたします。最後に、NTT 情報通信研究所の東田 正信氏にはこの論文をまとめる時間と機会を頂きました。

これらの皆様に感謝致します。

関連図書

[1] 青江 順一, “静的ハッシュ法とその応用”, 情報処理, Vol.33, No.11, pp.1359-1366 (1992-11).

[2] 荒木 哲朗, 村上 仁一, 池原 悟, “m 重マルコフモデルを用いた音 節ラティスからの候補絞り込みアルゴリズム”, 電子情報通信学会技術報告, CS90-55, (Dec. 1990).

[3] 荒木 哲朗, 村上 仁一, 池原 悟, “二重音韻マルコフモデルによる日本語の文節音韻認識候補の曖昧さの解消効果”, 情報処理学会論文誌, Vol.30, No.4, pp.467-477 (1989.4).

[4] X.D. Huang, Y. Ariki and M.A. Jack, “Hidden Marcov Models for Speech Recognition”, Edinburgh University Press, Edinburgh (1990).

[5] 有田 英一, 小暮 潔, 野垣内 出, 飯田 仁, “メディアに依存する会話の様式”, 情報処理学会研究報告, NL61-5, (1987).

[6] A.Asadi, R.Schwartz and J.Makhoul, “ Automatic Modeling for Adding New Words to a Large Vocabulary Continuous Speech Recognition System”, Proc. ICASSP91, (1991).

[7] A.Averbuch, “An IBM PC Based Large-Vocabulary Isolated-Utterance Speech Recognizer”, Proc. ICASSP86, Vol.1, 2.4.1, pp.53-56, (1986).

[8] G.Yu, M.Siu, H.Gish, “An Unsupervised, Sequential, Learning Algorithm for the segmentaion of Speech Waveform with Multiple Speakers”, Proc. of ICASSP92, 2-189 (Apr.1992).

[9] H.Gish, G.Yu, “Clustering of Speakers Engaged in Dialog”, Proc. of Speech Research Symposium XII (Jun.1992).

[10] 江原 暉将, 小倉 健太郎, 森本 逞 , “電話対話ベースの構築”, 情報処理学会第 40 回全国大会予稿集, Vol. 1, pp.486-487 (1990).

[11] 江原 暉将, “対話データベースからの統計情報の抽出”, 情報処理学会第 41 回全国大会予稿集, pp.3-83-84, (1990)

[12] G.D.Forney, “the Viterbi Algorithm”, Proc.of IEEE, Vol.61, pp.268-278 (1973).

[13] 古井 貞熙, “日本語単音節音声認識の検討”, 電子通信学会全国大会, No.1351, pp.5-329 (1981).

[14] 古井 貞熙, 板倉 文忠, “単語の統計的パラメータによる話者認識”, 電子通信学会論文誌, 56A-11, pp.717-724 (1973-12).

[15] Cambridge University Engineering Department Speech Group and Entopic Research Laboratories Inc., “HTK:Hidden Markov Model Toolkit V1.5”, (23 Sepember 1993).

[16] 箱田 和雄, 佐藤 大和, “文音声における音調規則”, 電子通信学会論文誌, D-104, Vol.J63-D, pp.715-720, (1980)

[17] 花沢 利行, 中島 邦男, “音声タイプライタを用いた未知語検出方法の改良検討”, 日本音響学会講演論文集, pp.219-220, (Oct. 1992).

[18] Higgins A.L., Wohlford R.E., ”Keyword recognition using template concatenation”, Proc.ICASSP85, pp.1233-1236, (March 1985).

[19] 飯田 仁, 野垣内 出, 相沢 輝昭, “通訳を介した電話対話の特徴分析”, 電子通信学会技術速報, NLC86-11 (1986).

[20] 池原 悟, 白井 諭, “単語解析プログラムによる日本文誤字の自動検出と二次マルコフモデルによる訂正候補の抽出”, 情報処理学会論文誌, Vol.25, No.2, pp.298-305, (1984.3)

[21] 池原 悟, 安田 恒雄, 島崎 勝美, 高木 伸一郎, “日本文訂正支援システム”, ＮＴＴ研究実用化報告第 36 巻第 9 号, (1987)

[22] 今村 明弘, 北井 幹雄, “事後確率を用いたフレーム同期型ワードスポッティング ”, 日本音響学会講演論文集, 1-4-2, pp.3-4, (Mar. 1993).

[23] 井ノ上 直己, 武田 一哉, 山本 誠一, “Garbage HMM を用いた自由発話文中の不要語処理手法”, 電子情報通信学会論文誌, Vol.J77-A, No.2, pp.215-222, (1994.2).

[24] 田中 和世, 板橋 秀一, 他, “音声の知的処理に関する調査報告書”, システム技術開発調査研究 3-R-2, 財団法人 機械システム振興協会, (平成４年３月)

[25] 伊藤, 中川 聖一, “確率オートマトンと品詞の３字組出現確率を用いた文節音声認識”, 日本音響学会講演論文集, pp.145-146, (1988.10)

[26] 伊藤 克亘, 速水 悟, 田中 穂積, “連続音声認識における未知語の扱い”, 電子情報通信学会 技術報告, SP91-96, (1991-12).

[27] F.Jelinek, “Self-Organized Language Modeling for Speech Recognition”, Readings in Speech Recognition, Morgan Kaufmann Publishers, Inc. San Mateo, California pp.450-506, (1990)

[28] F. Jelinek, R.L.Mercer, S.Roukos, “Principle of Lexical Language Modeling for Speech Recognition”, In S. Furui and M. M. Sondhi, editors, Advances in Speech Signal Processing, pp.651-699, Marcel Dekker, Inc., New York, New York. (1992).

[29] 甲斐 充彦, 中川 聖一, “日本語連続音声認識システム SPOJUS-SYNO の改良と評価”, 電子情報通信学会技術報告, SP93-20 pp.49-56, (1993-06).

[30] Kenji Kita, Takeshi Kawabata, Toshiyuki Hanazawa, “HMM Speech Recognition Using Stochastic Language Models”, ICASSP90, J. Acoust. Soc. Jpn. (E) 12, 3 pp.99-105, (1991)

[31] 北 研二, 江原 暉将, 森元 逞, “連続音声認識における未知語処理”, 日本音響学会講演論文集, 3-5-3, pp.93-94, (Mar. 1991)

[32] 北 研二, 森元 逞, “音声認識システムにおける確率文法の有効性”, 情報処理学会第 41 回全国大会予稿集, pp.2-237-238, (1990).

[33] 小林 聡, 山本 幹雄, 中川 聖一, “間投詞・言い直し等の出現に関する音響的特徴”, 情報処理学会, 音声言語処理研究グループ資料 93-SLP-1-2, pp.7-10, (1993).

[34] 小林 聡, 甲斐, 山本 幹雄, 中川 聖一, “間投詞の出現位置の特徴分析と音声認識システムの評価”, 情報処理学会, 音声言語処理研究グループ資料, 92-SLP-3-4, (1992).

[35] 児島宏明, 田中和世, 速水悟, “単語音声サンプルからの階層的な音韻概念の獲得”, 日本音響学会講演論文集, 2-P-4, (1991-10).

[36] Kucera.H., Francis, W.N., “Computational Analysis of Present-day American English”, Brown University Express. Providence, Rhode Island. (1967).

[37] T. Kuhn, et al., “Ergodic Hidden Markov Models and Polygrams for Language Modeling”, Proceeding of ICASSP94, Vol. 1, pp.357-360, (1994).

[38] J.Kupiec, “Robust Part-of-Speech Tagging Using A Hidden Markov Model”, Computer Speech and Language, Vol. 6, pp.225-242, (1992).

[39] K.Lari, S.J.Young, “The Estimation of Stochastic Context-Free Grammars Using The Inside-Outside Algorithm”, speech recoginition, Computer Speech and Language, Vol. 4, pp.35-56, (1990).

[40] Kai-Fu Lee, “Large-Vovabulary Speaker Independent Continuous Speech Recognitton: The SPHINX System”, 15213 CMU-CS-88-148, (April 18, 1988).

[41] G.Maltese, F.Mancini, “An Automatic Technique to Include Grammartical And Morphological Information in a Trigram-Based Statistical Language Model”, ICASSP92, Vol.1, pp.157-160, (1992).

[42] 松井 知子, 古井 貞熙, “エルゴード的 HMM による話者認識”, 日本音響学会講演論文集, 3-6-14, (1991-10).

[43] 松永 昭一, 好田 正紀, “branch&bound 法の効果と Bottom-up 音節認識を利用した候補選択”, 音声研究会資料, S85-79, pp.611-620

[44] 南 泰浩, 山田智一, 鹿野清宏, “番号案内を対象とした大語彙連続音声認識アルゴリズム ”, 電子情報通信学会技術速報, SP92-108, (Dec. 1992).

[45] 南 泰浩, 中川 正雄, “trigram モデルを用いた複数候補を求めるフレーム同期型 HMM 連続音声認識”, 電子情報通信学会論文誌, D-2, Vol.j73-D-2, No.9 pp.1383-1392, (1990).

[46] 宮崎 正弘, “係り受け解析を用いた複合語の自動分割法”, 情報処理学会論文誌, Vol.25, No.6, pp.970-979, (1984.11)

[47] 宮崎 正弘, 大山 芳史 “日本音声出力のための言語処理方式”, 情報処理学会論文誌, Vol.27, No.11, pp.1053-1061, (1986).

[48] 村上 仁一, “メモリ量および計算量を削減した Baum-Welch アルゴリズムの提案と言語モデルへの適用”, 日本音響学会講演論文集, 1-Q-6, pp.153-154, (1994-10) .

[49] 村上 仁一, 荒木 哲朗, 池原 悟, “日本文音節入力に対して 2 重マルコフ連鎖モデルを用いた漢字かな交じり文節候補の抽出精度”, 電子通信学会論文誌, D-2, Vol.J75-D-2, No.1, pp.11-20, (Jan.1992).

[50] 村上 仁一, 嵯峨山 茂樹, “自由発話音声認識における音響的および言語的な問題点の検討”, 電子情報通信学会技術報告 SP91-100, pp.71-78, (Nov. 1991)

[51] 村上 仁一, 坪井 俊明, “BIGRAM をもちいた音節 HMM による文節音声認識”, 日本音響学会講演論文集, 3-5-15, pp.117-118, (1991-04).

[52] 村上 仁一, 松永 昭一, “単語の trigram を利用した文音声認識アルゴリズムの改良と、非朗読発話認識への拡張”, 電子情報通信学会技術報告, SP93-127, pp.71-78, (1994-01).

[53] 村上 仁一, 荒木 哲朗, 池原 悟, “２重マルコフ連鎖確率モデルを 使用した単音節音声入力の改善”, 電子情報通信学会技術報告 , SP88-29, pp.63-70, (June 1988).

[54] 村上 仁一, 荒木 哲郎, 池原 悟 “音声におけるポーズ長およびアクセント位置の情報量の考察”, 日本音響学会講演論文集, 3-3-11, pp.89-90, (Oct. 1989)

[55] 村上 仁一, 嵯峨山 茂樹, “単語の trigram を用いた連続音声認識の一アルゴリズム”, 日本音響学会講演論文集, 2-Q-7, pp.185-186, (1992-10).

[56] 村上 仁一, 嵯峨山 茂樹, “HMM を用いた形態素解析”, 情報処理学会第 45 回全国大会予稿集, 4F-01, (1992-10).

[57] 村瀬 功, 上田 佳央, 中川 聖一 , “文脈自由文法と bigram, trigram による言語のモデル化の検討”, 電子情報通信学会技術研究報告, SP90-5, pp.49-56, (1990).

[58] 永井 明人, 他, “HMM-LR 連続音声認識装置の開発と性能評価”, 日本音響学会講演論文集, 1-5-23, pp.45-46, (Oct. 1991).

[59] 長尾 真 “日本語情報処理”, 電子通信学会 (昭和 59 年)

[60] 中川 聖一, “確率モデルによる音声認識”, 電子情報通信学会, 1988.

[61] 中川 聖一, “音声入力を想定したあいまいな発話文の理解システムに関する研究”, 文部省科学研究費補助金, 一般研究（B）研究成果報告書, pp.137-144, （平成６年３月）.

[62] C.Nakatani and J.Hirschberg, “A Speech-First Model for Repair Detection and correction”, In Proceedings 31st Annual meeting of the-Association for computational linguistics, pp.46-53, (1993).

[63] 岡田 美智男, “文脈自由な句構造文法による One-Pass DP 法の構文制御について”, 日本音響学会講演論文集, pp.91-92, (Mar. 1990).

[64] 大倉 計美, 杉山 雅英, 嵯峨山 茂樹, “混合連続分布 HMM を用いた移動ベクトル場平滑化話者適応方式”, 日本音響学会講演論文集, 2-Q-17, pp.191-192, (Mar. 1992).

[65] Lawrence R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, pp.267-295, IEEE (1989)

[66] 佐川 雄二, 大西 昇, 杉江 昇, “対話文における誤りの自動修復”, 情報処理学会自然言語処理研究会資料, 93-10, pp.71-78 (Jan. 1993)

[67] Toshiya Sakano, Tsuyoshi Morimoto, “Efficiency of Linguistic Grammar For Speech Recognition”, IASTED International Conferrence SIGNAL PROCESSING AND DIGITAL FILTERING (1990)

[68] 迫江博昭, 藤井浩美, 吉田和永, 亘理誠夫, “フレーム同期化、ビー ムサーチ、ベクトル量子化の統合による DP マッチングの高速化”, 電子通信学会論文誌, Vol.J71-D, No.9, pp.1650-1659, (Sep. 1988).

[69] Rechard schwarts, et., “Comparative Experiments on Large Vocabulary Speech Recongition”, ARPA Human Langauge Technology Workshop, (Mar. 1993)

[70] L.E.Shanon, “Predection and entropy of printed English”, Bell Syst. Tech. J., 30, pp.50-64, (1951)

[71] 柴田 武, 柴田 里程 “アクセントは同音語をどの程度弁別しうるかー日本語・英語・中国語の場合ー”, 計量国語学会 17-7, pp.317-322 (Dec. 1990)

[72] 鹿野 清宏, “Trigram Model による単語音声認識結果の改善”, 電子情報通信学会技術報告, SP87-23, pp.9-16, (1987).

[73] 金田 一京助, 他, “新明解 国語辞典 第 3 版”, (1987-10).

[74] 篠崎 直子, 小倉 健太郎, 森元 逞, “言語データベース作成のためのシュミ レーション会話”, 第 37 回情報処理全国大会, pp.1000-1001, (1988).

[75] E.Shriberg, J.Bear and J.Dowding, “Automatic detection and corrction of repairs in Human-Computer Dialog”, Proc. Speech and Natual Language Workshop, pp.419-424, (1992).

[76] 首藤 公昭, 吉村 賢治, “日本語の構造とその解析”, 情報処理 Vol.27 No.8, pp.947-953, (1986)

[77] H.Singer “Pitch Dependent Phone Modelling for HMM Based Speech Recognition”, Proc. ICASSP92, 36.1, pp.273-276, (Mar. 1992)

[78] 田中 信一, 伊藤 彰則, 牧野 正三, 曽根 敏夫, 城戸 健一, “日本語 Dictation システムにおける文節検出の高速化”, 電子情報通信学会技術報告書 Vol.90 No.374 SP90-70 pp.17-24, (1990)

[79] 杉山 雅英, 村上 仁一, 渡辺 秀行, “N 信号源モデルに基づく音声の区分化識別問題ー話者特徴の違いに基づく区分化音声の識別”, 電子通信学会論文誌 D-2 Vol.J76-D-2 No.12, pp.2477-2485, (1993-12)

[80] 高木 一幸, 保浦 直子, 板橋 秀一, “対話における話題展開と発話単位の性質”, 情報処理学会, 音声言語処理研究グループ資料, 93-SLP-1-3 pp.11-18, (1993).

[81] 高橋 敏, 松永 昭一, 嵯峨山 茂樹 “ピッチパタン情報を用いた単語音声認識”, 日本音響学会講演論文集, 1-3-20, pp.39-40, (Mar. 1990)

[82] 武田 一哉, 匂坂 芳典, 片桐 滋, 桑原 尚夫, “音韻ラベルの持つ日本語音声データベースの構築”, SP87-19, (1987).

[83] 瀧 保夫, “情報論 I ”, 岩波書店, (1978).

[84] 田本 真詞, 村上 仁一, 嵯峨山 茂樹, “HMM を利用した言語獲得の可能性について”, 人工知能学会研究会資料, SIG-SLUD-9201-6, pp.47-54, (1992).

[85] 徳永 豪, “ランダムアルゴリズムの話題から”, 電子情報通信学会誌, Vol.77, No.9, pp.957-967, (1994-9).

[86] 坪井 俊明, 管村 昇, 他, “文節発声の日本語入力システムにおける日本語変換法”, 電子通信学会論文誌, D-2, Vol.j72-D-2, No.8, pp.1284-1290, (1989.4)

[87] 坪井 宏之, 橋本 秀樹, 竹林 洋一, “連続音声理解のためのキーワードラティスの解析”, 日本音響学会講演論文集, 1-5-11, pp.21-22, (Oct. 1991).

[88] 坪井 俊明, 菅村 昇 “文章作成支援装置の評価”, 電子情報通信学会技術報告, SP90-36, pp.17-22, (1990.8).

[89] Wayne Ward and Sunil Issar, “Recent Improvements in the CMU Spoken Language Understanding System”, ARPA HUMAN LANGUAGE TECHNOLOGY WORKSHOP, pp.208-211, (Mar. 1994).

[90] 渡辺 隆夫, 塚田 聡, “音節認識を用いた尤度補性による未知発話のリジェクション”, 電子通信学会論文誌, D-2 Vol.J75-D-2 No.12 pp.2202-2009, (1992-12).

[91] 渡辺 隆夫, 畑崎, “音節をベースとする日本語音声認識”, 音声研究会資料, S85-62, pp.477-484

[92] Wilpon G., Lee C. and Rabiner R., “ Application of Hidden Markov Models for Recognition of a Limited Set of Words in Unconstrained Speech”, Proc. ICASSP89, pp.254-257, (May 1989).

[93] Xuedong Huang, et.al “An Overview of the SPHINX-2 Speech Recognition System”, ARPA Human Language Technology Workshop (Mar. 1993).

[94] 山本 寛樹, 村上 仁一, 嵯峨山 茂樹, “HMM を言語モデルに用いた連続音声認識の検討”, 情報処理学会第 45 回全国大会予稿集, Vol. 3, pp.227-228, (1992.10).

[95] 山本 寛樹, 村上 仁一, 嵯峨山 茂樹, “HMM を言語モデルに用いた連続音声認識の検討”, 日本音響学会講演論文集, Vol.1, pp.193-194, (1992.10).

[96] 山本 寛樹, “HMM による言語モデルの自動獲得の検討”, 早稲田大学理工学研究科修士論文 (1993).

[97] 山本 幹雄, 小林 聡, 中川 聖一, “音声対話文における助詞落ち・倒置の分析と解析手法”, 情報処理学会論文誌 Vol.11, No.11, pp.1322-1330, (1992).

[98] 吉本 啓, “日本語品詞の分類”, ATR Techinical Report TR-I-0008 (Nov, 1987).

[99] Victor Zue, Et al “ Pegasus: A Spoken Language Interface for On-Line Air Travel Planning”, ARPA Human Language Technology Workshop, pp.196-201, (Mar. 1994).

[100] Victor Zue, James Glass, David Goodine, et al. “The MIT ATIS System: Preliminary Development, Spontaneous Speech Data Collection, And Performance Evaluation”, Proc. Eurospeech 91, pp.537-540, Genova, Italy, (Sep. 1991)

[101] Victor Zue, Nancy Daly, et al, “The Collection and Preliminay Analysis of a Spontaneous Speech Database”, Proc. DARPA Workshop 1989, pp.126-134, (1989).

[102] 井ノ上 直己, 江原 暉将, 小倉 健太郎, “係り受け関係から見たキーボード会話と電話会話の比較”, 情報処理学会第 40 回全国大会 pp.1-490-491, (1990)

付 録A　品詞の出現頻度

この節では第 9.2 章において使用した言語データベースの品詞の出現頻度をまとめた。

この実験では、データベースの 8000 文を奇数番目の文の set と偶数番目の文の set とに分け、さらにそれぞれ先頭から 1000 文の set、先頭から 2000 文の set、4000 文の set に分けている。奇数番目の set3 種類をそれぞれ odd1000, odd2000, odd4000、偶数番目の set を even1000, even2000, even4000 と名づけ使用している。

表 A.1: 品詞別出現頻度 (odd1000)

順位	品詞名	出現数	割合 (%)	一文当たり出現頻度
1	普通名詞	1915	14.40	1.915
2	助動詞	1900	14.29	1.900
3	格助詞	1651	12.41	1.651
4	本動詞	1028	7.73	1.028
5	間投詞	912	6.86	0.912
6	接続助詞	780	5.87	0.780
7	補助動詞	596	4.48	0.596
8	感動詞	552	4.15	0.552
9	終助詞	514	3.86	0.514
10	副詞	512	3.85	0.512

表 A.2: 品詞別出現頻度 (odd2000)

順位	品詞名	出現数	割合 (%)	一文当たり出現頻度
1	普通名詞	2937	14.17	1.468
2	助動詞	2878	13.88	1.439
3	格助詞	2586	12.47	1.293
4	本動詞	1672	8.07	0.836
5	間投詞	1473	7.11	0.737
6	感動詞	1307	6.30	0.653
7	接続助詞	1208	5.83	0.604
8	補助動詞	858	4.14	0.429
9	副詞	775	3.74	0.388
10	終助詞	712	3.43	0.356

表 A.3: 品詞別出現頻度 (odd4000)

順位	品詞名	出現数	割合 (%)	一文当たり出現頻度
1	普通名詞	8590	14.98	2.147
2	格助詞	7832	13.66	1.958
3	助動詞	7778	13.56	1.944
4	本動詞	4886	8.52	1.222
5	間投詞	4404	7.68	1.101
6	接続助詞	3813	6.65	0.953
7	補助動詞	2849	4.97	0.712
8	副詞	2065	3.60	0.516
9	感動詞	1997	3.48	0.499
10	終助詞	1586	2.77	0.397

表 A.4: 品詞別出現頻度 (even1000)

順位	品詞名	出現数	割合 (%)	一文当たり出現頻度
1	普通名詞	2089	15.11	2.089
2	助動詞	1989	14.39	1.989
3	格助詞	1819	13.16	1.819
4	本動詞	1105	7.99	1.105
5	間投詞	938	6.79	0.938
6	接続助詞	816	5.90	0.816
7	補助動詞	646	4.67	0.646
8	感動詞	509	3.68	0.509
9	副詞	503	3.64	0.503
10	終助詞	463	3.35	0.463

表 A.5: 品詞別出現頻度 (even2000)

順位	品詞名	出現数	割合 (%)	一文当たり出現頻度
1	助動詞	3008	14.25	1.504
2	普通名詞	3004	14.23	1.502
3	格助詞	2662	12.61	1.331
4	本動詞	1756	8.32	0.878
5	間投詞	1473	6.98	0.737
6	接続助詞	1258	5.96	0.629
7	感動詞	1237	5.86	0.619
8	補助動詞	913	4.32	0.457
9	副詞	780	3.69	0.390
10	終助詞	664	3.14	0.332

表 A.6: 品詞別出現頻度 (even4000)

順位	品詞名	出現数	割合 (%)	一文当たり出現頻度
1	普通名詞	8424	14.82	2.106
2	助動詞	7948	13.99	1.987
3	格助詞	7744	13.63	1.936
4	本動詞	4932	8.68	1.233
5	間投詞	4281	7.53	1.070
6	接続助詞	3723	6.55	0.931
7	補助動詞	2835	4.99	0.709
8	副詞	2072	3.65	0.518
9	感動詞	1973	3.47	0.493
10	終助詞	1547	2.72	0.387

表 A.7: 文構成単語数 odd

	odd1000		odd2000		odd4000	
文の単語数	頻度	(%)	頻度	(%)	頻度	(%)
1	127	12.70	399	19.95	610	15.25
2	18	1.80	86	4.30	140	3.50
3	77	7.70	125	6.25	206	5.15
4	152	15.20	222	11.10	356	8.90
5	52	5.20	92	4.60	190	4.75
1〜 10	570	57.00	1273	63.65	2136	53.40
11〜 20	201	20.10	453	22.65	901	22.53
21〜 30	113	11.30	153	7.65	449	11.22
31〜 40	62	6.20	66	3.30	252	6.30
41〜 50	30	3.00	30	1.50	111	2.77
51〜 130	24	2.40	25	1.25	148	3.70

表 A.8: 文構成単語数 even

	even1000		even2000		even4000	
文の単語数	頻度	(%)	頻度	(%)	頻度	(%)
1	107	10.70	375	18.75	547	13.68
2	15	1.50	79	3.95	131	3.27
3	75	7.50	134	6.70	259	6.47
4	139	13.90	210	10.50	362	9.05
5	51	5.10	87	4.35	172	4.30
1〜 10	526	52.60	1239	61.95	2107	52.68
11〜 20	232	23.20	482	24.10	943	23.57
21〜 30	128	12.80	160	8.00	454	11.35
31〜 40	64	6.40	66	3.30	250	6.25
41〜 50	30	3.00	32	1.60	116	2.90
51〜130	20	2.00	21	1.05	130	4.25

付 録B　8状態Ergodic HMMの特徴

ここでは第 9.2 章において得られた 8 状態 Ergodic HMM から抽出される細かい特徴を述べた。

1. 間投詞、感動詞について

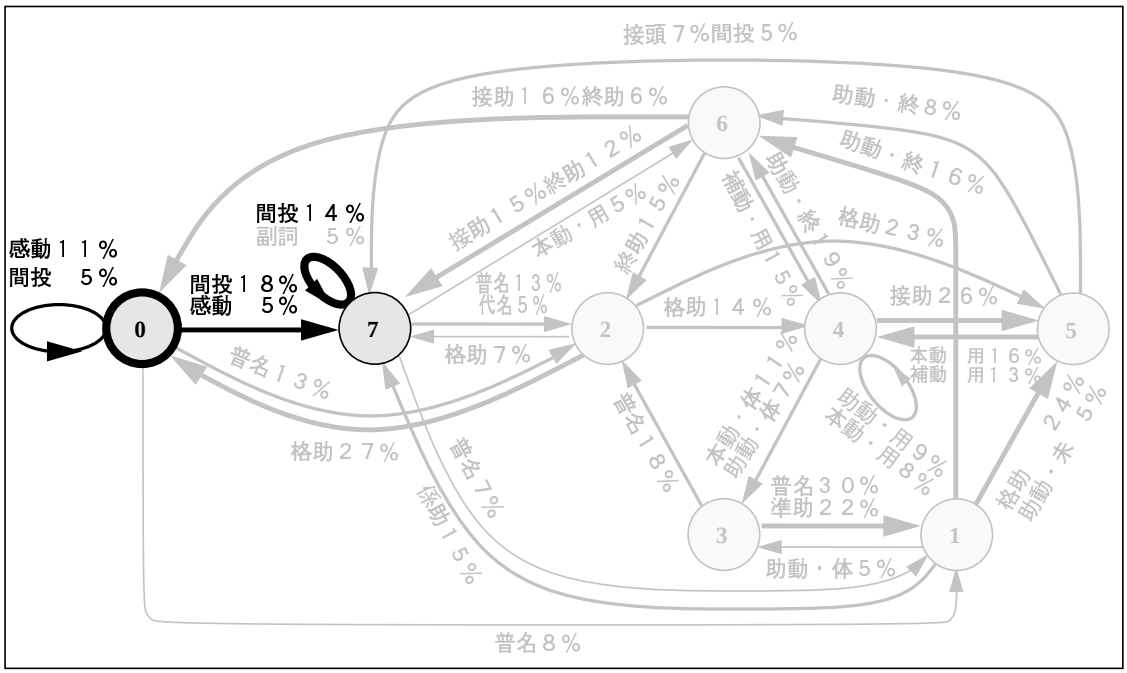


図 B.1: 間投詞・感動詞の出力

8 状態の Ergodic HMM でも、2 状態、4 状態と同様に、間投詞や感動詞がイニシャルノード○0からの遷移 (○0⇒○0, ○0○7⇒○7) で出力されている。これは、学習データの特徴を表現していといえる。

出力される主な単語を以下に示す。括弧内の数字は、シンボル出力確率である。

- 状態○0⇒状態○0 「はい」 (50%) 「あ」 (10%)

- 状態○0⇒ 状態○7 「え」 (19%) 「あのー」 (10%) 「はい」 (8%) 「え」 (7%)
- 状態○7⇒ 状態○7 「あの」 (16%) 「えー」 (10%) 「あのー」 (9%) 「はい」 (6%)

2. 体言について

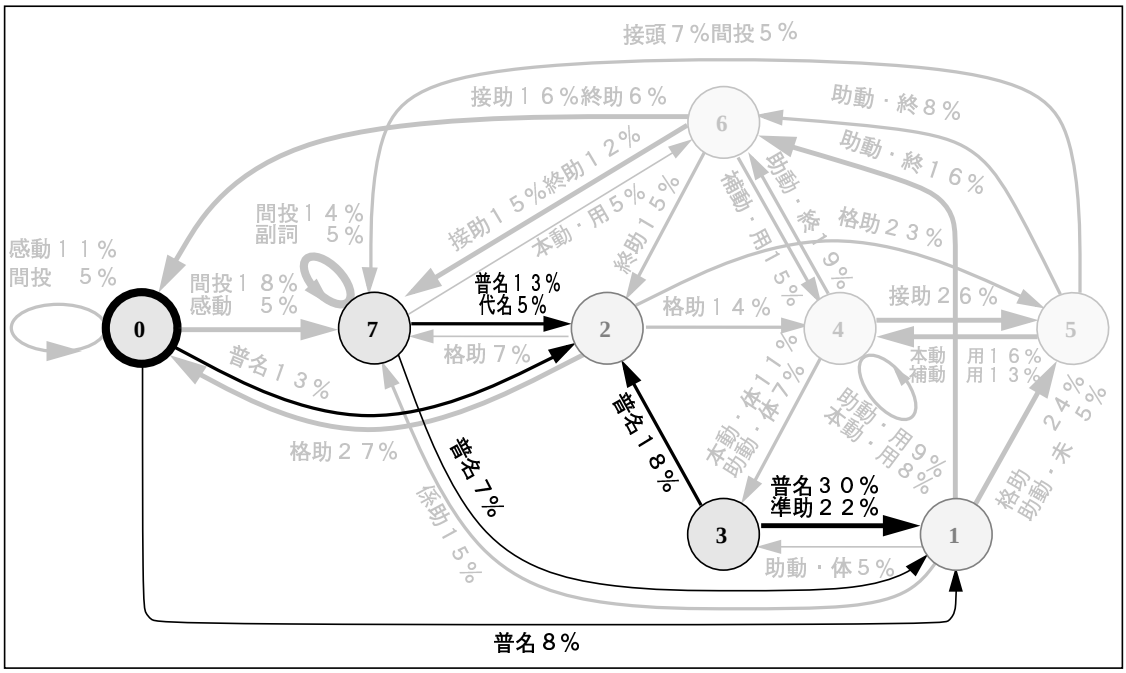


図 B.2: 体言の出力

図 9.9 では、複数の品詞を出力する遷移が見られる。しかし、Ergodic HMM は活用する品詞、体言を分離し、別々の遷移で出力している。体言は主として状態○0○3○7から状態○1○2に遷移する際に出力されている (図 B.2 参照)。出力されている単語は以下の通りである。

- 状態○0⇒ 状態○1
  「方（ほう）」 (11%) 「方（かた）」 (5%) 以下 「それ」「私」「これ」
- 状態○0⇒ 状態○2
  「方（ほう）」 (23%) 「会議」 (7%) 以下 「事務」「方（かた）」「中」
- 状態○3⇒ 状態○1
  「ん」 (準体助詞, 31%) 「こと」 (18%) 以下「の」「もの」「わけ」

- 状態○3⇒ 状態○2
  「こと」 (13%) 「時」 (4%) 以下 「ふう」「用紙」
- 状態○7⇒ 状態○1
  「私」 (9%) 「それ」 (5%) 以下 「結構」「これ」「名前」
- 状態○7⇒ 状態○2
  「こちら」 (7%) 以下 「会議」「そちら」「先生」「私」

上記の単語以外でも、多くの状態遷移において体言の単語が多数出力されているのが観測された。

状態○3からの遷移では、形式名詞（ADD では普通名詞として扱っている）が他の名詞類よりもやや高い出力確率になっている。また、○3⇒○1の遷移では準体助詞「ん」と同様に用いられる準体助詞「の」が出力されている。

状態○7からの遷移では、出力確率の合計では普通名詞が最も高いが、個々の単語出力の上位は代名詞が多い。人に対して用いられる名詞類が多く出力されていることもわかる。

状態○0からの遷移では、主として形式名詞「方」が出力されている。また、体言以外の単語では副詞の出力が多く見られた (○0⇒ ○1の遷移では、副詞「そう」の出力確率 25%である。)。これは、状態○1からの遷移で出力される「です」や「でしょ」をともなって学習データに多く見られた「そうです」「そうですか」などを表現しているため、文頭に当たる状態からの遷移で出力されていると考えられる。

遷移によって出力される単語や品詞が異なり、複数ある体言を出力する遷移でも、個々の表現する内容は異なっていることがわかる。

3. 活用する品詞について

   8 状態の Ergodic HMM では、活用する品詞が、品詞ごとではなく、活用形ごとに集まって出力されている。連体形のものは状態○3への遷移 (○4, ○1⇒ ○3) で (図 B.3 参照)、連用形のものは状態○4への遷移 (○4, ○5, ○6⇒ ○4) で (図 B.4 参照)、助動詞の終止形は状態○6への遷移 (○1, ○4, ○5⇒ ○6) で (図 B.5 参照)、それぞれ出力されている。以下に、活用形ごとに見られる特徴を述べる。

   (a) 連体形

   図 B.3 に連体形の出力される遷移を示し、以下にその遷移で出力される主な単語を示す。

   - 状態○4⇒ 状態○3
     「いう」 (50%) 「た」 (26%) 「たい」 (10%) 以下「思う」「ます」

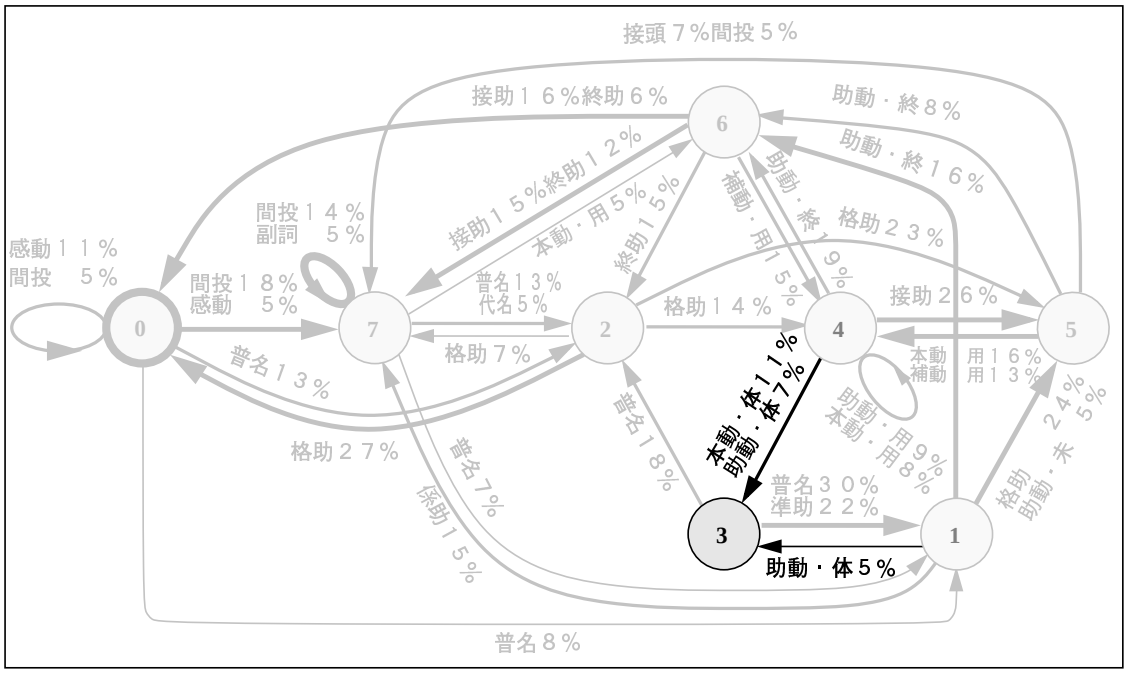


図 B.3: 連体形の出力

図 B.3 から連体形を出力して遷移する先の状態○3は、連用形を出力する遷移と体言を出力する遷移の節点になっていることがわかる。状態○3からの遷移では、体言 (形式名詞、準体助詞) が多く出力されていた。ここで、「〜(と) いうこと」や「〜(し) たこと」、「〜(し) たいん (です)」などの「連体形＋体言」の接続が表現されていると思われる。

(b) 連用形

図 B.4 に連用形の出力される遷移を示し、以下にその遷移で出力される主な単語を示す。

- 状態○4⇒ 状態○4
  「まし」 (50%) 「思い」 (23%) 以下 「申し」「思っ」「いい」
- 状態○5⇒ 状態○4
  「おり」 (15%) 「頂き」 (9%) 以下 「し」「ございまし」「頂い」
- 状態○6⇒ 状態○4
  「し」 (64%) 「いたし」 (21%) 以下「申し上げ」「でき」
- 状態○7⇒ 状態○6
  「願い」 (16%) 「送り」 (8%) 以下「待ち」「伺い」

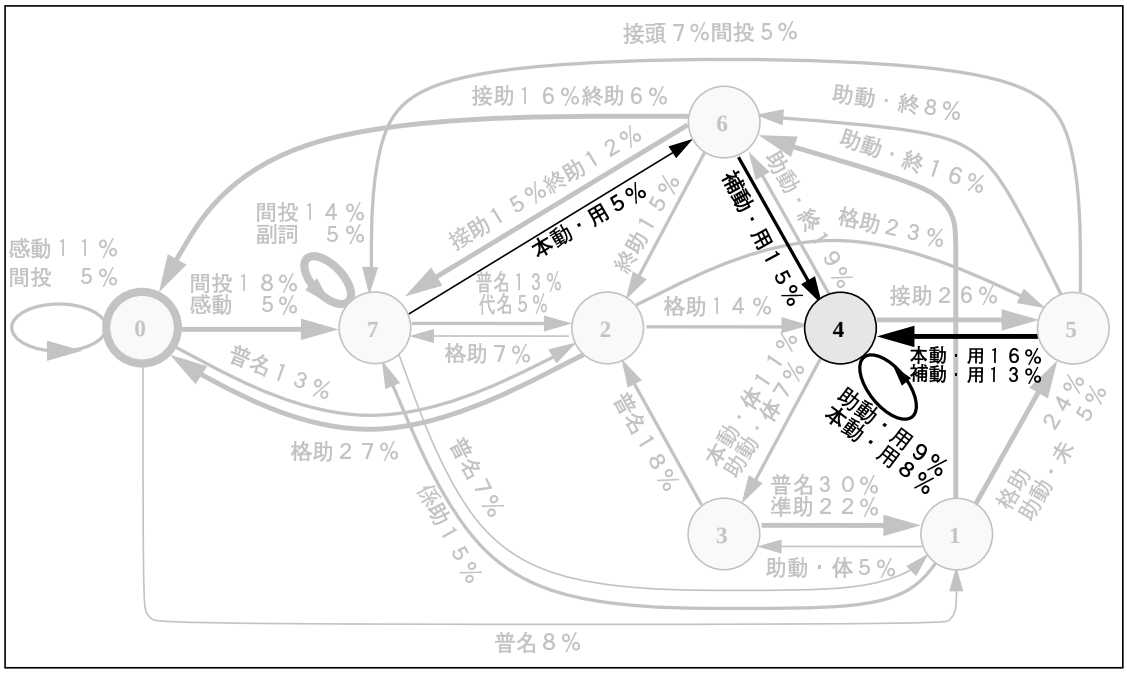


図 B.4: 連用形の出力

状態○4への遷移は連用形を出力するものが多く、状態○4からの遷移は活用する品詞、接続助詞 (○4⇒ ○5) を出力している。状態○4は連用形の単語と助動詞・補助動詞との節点と考えられる。○4⇒ ○4○6の遷移で、助動詞の「ます」「まし」が多く出力され、「～思います」「～おります」のような接続が考えられる。○4⇒ ○5の遷移では 97%の確率で「て」が出力され、「～頂いて」「～して」「思って」など活用する品詞の「連用形＋接続助詞 て」を表現している。○7⇒ ○6の遷移で出力される単語の多くは、本動詞で、○5⇒ ○7で出力される (出力確率 34%) 接頭辞「お」と接続して、「お待ち (して)～」「お願い (いたし）～」などの謙譲表現を形成している。

(c) 終止形

図 B.5 に終止形の出力される遷移を示す。終止形の単語の多くは助動詞である。これは、学習データの内容が会議に申し込みに関する電話対話であるため、本動詞の言いきりの表現が少なく、「～です」「～ます」などの助動詞をともなった丁寧な表現が多いためと考えられる。

以下に終止形を出力する遷移で出力される主な単語を示す。

- 状態○1⇒ 状態○6

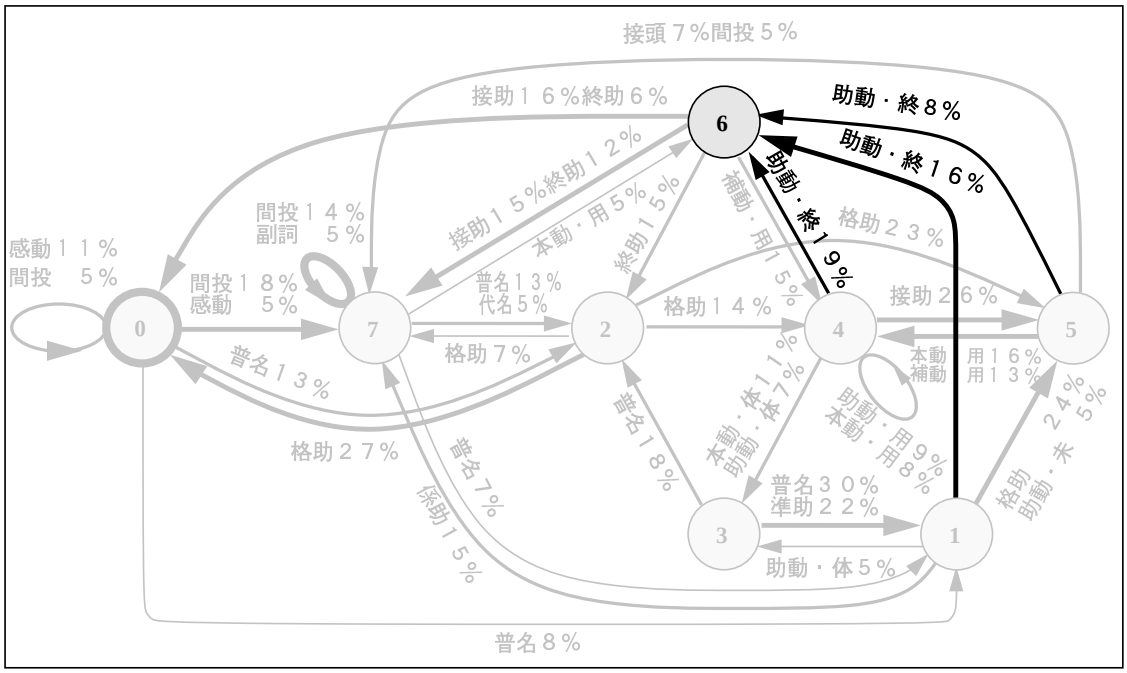


図 B.5: 終止形の出力

「です」 (94%)

- 状態○4⇒ 状態○6
  「ます」 (69%) 「ます」 (助動詞連体形, 17%) 以下 「た」
- 状態○5⇒ 状態○6
  「う」 (37%) 「ございます」 (14%) 「です」(7%) 以下 「ない」

単語の出力をみると、「です」「ます」が分離して出力されているのがわかる。体言から接続する「です」を出力する遷移が状態○1から遷移していて、連用形の単語から接続する「ます」が状態○4から遷移の遷移で出力されている。状態○1は体言を出力した遷移の集まる状態であり、連体形の単語を出力した遷移は状態○4に集まっていることから、これらはいずれも自然言語の文法に沿った接続を表現している。状態○5から遷移する「う」は推量や意志の意味を持つ助動詞で、未然形の単語から接続する。状態○1⇒ 状態○5の遷移で助動詞の未然形「でしょ」が 13%出力されており、「〜でしょう (か)」を表現していると考えられる。

4. 格助詞の分離について

体言を出力する遷移の集まる状態○2からの全ての遷移は、格助詞を出力

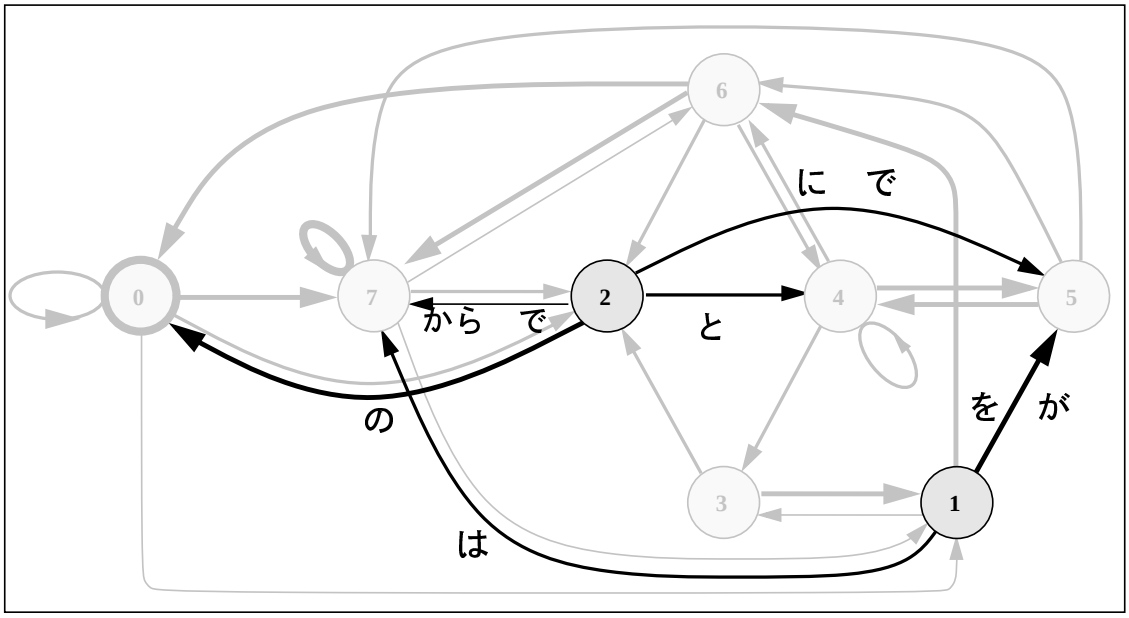


図 B.6: 格助詞の出力

している。

(◯2⇒ ◯0, ◯4, ◯5, ◯7, 図 B.6 参照)。体言を出力する遷移が集まるもう一つの状態◯1からの遷移も、格助詞や係助詞を出力している (◯1⇒ ◯5, ◯7)。この連鎖は「私は」「会議の」など 名詞 + 格助詞・係助詞 を表現している。

格助詞は複数の遷移で出力されている。しかし、後接する品詞の違い (文中での機能の違い) によって各遷移は別々の格助詞を出力していることが、各遷移の単語出力確率を調べた結果示された。

- 状態◯2⇒ 状態◯0 「の」 (94%)
- 状態◯2⇒ 状態◯7 「から」 (28%) 「で」 (25%) 以下 「が」「まで」
- 状態◯2⇒ 状態◯4 「と」 (94%)
- 状態◯2⇒ 状態◯5 「に」 (73%) 「で」 (18%)
- 状態◯1⇒ 状態◯5 「を」 (39%) 「が」 (27%)
- 状態◯1⇒ 状態◯7 「は」 (80%) 「も」 (7%)

付 録C　16状態Ergodic HMMの特徴

ここでは第 9.2 章において得られた 16 状態 Ergodic HMM から抽出される細かい特徴を述べた。

1. 間投詞、感動詞について

   図 C.1 に間投詞・感動詞を出力する遷移を示す。間投詞、感動詞は 2、4、8 状態の場合と同様で、初期状態確率の高い状態 (○2, ○3) からの遷移で出力される。

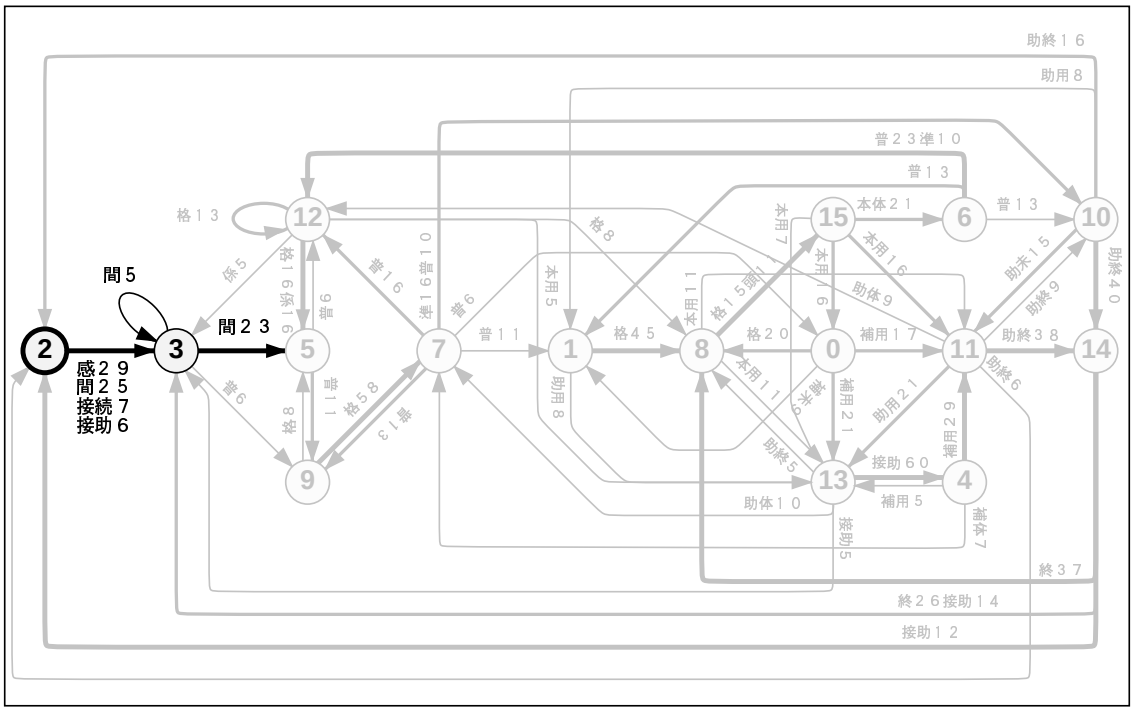


図 C.1: 間投詞・感動詞の出力

- 状態○2⇒ 状態○3
  「はい」 (33%)　「えー」 (16%)　以下 接続詞「けれども」「え」

「えーと」

- 状態○3⇒ 状態○3
  「え」 (8%) 「ま」 (8%) 以下 「まあ」 「あの」
- 状態○3⇒ 状態○5
  「あの」 (32%) 「あのー」 (24%) 「えー」 (19%)

状態数の少ない場合と異なり、初期状態確率が最も高い状態○2での自己ループが見られない。初期状態確率が状態○2○3の 2 状態が値を持つことと考え合わせると、8 状態の場合にイニシャルノードからの遷移で表現していたものを、16 状態では 2 つの状態からの遷移で表現していると考えられる。

2. 体言について

図 C.2 に体言が出力される遷移を示す。

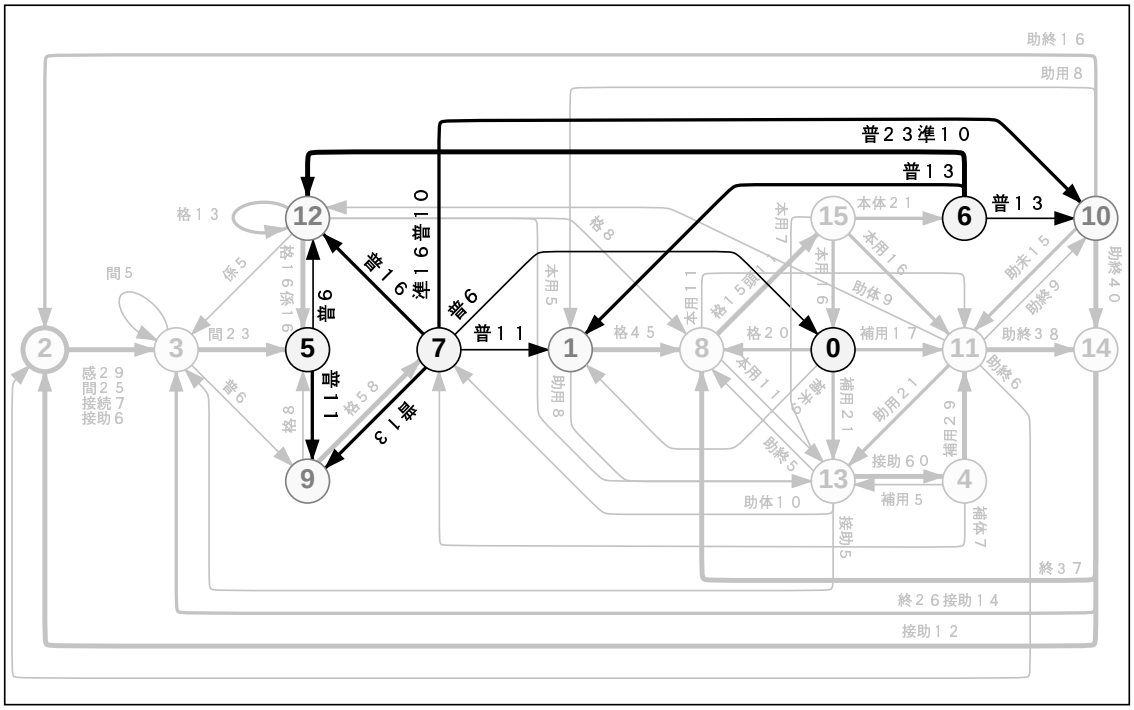


図 C.2: 体言の出力

体言は主として、○5, ○6, ○7から○1, ○9, ○12への遷移で出力されている。主に出力されているのは以下の単語である。

- 状態○5⇒ 状態○9
  「会議」 (10%) 「こちら」 (8%) 以下「先生」「そちら」

- 状態◯5⇒ 状態◯12
  「会議」「１００」「語」(接尾辞)「それ」「私」
- 状態◯7⇒ 状態◯0
  「方 (ほう)」 (31%) 「ありがとう」 (感動詞, 18%) 以下「中」「方 (かた)」
- 状態◯7⇒ 状態◯1
  「方 (ほう)」 (20%) 「事務」 (9%) 以下「中」「方 (かた)」
- 状態◯7⇒ 状態◯9
  「方 (ほう)」(15%) 「共」(接尾辞, 10%) 以下「会議」「方 (かた)」
- 状態◯7⇒ 状態◯10
  「ん」(準体助詞, 67%) 以下「わけ」「の」「方 (かた)」「もの」
- 状態◯7⇒ 状態◯12
  「方 (ほう)」 (19%) 「方 (かた)」 (10%) 以下「時間」「こと」「もの」
- 状態◯6⇒ 状態◯1
  「こと」(31%) 「ふう」 (14%) 以下 「一」
- 状態◯6⇒ 状態◯10
  「こと」(73%) 以下「ん」(準体助詞)「もの」「ところ」
- 状態◯6⇒ 状態◯12
  「こと」(38%) 「の」(準体助詞, 29%) 以下「もの」「必要」「ところ」

遷移の起点となる状態によって出力される単語が異なり、状態◯6◯7からの遷移では主として形式名詞が出力され、状態◯5からの遷移では、形式名詞以外の名詞が出力される。形式名詞を出力する遷移でも、普通名詞は出力されるが、8 状態の場合と比べ出力確率の差が大きく、普通名詞と形式名詞の分離が著しい。

状態◯5からの遷移では、出力確率が特に高い単語はなく、多種類の普通名詞の単語がこの遷移で出力され、全体の和が他の品詞よりも高くなっている。これは Ergodic HMM が学習によって、単語のカテゴリーを獲得していることを示している。

3. 活用する品詞について (品詞から見た分析結果)

図 C.3 に活用する品詞が出力される遷移を示す。4 状態の Ergodic HMM では、活用する単語が品詞・活用形に関係なく同じ遷移で出力され、8 状態の Ergodic HMM では、活用形が同じ単語が品詞に関わりなく同じ遷移に出力されていた。これに対し、図 C.3 〜図 C.3 を見ると 16 状態ではさらに細かく分類して品詞の異なるものを別々の遷移で出力

している。本動詞は状態○8,○15からの遷移で、補助動詞は状態○0,○4からの遷移で、助動詞は状態○0,○1,○13からの遷移でそれぞれ出力されている。これは、状態数が増えるにしたがい、単語の分類がより詳細になっていくことを示している。以下では各品詞ごとに出力される単語を示し、考えられる単語連鎖や特徴を述べる。

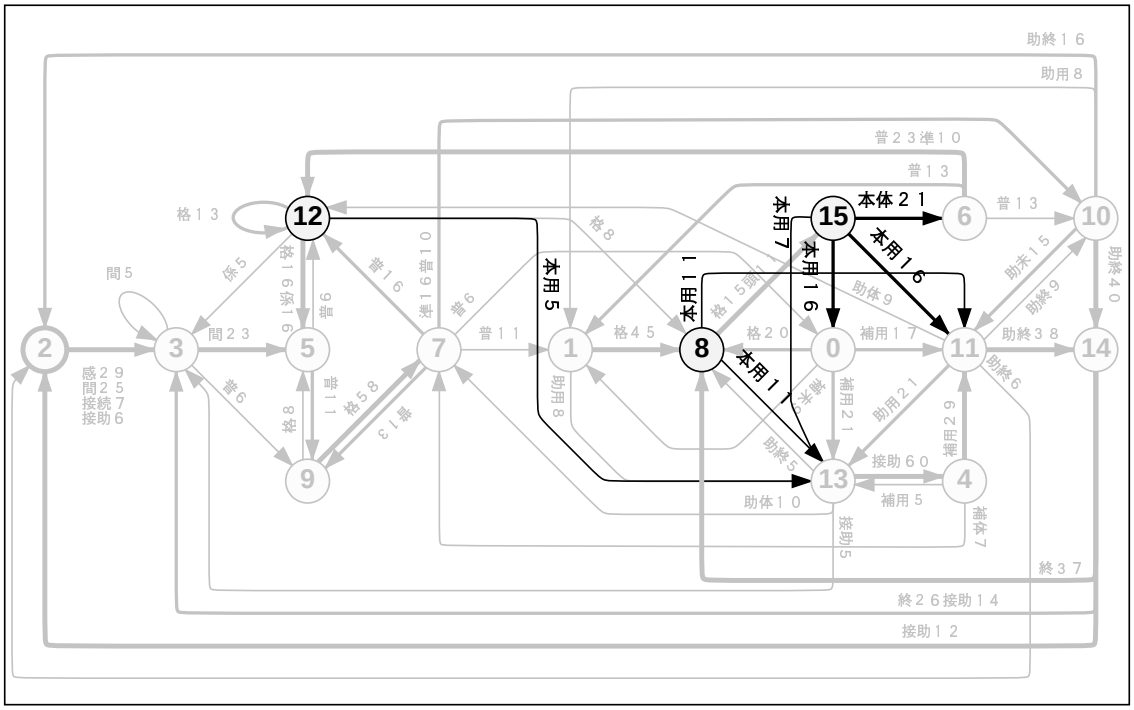


図 C.3: 本動詞の出力

(a) 本動詞

本動詞は、主に○8,○15からの遷移で出力される (図 C.3 参照)。複数の遷移で本動詞が出力されているが、各遷移によって出力される単語に違いが見られ、次のような単語が出力されている。

- 状態○8⇒ 状態○11
  「なり」 (21%) 「つき」 (10%) 以下「関し」「致し」
- 状態○8⇒ 状態○13
  「なっ」 (21%) 「し」 (14%) 以下「関し」「送っ」「書い」
- 状態○12⇒ 状態○13
  「し」 (21%) 「持っ」 (10%) 以下「でし」「送っ」
- 状態○15⇒ 状態○0
  「願い」 (28%) 「送り」 (13%) 「待ち」 (11%) 「伺い」

(8%)

- 状態○15⇒ 状態○6
  「いう」 (94%)
- 状態○15⇒ 状態○11
  「思い」(60%) 「申し」 (14%) 「し」 (12%) 以下「いい」
- 状態○15⇒ 状態○13
  「いっ」 (55%) 「思っ」 (24%) 以下「し」「なっ」「考え」

本動詞は、連用形の単語の出力が他の活用形に比べ非常に高く、言い切り（終止形）や体言への接続よりも、助動詞、補助動詞をともなうことが多いことがわかる。

意志を伝える本動詞「いう」「思う」などが状態○15から出力され、状態○8からは「〜になる」など格助詞「に」に続く本動詞が多く出力されている。

また、8 状態で見られた「お願い (いたし)〜」などの接頭辞「お」をともなう謙譲表現が 16 状態でも見られ、状態○8⇒ ○15の遷移で出力される接頭辞「お」に状態○15⇒ 状態○0で出力される本動詞が接続すると考えられる。

(b) 補助動詞

図 C.4 に補助動詞の出力される遷移を示す。

補助動詞は、○0, ○4からの遷移で出力されている。出力されている単語を以下に示す。

- 状態○0⇒ 状態○1
  「さ」 (66%) 「ください」 (11%) 以下「し」
- 状態○0⇒ 状態○11
  「いたし」 (44%) 「し」 (38%) 以下「でき」「申し上げ」
- 状態○0⇒ 状態○13
  「し」 (77%) 「ございまし」 (19%)
- 状態○4⇒ 状態○7
  「いる」 (34%) 「る」 (14%) 「いただける」 (11%) 以下「ない」
- 状態○4⇒ 状態○11
  「おり」 (51%) 「いただき」 (28%) 以下「いただけ」
- 状態○4⇒ 状態○13
  「いただい」(62%) 「しまっ」 (28%) 以下「き」「行」

本動詞と同様に連用形の出力が多く、助動詞をともなうことが多いのがわかる。状態○0から出力される単語は表記が異なるものの意味的には「する」と同じものが多い。

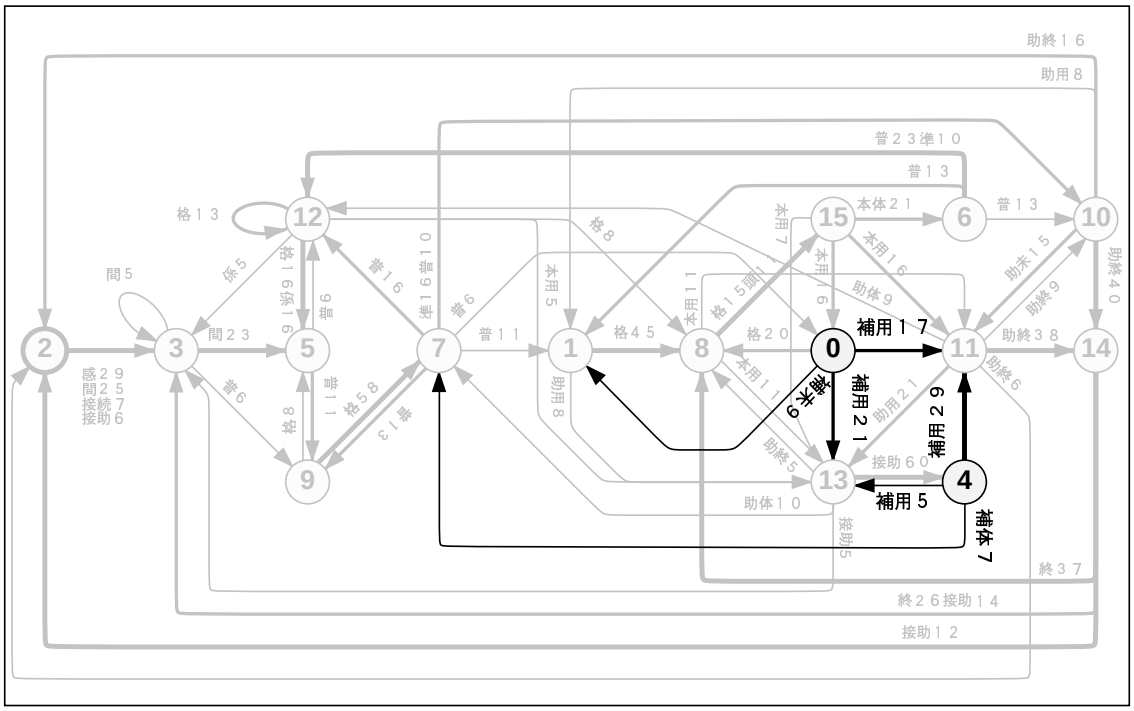


図 C.4: 補助動詞の出力

(c) 助動詞

図 C.5 に助動詞を主力する遷移を示す。

助動詞は、主に○10,○11,○13からの遷移で出力される。各遷移での単語の出力を以下に示す。

- 状態○11⇒ 状態○13「せ」 (36%) 「し」 (23%) 「れ」 (22%)
- 状態○10⇒ 状態○1 「で」 (93%)
- 状態○10⇒ 状態○2 「です」 (97%)
- 状態○10⇒ 状態○11「でしょ」 (96%)
- 状態○10⇒ 状態○14「です」 (100%)
- 状態○11⇒ 状態○2 「ます」 (99%)
- 状態○11⇒ 状態○10「ます」(98%)
- 状態○11⇒ 状態○12「ます」 (97%)
- 状態○11⇒ 状態○13「まし」 (99%)
- 状態○11⇒ 状態○14「ます」 (58%) 「う」 (41%)
- 状態○13⇒ 状態○7 「た」 (87%) 「たい」 (13%)
- 状態○13⇒ 状態○8 「た」 (67%) 「たい」 (13%)

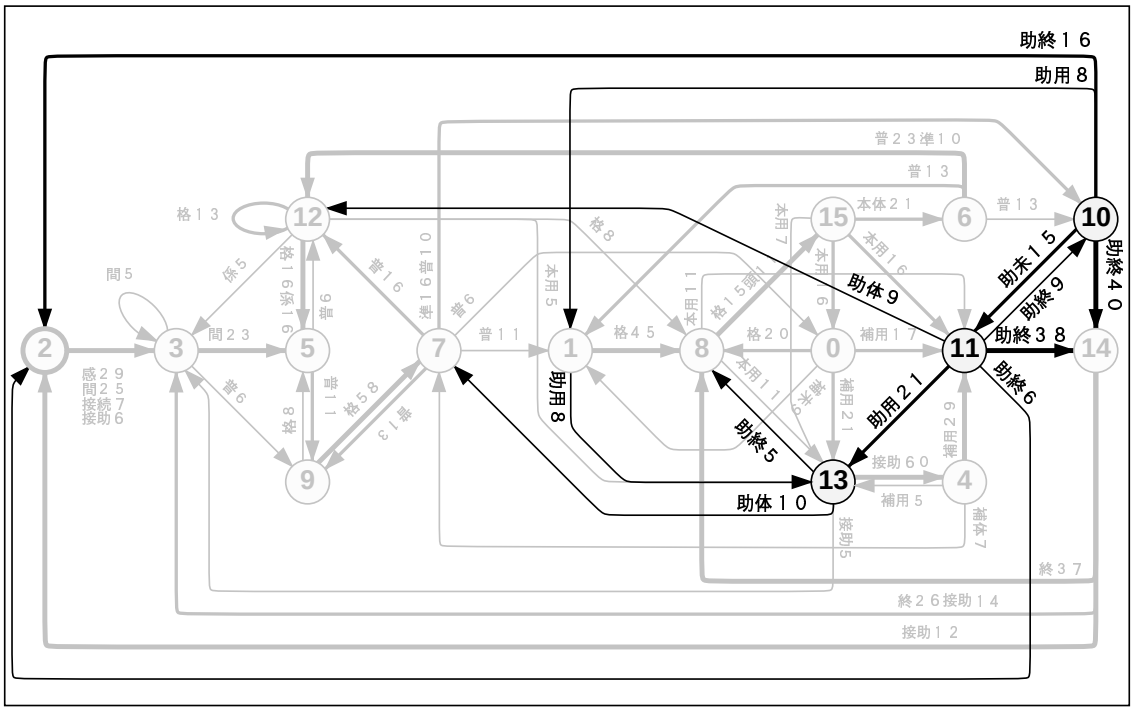


図 C.5: 助動詞の出力

遷移の起点となる状態によって全く異なる助動詞を出力しており、8 状態と同様に「です」「ます」が分離され、さらに 16 状態では「た」「たい」が他の単語から分離して出力されている (8 状態では本動詞「いう」と同じ遷移で出力されていた)。また、8 状態の場合には連体形と連用形の「ます」が同一遷移で出力されていたが、上記の出力確率から、「です」も「ます」も活用形ごとに明確に分離されていることがわかる。これらのことから、HMM の状態数が増えることによって品詞分類が細分化されることがわかる。

4. 活用する品詞について （活用形から見た分析結果）

   活用する単語の活用形ごとの出力を以下に示す。

   (a) 連体形

   図 C.6 に連体形の単語が出力される遷移を示す。

   8 状態の場合と同じく、主として体言を出力する状態○6○7への遷移で出力される。助動詞連体形の「ます」は（体言をほとんど出力していない）状態○12への遷移で出力されている。ネットワーク上には現れないが、状態○12→ 状態○2○3で接続助詞「ので」が出力されており、これに接続するものと思われる。

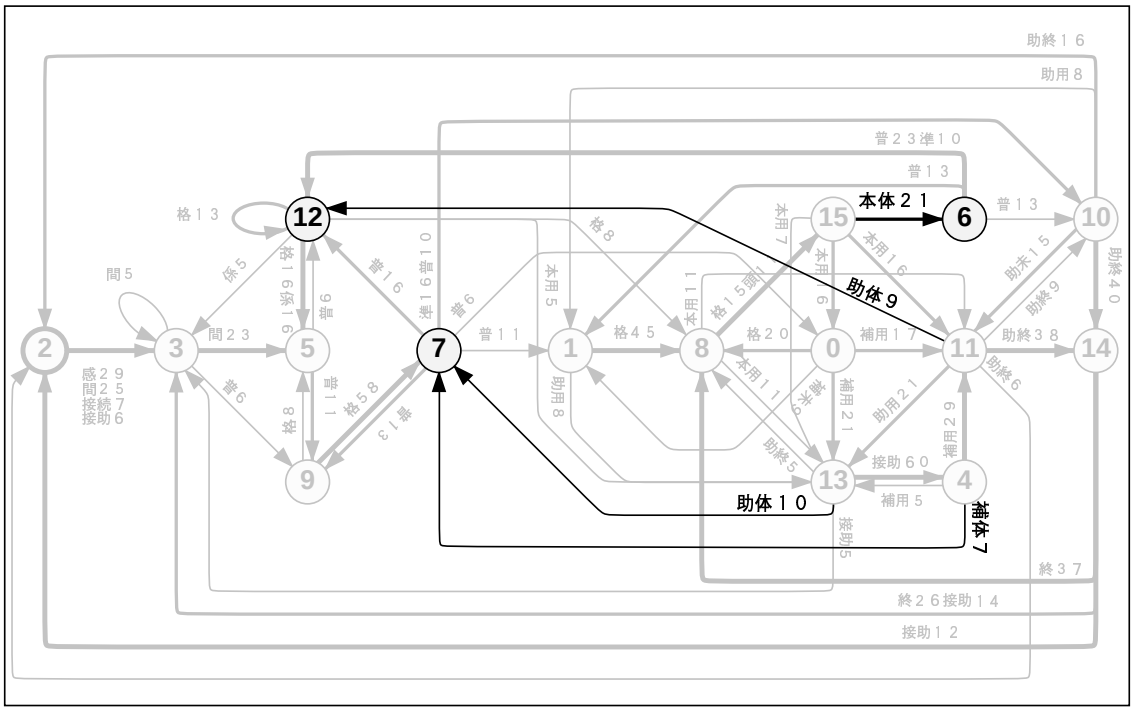


図 C.6: 連体形の出力

(b) 連用形

連用形が出力される遷移を図 C.7 に示す。

図 C.7 から、活用する品詞間の接続の様子がわかる。「(お) 願いいたします」のような「本動詞＋補助動詞＋助動詞」は状態○15⇒ ○0で本動詞を出力し、さらに状態○10○18の助動詞を出力する状態へと遷移すると考えられる。また補助動詞をともなわない「本動詞＋助動詞」のような場合は、状態○8○15⇒ 状態○10○18のように遷移すると思われる。

また、8 状態で見られた、「〜いただいて」「思って」などの「連用形＋接続助詞 て」の連鎖が、状態○4⇒ 状態○13⇒ 状態○4で見られる。

(c) 終止形

終止形を出力する遷移を図 C.8 に示す。

8 状態と同じく終止形は助動詞がほとんどであり、終助詞や接続助詞、間投詞を出力し文を終了あるいは文節頭に接続する状態○10○14○12への遷移で出力される。

5. 格助詞について

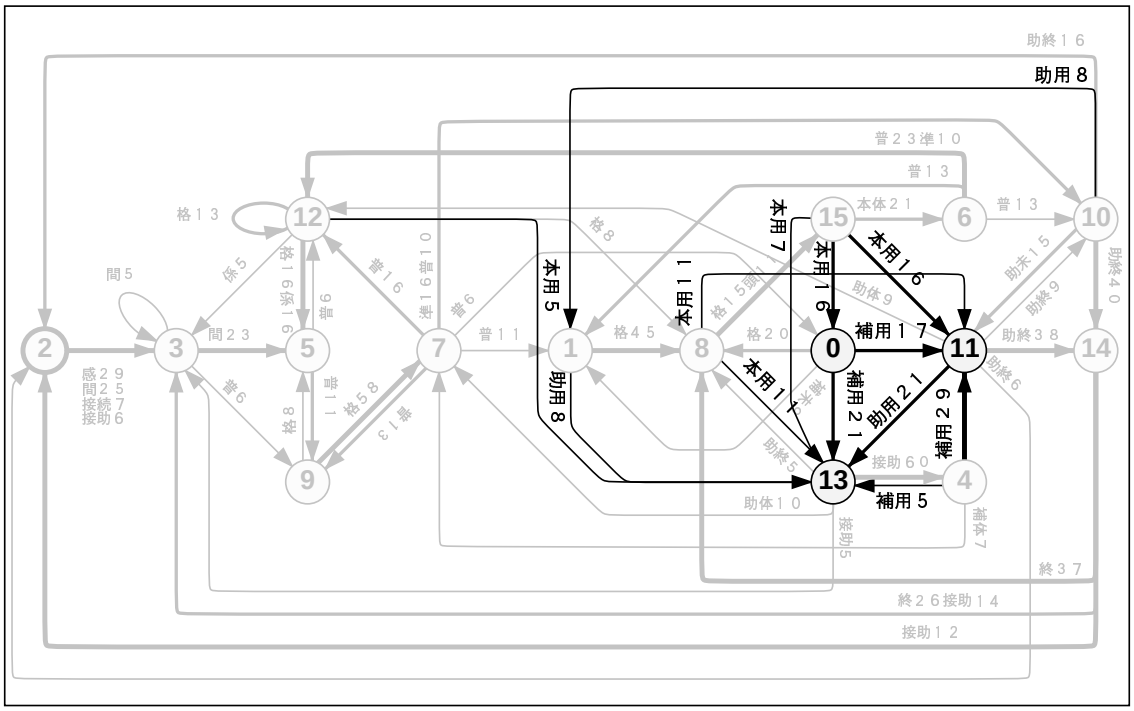


図 C.7: 連用形の出力

格助詞の出力は、8 状態 Ergodic HMM と同様に、主に体言を出力した遷移の集まる状態○8○9○12からの遷移で出力される。

- 状態○0⇒ 状態○8 「で」 (86%)
- 状態○1⇒ 状態○8 「に」 (87%)
- 状態○8⇒ 状態○15 「と」 (63%) 「お」 (接頭辞,28%)
- 状態○9⇒ 状態○5 「から」 (57%) 「で」 (23%) 以下「が」「として」
- 状態○9⇒ 状態○7 「の」 (93%)
- 状態○12⇒ 状態○3 「は」(46%) 「ので」 (接続助詞,35%) 以下「を」「と」
- 状態○12⇒ 状態○5 「が」 (55%) 「は」 (24%) 以下「は」「も」
- 状態○12⇒ 状態○8 「を」 (75%) 以下「について」
- 状態○12⇒ 状態○12 「を」 (74%) 「について」 (24%)

8 状態では同じ遷移で出力されていた「で」と「に」、「が」と「を」が別々の遷移に分かれて出力され、ここでも、状態数が増えることによって単語が細かく分類されていることがわかる。

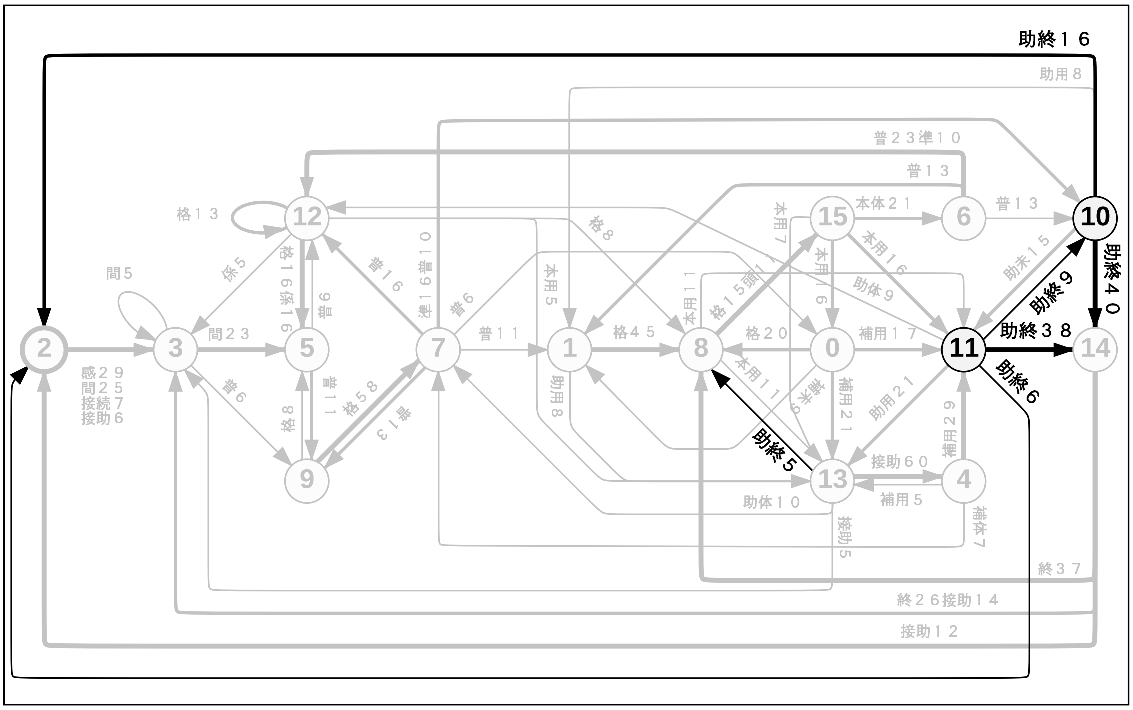

助終１６
助用８
普２３準１０
普１３
格１３
12
15
本体２１
6
普１３
10
間５
係５
格１６係１６
普６
普１６
準６普１０
普６
格８
本用５
本用１１
格１５頭１
本用７
本用１６
本用１６
助体９
助末１５
助終９
助終４０
2
3
間２３
5
7
普１１
1
格４５
8
格２０
0
補用１７
11
助終３８
14
感２９
間２５
接続７
接助６
普６
普１１
格５８
普１３
格８
助用８
本用１１
補用２１
助用２１
助終６
補用２９
助終５
9
13
接助６０
4
助体１０
補用５
補体７
接助５
終３７
終２６接助１４
接助１２

図 C.8: 終止形の出力

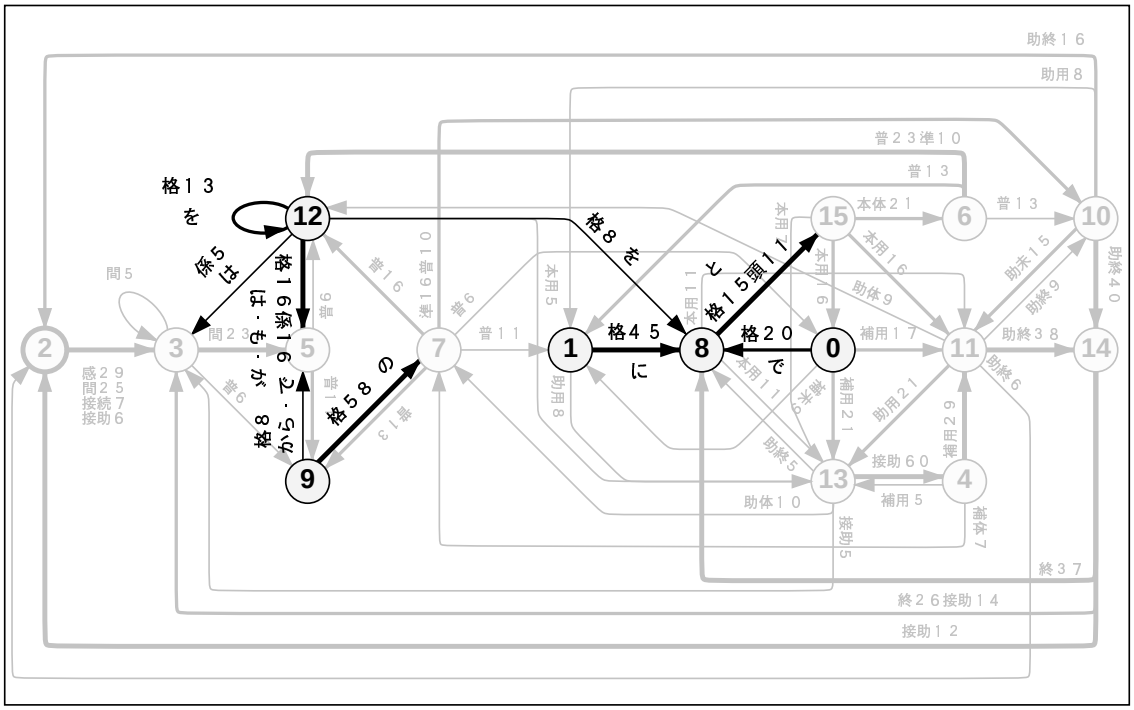

助終１６
助用８
普２３準１０
普１３
格１３
を
ならば
格１６格１７と・も・が
格８の
格８を
格１５頃１１
と
格４５
に
格２０
で
本体２１
普１３
助終４０
助終３８
補用１７
助用２１
助用２６
接助６０
補用５
助体１０
終３７
終２６接助１４
接助１２
間５
感２９
間２５
接続７
接助６

図 C.9: 格助詞の出力