音声インタフェースにおける
高度な音声言語処理モデルに関する研究

2016年3月

長野　徹

音声インタフェースにおける
高度な音声言語処理モデルに関する研究

長野　徹

システム情報工学研究科

筑波大学

2016年3月

概要

近年の高速移動体通信の発達により，音声合成および音声認識の仕組みは大きく変わりつつある．従来，端末側で行われていた音声データの処理は通信先の高性能計算機により行われるようになった．この音声技術の提供形態の変化は，サービス提供者・利用者双方に対してメリットをもたらしている．計算資源を集中化することで豊富な計算資源が利用可能となり音声認識・音声合成の高精度化が進んだこと，また，ソフトウェアの更新なしに常に最新のサービスを利用できることからソフトウェア管理の必要性がなくなったこと，そして，リアルタイムに音声データを収集できる仕組みが整ったこと等が挙げられる．リアルタイムに蓄積されるデータは，音声認識・音声合成システムの学習データとして，また顧客の生の声を含む高度な顧客情報として利用できる．本論文では，音声言語処理の観点から，これら蓄積されていく音声データの「コーパスおよび辞書を入力とする統計的音声合成フロントエンド」と「2種類の認識単位の異なる音声認識システムを用いた効率的に音声検索語検出が可能なシステム 」の2つのシステムにおける音声データの利用方法と必要とされる音声言語モデルについて論じる．前者では蓄積された音声データを音声合成の言語処理部の学習データとして用い，後者では音声認識アプリケーションの一つである音声検索語検出の検出対象として音声データが用いられる．統計的音声合成フロントエンドでは，学習データのみから読みとアクセントの付与モデルを構築できることを示し，音声検索語検出システムでは複数の音声認識器の組み合わせにより，音声検索作業の効率化を実現できること示す．

第1章にて全体の概要を述べる．第2章では，本論文の対象とする音声合成・音声認識技術の近年における応用とトレンドについて解説を行う．

第3章では，コーパスおよび辞書を入力とする統計的音声合成フロントエンド について述べる．音声合成の最終的な目的は，任意のテキストを人間と変わらない自然さで音声に変換することである．音声合成は入力テキストを解析する言語処理部と言語処理部の結果をもとに音声を合成する波形生成部からなるが，言語処理部の出力する音韻・韻律情報シンボルによって最終的に生成される音声の言い回しやアクセントといった発話スタイルが制御できる．音声認識・分析技術を用い

て学習データの音韻・韻律特徴を抽出することができれば，得られた言い回しやアクセントを用いて，特定の話者に特化した言語処理部の構築が可能になる．そこで従来ルールベースで処理されていた音声合成フロントエンドに対して，統計的言語モデルを用いて読みとアクセントの付与を試みた．実験の結果，読みとアクセントが付与されたコーパスおよび辞書を入力とした音声合成フロントエンドによる読み・アクセント付与の精度が従来手法の精度を上回った．また，本章では，蓄積された音声データを学習データとして利用することでコーパスからの音声合成システム全体の構築を可能にする 全自動構築可能な音声合成システム のコンセプトについても説明する．

さらに第4章では，この統計的フロントエンドの実用性を高めるためのアクセントクラスを用いた精度改善について述べる．学習コーパスを増やすことなく，日本語のアクセントの特徴を利用したモデルを構築する．基本的な統計的フロントエンドの枠組みに対して従来研究で利用されてきた辞書アクセントおよびアクセントの変化に関する情報を組み込むことで，アクセントの推定精度を改善する．実験の結果，同一のコーパスを利用した場合，アクセントクラスを導入することで，アクセント付与の精度だけでなく読みの精度も向上することができた．また，比較的音声の知識に精通していないユーザーでも読みやアクセント誤りを容易に修正および調整できるチューニングツールについても述べる．

第5章では音声認識における音声言語処理モデルの利用として，蓄積された音声データを効率的に活用するための 音声検索語検出の効率化 について論じる．テキストデータと同様に大量の音声データに対して検索を行いたいというニーズがある．特にコールセンターのような膨大な音声が集約される場所では，効率的な音声検索語検出が必要とされている．基本的な音声検索語検出では一般的な音声認識器である単語音声認識を用いてテキスト化し，テキスト化された音声データに対して文字列検索を行うのが一般的であるが，音声認識結果には認識誤りが含まれるため，人手による聴取作業が必要となる．この聴取作業の作業量削減のためには音声認識率の向上だけではなく，効率的に音声を聴取できる仕組みが欠かせない．本章では単語音声認識器と音節音声認識器を併用して計算量を大きく増やすことなく，効率的に音声聴取を行える仕組みと，実験結果について述べる．

最後に第6章にて全体のまとめを行うとともに，大量のデータの利用を前提とした高度な音声言語処理モデルの展望について論じる．

謝辞

本論文は筆者が日本アイ・ビー・エム株式会社東京基礎研究所ならびに筑波大学大学院システム情報工学研究科において行った研究成果をまとめさせて頂いたものです．社会人の筆者を快く受け入れて頂き，研究成果のとりまとめにあたって多くの教示，ご指導を賜りました 宇津呂武仁 教授に心から感謝いたします．ならびに，本研究を進めるにあたり，非常に有益なご指摘と助言をいただきました システム情報工学研究科 丸山勉 教授， 古賀弘樹 教授， 山本幹雄 教授， 矢野博明 准教授の皆様方に深く感謝申し上げます．また，博士後期課程の進学にあたり，再び大学院にて研究することを勧めていただいた 水谷孝一 教授， 藪野浩司 教授に深く御礼申し上げます．また 15 年以上前になりますが，修士課程の研究に際しご指導いただき，研究の基礎を教えて頂いた 星野力 筑波大学名誉教授に心から感謝いたします．また 丸山勉 先生には日本アイ・ビー・エム株式会社への就職に関して助言をいただきました．皆様方の惜しみないご支援により本学位論文をまとめ上げることができました．

社会人として研究活動を遂行するにあたり，長きに渡り日本アイ・ビー・エム東京基礎研究所の音声グループを率いられた 西村雅史 氏（現 静岡大学教授）には一企業の社会人としてまた研究者として，企業人としてのあり方から研究の進め方まで，多岐に関しご指導をいただきました．また， 森信介 氏（現 京都大学准教授）には，本論文の基礎となる音声言語処理の理解と習得に関して，自然言語処理の基礎から論文の書き方まで，多くのことを教えていただきました．心から感謝いたします．現音声グループリーダーの 立花隆輝 氏には論文の共著者として，多くの示唆をいただくとともに社会人博士課程への入学を快諾して頂きました．音声グループの 伊東伸泰 氏には音声と言語処理の両方の視点からさまざまなアドバイスをいただき，論文の共著者としてもご指導をいただきました． 市川治 氏には信号処理の専門家として，音声認識におけるチューニング方法などさまざまなご支援をいただきました． 倉田岳人 氏には音声認識モデルのモデリング方法・音声認識言語モデルの性能向上に関してご助言をいただくとともに，新しい技術の開拓に関してもさまざまな協力をいただきました． 福田隆 氏には音声認識音響モデルの専門家として，音声認識の高精度化に関して多くの技術的な示唆をいた

だくとともに，公私にわたりご支援をいただきました． 鈴木雅之 氏には実用的な音声技術の研究・開発において，さまざまなご協力をいただくとともに，研究に対してのチャレンジングな姿勢に良い刺激を受けました．グループの皆様のたゆまないご支援・ご協力のもとに，社会人として研究活動を行うことができました．あらためて感謝申し上げます．

また，現コグニティブ・コンピューティング マネージャーの 吉永秀志 氏，前マネージャーの 浅川智恵子 氏，東京基礎研究所 福田剛志 所長，ならびに 森本典繁 前所長には社会人博士への進学に関して，ひとかたならぬご支援をいただきました． 森本典繁 前所長， 丸山宏 元所長（現 統計数理研究所教授）にはマネージメントの観点から， 武田浩一 氏， 渡辺日出雄 氏， 那須川哲哉 氏， 浦本直彦 氏の各氏には，自然言語処理のプロフェッショナルとしてさまざまなご指導と暖かいご支援をいただきました．重ねて感謝申し上げます．

最後に，いついかなるときも私を支えてくれた家族に心から感謝します．

目次

表 目 次

図 目 次

第1章　序論

音声は人間にとって最も自然なインターフェースの一つであり，音声をテキスト化する音声認識技術，テキストを音声に変換する音声合成技術に関して，古くから研究が行われてきた．音声認識は1950年代にベル研究所による数字音声認識の試みから始まり，その後1990年代にディクテーションソフトウェアの発売，カーナビゲーションシステムへの搭載を経て一般に認知される技術となった．近年はスマートフォンに搭載された音声認識アプリケーションにより，その用途は大きく広がってきている．一方，音声認識と対になる音声合成は1930年代にベル研究所にて研究が始められ，アナログ音声合成からデジタル音声合成への変遷を経て，1960年代にはテキストを計算機に入力し音声を合成するテキスト音声合成システムが開発されるに至った．1980年代にはパーソナルコンピュータ上での音声合成が可能となり，その後はカーナビやアクセシビリティーへの応用に続き，近年はロボットの対話インターフェースとして身近になってきている．

従来，カーナビ等に用いられる音声認識システムおよび音声合成システムでは，静的なデータを元に音響・言語モデルを構築し，必要に応じこれらモデルを人手で変更・修正していく，という方法が一般的であった．例えば，音声認識モデルを構築する場合，音素バランスを考慮した文章の読みあげを収録，または既に収録された音声データに対して人手にて書き起こしを行い学習データを作成し，音声認識モデルを構築する．音声合成においても同様に，あらかじめ用意された文章とその読み上げを収録した音声を元に，音声に対して音韻情報の音素のアラインメントを行い，音声合成モデルを構築する．こうして作られた音声認識・音声合成モデルは組み込み用機器やパソコン等にインストールされ，限定された語彙の認識・合成を組み込み用CPUや汎用CPUで行うというのが一般的であった（図1.1左）．ところが，この数年における高速移動体通信ネットワークの普及により，音声認識・音声合成ともに端末側でデータを処理するのではなく，サーバー側でGPGPU等を用いた高度な処理を行うことが一般的となった（図1.1右）．端末からサーバーへ処理が移行することにより，利用者にとっては「ネットワークに接続されていないとサービスが利用できない」「サービス提供者の都合によりサービスが利用できない」等の問題が生じる可能性がある一方，端末の計算リソースに

依らず飛躍的に精度の向上した処理結果を利用することができるようになった．

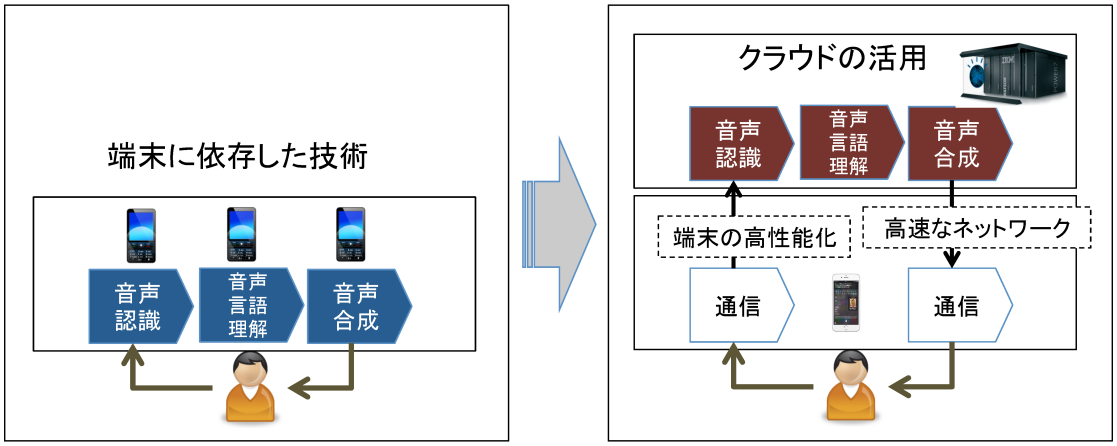

図 1.1: データ処理機能を端末に内包する形態から，クラウドコンピューティングを利用する形態への移行

サービス提供者にとっては，従来のソフトウェアを配布する形態とは異なり，任意のタイミングでサーバー側のソフトウェアを更新・改善することができるため，ソフトウェアの管理が容易になる．さらに，サービス提供者は大量のデータを全世界からリアルタイムに集めることが可能になった．言い換えると，音声インターフェースが単なる「代替インターフェース」から「新たな知識源」に変化してきた．蓄積されたデータの利用方法は大きく分けて以下の 2 つがある．

1. 学習データとしての利用
   例）音声認識モデル構築：認識率の高い音声認識モデルを構築するためには大量の音声データが必要になる．蓄積されたデータを音声認識器を用いて音声認識を行い，音声認識信頼度の高い音声箇所のみを学習データとして用いることで音声認識率を向上させる．

2. 高度な顧客情報としての利用
   例）コールセンター通話録音データからの知識発見：音声データから言語情報・パラ言語情報を抽出することで発話者の意図を推測する．「顧客は前回どのようなことを話していたか」「今顧客は怒っているか」「何を話せば喜ぶか」「どのような対話をすれば商品を購入するのか」「どのような問い合わせが多いのか」

このような背景のもと，これら知識源を活用すべく，蓄積されたデータをいかに効率的に活用できるかという観点で二つの利用方法に関して研究を行った（図

1.2）．一つは音声合成言語処理部の学習データとして音声データを用いる際に必要とされる言語モデルに関して，もう一つは音声検索語検索の効率化を目的とした言語モデルに関する研究である．音声認識および音声合成は，ともに音声波形の分析・生成を行う音響処理部と，これら音響処理部とユーザーをつなぐ言語処理部から構成される．本論文ではこれら二つの言語処理部に対して新たな音声言語処理を導入することで，蓄積されたデータの効率的な利用が可能になることを示す．

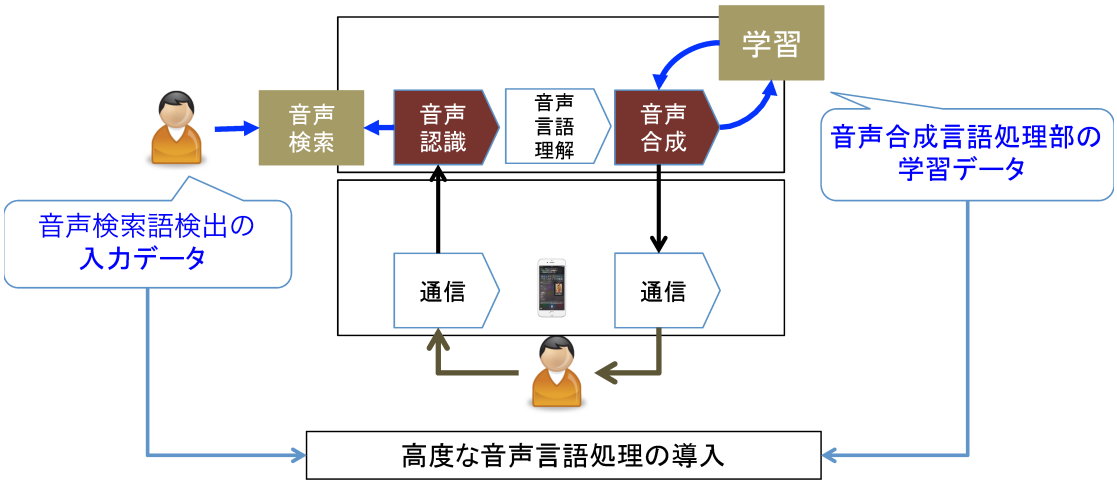

図 1.2: 本論文で取り扱う 2 つの言語モデル適用部：音声合成言語処理部および音声検索語検出

コーパスおよび辞書を入力とする統計的フロントエンド

音声合成の最終的な目的は，人と区別のつかない音質・発話スタイルによる発話を実現することである．音声合成システムによって生成される音声は，未だ「人と区別のつかない」音声品質には至っていないが，単に文字を音声に変換するだけではなく，様々な特徴的な音声合成システムを作りたいという要望がある．例えば，ユーザーの好みの声で音声を合成したい，東北弁で音声を合成したい，といったものである．このような音響的・言語的な特徴を持った音声合成システムを容易に構築することができれば，音声合成技術の応用範囲も格段に広まると考えられる．現在市販されている人型ロボットは単一の音声が用いられることが多いが，子供の声や関西弁などでの音声合成が実装されれば，ロボットの個性を表現する手段として，また個体を識別する手段として役に立つ．

実際の人間の発話から音響・言語的な特徴量を抽出し，それをもとに音声合成システムを構築することができれば，人に近い音声合成を実現できるが，そのた

めには音声から高精度に特徴量を抽出するだけでなく，抽出された特徴量を利用できる統計的な枠組みが必要となる．一般的な音声合成はフロントエンドと呼ばれる言語処理部とバックエンドと呼ばれる波形合成部から構成される．バックエンドで用いられる音響特徴量のモデル化は，一般的に自動的また半自動的に構築できるようになっていることが多く．波形重畳型の音声合成システムにおいても，発音・ポーズ位置・音素アラインメントを推定することで，音声合成に必要な音声素片データベースを自動構築できる．一方，従来，フロントエンドはルールに基づく処理が基本となっており，フロントエンドと言語モデルはルールに精通した作成者によって作られた固定した単一のモデルが用いられることが多かった．発話者の特徴を音声合成システムに反映させるためには，機械学習に基づいた統計的なフロントエンドの開発が必須である．そこでコーパスおよび辞書を入力とする統計的フロントエンドにより，ルールによらない汎用的な枠組みによる音声合成システムを実現する．

実験では，人手によって $\langle$ 単語境界, 品詞, 読み, アクセント $\rangle$ の 4 つ組が付与された 8,800 文の学習コーパスを用いて言語モデルの構築を行い，従来手法であるルールに基づきアクセント句およびアクセント核を決定する手法，逐次的に読みおよびアクセントを決定する手法，との比較を行った．

さらに実験結果を詳細に検討したところ，正解の 4 つ組を持つ単語または履歴を含む単語列が学習コーパス中に存在しないという問題があることが分かった．しかしながら，同じ表層をもつ単語であっても日本語のアクセントは文脈によって変化するため，学習コーパスから単語の取り得るアクセント型を全て収集することは一貫性およびコストの面からあまり現実的ではない．そこで，学習コーパスを増やすことなく，日本語のアクセントの特徴を利用したモデルを構築する．4 つ組を 1 つの予測単位とする単語 N-gram モデルでは，アクセントが文脈によって大きく変化するため学習コーパスおよび辞書中にテストコーパス中の出現単語を直接的に網羅することは期待できない．そこで,「学習コーパス中に現れないアクセント型を辞書を用いて列挙しそれらのアクセント型を持つ単語に適当な確率を割り当てる」モデルを作る．テストコーパスに含まれる単語が，学習コーパスに存在しない場合，コーパスに出現しない語のアクセントのふるまいを，アクセント特徴が同一である既知語の統計モデルから予測する．

2 種類の認識単位の異なる音声認識システムを用いた効率的に音声検索語検出が可能なシステム

音声認識精度の向上に伴い，様々な場面で音声認識が用いられるようになった．一般的なスマートフォン等に用いられる音声インターフェースとしてだけではな

く，企業のバックエンドシステムにおいても音声認識が用いられるようになってきている．例えば，企業内で音声が集約されるコールセンター業務においても音声認識技術が用いられている．コールセンターにおけるコールモニタリング業務では，大量の音声通話の中から特定の単語や不適切な発言等をチェックすることで，コールセンターの品質向上やコミュニケータ（オペレータ・販売営業員）の評価を行っている．従来は対象音声データをサンプリングし，エキスパートによる音声聞き取りが中心であったが，近年音声認識システムを用いたコールモニタリングが実用化されており，全通話のモニタリングができるようになった．

音声データを対象にした検索は音声検索語検出ともよばれ，一般的には単語を単位とした音声認識器の出力に対してテキスト検索を行う．ただし，テキストデータを対象とした検索と異なり，音声認識誤りに起因する過検出・誤検出が問題となる．どの程度の誤りが許容されるかは業務内容によって異なるが，検出結果の再現率重視（音声認識誤りによる過検出を許容するができるだけ漏れなく検出したい），または適合率重視（できる限り正確に認識されているもののみを検出したい）といった要求がある．特に，大量の検出結果に対して人手にて聴取作業を行うような場合，正確に認識されている可能性の高い音声から聴取できると作業効率が高まる．そのため，主に適合率を高める方向でキーワード検出の性能を調整できる仕組みが望まれている．一般的には音声認識パラメータの変更，単語の出現確率を変える等の操作を行った後，再度音声認識を行えば，ある程度，再現率・適合率の調整が可能であるが，大量のデータに対してパラメータを調整しつつ再び音声認識処理を行うことは運用上困難なため，計算量にも考慮した方法が必要である．そこで 2 種類の認識単位の異なる音声認識システムを用いた効率的に音声検索語検出が可能なシステム を提案する．ここでは音声認識率の良い単語音声認識結果をキーワードの検出に用い，マッチしたキーワード区間の確からしさを計算するために，より詳細な単位で認識を行う音節音声認識を用いる．単語音声認識結果にキーワードが一致する区間のみ信頼度計算を行うことにより，計算量を大きく増やすことなく聴取作業を効率化することができる．

本論文の構成

第 2 章にて近年における音声技術応用のトレンドについて俯瞰し，本論文の対象とする音声合成・音声認識技術について一般的な解説を行う．第 3 章では，高度な音声言語処理モデルの音声合成への応用として，蓄積された音声データを学習データとして利用し，音声合成インタフェースを洗練する全自動構築可能な音声合成システムのコンセプトと，統計的日本語音声合成フロントエンドに関して行った実験について述べる．さらに第 4 章では，この統計的フロントエンドの実

用性を高めるためのアクセントクラスを用いた精度改善について論じる．第 5 章では，音声検索語検出システムの利便性を高めるための，2 種類の認識単位の異なる音声認識システムを用いた効率的に音声検索語検出が可能なシステムについて論じ，第 6 章にて全体のまとめを行うとともに，大量のデータの利用を前提とした高度な音声言語処理モデルの展望について論じる．

第2章　統計的言語モデルと音声技術

近年における音声技術の動向について述べ，確率的言語モデルである N-gram モデル，および音声合成技術と音声認識技術について以下に説明する．以降の節では，まず本研究の基礎となる N-gram モデルの定式化について確認し，生成確率の推定および，スムージングの方法について説明する．また確率的言語モデルの評価について述べる．次に音声合成システムの一般的な構成について説明し，その評価方法および関連研究に関して述べる．最後に音声認識システムについて簡潔に説明し，その評価方法について述べる．

2.1　音声技術の進展

音声技術の実社会での応用が盛んになってきている．計算機性能の向上，認識・合成方法の確立により，スマートフォンやカーナビゲーションシステムなどに音声認識・音声合成機能が搭載され一般消費者に認知される技術となってきた．人間の音声を「人間が聞き取るように」聞き取る,「人間が声を出すように」発声することは一見簡単に思えるが，技術レベルが現在のレベルに達するまで，当初想像していたよりはるかに長い時間がかかった．音声認識技術を例にとると，コマンド音声入力を目的とした孤立単語音声の認識から始まった音声認識は，自然発話の音声認識を目指し，不特定話者発話に対応した大語彙連続音声認識へ向けて音響モデル・言語モデルの改善が行われてきており，近年ではディープラーニング技術の利用により，限られた条件の下では人間に近い認識性能を示している [1]．ただ様々な条件を考慮すると大量の計算資源を投入しても未だ人間と同じレベルには達しておらず，達するにはまだ長い時間がかかると考えられている[1]（図 2.1）．

一方，計算能力の向上とは別に，3G/4G（3rd Genration/4th Generation），LTE（Long Term Evolution）とよばれる高速移動体通信が一般に普及してきた．あらゆる場所から高速な通信ができるようになり，従来端末側で行われていた様々な

[1]参考文献 [2] から引用．縦軸は音声認識単語誤り率．人間の誤り率は 2 〜 4% といわれている．読み上げ（Read Speech）に関しては人間の聞き取りに近い性能を示しているが，放送（Broadcast Speech）・会話（Conversational Speech）・会議（Meeting Speech）のような音声に関しては人間の音声認識精度には達していない．(©National Institute of Standards and Technology)

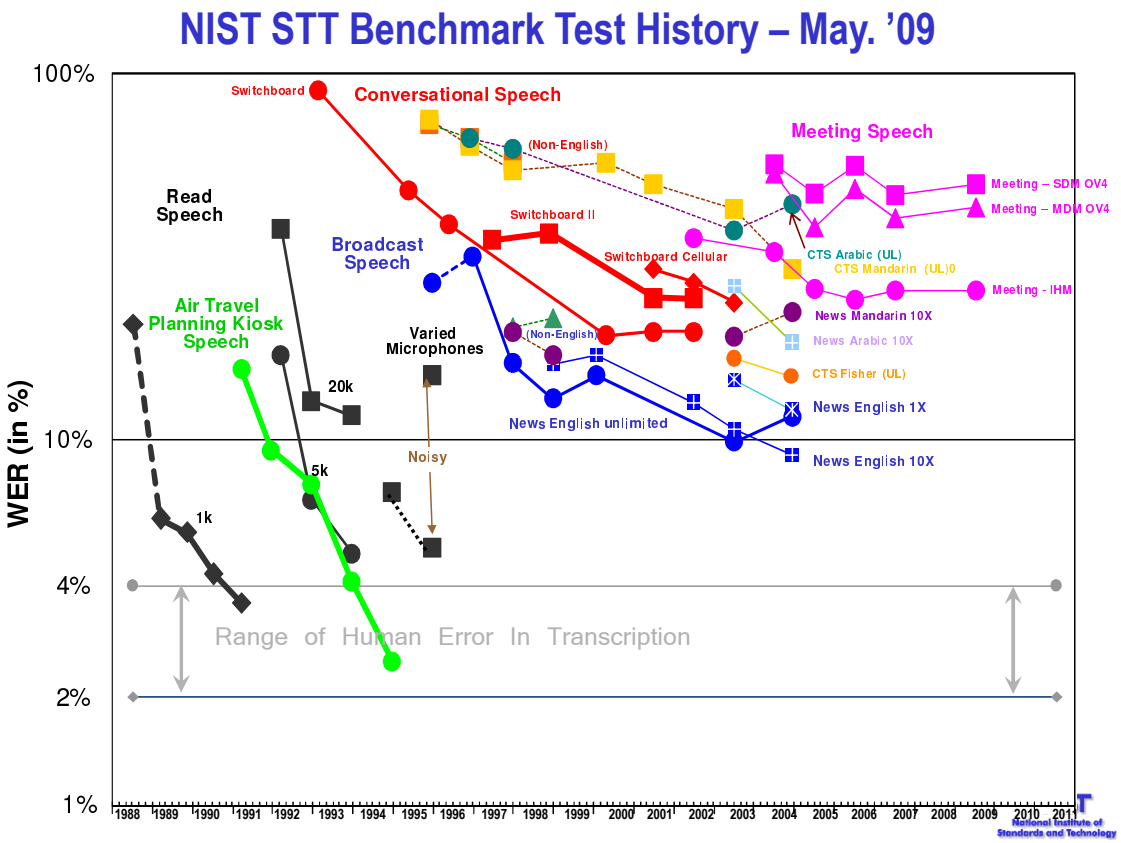

図 2.1: 音声認識精度の推移

処理を通信先のサーバーで行うことが一般的となった．このように必要な機能を必要な分だけサービスとして利用できる形態は Software-as-a-Service（SaaS）と呼ばれており，従来の音声ソフトウェアベンダーがこぞって SaaS 型の音声認識・音声合成サービスを提供するようになった [3][4][5]．2011 年には Apple 社のパーソナルアシスタントアプリケーション Siri[6] に音声認識・音声合成機能が組み込まれ，高精度な音声認識とユーモアのある対話・音声合成などを組み合わせて人気となった．2014 年にはソフトバンクロボティクス社から人型ロボット Pepper[7] が発売され，基本的な入出力インターフェースとして，音声認識・音声合成が大きな役割を担っている．クラウド型の音声認識・音声合成サービスによってサービスの利用者は，常に最新の技術を利用することができるようになった．このような音声技術の発展，計算機性能の向上，ネットワーク環境の成熟が相まって，音声技術が新たなビジネス・サービスを生み出す時代になってきている．

音声の収集とその利用

サービス提供者側には音声データが随時世界中から集まり，蓄積されるようになった．蓄積された音声データの利用目的は大きく分けて 2 つある．1 つ目は，サービスとして提供している音声認識・音声合成モデルの改善に用いるという目的である．例えば音声認識の性能向上には大量の学習データが必要となる．求められる精度にもよるが数十時間程度から数百時間の書き起こしが付与された音声データを用い，収集した音声に対して人手で書き起こしを行うことで，対象音声の発話環境（背景雑音や録音帯域，回線歪みなど）にマッチした音声認識モデルを構築することができる．2 つ目は音声データを音声認識し，音声認識結果を別のアプリケーションの入力とし，利用することであり，キーワードをクエリとした検索連動型広告，音声認識結果を音声にタグ付けする音声インデクシング，さらには大量の音声認識結果から新たな法則や知見を発見することを目的とした音声データテキストマイニング，などがある．

ソーシャル音声データ収集システム

上記いずれの目的にも可能な限り正確な学習データがあることが好ましい．音声認識・音声合成ともに，大量の音声データの書き起こしを必要とする [8]．大語彙連続音声認識のための学習コーパスには対象音声の書き起こし，音声合成には書き起こし音声の読み上げが一般的であるが，常に書き起こしが得られるわけではない．そこで，大量の学習データを作成するために，ユーザー参加型の音声データ蓄積システム [9] が提案されている．これは音声認識結果をユーザーに提示し，認識誤りを容易に訂正できる編集機能を提供することで，ユーザーは周辺の音声を聞きながら誤りを訂正することができる．また人間の知識を用いることで，音声認識モデルの作成に必要な新たな語の獲得も可能になる．

また Amazon Mechanical Turk を用いたクラウドソーシングによる音声書き起こしに関しても研究が行われている [10]．書き起こしを専門としない人が書き起こしを行い，その書き起こしを使って音声認識言語モデルを構築した場合について検討を行っている．その結果，正解書き起こしと比較して 23% の disagreement が発生するが，音声認識誤り率は，全て人手書き起こしを用いて構築した音声認識モデルを用いた場合の 39.5% から 42.0% へと 2.5% の性能悪化にとどまっている [11]．さらに [12] は Amazon Mechanical Turk の結果に対して ROVER[13] を適用し，書き起こし誤りの非常に少ない書き起こしが書き起こしが可能であるとしている．利点として，従来の書き起こしに対して最低 1/40 のコスト（費用）で書き起こしが行えること，また初期書き起こしでは書き起こし誤りが 5% 程度であるが

ROVER の適用で 1.5% ∼ 2.5% まで書き起こし誤りが減らせることを挙げている．

教師なしデータによる音声認識精度の向上

しかし，音声を正確な書き起こし学習データの作成には少なからずコストおよび時間を要するため，全自動化に関しても研究がなされている．音声認識モデルの構築に必要な音声を収録し，それらを単純に人手に書き起こすことには限界がある．たとえ，一度書き起こしを行ったとしても，様々な音響的・言語的な変化に対して対応できない．そのため全く人手を介さず学習データを構成する方法が必要とされている．既存の音声認識モデルを用いて，書き起こしの存在しない音声に対して音声認識を行い，その音声認識結果を正解書き起こしとみなして，音声認識モデルを構築する．ただし，そのままでは音声認識誤りを多く含むため，音声認識信頼度を用いて認識結果をフィルタリングする [14][15]．つまり音声認識結果の「正しそうな」部分だけを学習データとして用いる方法が用いられる．Broadband News データを対象として，(1)72 時間音声データのうち 5.6 時間の音声を学習データとして用い初期音声認識モデルを作成し，(2) 残りの 66.4 時間を音声認識し適切なフィルタリングを行った結果を再度学習データとして加え音声認識モデルを構築 (3) 別のテストデータに対して音声認識をおこなったところ，単語誤り率は 37.4% であり，これは 72 時間すべての人手による正解書き起こしを用いて音声認識モデルを作成した結果の 33.5% に対して，約 4% の精度劣化にとどまっている．

また，英語・フランス語・ドイツ語・スペイン語の音声認識器を用いて，チェコ語の音声認識を行い，その音声認識結果をもとにチェコ語の音声認識モデルを構築する，異言語間ブートストラップについても研究が行われている [16]．

2.2 言語モデル

2.2.1 N-gram 言語モデル

確率的な言語モデルである N-gram モデルは，自然言語は確率的に生成され，ある文字の生成確率は直前の N-1 文字に依存するというモデルである．確率モデルの単位は任意であり，文字を単位としたモデルは文字 N-gram ，単語を単位としたモデルは単語 N-gram と呼ばれる．文字 1-gram は個々の文字 x ($\in \mathcal{X}$) の出現確率 $P(x)$ のみから計算され，

$$P(x_1 x_2 \cdots x_h) = \prod_{i=1}^{h+1} P(x_i) \tag{2.1}$$

で表される．ここで h は文字列長であり，x_{h+1} は文末に対応する特別な記号を表す．同様に文字 2-gram は，

$$P(x_1 x_2 \cdots x_h) = \prod_{i=1}^{h+1} P(x_i | x_{i-1}) \tag{2.2}$$

で表される．ここで x_0 は文頭に対応する特別な記号を表す．これを N-gram として一般化すると，

$$P(x_1 x_2 \cdots x_h) = \prod_{i=1}^{h+1} P(x_i | x_{i-k} \cdots x_{i-1}) \tag{2.3}$$

となる．ここで，$k = n - 1$ であり，x_i $(i \leq 0)$ は，文頭に対応する特別な記号である．また，x_{h+1} は文末に対応する特別な記号を表す．

例えば，文「我輩は猫である」の文字 1-gram による生成確率は式 2.1 から，

$$\begin{aligned} &P(\text{我輩は猫である}) \\ &\quad = P(\text{我}) \times P(\text{輩}) \times P(\text{は}) \times P(\text{猫}) \\ &\qquad \times P(\text{で}) \times P(\text{あ}) \times P(\text{る}) \times P(\texttt{EOS}) \end{aligned}$$

のように計算される．また同様に文「我輩は猫である」の文字 2-gram による生成確率は式 2.2 から，

$$\begin{aligned} &P(\text{我輩は猫である}) \\ &\quad = P(\text{我} \,|\, \texttt{BOS}) \times P(\text{輩} \,|\, \text{我}) \times P(\text{は} \,|\, \text{輩}) \times P(\text{猫} \,|\, \text{は}) \\ &\qquad \times P(\text{で} \,|\, \text{猫}) \times P(\text{あ} \,|\, \text{で}) \times P(\text{る} \,|\, \text{あ}) \times P(\texttt{EOS} | \,\text{る}) \end{aligned}$$

のように計算される．

2.2.2 生成確率の推定

真の文字の生成確率 $P(x_i)$ を知ることはできないため，文字生成確率は一般的に学習コーパス中に出現する文字の頻度を数え上げる最尤推定によって行われる．文字 1-gram モデルにおける各文字の生成確率は，

$$P(x_i) = \frac{C(x_i)}{C(\cdot)} \tag{2.4}$$

で表される．ここで $C(x_i)$ は学習コーパス中での文字 x_i の出現回数を表し，$C(\cdot)$ は学習コーパス中に全ての単語の出現回数 $\sum_{x \in \mathcal{X}} C(x)$ である．文字 2-gram モデルにおける各文字の生成確率は，

$$P(x_i | x_{i-1}) = \frac{C(x_{i-1} x_i)}{C(x_{i-1})} \tag{2.5}$$

で表される．さらに文字 N-gram モデルにおける各文字の生成確率は，

$$P(x_i|x_{i-N+1}^{i-1}) = \frac{C(x_{i-N+1}^{i-1}x_i)}{C(x_{i-N+1}^{i-1})} \tag{2.6}$$

で表される．ここで x_{i-N+1}^{i-1} は文字連接 $x_{i-N+1}x_{i-N+2}\cdots x_{i-i}$ である．

N を大きくすれば長いコンテキストをもとに次の単語をの生成確率を計算することができ，N-gram モデルは精緻化されることになる．ところが文字の組み合わせは文字の種類数 $|\mathcal{X}|$ に対して $|\mathcal{X}|^N$ となり，学習コーパス中に含まれない文字の連接に対しては確率が 0 になる・計算ができなくなるといった問題を生じる．文字 1-gram の場合コーパス中に現れない文字（文字 1-gram ）があれば最尤推定される生成確率は 0 になる．また文字 2-gram の場合コーパス中に現れない文字 1-gram があれば，式 2.5 の分母が 0 になり，コーパス中に現れない文字 2-gram があれば生成確率が 0 になる．

そこで，学習コーパスに出現しない文字連接に対しても非ゼロ確率を付与するために，スムージングを行う．スムージングの方法として，定数で非ゼロ確率化を行う加算スムージング，常に低次の N-gram を利用する補間法（Interporation），高次の N-gram が存在しない時に低次の N-gram によって高次の N-gram を近似するバックオフ・スムージング（Back-off）という方法が知られている．N が大きくなるほど頻度が 0 になりやすいため，高次の N-gram をより低次の N-gram によって補完する．各手法について以下で解説する．

加算スムージング

加算スムージング（Addictive Smoothing）は確率が 0 にならないように出現回数に定数 δ $(0 < \delta \leq 1)$ を加算する．つまり未知の（学習コーパスに含まれない）事象がコンテキストによらず同確率で生起すると仮定している．加算スムージングにより修正された確率は，

$$P_{Add}(x_i|x_{i-N+1}^{i-1}) = \frac{C(x_{i-N+1}^{i-1}x_i) + \delta}{C(x_{i-N+1}^{i-1}) + \delta|\mathcal{X}|} \tag{2.7}$$

で定義される．加算スムージングはコンテキストによらず定数を用いているので以下に述べる手法と比較して性能が良くないことが知られているが，簡易的にゼロ除算を避けることができる．

線形補間

線形補間法（Linear Interporation）がよく知られている．Jelinek-Mercer Smoothing とも呼ばれ，

$$\begin{aligned} P_{Lin}(x_i|x_{i-N+1}^{i-1}) &= \lambda_a P(x_i|x_{i-N+1}^{i-1}) + \lambda_b P(x_i|x_{i-(N-1)+1}^{i-1}) \\ &= \lambda_a \frac{C(x_{i-N+1}^{i-1}x_i)}{C(x_{i-N+1}^{i-1})} + \lambda_b \frac{C(x_{i-(N-1)+1}^{i-1}x_i)}{C(x_{i-(N-1)+1}^{i-1})} \end{aligned} \tag{2.8}$$

で定義される．$\lambda_a + \lambda_b = 1$ であり，一般的にはより低次のモデルと補間することで，スムージングを行う．補間係数 λ_a, λ_b は EM アルゴリズムによって求められる．補間を行うモデルは 3 以上であっても良く，単に低次のモデルでなく，異なるスムージングを行うモデル同士を補間することもある．

2.2.3 スムージング

バックオフ・スムージンズは学習コーパスに出現する N-gram の確率の一部を学習コーパスに出現しない N-gram の確率として割り当てる（ディスカウント）．一般的なバックオフ・スムージングは，

$$P_{Bak}(x_i|x_{i-N+1}^{i-1}) = \begin{cases} \lambda(x_{i-N+1}^{i})P(x_i|x_{i-N+1}^{i-1}) & \text{if } C(x_{i-N+1}^{i}) > 0 \\ (1-\lambda_0(x_{i-N+1}^{i-1}))\ \alpha P_{Bak}(x_i|x_{i-(N-1)+1}^{i-1}) & \text{else if } C(x_{i-N+1}^{i-1}) > 0 \\ P_{Bak}(x_i|x_{i-(N-1)+1}^{i-1}) & \text{otherwise} \end{cases} \tag{2.9}$$

で計算される．式中の λ はディスカウント係数と呼ばれる．また $\lambda_0(x_{i-N+1}^{i-1})$ は，

$$\lambda_0 = \sum_{x \in \mathcal{X}} \lambda(x_{i-N+1}^{i-1}x)P(x|x_{i-N+1}^{i-1}) \tag{2.10}$$

で表される x_{i-N+1}^{i-1} に対する λ の総和である．N-gram が学習データ中に存在する場合は最尤推定により計算された確率値よりもディスカウントされた（小さな）確率を割り当て，残りの確率をより低い $N-1$-gram モデルに割り当てる．$N-1$-gram も学習データに存在しない場合は，再帰的に繰り返すことで非ゼロの確率が得られる．なお，α は確率の総和を 1 にするための正規化係数であり，

$$\alpha = \frac{1}{1 - \sum_{C(x_{i-N+1}^{i})>0} P_{Bak}(x_i|x_{i-(N-1)+1}^{i-1})} \tag{2.11}$$

として求められる．現在，この一般的なバックオフ･スムージングを改良したKneser-Neyスムージングの性能が良いとされており，

$$P_{KN}(x_i|x_{i-N+1}^{i-1}) = \begin{cases} \frac{\max\{C(x_{i-N+1}^{i})-D,0\}}{C(x_{i-N+1}^{i-1})} & \text{if } C(x_{i-N+1}^{i}) > 0 \\ \gamma(x_{i-N+1}^{i-1})\ P_{KN}(x_i|x_{i-(N-1)+1}^{i-1}) & \text{otherwise} \end{cases} \tag{2.12}$$

で表される．一般的な最尤推定との違いは，コンテキストの文字の出現回数ではなく文字の種類数を重視することにある．本論文では，線形補間およびKneser-Neyスムージングを用いた．

2.2.4 言語モデルの評価

確率的言語モデルである N-gram モデルの評価尺度としてクロスエントロピーおよびパープレキシティが広く用いられている．エントロピーは確率分布 $P(x)$ に対して，

$$H_0(P) = \sum -P(x)\ \log_2 P(x) \tag{2.13}$$

のように定義される．一様分布 $P(x) = \frac{1}{|\mathcal{X}|}$ のときに最大となり $0 \leq H_0 \leq 1$ である．これに対して，テストコーパスの長さ n の文字列 $x_1x_2\cdots x_n$ に対するクロスエントロピーは，

$$H(P, \boldsymbol{x}) = -\frac{1}{n}\sum \log_2 P(x_1x_2\cdots x_n) \tag{2.14}$$

と定義される．クロスエントロピーの計算はテストコーパスに依存し，テストコーパスをどれだけよくモデル化したかということを評価する指標になる．パープレキシティPPはエントロピーおよびクロスエントロピーから一意に変換することができ，特にテストコーパスに対するパープレキシティは，テストセットパープレキシティともよばれ，

$$PP = 2^{H(P,\boldsymbol{x})} \tag{2.15}$$

で計算される．これはある文字 x_{i-1} に対して平均 2^H 種類の文字 x_i が後続することを示している．後続する文字が少ないほうがテストデータをよりよくモデル化できていると言える．既存研究 [17] によると日本語の場合，単語単位で N-gram モデルを作成した場合の文字あたりのクロスエントロピーは 4.3 程度，つまり $2^{4.3} \sim 19.7$ であり，およそ 20 個の文字が後続する．

計算機上で表される文字の集合は Unicode 等の文字コード規格により定められており，たかだか数万程度の文字集合にて表現可能であるが，文字の連接である単語の集合を一意に定めるのは困難であるため，未知語の問題が生じる．クロスエ

ントロピーの計算において，未知語に対しては特別な記号と確率を割り当て，この未知語モデルをもとに確率が計算されることが一般的である．また，クロスエントロピーおよびパープレキシティとは別に未知語に関する指標として，被覆率（カバレージ）が用いられることもある．未知語とは言語モデルに含まれない語の全てを指すこともあるが，既知語 : 言語モデルに含まれる語，未知語 : テストコーパスに出現するが既知語の集合に含まれない単語，と定義すると，単語被覆率は，

$$\text{単語被覆率} = \frac{|\,\text{既知語}\,| - |\,\text{未知語}\,|}{|\,\text{既知語}\,|} \tag{2.16}$$

と定義される．

2.3 音声合成

2.3.1 音声合成システムの概要

語彙の制限のない任意のテキストを入力として，人間の発する音声と同等の音声を出力することが，規則音声合成 [18] の一つの最終目標である．図 2.2 に規則音声合成システム（以下，単に音声合成システム）の処理の概要を示す．

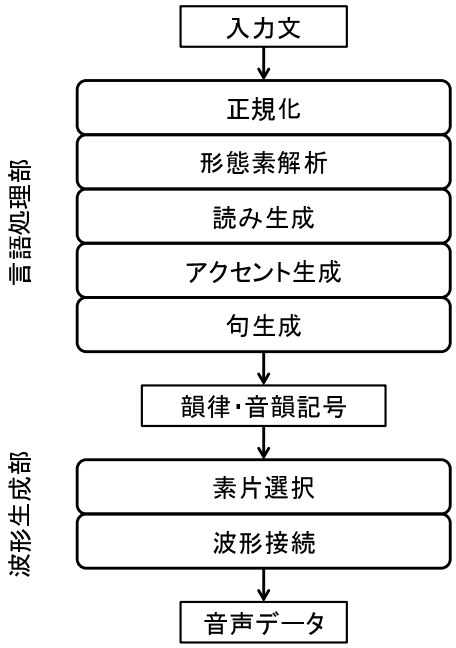

図 2.2: 音声合成システム

一般的に，テキストつまり文字列を入力として音声を合成するためには，主に

自然言語処理を用いた言語処理部と，信号処理を用いた波形生成部を組み合わせて実現される．言語処理部においては，任意の入力テキストに対し，正しい音韻情報と韻律情報を生成することが，自然な合成音声を得るための基本的な要件である．この言語処理部で生成された音韻情報と韻律情報を元に，波形生成部では音声素片を組み合わせ・接続することで，音声が合成される [19]．現在では，任意のテキストを入力としていかに人間と変わらない自然さで合成できるかが研究課題となっている．言語処理部は音声合成フロントエンドとよばれ，入力されたテキストを言語的に解析し，読みやアクセントといった音韻，韻律情報を付与する．言語によって必要な処理は異なるが，分かち書きされていない日本語や中国語の場合，単語分割も同時に行うことが多い．一方，波形生成部は音声合成バックエンドとよばれ，読みやアクセント情報を元に基本周波数等のパラメータを決定し，音声波形を生成する．

言語処理部が出力するシンボル列の例を図 2.3 に示す．入力文「冬は過ごせない」に対して形態素解析・読み付与・アクセント付与・句生成の結果，分割された 6 個の単語 $w_1 \cdots w_6$ に対してシンボルが付与される．日本語の場合，アクセントは高低 2 値アクセント [2] を取ることが多い．

言語処理部	シンボル	*w1*	*w2*	*w3*	*w4*	*w5*	*w6*
形態素解析	単語	冬	は	過ご	せ	な	い
	品詞	名詞	助詞	動詞	助動詞	助動詞	語尾
読み付与	読み	fu | yu	wa	su | go	se	na	い
アクセント付与	アクセント	低 | 高	低	低 | 高	高	低	低
句生成	アクセント句	*ap1*		*ap2*			
	イントネーション句	*pp1*					

図 2.3: 言語処理部の出力するシンボル列

アクセント句とは，発話する際のひとつのまとまりであるとされ，日本語標準語のアクセントの場合，各アクセント句に対してアクセント型が定義される．T 型アクセントで定義されるモーラ長 N ののモーラ列 $m_1 m_2 \cdots m_N$ の高低アクセントは以下のようになる．

[2] 本来連続値をとるアクセントの 2 値への離散化には議論の余地がある．例えば，日本語の単語が連鎖した場合，ここで示した高と低 の中間に位置する副（次）アクセントが発生することがある．より質の高い音声合成を出力するためには，離散化の手法についても考慮すべきであろう．

$$m_i = \begin{cases} 0\text{型} \begin{cases} 0 & \text{if } i = 1 \\ 1 & \text{otherwise} \end{cases} \\ 1\text{型} \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{otherwise} \end{cases} \\ T\text{型} \begin{cases} 1 & \text{if } 2 \leq i \leq T \text{ , } T < N \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{2.17}$$

例えば，図2.3において「冬」のアクセントは「fu 低, yu 高」なので，アクセント型は2モーラ0型となる．アクセント句 ***ap*1**「冬は」のアクセント型は3モーラ2型となる．同様に ***ap*2**「過ごせない」のアクセント型は5モーラ3型となる．イントネーション句は呼気による区切りであり，人間が1回の呼吸で発話する単位に相当する．イントネーション句区切りは，波形合成時に前後のコンテキストに依存した短い無音区間に変換される．波形生成部は，これらシンボルを入力として，音声を合成する（図2.4）．

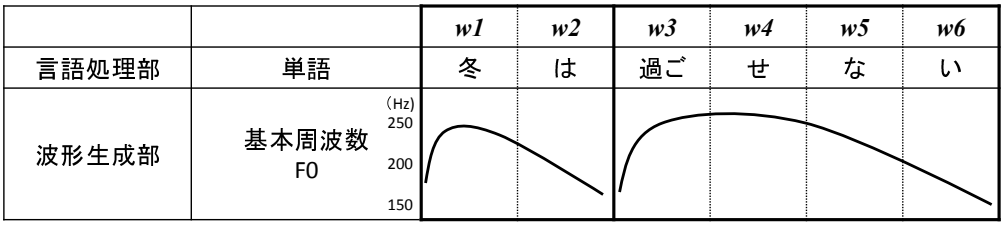

図 2.4: 言語処理の出力シンボル列に対する波形生成

2.3.2 音声合成の評価

最終的な音声合成の品質は．人間の聴取作業による主観評価で行う．ただ被験者間によるスコアのばらつきもあり，統計的に有意である統計結果を得るためには相当の音声を複数の被験者によって評価する必要があり効率が悪い．また音声合成の声質などにも影響を受けるため，異なる音声合成システムにおける比較評価が難しい．そのため，声のトーン・声質などに関連するパラメータを合成音から抽出して元音声と比較する手法が取られることが多い．また，言語処理部の出力は離散化された記号列であるので，正解記号列との比較を行うことで評価できる．以下に一般的な評価基準を示す．

- 主観評価
  - 平均オピニオン評点（Mean Opinion Score : MOS） [20]
    複数の被験者による 5 段階評価（非常に悪い 1 悪い 2 普通 3 良い 4 非常に良い 5）の算術平均
- 客観評価（言語的指標）
  - 一致率（再現率・適合率・誤り率・精度など）

- 客観評価（音響的指標）
  - 非周期成分歪み（Aperiodicity distortion） (db)
  - 有声・無声エラー率（Voiced/Unvoiced error rates） (%)
  - メルケプストラル歪み（Mel-cepstral distortion） (db)
  - 対数 F_0 残差（RMSE in $\log F_0$）

本論文では評価基準は 2.4.2 節と同様に出力単位ごとの誤り率（モーラ誤り率）で評価を行う．

2.3.3 研究動向

音響的な面での性能向上としては，ディープニューラルネットワーク技術を用いた音声合成システムの研究が行われている．特にリカレントニューラルネットのユニットを任意の時間記憶が可能な LSTM（Long Short-Term Memory）に置き換えた LSTM-RNN（Long Short-Term Memory in Recurrent Neural Networks）を用いた音声合成システム [21] が提案され良い性能を示している．また音質変換 [22] などでも LSTM が用いられる．また言語処理部においては，従来はルールベースによる，人名辞書から名前読み付与規則を抽出するアルゴリズム [23]，SVM による日本語テキストにおけるアルファベット文字列の読みクラス分類 [24]，などの研究があるが，近年は言語処理部にもニューラルネットが導入され，LSTM を用いた英語の読み付与（Grapheme-to-Phonemes: G2P）[25] などの研究が行われている．

2.4 音声認識

2.4.1 音声認識システムの概要

音声データからテキストを推定することが，音声認識の目標である．一般的な音声認識器の構成を図 2.5 に示す．マイクで録音された音声信号は 8KHz（125μsec/サンプル），16KHz（62.5μsec/サンプル）といったサンプリングレートで量子化され，入力音声データとなる．音声認識器は特徴抽出部とデコーダ部からなり，前段では人間の発声機構からくる音声の特性や聴覚特徴を考慮しつつ，計算機で利用しやすい音響特徴量を抽出する．後段では 20msec 程度のフレームに区切られた信号に対応する特徴量をもとに尤もらしいテキストを出力する．

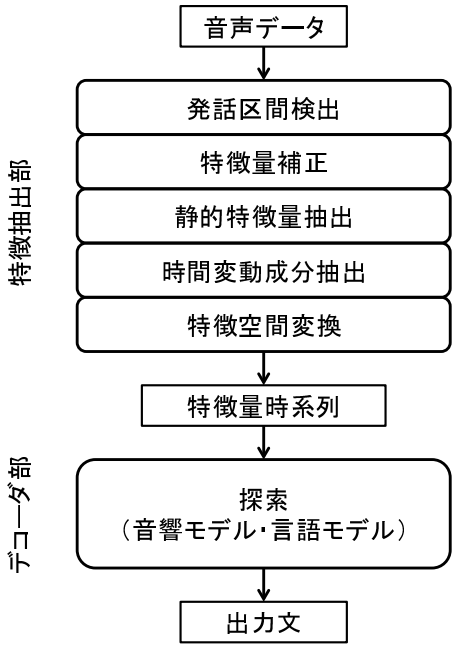

図 2.5: 音声認識システム

特徴抽出部は主に信号処理であり，フレームごとにパワースペクトル抽出し，MFCC や LPC メルケプストラムといった静的特徴量が計算される．これら静的特徴に対して時間方向の差分，もしくは線形回帰にて短時間変動の動的特徴量を計算する．さらにこれらを LDA（線形判別分析）などの手法により別の空間に写像したのち，デコーダの入力とする．デコーダは，フレームごとに数十次元の特徴量ベクトルが時間軸方向に連なったデータを入力とし，フレームごとに対応する音素・単語列を決定する．

このデコーダにおけるプロセスは，長さ T フレームの入力特徴量 $\boldsymbol{X} = \{\boldsymbol{x}_t\}$ $(t =$

$1 \cdots T$)，から $\boldsymbol{X}$ に対応する単語列 $\hat{W} = \{\hat{w}_i\}$ を推定する問題として定式化され，

$$\begin{aligned} \hat{W} = \text{argmax } P(W|\boldsymbol{x}) &= \text{argmax } \frac{P(\boldsymbol{X}|W)P(W)}{P(\boldsymbol{X})} \\ &= \text{argmax } P(\boldsymbol{X}|W)P(W) \\ &= \text{argmax } \log(P(\boldsymbol{X}|W) + \log P(W) \end{aligned} \tag{2.18}$$

$\log(P(\boldsymbol{X}|W)$ を計算するためのモデルを音響モデル，$\log P(W)$ を計算するためのモデルを言語モデルという．

2.4.2 音声認識の評価

音声認識の精度は単語誤り率（Word Error Rate: WER）または文字誤り率（Charactor Error Rate: CER）によって評価されることが多い．英語のような単語境界がおよそ明確な場合は，正解単語列に対してどれだけの単語数が誤った（正解したか）を評価する単語誤り率が用いられるが，単語境界に曖昧性が大きい日本語の場合，文字誤り率が用いられることもある．また送り仮名（例：振り込み，振込み，振込）の自由度の大きい日本語の場合，カナ読みでの誤り率などが用いられることもある．本論文では音声合成のフロントエンドにおいてはモーラ単位のモーラ誤り率（MER）を用い，音声合成の認識結果においては文字単位の文字誤り率（CER）を用いた．誤り率の算出単位によらず計算方法は同一であり，文字誤り率の場合，以下の式で計算される．

$$CER = \frac{S : \text{置換誤り} + D : \text{削除誤り} + I : \text{挿入誤り}}{N : \text{文字数}} \tag{2.19}$$

例えば，入力された音声（人手で音声を聴取して得られた正解文字列）が以下で与えられ，

東	京	特	許	許	可	局

音声認識器の出力した結果が

東	京	お	特	許	可	曲

だった場合，置換誤り:「局 $\rightarrow$ 曲」，削除誤り:「許 $\rightarrow \emptyset$」，挿入誤り:「$\emptyset \rightarrow$ お」，という 3 文字の誤りが発生している．この場合，$CER = \frac{S:1+D:1+N:1}{N:7} = 3/7 = 0.42 = 42\%$ となる．

第3章　全自動構築可能な音声合成システム

ユーザーの意図した話し方での発話を容易に実現できる音声合成システムを構築したい．テキスト音声合成システムでは，一般に入力テキストに対して言語処理部（音声合成フロントエンド）において，読みやアクセント，呼気段落等を出力し，これらの音韻，韻律情報を元に，後続する波形生成部で音声波形を生成するが，入力者の言い回しやアクセントといった特徴を反映することは難しかった．従来，フロントエンドはルールに基づく処理が基本となっており，フロントエンドと言語モデルはルールに精通した作成者によって作られた固定した単一のモデルが用いられることが多かった．しかし，さらなる精度の向上，また特定の分野に特化したチューニングなどが必要とされることもあり，コーパスおよび辞書のみを入力とする統計的フロントエンドを作成する．

3.1　はじめに

ユーザーの意図した読みやアクセントを実現する言語処理部を構築するには，音韻，韻律情報を高い精度で指定できるフロントエンドと，人手による情報の付与を効率良く行えるシステムの開発が必要である．本章では，入力テキストに対し，最も基本的な音韻情報と韻律情報である読みとアクセントを付与する問題を取り扱う．日本語の場合，入力テキストは一般的に漢字仮名交じり文であり，複数の読み候補から正しい読みを推定する必要があるとともに，その読みに対して正しいアクセントを推定する必要がある．従来，日本語テキストに対しては，形態素解析・読み付与・アクセント句決定・アクセント核決定，という手順を段階的に行うことで，読みとアクセントを決定していたが，本研究では，表層（単語境界)・品詞・読み・アクセントの 4 つ組を 1 つの単位とみなし，N-gram モデルを用いて同時に推定する．実験では，従来手法である (1) ルールに基づきアクセント句およびアクセント核を決定する手法，(2) 逐次的に読みおよびアクセントを決定する手法，との比較を行った．その結果，提案手法による読みおよびアクセント付与の精度が従来手法の精度を上回った．

3.2 全自動構築可能な音声合成システム

3.2.1 システム概要

図3.1は全自動構築可能な音声合成システムの概要である．初期データ作成フェーズにおいては一般的な波形重畳型テキスト音声合成システムと同様に，収録された音声データを元に音声認識，特徴抽出を行い，音響パラメータを推定，モデルの構築を行う．同時に，対象話者言語コーパスも構築する．対象話者コーパスは，特徴的な言い回しやアクセントを実現させたい文章や，誤りの少ない読み上げを実現したい分野の文章を想定している．特徴抽出および音声認識には誤りも含まれるため，初期の段階においては網羅的に手作業にて修正を行う．フロントエンドに関しては，この対象話者の言語コーパスに加えて，予め作成してある一般的な言語コーパスと混合し，合せてランタイム処理用の言語モデルとする．対象話者のみで言語モデルを作成できれば，話者性の再現可能性という面では好ましいが，未知語も含めた全ての語彙，文脈を網羅する音声コーパスを録音し，正確に読みやアクセントを認識することは不可能である．そのため一般的言語モデルにおいて，固有名詞辞書や数字の読み上げなどを含む一般的な言い回しを含むコーパスを用意し，広範囲の語彙を網羅する．また，言語コーパスにはアクセント句区切りも含まれ，これらの情報からアクセント句等の句構造を決定するための句決定モデルも生成する．

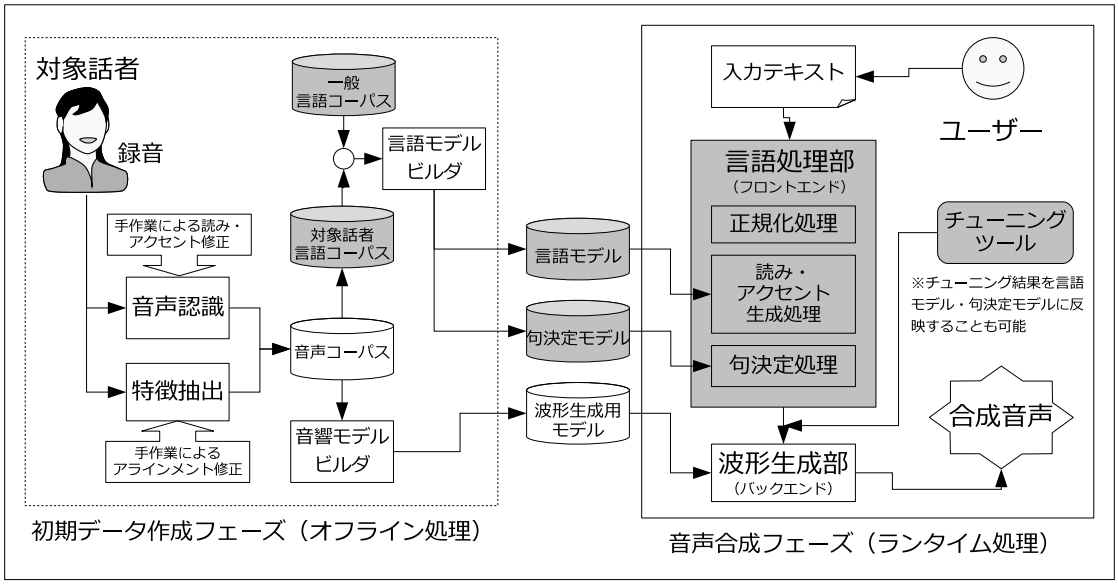

図 3.1: 全自動構築可能な音声合成システム：システム構成図

3.2.2 音声合成フロントエンド

バックエンドで用いられる音響特徴量等のパラメータやモデルは，一般に自動的または半自動的に構築できるようになっていることが多く，音声認識の技術を応用し，正確な発音，ポーズ位置，ならびに音素アライメントを推定することで音声素片 DB を自動構築可能な波形重畳型のテキスト音声合成システムの開発が行われている．この手法は英語，ドイツ語，フランス語，スペイン語，イタリア語などの西欧言語に加えアラビア語，韓国語，日本語にも適用されてきた [26][27]．素片の選択方法に加え，音素継続長，ピッチ，強度といった韻律制御のモデルも決定木に基づく方法に統一されており，自動構築が可能となっている．一方，フロントエンドはいずれの言語もルールに基づく処理が基本となっている．日本語についても過去に開発したフロントエンド部を逐次改良しながら使用していたが，高品質な合成音声を追求するにつれ，フロントエンドの読み及びアクセント推定の誤り率の高さが問題となった．また，適用分野としては汎用的なテキスト読上げの要求も残るが，特定分野に特化し，さらに肉声に近い精度と品質を達成できた方がよいとされるケースも多い．

テキストのみを入力として，正しい音声を生成するためには，テキストの構成要素である単語だけでなく，単語列として表される文全体が，言語的・音声的にどのような性質であるかを知る必要がある．日本語テキストを発話する際に，読みが重要であることは言うまでもないが，アクセントも同様に重要な要素である．例えば,「庭には二羽，鶏がいる.」（にわにはにわにわとりがいる）という文には「にわ」という音が 4 回出現するが，このいずれかのアクセントを誤ると，単に発話の自然さに欠けるというだけではなく，文として意味が通じない，または違う意味になる．

日本語テキストに対して音韻・韻律情報を付与する手法としては，従来，形態素解析・読み付与・アクセント句境界決定・アクセント核決定，という手順を段階的に行うことで，読みとアクセントを決定する手法が提案されている [28]．また，他言語においても，テキスト処理技術に関しては，イントネーション句・アクセント句，といった階層構造を仮定し，各階層構造をルールまたは決定木等の統計的な手法を用いてトップダウンに決定し，読み及びアクセントを決定する手法 [29] が提案されている．しかし，読み及びアクセントが発話順に順次決定するモデルであるとすると，前者の場合，本来事後的に決定すべきアクセント句境界を先に決定していることから，最適性が保障されない，という問題がある．最適性については，各段階で N-best の解を出力して，順次，逐次的な処理を行うことで同じ効果を期待出来るが，組み合わせのモデルが複雑になる上，やはり，アクセント句境界を各単語のアクセントに先行して求めることになる．

上記のことから，本研究では，⟨ 表層, 品詞, 読み, アクセント ⟩ の組を 1 つの単位とみなし，N-gram モデルを用いて同時に推定する手法（4 つ組 N-gram モデル）を提案する．つまり，逐次的な処理ではなく，1 つの確率モデルで 4 つの値を同時に推定する．実験では,「確率モデル」を用いて「同時」に推定を行うことの有効性を検証するため，従来手法である （1）ルールによるアクセント付与,（2）逐次的な読み・アクセント付与，との比較を行った．前者ではルールに基づきアクセント句境界及びアクセント核を逐次的に決定するモデルを用い，後者では逐次的に形態素解析・読み付与・アクセント付与を行うモデルを用いて比較を行った．その結果，本手法による推定精度が両モデルに基づく手法の精度を上回ることを確認した．

さらに，読みとアクセントの推定に品詞情報を用いないモデル，つまり ⟨ 表層, 読み, アクセント ⟩ の組を 1 つの単位とみなしたモデル（3 つ組 N-gram モデル）に関しても読みとアクセントの推定精度を検証した．その結果，読みとアクセントの精度に関して，4 つ組 N-gram モデルの精度が，3 つ組 N-gram モデルの精度を上回った．ただし，コーパス作成の労力を考慮すると，読みとアクセントの推定というタスクに関しては 3 つ組 N-gram モデルが適していた．

3.3 日本語における読みとアクセント

任意の入力テキストに対して，単語分割を行い，読みとアクセントを割り当てる問題を考える．読みに関しては，他の言語も日本語と同様に，同じ表記を持ち読みの異なる単語が存在するが，他の言語（例えば，英語や中国語）に比べて，読みの種類が多い．本章では，日本語の読みとアクセントの特性について説明するとともに，読みとアクセントの推定に関する従来手法について述べる．

3.3.1 読み

日本語の表記は主にカタカナ・ひらがな・漢字から構成されるが，このうち漢字には複数の読みの可能性が存在する．手元の辞書を用いて JIS 第一・第二水準の漢字 6,355 文字に対して，漢字 1 文字あたりの読みの異なり数を調べたところ，漢字 1 文字あたり平均 1.84 個の異なる読みが存在し，例えば最も読みの多い漢字の一つである「端」に対しては，10 個の読み（*ha* , *ha shi* , *ta n* 等） が存在する．単語は，これらのカタカナ・ひらがな・漢字を構成要素とする文字列であり，単語の読みは，多くの場合構成要素である各文字の読みの組み合わせであるが,「流石」*sa su ga* のように単文字の読みの組み合わせではない読みも存在するため，読みを

含む単語辞書が必須となる．また，同じ表層をもつ単語であっても文脈によって読みが異なるため，単に辞書を用いるのではなく，文脈に即して読みの候補から尤もらしい読みを推定する必要がある．

また，形態素解析における未知語の品詞推定と同様に，読みの推定に関しても未知語（未知読み語）の読み推定の問題が避けられない．日本語における未知語の読み推定に関しては，英語の文字列を日本語の音素列に変換するモデル [30] 等をベースにした，文字 N-gram モデルを用いた手法 [31] などが提案されており，本研究においても $\langle$ 文字列, 読み $\rangle$ を単位とした N-gram モデルを用い未知語の読み推定を行う．

3.3.2 アクセント

日本語のアクセントは多くの場合，高低 2 値ピッチ H 及び L の列で表され，各モーラに 1 つ付与される．例えば，3 モーラの単語「京都」/*kyo o: to*/ に対しては，/H L L/ という 3 つの高低 2 値ピッチ列が付与される．本研究においても，この高低 2 値ピッチ列として表されるアクセントを用いる．ここで注意すべきことは，同じ表層を持ち，品詞も読みも同じ単語であっても，文脈によってアクセントは異なるということである．例えば，名詞「タワー」が「京都」に後続した単語「京都 , タワー」/*kyo o: to , ta wa a:*/という複合名詞中では,「京都」/*kyo o: to*/のアクセントは/L H H/ であり，全体のアクセントとしては，/L H H , H L L/ である（表 3.1）．さらに，名詞「ホテル」が後続した単語「京都 , タワー , ホテル」/*kyo o: to , ta wa a: , ho te ru*/の場合,「タワー」のアクセントは/H L L/ ではなく/H H H/ であり，全体のアクセントとしては，/L H H , H H H , H L L/ である（表 3.2）．

表 3.1: 複合名詞「京都タワー」における「京都」及び「タワー」のアクセント

表層	w	京都			タワー		
品詞	t	固有名詞			一般名詞		
読み	$\boldsymbol{s}$	*kyo*	*o:*	*to*	*ta*	*wa*	*a:*
アクセント	$\boldsymbol{a}$	L	H	H	H	L	L

日本語の音声合成を目的としたアクセント付与に関する研究は幾つか行われており，匂坂 [32] らは，日本語の単語連鎖におけるアクセントの変化（アクセント核の移動）規則に関して体系化を行った．具体的には前後に文脈が存在しない時の単語のアクセント型と，単語毎に決まるアクセント移動関数によって，アクセ

表 3.2: 複合名詞「京都タワーホテル」における「京都」・「タワー」及び「ホテル」のアクセント

表層 w	京都			タワー			ホテル		
品詞 t	固有名詞			一般名詞			一般名詞		
読み $\boldsymbol{s}$	*kyo*	*o:*	*to*	*ta*	*wa*	*a:*	*ho*	*te*	*ru*
アクセント $\boldsymbol{a}$	L	H	H	H	H	H	H	L	L

ントを決定している．文をアクセント句の列 $\boldsymbol{p} = (p_1 p_2 \cdots p_l)$ とみなし，下記の手法でアクセント列を決定する．

1. まず，形態素解析器等を用いて入力テキストを解析し，表層（単語境界）w ・品詞 t ・読み $\boldsymbol{s}$ を決定し，この 3 つ組を $v = \langle w, t, \boldsymbol{s} \rangle$ とする．

2. この 3 つ組列 $\boldsymbol{v} = (v_1 v_2 \cdots v_h)$ を，アクセント句決定ルールを用いて，$\boldsymbol{v}$ を 1 つ以上のアクセント句の列に分割する $\boldsymbol{v}_1^h \mapsto \boldsymbol{p}_1^l (h \leq l)$ ．ここで各アクセント句 $p_i (1 \leq i \leq l)$ は $\boldsymbol{v}$ の部分列 $v_{r_{i-1}+1} v_{r_{i-1}+2} \cdots v_{r_i} (r_0 = 1, r_l = h)$ である．

3. 各アクセント句 p_i に対して，アクセント句内の各 3 つ組 $v_j (j = r_{i-1}+1, r_{i-1}+2 \cdots r_i)$ の単独でのアクセント（アクセント核の位置） 及び，アクセント移動関数を辞書を参照して取得し，アクセントをアクセント句 p_i 内で変化させる．この結果，各アクセント句に対して 1 つのアクセント型が割り当てられ，最終的にアクセント型はアクセント（L, H の列）に変換される．

この手法は，標準語において辞書が十分に整備されている状況の下では，比較的高い精度が得られる．ただし，辞書の各見出し語に対して，アクセント・アクセント移動関数が必要であり，新たに語を追加する場合にこの両方を辞書に登録する必要がある．また，アクセント句の決定ルールも追加する必要があり，専門的な知識を要求される．また，これらのルールは形態素解析器の品詞体系に依存するため，汎用性という面で不利である．

また，Seto らは，決定木 [33] を用いて，アクセント句の推定，アクセント句に対するアクセントの推定，ポーズ推定を目的とした単語の係り受けの推定を行っている [28] が，形態素解析を前もって別に行う必要があり，さらにアクセント句等，各段階の値を推定するためのモデルが個別に必要となる．他に，日本語の姓名を対象にした未知語モーラ列からの統計的なアクセント推定も行われている [34] が，文脈中のアクセントを推定するものではない．

3.3.3 解くべき問題

本研究の目的は，文脈を伴って現れる文字列に対して，正しい読みとアクセントを付与することである．ただし，音声合成システム全体を考えると，この読みとアクセント付与の処理に，ポーズ推定等の処理が後続する．一般的に発話時のポーズは単語と単語の間に発生する．ポーズ推定等の後続する処理で用いる言語単位が文字単位ではないため，文字単位の処理ではこれらの処理との整合性が良くない．また，文字単位の処理では，読み及びアクセントがどのような言語的な特徴を持つかを調べるのが困難である．このような点を考慮すると文字単位でなく，単語単位で読み及びアクセント付与を行った方が都合が良い．したがって，解くべき問題は，文脈を伴って現れる文字列に対して，正しい読みとアクセントを付与された単語列を出力することである．つまり，入力文字列 $\boldsymbol{x}$ から，正しい単語境界・読み・アクセントの組 $\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle$ の列を推定することである．ここで，読み $\boldsymbol{s}$ は音素 $s \in \mathcal{S}$ の列であり，本研究では，日本語に対応した約 120 種類の音素をもつ集合 $\mathcal{S} = \{\ a\ ,\ i\ , \cdots ,\ n\ \}$ を用いた．また，アクセント $\boldsymbol{a}$ は 2 値ピッチ $a \in \mathcal{A}$ の列であり，上述したように，$\mathcal{A} = \{\mathsf{H},\mathsf{L}\}$ とした．以下に例を示す．

入力

京都タワーホテルですね

出力

京都, /*kyo o: to*/, /L H H/
タワー, /*ta wa a:*/, /H H H/
ホテル, /*ho te ru*/, /H L L/
で, /*de*/, /H/
す, /*su*/, /L/
ね, /*ne*/, /H/

3.4 統計的日本語音声合成フロントエンド

本章では，確率モデルを用いた読み及びアクセント付与の枠組みを提案する．本モデルでは，読みとアクセントを推定するのみでなく，単語境界及び品詞も同時に推定する．

3.4.1 N-gram モデルに基づく形態素解析

確率的な言語モデルである N-gram モデルは，英語や他のヨーロッパ言語のような空白で分かち書きされた文に対する品詞タグ付けのモデルとして用いられており，永田 [35] によって，日本語や中国語のような分かち書きされない言語に対しての形態素解析のモデルとして一般化された．表層 w と品詞 t の組を一つの単位として，形態素解析のモデルとなっている．

$$\begin{aligned} & P(\langle w_1, t_1\rangle\langle w_2, t_2\rangle \cdots \langle w_h, t_h\rangle) \\ = & \prod_{i=1}^{h+1} P(\langle w_i, t_i\rangle | \langle w_{i-k}, t_{i-k}\rangle \cdots \langle w_{i-1}, t_{i-1}\rangle) \end{aligned} \tag{3.1}$$

ここで， $k = n - 1$ であり， $\langle w_i, t_i\rangle$ $(i \leq 0)$ は，文頭に対応する特別な記号である．また，$\langle w_{h+1}, t_{h+1}\rangle$ は文末に対応する特別な記号を表す．

確率モデルを用いた形態素解析器は，文字 x の列として表される文 $\boldsymbol{x}$ と，形態素 $\langle w, t\rangle$ の列として表される文の表層 $\boldsymbol{w}$ が等しい $\boldsymbol{x} = x_1x_2\cdots x_l = w_1w_2\cdots w_h = \boldsymbol{w}$ という制約条件下で，最も確率値の高い品詞と表層の組の列を出力する．

$$\begin{aligned} & (\langle w_1, t_1\rangle\langle w_2, \widehat{t_2}\rangle \cdots \langle w_h, t_h\rangle) \\ = & \ \mathbf{argmax}\, P(\langle w_1, t_1\rangle\langle w_2, t_2\rangle \cdots \langle w_q, t_q\rangle | x_1x_2\cdots x_l) \end{aligned} \tag{3.2}$$

3.4.2 読みおよびアクセント推定

読み及びアクセント推定を目的として，形態素 N-gram モデルを拡張する方法を提案する．まず，表層 w ・品詞 t ・読み $\boldsymbol{s}$ ・アクセント $\boldsymbol{a}$ の 4 つ組を一つの単位 u とした N-gram モデルを考える．つまり $u = \langle w, t, \boldsymbol{s}, \boldsymbol{a}\rangle$ となり，例えば，$\langle$ 京都, 固有名詞, *kyo o: to*, H L L$\rangle$ である．この 4 つ組 N-gram モデル M_u による，4 つ組列 $u_1, u_2\cdots u_h$ の生成確率は以下の式で表される．

$$M_u(u_1u_2\cdots u_h) = \prod_{i=1}^{h+1} P(u_i | u_{i-k}\cdots u_{i-2}u_{i-1}) \tag{3.3}$$

ここで，$k = n-1$ であり，u_i $(i \leq 0)$ は，文頭に対応する特別な記号である．また，u_{h+1} は文末に対応する特別な記号を表す．形態素解析と同様に，4 つ組 N-gram モデルの確率値も，学習コーパス中に出現する 4 つ組を計数した頻度から最尤推定される．文字 x の列として表される文 $\boldsymbol{x}$ と，4 つ組 $u = \langle w, t, \boldsymbol{s}, \boldsymbol{a}\rangle$ の列として表される文の表層 $\boldsymbol{w}$ が等しい $\boldsymbol{x} = x_1x_2\cdots x_l = w_1w_2\cdots w_h = \boldsymbol{w}$ という制約条件

下で，下記の式において解探索を行う．

$$\hat{\boldsymbol{u}} = \mathbf{argmax}\, M_u(u_1 u_2 \cdots u_h | x_1 x_2 \cdots x_l) \tag{3.4}$$

解探索に関しては，動的計画法 [36] を用いて効率的に解けることが分かっており，解探索の計算量は入力文字列長に比例する．

3.4.3 未知語モデル

4 つ組 N-gram モデルは，文を 4 つ組の列 $\boldsymbol{u} = u_1 u_2 \cdots u_h$ の表層を連結したものとみなし，各 4 つ組を文の先頭から順に予測する．しかし，日本語の $\langle$ 表層, 品詞, 読み, アクセント $\rangle$ の組を全て列挙することは出来ないので，未知 4 つ組の扱いが避けられない問題となる．通常，式（3.3）の確率値は，学習コーパス中に出現する単語を計数した頻度から最尤推定されるが，文に未知 4 つ組 （学習コーパスに出現しない 4 つ組）を含む場合， M_u の確率値は 0 となり，未知 4 つ組を含む 4 つ組列が，式（3.4） によって選ばれることは無い．しかしながら，実際の問題として，入力テキスト中に出現する可能性のある全ての 4 つ組が，学習コーパス中に出現することはない．

この問題に対処するため，未知 4 つ組に対応する特別な記号 $u^{\texttt{UNK}}$ を用意し，既知の 4 つ組以外はこの記号を用いた未知語モデルにより与えられる確率で生成されることとする．未知 4 つ組に対応する特別な記号は，かならずしも唯一である必要はなく，品詞などの情報を用いて区別される複数の記号であってもよい．以下の説明では，各品詞 t に対して未知 4 つ組に対応する記号 $u_t^{\texttt{UNK}}$ を設ける．式（3.3）における確率値 P は 未知語モデル M_x を含み，式（3.5）のように表される．

ここで $\mathcal{U}$ は学習コーパス中に出現する 4 つ組の集合であり，u_i が未知 4 つ組の場合，その未知 4 つ組の品詞毎に対応する記号 $u_t^{\texttt{UNK}}$ が用いられる．任意の入力テキストに対して未知語モデルが適用可能であるためには，任意の文字に対して，出現確率が 0 より大きくなければならない．

$$P(u_i | u_{i-k} \cdots u_{i-2} u_{i-1}) \tag{3.5}$$
$$= \begin{cases} P(u_i | u_{i-k} \cdots u_{i-2} u_{i-1}) & \text{if } u_i \in \mathcal{U} \\ P(u_t^{\texttt{UNK}}{}_i | u_{i-k} \cdots u_{i-2} u_{i-1}) M_x(u_i | t_i) & \text{if } u_i \notin \mathcal{U} \end{cases}$$

未知語読み

まず，文字列と読みの組 $\langle x, \boldsymbol{s} \rangle$ を単位とした未知語読み N-gram モデルを各品詞毎に定義する．

$$
\begin{aligned}
& M_x(\langle x_1, \boldsymbol{s}_1 \rangle \langle x_2, \boldsymbol{s}_2 \rangle \cdots \langle x_{h'}, \boldsymbol{s}_{h'} \rangle | t) \\
= & \prod_{i=1}^{h'+1} P(\langle x_i, \boldsymbol{s}_i \rangle | \langle x_{i-k}, \boldsymbol{s}_{i-k} \rangle \cdots \langle x_{i-1}, \boldsymbol{s}_{i-1} \rangle, t)
\end{aligned}
\tag{3.6}
$$

ここで， $k = n - 1$ であり， $\langle x_i, \boldsymbol{s}_i \rangle$ $(i \leq 0)$ は，単語頭に対応する特別な記号である．また，$\langle x_{h+1}, \boldsymbol{s}_{h+1} \rangle$ は単語末に対応する特別な記号を表す．

未知語アクセント

次に，未知語のアクセントに関しては学習コーパス中の全単語で最も頻度の高いアクセント L H H $\cdots$ H を用いることにする．これは先頭のモーラのアクセント要素のみが L で，2 モーラ目以降のアクセント要素が H であるアクセントである．このアクセントは学習コーパス中に現れる全形態素のアクセントのうち全体の 37.28% を占めるため，最も簡易な推定方法としては妥当であると考える [1]．

未知 4 つ組モデル

最後に，未知語モデルによって推定される 4 つ組の出現確率 $u = \langle w, t, \boldsymbol{s}, \boldsymbol{a} \rangle$ は，品詞 t 毎に，下記の式で与えられるとする．

$$
\begin{aligned}
& M_x(u|t) \\
= & \begin{cases}
M_x(\langle x_1, \boldsymbol{s}_1 \rangle \langle x_2, \boldsymbol{s}_2 \rangle \cdots \langle x_{h'}, \boldsymbol{s}_{h'} \rangle | t) \\
\qquad\qquad\qquad \text{if } \boldsymbol{a} = \mathsf{L\ H\ H} \cdots \mathsf{H} \\
0 \qquad\qquad\qquad\qquad \text{それ以外}
\end{cases}
\end{aligned}
\tag{3.7}
$$

ここで，未知 4 つ組の表層文字列は，各文字を結合した結果に等しく（$\boldsymbol{w} = x_1 x_2 \cdots x_{h'}$），音素列の長さはアクセント要素の長さに等しくなければならない（$|\boldsymbol{s}| = |\boldsymbol{a}|$）．

[1] アクセントの推定も同様に未知語モデルによって推定可能であると考えられ，さらなる精度の向上が期待出来る．

3.4.4 パラメータ推定

最終的に，確率的モデルに基づく処理部は，式（3.3）(3.5) （3.7）の生成確率を計算し，最終的に最も生成確率の高い解が式（3.4） によって与えられる．

式 （3.5） のパラメータは，学習コーパスの中の 4 つ組を計数した頻度から最尤推定される．コーパスの各文は予め単語に区切られており，各単語には，品詞・読み・アクセントが付与されている．学習コーパスは 9 個に分割され，4 つ組が分割されたコーパスの 1 個のみに出現する場合，コーパス中の 4 つ組を品詞毎の未知 4 つ組に対応する特別な記号 $u_t^{\mathtt{UNK}}$ に置き換える．

式（3.6）の未知語読み N-gram モデルのパラメータは 4 つ組の N-gram 確率値を計算する際に，未知形態素に対応する特別な記号によって置き換えられた低頻度の 4 つ組から計算される．これらの低頻度の形態素からは品詞毎に，文字列と音素列の組が取り出される．この文字列と音素列の組の対応付けは，辞書を用いて自動的に行われる．この辞書には，全ての文字に対して可能な読みが記述されている．式 （3.6） の各品詞に対するパラメータは，この対応付けされた形態素集合から推定される [2].

式 （3.5） における N-gram 確率値はより低次のモデルとの補間を行う．補間係数は，削除補間法 [37] によって推定される．式（3.6）の N-gram 確率も同様に低次のモデルと補間が行われる．

3.5 実験

本提案手法の「確率モデル」を用いて「同時」に推定を行うことの有効性を検証するため，(1）ルールによるアクセント付与，(2）逐次的な読み・アクセント付与，との比較を行った．さらに，4 つ組から品詞情報を取り除いた 3 つ組モデルに関しても評価を行った．

3.5.1 コーパス

実験に用いたのは，新聞記事・テレビニュースの書き起こし・電話応答文など，多様な内容を含むコーパスである．したがって，含まれる語彙も多岐にわたり，書き言葉だけでなく話し言葉も含まれる．各文は，予め人手によって単語列に分割されており，各単語は，表層 w・品詞 t ・読み $\boldsymbol{s}$ ・アクセント $\boldsymbol{a}$ の 4 つ組から構成される．コーパス全体における 1 文あたりの平均単語長は 21.6 語で，各単語の

[2]僅かの形態素において，アラインメントに複数の候補が存在する場合があったが，そのような例はパラメータ推定には用いていない．

平均文字列長は 1.91 文字である．学習用およびテスト用のコーパスのサイズを表 3.3 に示す．本研究で用いた品詞は名詞・固有名詞等の 17 種類で，加えて，文頭・文末に対応する特殊な品詞を用意してある．

表 3.3: 実験用コーパス

	文数	単語数	文字数
学習コーパス	8,800	190,318	285,082
テストコーパス	150	2,130	3,170

3.5.2 モデル

提案手法である 4 つ組 N-gram モデルおよび，4 つ組から品詞情報を除いた 3 つ組 N-gram モデル，加えて比較対象として，従来手法の 2 つのモデルを用意した．3 つ組モデルは，品詞情報の要否を検討するために用意した．また，今回実験で用いた各 N-gram モデルは，2-gram までを考慮した．

WT+S+A 逐次推定確率モデル

- 文を表層・品詞の 2 つ組 $\langle w, t\rangle$ の列とみなし，N-gram モデルを用いて，単語境界及び品詞を推定する．
- 推定された表層及び品詞の組 $\langle w, t\rangle$ の列を制約条件とし，表層・品詞・読みの 3 つ組 N-gram モデルを用いて読みを推定する．
- 推定された表層・品詞及び読みの組 $\langle w, t, \boldsymbol{s}\rangle$ の列を制約条件とし，表層・品詞・読み・アクセントの 4 つ組 N-gram モデルを用いてアクセントを推定する．
- それぞれの段階では，前段階での最尤解（1-best）を用いる．

WTS+A_r ルールによるアクセント付与

- 文を表層・品詞・読みの 3 つ組 $\langle w, t, \boldsymbol{s}\rangle$ の列とみなし，N-gram モデルを用いて，単語境界・品詞及び読みを推定する．
- アクセント句境界を予め作成しておいたルール（約 $1,000$ ルール）を用いて，順次適用することによって決定する．アクセント句境界を決定するためのルールの条件部には，主に品詞が用いられる．
- 各アクセント句内でアクセント核を 3.3.2 節で説明した方法に基づき決定し，各単語のアクセントを決定する．

WTSA 同時推定確率モデル （提案手法）

- 文を表層・品詞・読み・アクセントの 4 つ組 $\langle w, t, \boldsymbol{s}, \boldsymbol{a} \rangle$ の列とみなし，N-gram モデルを用いて，単語境界・品詞・読み及びアクセントを同時に推定する．

WSA 品詞なし同時推定確率モデル （提案手法）

- 文を表層・読み・アクセントの 3 つ組 $\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle$ の列とみなし，N-gram モデルを用いて，単語境界・読み及びアクセントを同時に推定する．

3.5.3 評価

文脈を伴って現れる単語列に対して，正しい読みとアクセントを付与することが本研究の目的である．つまり，入力文字列 $\boldsymbol{x}$ から，正しい単語境界・読み・アクセントの組 $\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle$ を推定することである．表 3.4 に各モデルに対する読みとアクセントの精度（6 列目）を示す．これは単語列として表される文において，単語境界がテストコーパスの正解と一致し，かつ読みとアクセントが，同じくテストコーパスの正解と一致した割合を示す（以下,「読み＋アクセント精度」とする）．また，参考として単語分割の精度（3 列目）と，品詞付与の精度（4 列目）も記す．

読み＋アクセント精度に関しては，モデル ***WTSA*** が 3 モデルの中で最も高く，90.26% となっている．モデル ***WTSA*** から品詞を取り除いたモデル ***WSA*** の読み＋アクセント精度は 89.72% であり，これはモデル ***WTSA*** に比べて 0.54% 低い．***WTSA*** と ***WSA*** の比較から，品詞情報は精度向上に寄与することが分かるが，その差はそれほど大きくない．この同時推定確率モデルを用いた両モデルのいずれも，逐次推定モデル ***WT+S+A*** ，ルール推定モデル ***WTS+***A_r よりも精度が高い．

また，読みとアクセントは同時に決定するため，アクセント付与単体の精度を正確に求めることは出来ないが，参考値として，読みが正解と一致した単語に対してアクセントが一致した割合 $accuracy(\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle | \langle w, \boldsymbol{s} \rangle)$ をアクセント付与単体 $\langle \boldsymbol{a} \rangle$ の精度（7 列目）とした．その結果，アクセント付与の精度は ***WTSA*** が最も高く，92.63% となっている．単体の形態素解析器としてみた場合，モデル ***WT+S+A*** ，モデル ***WTS+***A_r の順に精度が良い．この結果は，異なり語数が少ないほど精度が良く，読みやアクセントの情報は品詞の推定に関しては寄与しておらず，学習コーパスのデータスパースネスを引き起こしている．

図 3.2 及び図 3.3 は，学習コーパスのサイズと精度の関係を示してある．図 3.2 において．実線は読み $\langle w, \boldsymbol{s} \rangle$ の精度．破線は読み＋アクセント $\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle$ の精度を

表 3.4: モデル毎の精度（単語境界・品詞付与・読み付与・アクセント付与）

	語数	$\langle w \rangle$	$\langle w, t \rangle$	$\langle w, \boldsymbol{s} \rangle$	$\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle$	$\langle \boldsymbol{a} \rangle$
WT+S+A $\langle w, t \rangle$	15,211	*98.07%*	*96.13%*	96.70%	**88.73%**	91.76%
WTS+A_r $\langle w, t, \boldsymbol{s} \rangle$	15,723	*97.61%*	*96.08%*	97.69%	**89.53%**	91.65%
WTSA $\langle w, t, \boldsymbol{s}, \boldsymbol{a} \rangle$	21,164	*97.87%*	*95.79%*	97.45%	**90.26%**	92.63%
WSA $\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle$	19,560	*97.64%*	*N/A*	97.08%	**89.72%**	92.42%

語数:異なり語数，$\langle w \rangle$:単語境界，$\langle w, t \rangle$:単語境界&品詞，$\langle w, \boldsymbol{s} \rangle$:単語境界&読み，$\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle$:単語境界&読み&アクセント，$\langle \boldsymbol{a} \rangle \sim \langle w, \boldsymbol{s}, \boldsymbol{a} \rangle | \langle w, \boldsymbol{s} \rangle$:アクセント

示す．図 3.2 の各線の最も右の点は，表 3.4 の左から 5 番目の列 $\langle w, \boldsymbol{s} \rangle$ と同 6 番目の列 $\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle$ の精度と一致する．図 3.3 の各線の最も右の点は，表 3.3 の一番右の列 $\langle \boldsymbol{a} \rangle$ の精度と一致する．実線は近似されたアクセントの精度を示す．***WTSA*** の精度および ***WSA*** の読み＋アクセントの精度に関する学習曲線に注目すると，8,000 文あたりで， ***WTS+***A_r の精度を超えることが分かる．

品詞の付与に関しては，品詞を用いた場合の ***WTSA*** に比べ ***WSA*** の精度は，読みに関しては 0.37% ，アクセントに関しては 0.54% 精度が低い．ただし，精度と，学習コーパスを用意する際の作成に関わる労力の双方を考慮すると，品詞を付与しない学習コーパスを作成すべきか，品詞を付与する学習コーパスを作成すべきかを検討する余地がある．この点については，3.6.2 節にて詳述する．

3.6 考察

今後のさらなる精度向上に必要な改良点を明らかにするために，誤り原因を調べるとともに，読みとアクセントを推定するためのモデルとして，品詞情報の要否について検討を行った．

3.6.1 誤り解析

3.5 章では，N-gram モデルが読みとアクセントを推定するためのモデルとして有効であることを示したが，依然として読みおよびアクセントに関しては誤りが存在する．そこで，4 つ組モデル ***WTSA*** において，どのような文脈において誤りが多いかを調査した．

モデル ***WTSA*** において，どのような場合に解析が失敗するのか調べた．表 3.5 は，誤った単語の前 1 単語に着目して，品詞毎にどのような文脈において読み＋アクセントの誤りが多いのかを示す（誤りの多かった品詞のみについて示してある）．誤りの多い単語 N-gram の品詞は，多いほうから順に，名詞- 助詞，名詞-名詞，動

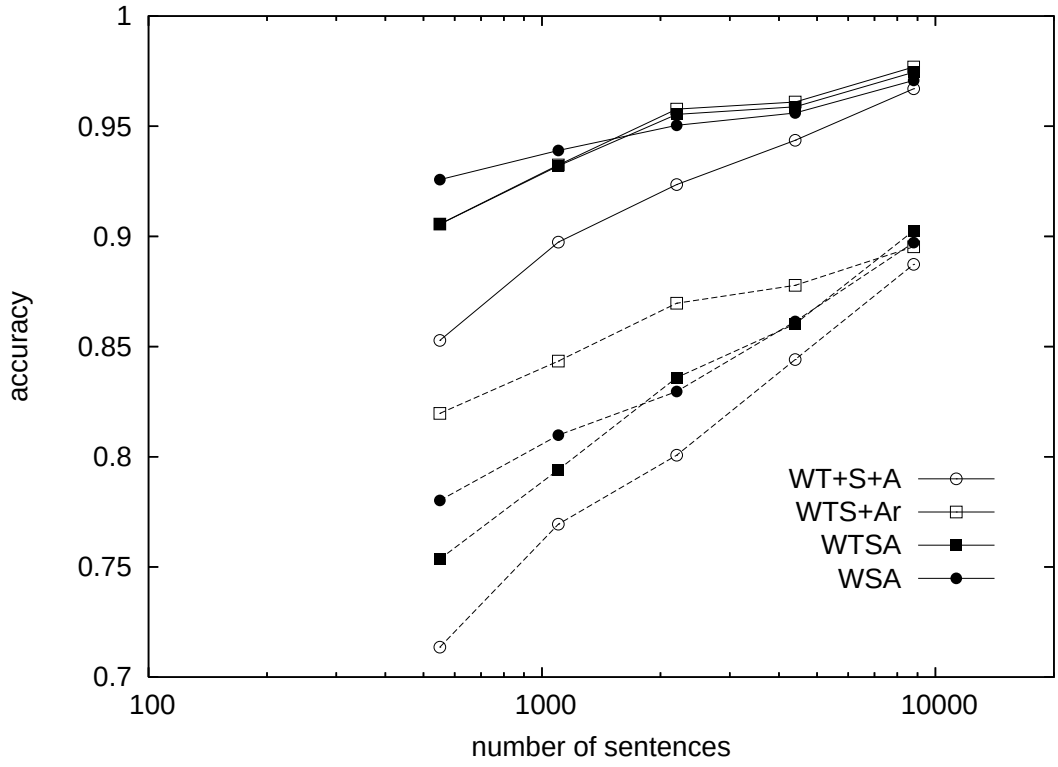

図 3.2: 読みとアクセントに関する学習曲線

詞-語尾，助詞-名詞，であった．特に名詞の関連する誤りが多く，これは名詞の異なり語数が他の品詞に比べて多いことが原因であると推測できる．表 3.5 の場合，206 単語の誤りがあったが，このうち名詞の関連するものは，74 単語（約 36%）であった．この種の誤りに関しては，単語クラスタリング等の低頻度 N-gram への対処が有効である．また，現在のコーパス量から考えると，さらなるコーパスの追加により，推定精度を向上できる余地が大きい．

また，モデルの違いによる誤り分布の違いを調べるため，アクセント推定にルールを用いたモデル ***WT+S+A*** に関しても調査を行ったが，モデル ***WTSA*** と比較して，誤り原因の分布に関しては，有意な差は見られなかった．

次に，モデル ***WTSA*** における読み誤りについて調査を行った．全部で 58 単語の誤りがあったが，このうち，単語分割の誤りに起因し，かつ分割誤りを許容すると読みがテストコーパスの読みと一致する場合（例：正解は $\langle$円安, 名詞, $e, n, ya, su\rangle$ ．モデル ***WTSA*** による出力が $\langle$円, 接尾辞, $e, n\rangle$ $\langle$安, 形容詞, $ya, su\rangle$.） が 42 単語（72%）であった．もし，この単語分割の誤りに起因する誤りを正解とみなすと，読みの正解率は 99.2% になる．また，明らかな読み誤りが 12 単語（21%），濁音化に関する誤りが 4 単語（7%）であった．最も多い誤りである単語分割の誤り

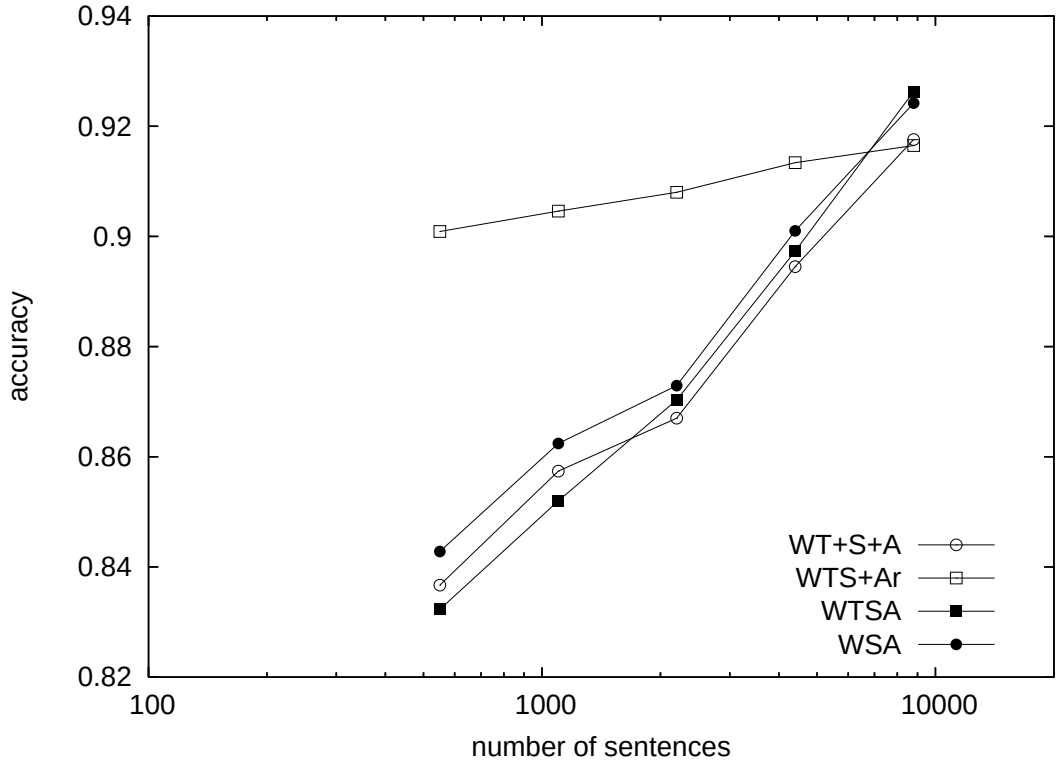

図 3.3: アクセントに関する学習曲線

に関しては，コーパスの追加・修正で対処できるであろう．さらに，***WTSA*** において，読みがテストコーパスの読みと一致するが，アクセントが誤った場合の $206 - 58 = 148$ 単語について調査を行ったが，読み＋アクセントが誤った場合（208 単語）と同様の傾向であった．

3.6.2 品詞付与の要否

音声合成全体を考えると，前段で，入力テキストから音韻・韻律情報を生成し，後段の音声合成部の入力とするが，この後段の音声合成部では品詞情報を必要としない．3.5 章における実験の結果，4 つ組モデル ***WTSA*** の精度は，3 つ組モデル ***WSA*** の精度を上回った．ただし ***WTSA*** モデルはコーパスに対して ⟨ 表層, 品詞, 読み, アクセント ⟩ を与える必要があり，対して，***WSA*** モデルは ⟨ 表層, 読み, アクセント ⟩ を必要とする．コーパスの作成を考えたとき，品詞を含む 4 つの属性をコーパスに付与する手間は，品詞を含まない 3 つの属性をコーパスに付与する手間より大きい．つまり，コーパス作成の労力と推定精度の双方を鑑みて，4 つの属性を付与するコーパスと 3 つの属性を付与するコーパスとのどちらを作成

表 3.5: 単語 N-gram における品詞毎の誤り分布

$t_{i-1} \setminus t_i$	動詞	語尾	助動	助詞	固有	名詞	接尾
動詞		**18**	2	1		1	
語尾	1		5	6		6	1
助動詞				1		2	
助詞	14			1	5	**17**	
固有名詞				1	2	4	2
名詞	1		1	**31**		**20**	14
接尾辞		1		9		2	1

すべきかをということが問題になる．そこで，実際に ***WTSA*** と ***WSA*** のためのコーパス作成の作業コストを比較するために 500 文に対して，コーパス修正に精通した 1 名の作業者によってコーパスの作成を行い，その作業にかかった時間を計測した（表 3.6 ）．

- 品詞を付与するコーパス $\langle w, t, \boldsymbol{s}, \boldsymbol{a} \rangle$
  人手で，新たに 500 文に対して，単語境界・品詞・読み・アクセントを付与する．実際には，前章の評価で用いた ***WTSA*** モデルを用いて，単語境界・品詞・読み・アクセントを付与し，それに含まれる誤りの修正を行った．
- 品詞を付与しないコーパス $\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle$
  人手で，新たに 500 文に対して，単語境界・読み・アクセントを付与する．実際には，前章の評価で用いた ***WSA*** モデルを用いて，単語境界・読み・アクセントを付与し，それに含まれる誤りの修正を行った．

表 3.6: コーパス修正に要する時間

	時間（min）	文数（/min）
WTSA	230	2.17
WSA	170	2.94

その結果，***WTSA*** の修正にかかる時間は，***WSA*** に対して 1.35 倍長い．一方，図 3.2 を元に対数回帰分析を行った結果，品詞を付与しない場合，品詞を付与する場合に比べて 1.01 倍のコーパスを用意すれば，ほぼ同じ精度を得ることが出来

る．このことから，品詞を付与しないモデル ***WSA*** のほうがコーパスの作成に要する労力という観点からは，効率が良い．限られた人的資源の中でどのようにして効率的にコーパスを作成するかということを考えると，品詞を用いないモデルが適している．

3.7 本章のまとめ

規則音声合成では，任意のテキストを入力とし，人間の音声に出来るだけ近い音声を出力することが求められる．音声合成の前段であるテキスト処理においては，任意の入力テキストに対し，出来るだけ正しい音韻情報と韻律情報を生成することが好ましい．本論文では，確率的な手法を用い，入力テキストに対し，読み及びアクセントを付与する手法について述べた．提案したモデルでは ⟨ 単語境界, 品詞, 読み, アクセント ⟩ の 4 つ組，または品詞を用いない ⟨ 単語境界, 読み, アクセント ⟩ の 3 つ組を 1 つの単位と捉え，N-gram モデルを用いて推定を行う．つまり，⟨ 単語境界, 品詞, 読み, アクセント ⟩ の 4 つの値，または ⟨ 単語境界, 読み, アクセント ⟩ の 3 つの値を同時に推定する．テストコーパスに対する推定結果に対し，コーパスに予め与えられた正解との単語毎の精度を計算した結果，4 つ組および 3 つ組確率モデルに基づく手法の精度は，既存手法である逐次的手法とルールを用いた手法の精度を上回った．また，4 つ組と 3 つ組のモデルを，学習コーパスの作成に要する時間と精度の 2 つの観点から考察を行った結果，品詞を用いない 3 つ組のモデルが適していた．

第4章　アクセントクラスの利用による音声合成フロントエンドの高精度化

本章では，さらなる精度の向上を目指し，言語モデルの改善を行う．第3章の4つ組モデル4.1に対し実験を行ったところ，正解の4つ組み⟨表層, 品詞, 読み, アクセント⟩の組を持つ単語または履歴を含む単語列が学習コーパス中に存在しないという問題があることが分かった．しかしながら，同じ表層をもつ単語であっても日本語のアクセントは文脈によって変化するため，学習コーパスから単語の取り得るアクセント型を全て収集することは一貫性およびコストの面からあまり現実的ではない．そこで，学習コーパスを増やすことなく，日本語のアクセントの特徴を利用したモデルを構築する．さらに，比較的音声の知識に精通していないユーザーでも容易に精度の高い言語モデルが構築でき，かつ読みやアクセント誤りを容易に修正および調整できるシステムを構築するためのチューニングツールについても述べる．

4.1　アクセントクラス N-gram モデル

4.1.1　アクセントクラス

標準的な日本語で記述された文章に対するアクセント推定としては，ルールを用いた手法 [32] が提案されている．ルールを用いた手法では，音声合成用辞書の各単語に対し，単独発声時のアクセント型（以下，辞書アクセント型）と，アクセント結合様式と結合アクセント価（以下，併せてアクセント移動型）を人手で付与し，入力文字列に対して，単語分割および品詞，読み，アクセント句推定を行った後，アクセント句中のアクセント核の位置を決定する．これら従来研究から，辞書アクセント型およびアクセント移動型が，出力単語列において観測されるべきアクセント型（以下，文脈アクセント型）を決定する特徴量の一つであることが分かり，本研究ではアクセントの変化をクラスの生成に用いることとした．表

4.1 に，入力文「今日京都タワーホテルに泊まる.」に対する表層 w，品詞 t，読み $\boldsymbol{s}$，文脈アクセント $\boldsymbol{a}$，および辞書アクセント $\boldsymbol{\acute{a}}$，アクセント移動型 g を示す．アクセントの特徴を表す要素の組をアクセント特徴 $f_{acc} = \langle t, \boldsymbol{a}, \boldsymbol{\acute{a}}, g \rangle$ と定義し，同じ f_{acc} を持つ単語を含む単語集合をアクセントクラスと定義する．

表 4.1: 入力文「今日京都タワーホテルに…」に対する言語情報

		1	2	3	4	5
表層	w	今日	京都	タワー	ホテル	に
品詞	t	名詞	固有名詞	名詞	名詞	助詞
読み	$\boldsymbol{s}$	*kyo o:*	*kyo o: to*	*ta wa a:*	*ho te ru*	*ni*
文脈アクセント	$\boldsymbol{a}$	H L	L H H	H H H	H L L	L
辞書アクセント	$\boldsymbol{\acute{a}}$	H L	H L L	H L L	H L L	L
アクセント移動型	g	C0	C1	C1	C1	I1

4.1.2 アクセントクラスの生成

4 つ組 $u = \langle w, t, \boldsymbol{s}, \boldsymbol{a} \rangle$（以下，単に単語と呼ぶ）を 1 つの予測単位とする単語 N-gram モデルでは，アクセントが文脈によって大きく変化するため学習コーパスおよび辞書中にテストコーパス中の出現単語を直接的に網羅することは期待できない．そこで，

- 学習コーパス中に現れないアクセント型を辞書を用いて列挙し
- それらのアクセント型を持つ単語に適当な確率を割り当てる

モデルを作る．テストコーパスに含まれる単語が，学習コーパスに存在しない場合，コーパスに出現しない語のアクセントのふるまいを，アクセント特徴が同一である既知語の統計モデルから予測する．

1. アクセント特徴の組 f_{acc} をラベルとするクラス集合を用意する（図 4.2 c_1, c_2, c_3）．
2. 既知語の各語に対し，アクセント特徴の組によって，合致するクラスに組み入れる（u_1, u_2, u_3, u_4）．
3. 辞書語の各語に対して，文脈アクセントが辞書アクセントと同一であるとみなし，既知語と同様にアクセントクラスに組み入れる（u_5, u_6, u_7）．

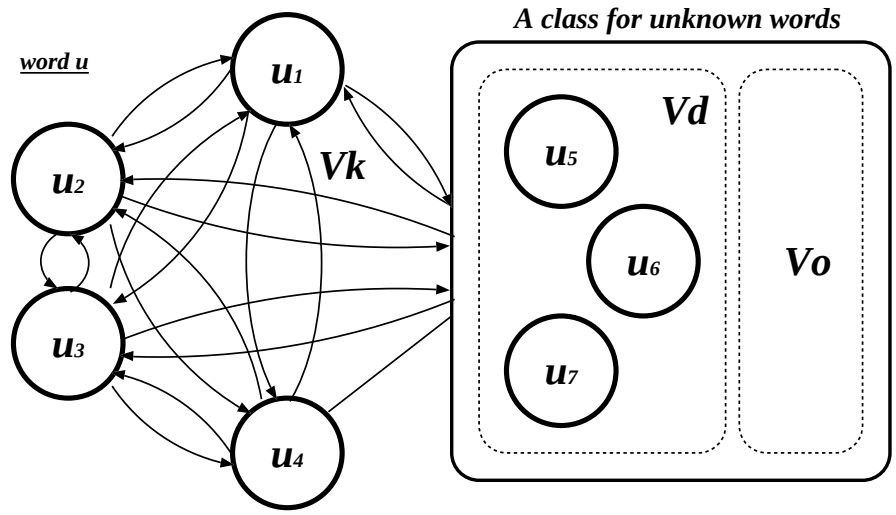

図 4.1: 単語 N-gram モデル

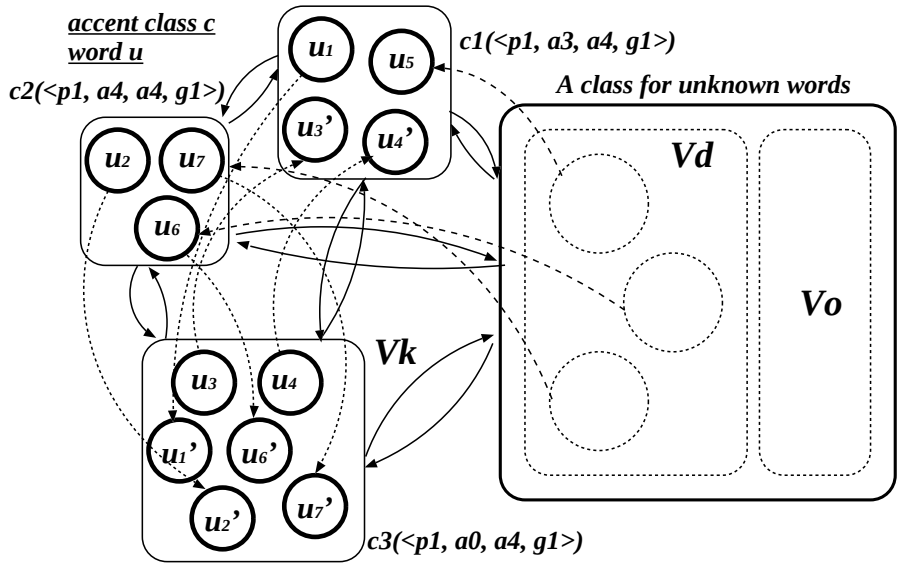

図 4.2: アクセントクラス N-gram モデル

4. 既知語および辞書語の各語に対して，取りうる文脈アクセント型をアクセント移動型から列挙し，文脈アクセント型のみを変化させた単語を生成し，アクセントクラスに組み入れる（u'_1, u'_2, u'_3, u'_4, u'_6, u'_7）.

5. アクセントクラス N-gram と N-gram の頻度を単語 N-gram と N-gram の表から計算する.

6. クラス内での単語の出現確率を計算する．手順 4 で追加される語に対しては非零確率を割り当てる.

4.2 アクセントクラス N-gram モデル

単語 N-gram とアクセントクラス N-gram モデルを構築し，これらを線形補間することで，最終的なモデルを構築する．4 つ組 N-gram モデルによる 4 つ組列 $u_1, u_2, \cdots, u_h$ の生成確率は，式 (4.1) によって表される.

$$P_u(u_1 u_2 \cdots u_h) = \prod_{i=1}^{h+1} P(u_i | u_{i-k} \cdots u_{i-2} u_{i-1}), \tag{4.1}$$

ここで $k = n-1$ であり，h は 4 つ組列の長さ，u_{h+1} は文末に対応する特別な記号を表す．この生成確率 $P_u(u_1 u_2 \cdots u_h)$ を最大化する 4 つ組列を最終的な解として選択する．同様に，クラス N-gram モデルでは，単語の生成確率は，クラスの生成確率とクラス内での単語の生成確率の積で表される.

$$\begin{aligned} & P_c(u_1 u_2 \cdots u_h) \\ = & \prod_{i=1}^{h+1} P(u_i | c(u_i)) P(c(u_i) | c(u_{i-k}) \cdots c(u_{i-2}) c(u_{i-1})), \end{aligned} \tag{4.2}$$

ここで $c(u)$ は単語 u を含むクラスを表す．単語 u のクラス c 内での生成確率は，単語 u の学習コーパス内の頻度を計数して求められる.

$$\begin{aligned} & P(u|c(u)) \\ = & \begin{cases} \alpha \dfrac{N(u, c(u))}{\sum_{u', N(u', c(u')) \neq 0} N(u', c(u'))}, & \text{if } N(u, c(u)) \neq 0 \\ (1-\alpha) \dfrac{1}{\sum_{u', N(u', c(u'))=0} 1} & \text{otherwise} \end{cases} \end{aligned} \tag{4.3}$$

ここで $0 \leq \alpha \leq 1$．この式で，コーパス中に出現する単語に対しては，クラス中での単語カウント $N(u, c(u))$ に比例した確率が与えられる．また，学習コーパス中に出現しない語に関しては，係数 α により非零の小さな値を確率として割り当てる．これらの 2 つのモデルを線形補間し，最終的なモデルを構築する．

$$\begin{aligned} & P(u_1 u_2 \cdots u_h) \\ = \ & \lambda_u P_u(u_1 u_2 \cdots u_h) + \lambda_c P_c(u_1 u_2 \cdots u_h). \end{aligned} \tag{4.4}$$

ここで，$0 \leq \{\lambda_u, \lambda_c\} \leq 1$, $\lambda_u + \lambda_c = 1$．単語 N-gram モデル（式 (4.1)）は $u = \langle w, t, \boldsymbol{s}, \boldsymbol{a} \rangle$ の組からなる単語から次の単語を予測し，学習コーパスで出現した単語列に対して尤もらしい単語列を与え，各単語に固有のアクセント，例外的なアクセントを反映することができる詳細なモデルである．アクセントクラス N-gram モデル（式 (4.3)）はアクセント特徴を同一にする単語集合から成るクラスから次のクラスを予測する．コーパスに出現しない単語に関しても，同じアクセント特徴をもつクラスの履歴を利用し，尤もらしいアクセントクラスから尤もらしい単語を予測する汎用的なモデルである．

4.2.1 一般的な語彙およびコーパスの追加

使用頻度の高い表現や，語彙の多い固有名詞表現に関しては，ルールを記述したプログラムを用意し，プログラムによりコーパスを生成した．これらに該当する表現としては下記のようなものがある．

- 数詞＋数詞
- 数詞＋数詞接尾
- 地名

例えば，数詞＋数詞の場合，下記のような表現が含まれる．全ての数詞の組を列挙して読み上げるわけにいかないので，自動的に全ての組に関して生成してある．

二百兆, 数詞, /*ni hya ku cho o:*/, /L H H L L/
四千億, 数詞, /*yo n se n o ku*/, /L H H L L L/
八十億, 数詞, /*ha chi ju u: o ku*/, /L H H L L L/
九十万, 数詞, /*kyu u ju u: ma n*/, /L H H H H L/

また，数詞＋数詞接尾に関しても同様に，数詞と数詞接尾の連接に関して読みとアクセントを与えてある．

五万, 数詞, /*go ma n*/, /L H H/ ‖ メートル, 接尾, /*me e: to ru*/, /H L L L/
六千, 数詞, /*ro ku se n*/, /L H H H/ ‖ 通, 接尾, /*tsu u:*/, /H H/

さらに，挨拶文や定型文のような使用頻度の高い文章や一般的な文章については別にコーパスを用意してある．最終的に，これらの一般言語モデル用コーパスと，対象話者コーパスを混合して用いる．

4.2.2 アクセント句，イントネーション句推定

アクセント句の推定は，読みおよびアクセント推定とは別に，決定木を用いたモデルにより行われる．読みおよびアクセント推定後に，決定木は注目した 4 つ組 u_i を含む前後 2 単語 $u_{i-2}, \cdots, u_{i+2}$ の計 5 単語の品詞および表層を入力として，この注目した 4 つ組 u_i の後にアクセント句境界が挿入されるか否かを文の先頭から逐次的に決定する．したがって，モデル構築時にはアクセント句境界が付与された言語コーパスから前後 2 単語を含む計 5 単語を用いて学習を行う．また，イントネーション句（呼気段落）は文中に含まれる句読点に対応して一意に決定する．

4.3 評価実験

4.3.1 実験用コーパス

精度評価実験に用いたのは，テレビニュースの書き起こし，電話応答文 など，多様な内容を含むコーパスである．各文は，予め人手によって形態素列に分割されており，各語は，表層 w，品詞 t，読み $\boldsymbol{s}$，アクセント $\boldsymbol{a}$ の 4 つ組から構成される．実験用コーパスの詳細を表 4.2 に示す．学習コーパスは約 60,000 文であり，辞書には 16 万語の単語が含まれる．テストコーパス 200 文には，10,826 個のモーラ毎の読み，アクセントが含まれる．用いた品詞セットは，名詞，固有名詞，動詞など 18 種類から構成されており，読みはアルファベット表記された a,i等からなる音素の列で記述されている．

また学習コーパスの内訳を表 4.3 に示す．対象話者からの音声書き起こしコーパスに加えて，別に用意した一般言語コーパスを加えてある．対象話者コーパスは，今回は十分な音声素片を収集する目的もあり 8851 文を収集したが，実用的な観点

表 4.2: 実験用コーパス

	文数	単語数	文字数
学習コーパス	59,351	1,045,803	1,755,004
テストコーパス	200	5,519	8,349

からは数百文程度が望ましいと考える．また，ルールを用いてプログラムにより自動的に生成したコーパスが，数詞＋数詞，数詞＋数詞接尾らを含めて約 2 万文相当用意した．また地名に関するコーパスを約 3,600 文用意してある．これら半自動的に構築されたコーパスに関しては，上記コーパス数の中には含めていない．数詞に関しては語彙が非常に限られており N-gram モデル上におけるサイズも小さいため，また地名モデルに関しては，辞書に近い位置づけであるためである．

表 4.3: 学習コーパス内訳

	文数
対象話者	8,851
一般言語コーパス	50,500
計	59,351
数詞モデル用コーパス	19,784
地名モデル用コーパス	3,677

4.3.2 モデル詳細

まずは，フロントエンドの推定精度を調査すべく実験を行った．複数の言語モデルを用意し比較を行った．表 4.4 に実験で用いたモデルを示す．

表 4.4: 言語モデル

	モデル	クラス数
$\mathcal{W}$	単語 N-gram モデル (ベースライン)	20,149
$\mathcal{AC}$	アクセントクラス N-gram モデル	1,062
$\mathcal{M}$	補間モデル (提案手法)	21,211
$\mathcal{R}$	ルールモデル	N/A
$\mathcal{C}$	自動クラスモデル	14,803

ベースラインを 4 つ組を 1 つの単語とする単語 N-gram モデルとし，アクセントクラス N-gram モデル，およびこれらの単語 N-gram とアクセントクラス

N-gram の 2 つのモデルを補間したモデルを用いる．また上記の 3 つのモデルに加え，ルールを用いた手法および，相互情報量によるクラスタリングを用いた方法 [38] についてもモデルを用意した．表 4.4 の右列は各モデルに含まれるクラスの数（$\mathcal{W}$ においては 1 単語を 1 クラスとみなした場合）である．

4.3.3 言語モデルの改善による精度評価実験

読みとアクセントの精度をモーラを単位とした誤り率（Mora Error Ratio :MER）で比較する．アクセントの誤り率は $\langle s, a\rangle$ の列を比較し，読みの誤り率はモーラごとの $\langle s\rangle$ の列を比較する．例えば,「日本人」$\langle$ 日本, 固有名詞, $/ni\ ho\ n/$, L H H$\rangle$ $\langle$ 人, 接尾, $/ji\ n/$, H L$\rangle$ に対して，アクセントの精度は（ni L, ho H, n H, ji H, n L）の列，読みの精度は（ni, ho, n, ji, n）の列から計算される．実験の結果を表 4.5 に示す．2 列目の H_{test} はテストコーパスにおける単語あたりのクロスエントロピーである．3 列目にアクセント推定の誤り率を示す．括弧内の数値は，ベースラインモデル $\mathcal{W}$ に対する改善率を示す．4 列目に読み推定の誤り率および，同じく改善率を示す．

表 4.5: モデル毎の予測力と精度

モデル	H_{test}	$MER\langle \boldsymbol{a}\rangle(\%)$		$MER\langle \boldsymbol{s}\rangle(\%)$	
$\mathcal{W}$	4.8962	9.64	-	1.20	-
$\mathcal{AC}$	5.5656	12.17	(-26.2)	2.05	(-70.8)
$\mathcal{M}$	4.9281	**8.01**	(16.9)	**0.94**	(21.6)
$\mathcal{R}$	*N/A*	*11.09*	(-15.0)	*1.28*	(-5.00)
$\mathcal{C}$	*4.8953*	*9.92*	(-2.90)	*1.18*	(1.66)

提案手法 $\mathcal{M}$ のモーラ毎のアクセント誤り率はベースライン $\mathcal{W}$ に対して 9.64% から 8.01% と 1.63 ポイント 改善しており，またモーラごとの読み誤り率は 1.20% から 0.94% と 0.26 ポイント 改善している．誤り率の改善としては，それぞれ 16.9% ，21.6% の改善となる．この結果，アクセント特徴でクラス化したモデルを加えることでアクセントだけではなく，読み推定の精度も改善していることがわかる．自動クラスタリングを用いた手法 $\mathcal{C}$ では，クロスエントロピーはわずかに改善したが，推定精度では，ベースラインモデルとほぼ同じ結果が得られた．アクセントクラスモデル単体を除くモデルでは，ルールベースの手法 $\mathcal{R}$ における推定精度を上回った．

実装上の観点から見ると，表 4.4 にあるように，補間モデル $\mathcal{M}$ はクラスの数が単語 N-gram モデル $\mathcal{W}$ に対して，わずか 5% の増加に留まっている．つまりメモリ量をほとんど増やすことなく，精度が向上している．

なお，直感的な理解のためには，単語を単位とした単語誤り率による評価も考慮されるべきであるが，日本語のような分かち書きされない言語の場合，単語単位の認定に曖昧性が残る．読みおよびアクセントを評価する際に，可能な限り曖昧性を含まない単位で評価を行うべく，モーラを単位とする誤り率を用いた．参考までに表 4.5 のモデル $\mathcal{W}$ の場合，単語単位の読みとアクセントと誤り率は，それぞれ 15.22%, 4.75%であった（単語境界が不一致の場合は不正解とする）．

4.3.4 英語を対象にした精度評価

本システムが他の言語に対しても有効かどうか調べるために，英語を対象に言語モデルを作り，精度の評価を行った．なお，日本語のように単語内に高低アクセントが存在し単語連鎖により高低アクセントが変化するようなことは無いが，英語の場合，文中の単語に強弱アクセントを付与する．つまり，各単語に関し $\langle$ 表層, 品詞, 読み, 強弱 $\rangle$ を推定する．ただし英語の場合，単語は空白文字によって区切られているため，単語境界は一意に決定される．

本来ならば英語の学習コーパスを作成して，そのコーパスに対して精度評価を行うべきであるが，英語のコーパスが十分でなかったため，ルールを基本とする現行の英語フロントエンドの出力を正解とみなし，どれだけの精度で現在のルールを基本とした出力と同じ出力が出来るかを測定した．英文約 75,000 文を入力とし，そのうちの 1%，744 文をテストコーパス，残り 99% を学習コーパスとした．表 4.6 に単語単位の誤り率（Word Error Ratio :WER）を示す．なお，現行英語システムは，単語の強弱を $[-2:5]$ の範囲の整数（8 段階）で出力するが，簡単のため，精度の判定は $-, 0, +$ の 3 値で評価した．

表 4.6: 英語を対象にした言語モデルの精度

	品詞 $WER(\%)$	読み $WER(\%)$	強弱 $WER(\%)$
英語	3.47	3.93	2.09

読み誤りの原因を調べたところ，誤り頻度全体の 19.8% が未知語の読みが付与できなかったことに起因するものであり，また既知語の誤りのうち 2 回以上現れる語は 98 語であり，出現頻度としては誤り全体の 50.3% を占める．このことから，英語に特徴的な誤りを修正，また英語の未知語読みモデルを構築することで，さらなる精度の向上が見込まれる．

4.4 実装

4.4.1 システムの実装

本システムは一般的なデスクトップコンピュータのみでなく，比較的計算能力およびメモリの限られた携帯端末やカーナビゲーションシステムでの動作も要求されている．また，日本語のみならず，他の言語でも動作するように設計することも必要とされている．したがって，システム要件の観点からは 計算リソースを小さくすること，対象言語に依存しないこと，ということが要求されている．実際の動作環境に関しては，4.4.1 節において述べる．

ランタイム処理において，フロントエンドおよびバックエンドは様々なプラットフォームで動作するように設計されており，AIX のようなサーバー用オペレーティングシステムから，Windows XP operating system のようなデスクトップ環境，また携帯端末等でも音声合成を行いたいというニーズがあり，Windows CE operating system 等の携帯端末上の環境でも動作するように設計されてある．CPU は POWER，x86, XScale 等をサポートする．実験は Windows XP 上にて行った．プログラミング言語は C/C++を用いた．また，携帯端末用途の CPU では，浮動小数点演算が行えない，プロセッサパワーやメモリの少ないといった制限がある．確率モデル上での計算はそのままでは多くの浮動小数点演算を行うため，これを避けるべく直接確率の計算を行わずに対数表を内部で含むこと，辞書や品詞ラベルの圧縮などの工夫を行っている．

メモリ効率向上および計算量削減

N-gram モデル自体は任意の長さ n の先行事象を扱うことが出来るが，実用上有効なのは $n = 3$ 程度までで，それ以上は緩やかに予測力が改善していく [38]．実装の都合上 $n = 2$ とした．辞書の圧縮方法は，ハッシュを用いる実装，Trie の実装の一つである Double Array[39] を用いる実装等，複数の実装を用意している．メモリ効率という面では，ハッシュを用いる実装が優れていた．

多言語対応

フロントエンドシステム全体としては，入力文に対して，正規化処理，読みおよびアクセント付与処理，句生成処理が行われ，システム後段の生成部の入力となるが，前段の正規化処理においては言語に依存した処理が必要となる．そのため，正規化処理においては言語毎に別個に用意する必要があり，日本語および英語に関しては，従来のシステムで用いたものを流用した []．一方，読みおよびアク

セント付与処理および句生成処理においては学習的な枠組みを用いて，言語依存性を極力排除することとした．読みおよびアクセント付与に関しては N-gram モデル（4.1.1 節）を，後続するアクセント句境界の決定は決定木を用いた学習的な枠組み（4.2.2 節）を用いており，対象言語に依存しない．また，プログラム上の内部表現としては UCS2 を利用しており，多くの言語で言語モデルの構築が可能である．

パフォーマンス

ランタイム処理（フロントエンド部）のパフォーマンスを専用のテストプログラムを用いて測定した．今回測定に用いたシステムは，組み込み用途の構成であり，辞書の実装は辞書サイズが最も小さくなるハッシュを用いた．開発時における実装環境およびパフォーマンスを表 4.7 にまとめる．使用メモリの計測では，一度に N-gram モデルにおける推定処理を行う単位として句読点から句読点までの文字列長が最大約 20 文字程度の入力テストパターンを用いた．また，実行速度は，約 324K 文字（10,000 文）のテストデータを用い，5 回の計測の平均をとった．

表 4.7: フロントエンド システム実装環境（組み込み用構成）

CPU	Pentium4 3.60GHz
OS	Windows XP SP3
Compiler	Microsoft C++ 8.0
RAM	100K バイト
ROM	4.75M バイト
実行速度	1,400 文字/秒

4.5 テキスト音声合成チューニングツール

一般的な言語モデルでは人間のように文の意味内容に応じて感情表現豊かなイントネーションをつけることはできない．また，言語モデル生成後，実際に音声を合成する際，漢字の読みの推定や標準語アクセントの再現において少数の誤りが生じることは避けられない．そのため，合成音声を自動的に生成した後に，人手によって読みやアクセントの誤りを修正し，文の意味内容に応じたイントネーションを与えるというチューニング作業を行う余地がある．しかしその作業は言語処理の流れや言語構造に対する専門知識が必要であり，時間と手間を要していた．そこで，言語処理に精通していないユーザーでも容易に編集作業が行えるチューニングツールの開発を行った．

4.5.1 要求される機能および実装上の制約

チューニングツールに関しては UI を用いるため，Adobe Flex を用いて開発を行った．基本的に Adobe Flash の動作する環境であればマルチプラットフォームで動くが，チューニングツールに関しても実験は Windows XP 上で行った．

作成したチューニングツールの概観を図 4.3 に示す．

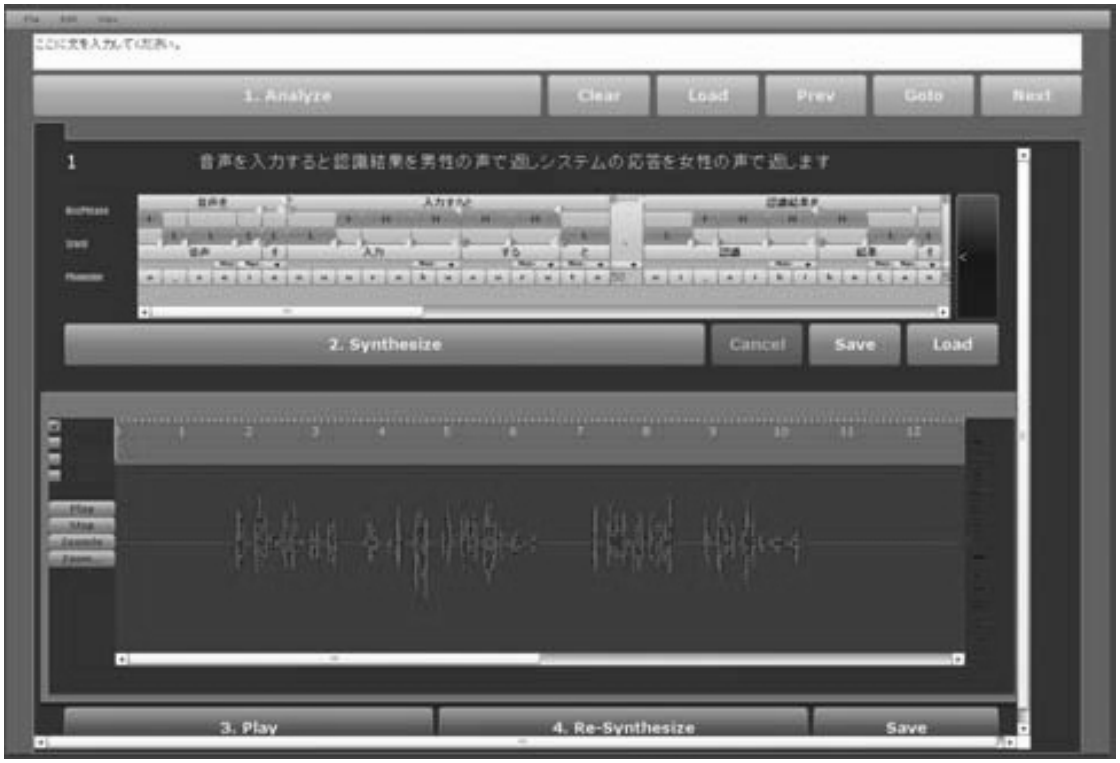

図 4.3: 合成音声チューニングツール概観

図 4.4: 合成音声チューニングツール修正部

画面上部に文章とそれに対応したアクセント句，アクセント，単語表層，品詞および読みを表示してあり，自由にこれらの値を修正，変更できる．画面下部には実際に音声を聞きながら修正作業を行うために，音声波形を表示するためのビューを用意してある．テキスト解析の結果である文構造がそのまま視覚的にも階層的に表示されるようにした（図 4.4）．

また，アクセントを視覚的に表現し，1 クリックで句のアクセントを変更できるようにした（図 4.5）．この時に選択可能なアクセントは，一般的に日本語で用いられる N モーラ M 型のいずれかにしか変更できないようにしており，1 クリックで句のアクセント型を変更できるようにしてある．図 4.5 の場合，左から 0 型，1 型，2 型である．

図 4.5: アクセント変更例

4.5.2 チューニングツール修正実験

チューニングツールによるテキスト解析結果の修正効率向上を実験によって検証する．被験者はアクセントの聴取やテキスト解析結果の修正の経験が全くない 1 名のユーザーとした．比較対象の旧ツールとしては通常のテキストエディタを用いる．どちらのツールも，表層，品詞，読み，アクセント，句境界を自由に編集可能である．通常のテキストエディタを用いる場合，図 4.6 の形式で保存された解析結果を用いる．解析結果は見易さを考慮して空白文字を挿入してあるが，任意個の空白文字を挿入しても構わない．形式は 4 行が 1 組となっており，上から各文の表層，品詞，読み，アクセントを表す．通常の単語境界は “|” で，アクセント句境界は “/” で区切り，イントネーション句の境界は読点または句点からなる語が対応する．各行の先頭は文番号を示す．任意のテキストエディタが利用可能であるが，ユーザーの選択により Windows に付属のテキストエディタ（メモ帳）を用いた．

```
1  |音声        |を     /入力          |する    |と     |、
1  |Mesi        |Josi   /Dosi          |Gobi    |Josi   |Kigo
1  |o,_n,se,i   |o      /nyu,u,ryo,ku  |su,ru   |to     |*
1  |1000        |0      /0111          |11      |1      |*
```

図 4.6: テキストエディタのための出力形式

実験では 1 セッション 15 分間で被験者が修正できる音声コーパスの分量とその精度を測定した．あらかじめ作成したコーパスを正解とし，モーラ単位のアクセントの一致率，および，句境界精度として単語境界からアクセント句境界を特定する精度（F 値）を測定した．セッションの順番は (1) 旧 (2) 新 (3) 旧 (4) 新とした．結果を表 4.8 に示す．なお，ここで用いたコーパスは 4.3.1 節で用いたコーパスとは別のコーパスを用いた．アクセント誤りは比較的少なく，修正対象のコーパス全体の平均アクセント誤り率（MER）は 3.96% であった．

表 4.8: チューニングツールの作業効率

	ツール	修正文数	文平均モーラ長	修正前		修正後	
				$\langle \boldsymbol{s}, \boldsymbol{a} \rangle$ MER(%)	$\boldsymbol{ap}$ ERR(%)	$\langle \boldsymbol{s}, \boldsymbol{a} \rangle$ MER(%)	$\boldsymbol{ap}$ ERR(%)
(1)	旧	9	29.8	6.72	17.65	4.48	3.45
(2)	新	16	31.8	1.31	10.42	2.81	2.27
(3)	旧	16	33.3	3.54	14.81	2.17	3.61
(4)	新	29	30.9	3.80	13.93	2.46	3.97

$\langle \boldsymbol{s}, \boldsymbol{a} \rangle$:読みとアクセントの組，$\boldsymbol{ap}$:アクセント句

若干の違いはあるが両ツールで精度は同程度であった一方，修正文数による作業効率には違いを見出すことができた．この被験者は音声アクセントの聴取経験がなかったために，音声を繰り返し聞きなおす時間が作業時間の大半を占めていた．セッションが進むにつれアクセントの判断が速くなり，ツールに無関係に作業効率が改善されている．しかし新ツールでは 1 回目のセッションで，旧ツールの 2 回目のセッションと同じ分量の修正を達成し，さらに 2 回目のセッションでは 1 回目のほぼ倍を修正することができた．これらの結果から新ツールの作業効率が良いことがわかる．精度においても，セッション (2) のアクセント精度以外は大きく改善している．たた，修正後のアクセントの精度は必ずしも 100% にならず，2.17 ～ 4.48% の誤りが含まれている．これらは，ユーザーが誤ったアクセントを付与したわけではなく，ユーザーによるアクセントに対する感覚の違いに起因するものであった．アクセント句に関しても同様であり，修正後も 2.27 ～ 3.97% のアクセント句において，正解コーパスとの差異が認められた．

また，網羅的に音韻，韻律情報のチェックを行う際の作業効率を調査するため，ランタイム時だけでなく，現在既に対象話者コーパスとして用意されている 8,851 文の分量のある音声コーパスについてすべて音声を聴取し，テキスト解析結果の修正を行う実験も行った．この音声コーパスは既にその大部分につき別のツールを用いてアクセントの修正を行ったものであったが，今回の作業によってさらに 2.77% のモーラにつきアクセントを修正，14.5% のアクセント句境界を単語境界

に修正できた．このツールを用いることで大規模の音声コーパス修正を短期間に一人で行うことができ，修正内容の一貫性の向上も期待できる．

4.6 関連研究

日本語に対するアクセント推定としては，ルールを用いた手法 [32] および，学習可能なモデルを使った手法 [28] [40] 等が提案されている．従来研究 [41] により，形態素単位でアクセント推定を行う場合，形態素解析，読み推定，アクセント推定を逐次的に行うより同時に推定を行ったほうが精度が良いという知見を得ており，本論文でも同時に推定する手法を用いたが，精度という面では，CRF を用いた手法 [40] により，アクセント句が与えられた状態で約 95%（単語単位）のアクセント推定精度を得ており，学習器および入力素性に関しては改善の余地が大きい．また，一般的なモデルの改善という観点では，単語クラスタリングに関する研究 [42]，文脈情報を使ったクラスの作成方法に関しての報告 [43]，統計モデルと文法を用いたモデルのハイブリッドモデル [44] などが行われており，改善の余地がある．

一方，音声合成システムのチューニングツールとしては，いくつかの商用音声合成システムがツールを提供している．Nuance 社の PromptSculptor[45] は，読みや強調箇所を変更するだけではなく，ピッチや話速といった制御情報を与えることができる．また，Luquendo 社の TTS Director[46] は，息継ぎや咳といったパラ言語情報をタグ付けできるようになっている．4.5 章にて述べたチューニングツールは，日本語特有のアクセント型を考慮したアクセント設定機能や，単語境界および句境界の簡便な設定機能を提供する．利用対象者によって必要な機能は異なると思われるが，比較的音声の知識に精通していないユーザーが容易に合成音声を修正および調整できる仕組みとして有効であると考える．

4.7 本章のまとめ

テストコーパスに含まれる様々な分野の文章を対象とした場合，日本語の読みおよびアクセントの推定精度は，それぞれ約 99%，92% であった．

日本語のアクセントの特徴を用いることで，ベースモデルに対して精度は改善したが，ユニバーサルな分野を対象とした場合の読み上げシステムとしては，現在の自動推定の精度は十分ではない．システム全体としては，補正が容易なチューニングツールを用いることで，効率的な精度の改善を図っているが，さらなる推定精度向上のため，推定モデルの一層の改善が必要である．同時に，話者の特徴を

反映するために，どのような対象話者言語コーパスを学習データとして加える必要があるかといった，学習データ選択についても考慮する必要がある．またチューニングツールにより効率的なチューニングが可能であることが分かったが，一方，ユーザーが正しいと判断したアクセントと，正解とするアクセントとの間には約 3% の差異があり，異なり原因を解析することで，さらなる精度の向上が見込める．

ユーザーの意図した話し方での発話を容易に実現できる音声合成システムを開発した．本研究では，特にユーザーの特徴を反映する要素として，読みおよびアクセント，およびアクセント句に着目し，ユーザーが自由に修正，変更ができるようなシステムを構築した．モデルの修正，変更が容易な，統計的フロントエンドを開発し，精度の面においては，言語モデルを改善することで，より良い読みやアクセントを付与することができるようになった．また，読みやアクセント誤りの修正や，任意の読みやアクセントを付与するためのチューニングツールを作成し，有効に修正，変更が行えることを確認した．

第5章　音声検索語検出の効率化

企業のコールセンターでは，音声通話に含まれる特定のキーワードを含む発言をチェックするコールモニタリング業務によりコールセンターの品質向上を図っている．一方，一部のコールセンターでは，大語彙連続音声認識技術の利用により日々大量に蓄積される音声データに対するキーワード検索が可能となってきた．このような現場では，検索キーワードや業務内容に応じて,「再現率を重視したい」,「適合率を重視したい」といった要望がある．本論文では，認識単位の異なる二種類の音声認識システムを用いて，各検出区間に対して信頼度を与え，検索時に再現率・適合率のバランスを調整できるシステムを提案する．

5.1　はじめに

キーワード検出では，一般的に単語を認識単位とする大語彙連続音声認識 (以下，単語音声認識) の書き起こし結果に対して文字列での比較または単語列での比較を行うが，未知語 (辞書に含まない語) や認識誤りへの対応として，サブワードや音素・音節またはそれに準じる単位での音声認識 (以下，音節音声認識) の結果に対して単位列での比較を行う方法 [47][48][49] が知られている．また，単語単位の音声認識において直接的に信頼度の算出が困難な場合，音素や音節を認識単位とした認識システムを用いて信頼度を推定する方法 [50] も提案されており，未知語の棄却に良い性能を示している．音節音声認識は単語音声認識を用いる手法と比較すると，言語情報を利用していないためキーワード検出性能が低い [51] が，各種認識誤りに対応した距離を定義することで再現率・適合率の調整が可能である．距離の定義としては，編集距離 [47]・Word Confusion Network (WCN)[48] により距離を定義する方法，音素ラティスから生成される N-gram インデックスと編集距離を組み合わせた方法 [49] 等が知られている．これら両方の利点・欠点を考慮して,「単語音声認識システム」と「未知語を扱うことのできる単語・音節音声認識システム」を併用したシステム [51] も提案されており，単語音声認識と音節音声認識の併用が未知語・既知語を含めた検出精度の向上に役立つことが分かっている．システム統合という観点からは，信頼度の高い認識区間を推定するために複数の音声認識システムの認識結果単語列の論理積部分を抽出する方法 [52] や，音

声認識精度を向上させるためのシステム統合手法 (ROVER 法)[13]，各システムから得られるスコアの正規化方法 [53] などについて研究が行われており，それぞれ単体のシステムを用いた場合に比べ良い性能を示している．ただし，5 ～数十種類のパラメータの異なる音声認識器を用いており，計算コストが高い．

本論文では，音声検索語検出システムの利便性を高めるため,「計算量を大幅に増やさない」「作業効率を向上させる」「高い適合率に調整が可能である」ことを目的として，単語音声認識と音節音声認識を組み合わせたシステムを提案する．提案法では音節音声認識システムを単語音声認識システムにより検出された区間の信頼度の計算に用いることで，単語音声認識単体ではカバーできなかった適合率を優先させたキーワード検出を実現する．具体的には，(1) 単語音声認識結果を用いてキーワード文字列の一致する区間を検出し，(2) それら検出区間に対応する音節音声認識結果とキーワード音節列を用いて信頼度を計算する．従来，音節音声認識は未知語への対処として用いられることが多かったが，既知語 (辞書に含まれる語) に信頼度を与えるシステムとして用いる．信頼度の与え方として，単語グラフ事後確率 [54]，N-best [55]，音響尤度，言語尤度を利用した方法など様々な指標が提案され，その有効性が検証されている [56][57] が，中でも単語グラフを用いた単語事後確率による信頼度付与および N-best を用いた単語事後確率を求める方法の性能が高いとされている [58][59]．本研究においても性能が高く，単語グラフを直接扱うより計算コストの低い，N-best を用いた音節事後確率による方法を用いて信頼度の計算を行った．また計算量の観点から，計算量の少ない，N-best の N 位以内に含まれるかどうか (ランキング) を信頼度とした指標についても検討を行った．実験ではまず，ベースシステムとして単語音声認識 1-best を用いた場合と，単語音声認識 1-best に音節音声認識 N-best を用いて信頼度計算を行った場合とを比較し，二つの音声認識器の組み合わせが有効であることを示す．次に，ベースシステムとして単語音声認識 N-best を用いて信頼度計算を行った場合と，このベースシステムの結果に対してさらに音節音声認識 N-best を用いて信頼度計算を行った場合とを比較し，ベースのシステムによらず二つの音声認識器の組み合わせが有効であることを示す．さらに単語音声認識 N-best から計算される信頼度と音節音声認識 N-best から計算される信頼度を組み合わせることにより，作業効率の良い信頼度を付与できることを示す．有効性の評価は，各システムで得られる適合率の範囲と，実際のコールモニタリング作業を模した作業効率 (信頼度上位から音声を聴取した際に誤った音声が含まれる割合) により行った．実験の結果，組み合わせにより適合率の上限は，それぞれのベースシステムに比べて 0.248, 0.035 ポイント向上し，また誤検出の割合を再現率 0.5 のポイントにおいてそれぞれ 76.7%, 42.1% 減らすことができた．

5.2 大語彙連続音声認識と音節音声認識を併用したキーワード検出

音声検索語検出システムでは単なる音声の書き起こしと異なり，未知語や認識誤り対策のため音節音声認識が併用されることが多い [51][60] [61]．本研究では，このように二種類の音声認識システムが利用可能な状況で，適合率を高めることのできるシステムについて検討を行う．音節音声認識は未知語検出に対して適用するのではなく，既知語に対する検出区間への信頼度付与に音節音声認識結果を用いるため，本論文では既知語のみを検出対象にし未知語を対象としない．システムの概要を図 5.1 に示す．システムは大きく分けて，検出対象データに対して音声認識を行い各キーワードが含まれているかを示すインデックスを付与するインデックス作成部と入力されたキーワードを元に検出対象データからキーワードの検出を行うキーワード検出部から構成されている．

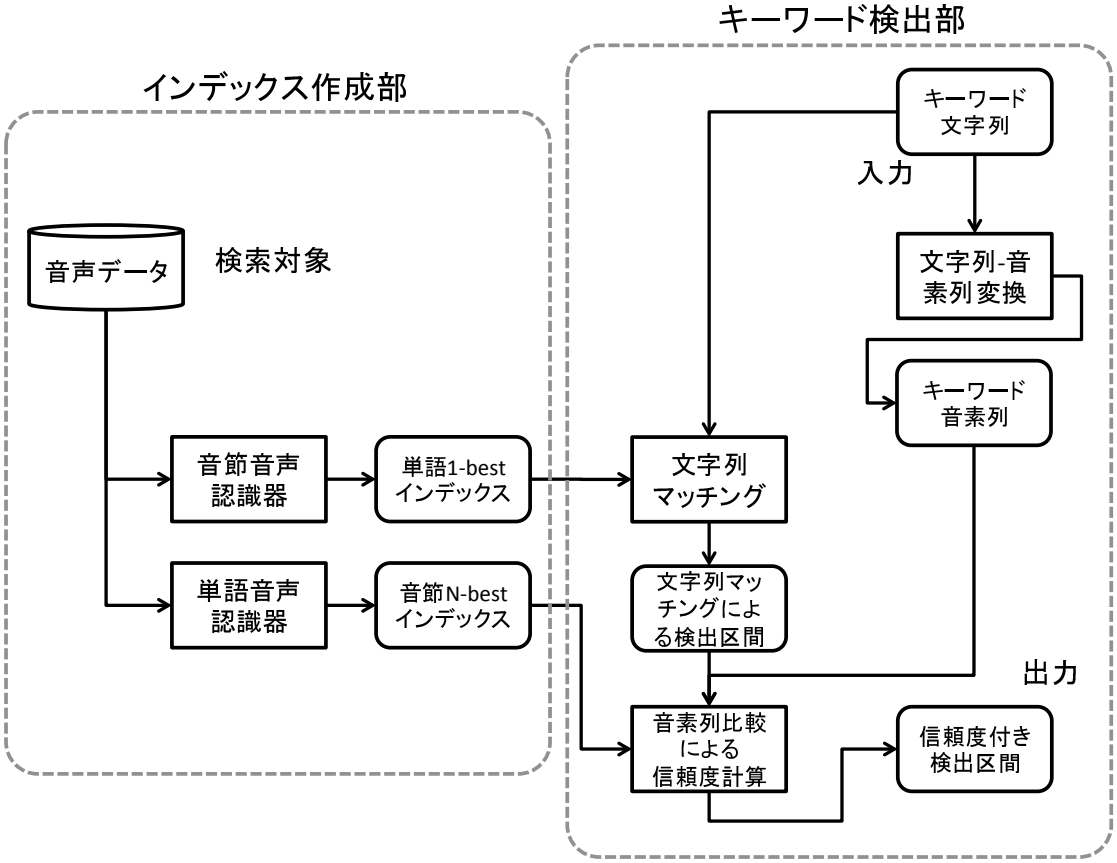

図 5.1: 大語彙連続音声認識と音節 N-best 音声認識を用いたキーワード検出

5.2.1 インデックス作成

同一のモデル構築用の音声コーパスから単位の異なる二種類の音声認識システムを構築する．音声コーパスには音声に対応する単語列が付与されており，辞書と音声コーパスを用いて音声認識モデルを構築できる．また，この音声コーパスと辞書を用いて，音声コーパスに対して音節列を付与し音節単位の音声認識モデルを構築できる．音声認識モデルの音響モデル・言語モデルをそれぞれのシステムで個別に構築することもできるが，本研究では同一の音響モデルを用い，言語モデルのみを単位の異なる二種類の言語コーパスから構築する．下記に二種類の音声認識システムを示す．

$\mathcal{W}$ 単語を認識単位とした音声認識システム

単語は漢数字を含む漢字と平仮名，片仮名，アルファベットから構成される．
例: 右 , クリック, ですね

$\mathcal{S}$ 音節を認識単位とした音声認識システム

音節は日本語の音節 451 種類から構成される．一般的な日本語音節に対し，長音化した母音，促音 Q を含む音節は別の音節として取り扱う．
例: *mi* , *gi* , *ku* , *ri* , *Qku* , *de* , *su* , *ne*

検出対象の音声を単語単位の音声認識システム $\mathcal{W}$ および音節単位の音声認識システム $\mathcal{S}$ を用いて音声認識を行う．システム $\mathcal{W}$ の認識結果は 1-best のみの出力として単語列 w^1 を，システム $\mathcal{S}$ は N-best の認識結果 $s^1 \dots s^N$ を出力する．ここで N-best は検出対象音声から音声認識システムを用いて得られる N 個の仮説の集合である．N は得られる仮説の個数の最大値とし，N-best 出力は認識尤度の降順に出力されているものとする．また，キーワードは，文字列 w_K (例:「クリック」) で与えられ，辞書またはテキスト音素列変換 [62][63] により音節列 s_K (例:*ku ri Qku*) に変換され信頼度計算装置に入力される．本実験では辞書を用いて変換を行ってある．各認識結果は，単語アラインメント結果 $d = \langle$ 開始時間 t_{beg}, 終了時間 $t_{end}\rangle$ の列からなり，これらを元に検出用のインデックスを作成する．

5.2.2 キーワード検出

次に以下の手順でキーワードに一致する音声区間を検出する (図 5.2)．キーワードは，文字列 w_K および音節列 s_K で与えられるものとする．

1. 文字列による検出

   キーワード文字列 w_K と単語音声認識システム $\mathcal{W}$ の認識結果 w^1 とを比較し，

一致する区間を検出する．一致する区間のタイムスタンプは検出用インデックスに含まれる単語アラインメント結果 $d = \langle t_{beg}, t_{end} \rangle$ から得られる最終的に M^{w_K} 個の一致した検出区間リスト $D(w_K) = d_1^{w_K} \ldots d_{M^{w_K}}^{w_K}$ を得る．

2. **音節 N-best による信頼度評価**
   単語音声認識の検出区間 $d_i^{w_K} (1 \leq i \leq M^{w_K})$ に含まれる音節音声認識システム $\mathcal{S}$ の認識結果 $s^1 \cdots s^N$ とキーワード音節列 s_K を用いて検出区間 $d_i^{w_K}$ の信頼度 $CM(s_K, d_i^{w_K})$ を求める．検出区間の信頼度の計算は 5.2.3 節に記す 2 種類の信頼度を用いて行う．

3. **閾値による抽出**
   検出時には閾値 T を与え，信頼度 $CM(s_K, d_i^{w_K}) \geq T$ となる検出区間のみを最終的な検出区間とする．

なお，文字列検出のタイムスタンプの開始時間と終了時間は単語アラインメントの誤差を考慮してわずかな時間 Δ だけ広げてある．以降の実験においては Δ の幅は検出評価用データにおける 1 音節あたりの継続長が 0.15 秒であったことから 2 音節分の 0.3 秒とした．

5.2.3 信頼度

信頼度として N-best を用いた二種類の尺度について検討を行う．以下に，本研究で用いる二種類の尺度について示す．

ランキング

計算量の少ない N-best の評価方法として N-best の N を変えたときに N-best に含まれるかどうかを判断する．1-best から順にキーワードとの比較を行うだけなので，最も検出時間が短くなると期待できる．検出区間 d_i の評価は音節音声認識 N-best の結果 $s^1 \ldots s^N$ とキーワード s_K の一致する最小の順位 $n (1 \leq n \leq N)$ を用いる (式 5.1)．$\mathcal{L}_{d_i}^{s^n}$ は $d_i^{w_K} (1 \leq i \leq M^{w_K})$ 中に含まれる N-best の n 番目の部分音節列集合を表し，δ は一致するときに 1，それ以外のときに 0 を返すクロネッカーのデルタである．

$$
\begin{aligned}
& CM_{Rank}(s_K, d_i^{w_K}) \\
&= 1 - \frac{min_{n=1}^{N}(n \cdot \delta(s_K, \mathcal{L}_{d_i^{w_K}}^{s^n})) - 1}{N}
\end{aligned}
\tag{5.1}
$$

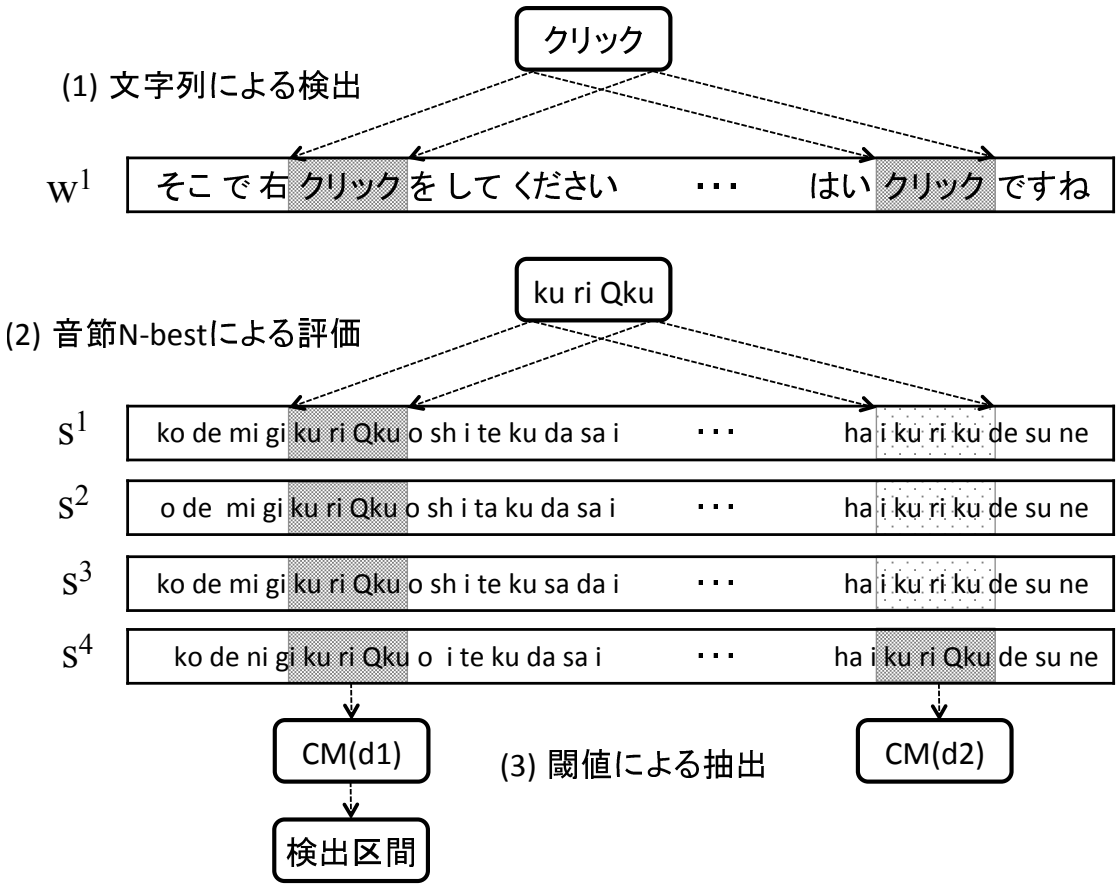

図 5.2: キーワード検出

事後確率

N-best を用いた事後確率の値を信頼度とする．あるキーワード s_K が検出区間 $d_i^{w_K}(1 \leq i \leq M^{w_K})$ に出現すると仮定したときの文の事後確率を，区間 $d_i^{w_K}$ に s_K を含む文の生成確率を仮説集合全体 N-best の生成確率の和で除したものとして定義する (式 5.2)．x は入力音声を表し，$p(s)$ は音節を単位とした言語モデルによる尤度，$p(x|s)$ は音響モデルによる尤度を表す．N-best 全体から事後確率が計算されるので，ランキングを用いた信頼度より計算量が多い．

$$CM_{Post}(s_K, d_i^{w_K}) = \frac{\sum_{n=1}^{N} p(x|s^n) \cdot p(s^n) \cdot \delta(s_K, \mathcal{L}_{d_i^{w_K}}^{s^n})}{\sum_{n=1}^{N} p(x|s^n) \cdot p(s^n)} \tag{5.2}$$

5.3 評価実験

単語音声認識システムおよび音節音声認識システムを構築し，実データを用いて，単語音声認識システムの結果に対し音節音声認識システムの認識結果を用いて信頼度を与えることの有効性を検証する．

5.3.1 検出評価用データ

電話録音データを用いて検出評価を行った．各音声ファイルは 8khz / 16bit サンプリングで録音され，二話者の音声は予め別チャンネルで保存されているステレオ音声データである（表 5.1）．発話時間はパワーヒストグラムを用いた発話区間検出器によって計算した．音声には人手による書き起こしによる正解が付与されており，音声認識に用いられる言語モデルを用いて単語分割されている．

表 5.1: 検出評価用データ

通話数	100 通話
録音時間	29.97 時間
発話時間	13.57 時間
発話区間数	21853 セグメント
単語数	179K 語

5.3.2 キーワードおよび評価方法

キーワードは未知語を含まず，長さ 1 ～ 11 文字 (1 ～ 12 音節) からなる 40 語で構成される．キーワードの長さ毎の語数は長さ 1 から 11 まで順に $(2, 9, 5, 4, 6, 6, 1, 1, 2, 3, 1)$ である．各キーワードは予め文字列に対する音節列を与えてある．キーワードの例を表 5.2 に示す．検出評価用データにはのべ 3248 個のキーワードが含まれている．

表 5.2: キーワード例

表記 w_K	音節表記 s_K
値段	*ne da n*
東京	*to: kyo:*
おはようございます	*o ha yo: go za i ma su*
よろしくお願いします	*yo ro shi ku o ne ga i shi ma su*

最終的な評価は作業効率や検出に要する検出時間を含めて行うが，本章では基本的な検出性能である再現率，適合率，およびこれら 2 つを組み合わせた検出性能を示す F 値 (式 5.3) により評価を行う．キーワードが正しく検出されたかどうかの判定は発話単位で行う．検出評価用データの各発話ごとに書き起こし文字列および音節列が付与されており，それを元にキーワードが含まれているかどうかをあらかじめ各発話に対して正解ラベルを付与しておく．システムが検出した区間がラベル付けられた発話区間に含まれていれば正しく検出されたものとする．検出評価用データでの 1 発話区間の平均の長さは 2.24 秒であり，1 発話区間に同一キーワードが複数回含まれる可能性は小さいことから発話ごとに検出の判定を行うこととした．

$$\text{再現率} = \frac{\text{正しく検出された発話数}}{\text{検出評価用データ中の正解ラベル付き発話数}}$$

$$\text{適合率} = \frac{\text{正しく検出された発話数}}{\text{システムにより検出された発話数}}$$

$$\text{F 値} = \frac{2 \cdot \text{適合率} \cdot \text{再現率}}{\text{適合率} + \text{再現率}} \tag{5.3}$$

5.3.3 音声認識

音声認識システムを用いて検出評価用データを単語列および音節列に変換しておく．音声認識モデルは 150 時間の電話会話の録音音声を元に作成されており，GMM-HMM を boosted MMI で識別学習した音響モデルである．単語音声認識システムの言語モデルは単語 N-gram，音節音声認識システムの言語モデルは音節 N-gram を用いて構築してある．単語 1-best による検出評価用データの音声認識率は，文字誤り率で 20.4% である．また音節認識モデルを用いた出力として，信頼度計算に必要な N-best を出力しておく．また比較のため単語 N-best についても出力しておく．一発話あたりの N-best の出力数は，単語音声認識では (最小値, 最大値, 平均) = $(1, 877, 91.6)$ 個，また音節音声認識では (最小値, 最大値, 平均) = $(1, 495, 36.6)$ 個であった．表 5.3 に音声「右クリックですね」を入力としたときの単語音声認識 N-best 出力，また表 5.4 に音節音声認識出力の結果を示す．誤りなく認識されているのはそれぞれ w^1 および s^4 となる．

表 5.3: 単語 N-best 出力の例

n	認識単語列 w^n	単語列対数認識尤度
1	右 クリック ですね	−81.1235
2	右 クリック です よね	−81.3125
3	右 クリック です よ ね	−81.6375
4	右 クリック でしょう ね	−81.8335
5	右 クリック です よ	−81.9705

表 5.4: 音節 N-best 出力の例

n	認識単語列 s^n	音節列対数認識尤度
1	*mi gi ku ri Qku su su me*	−80.6225
2	*mi gi ku ri Qku o su su me*	−80.7070
3	*mi gi ku ri Qku su de*	−80.7140
4	*mi gi ku ri Qku de su ne*	−80.7615
5	*mi gi ku ri Qku de su ne:*	−81.4520

5.3.4 ランキングを信頼度として用いた実験

2 種類の信頼度 $CM_{Rank}(s_K, d^{w_K})$，$CM_{Post}(s_K, d^{w_K})$ のうち，まずランキング $CM_{Rank}(s_K, d^{w_K})$ を用いた場合の提案手法の評価を行う．キーワード文字列 w_K が単語音声認識の 1-best である $\boldsymbol{W}_{(1)}^{Rank}$ に一致した区間に対して，音節音声認識 N-best によって計算される $\boldsymbol{S}_{(T)}^{Rank}$ で信頼度付与した結果を示す．比較として，単語音声認識 N-best と音節音声認識 N-best 単体での評価も行う．単語音声認識 N-best と音節音声認識 N-best に関してはそれぞれ $CM_{Rank}(w_K, d^{w_K})$, $CM_{Rank}(s_K, d^{s_K})$ による評価を行い，単語音声認識システム単体でどのような再現率・適合率の傾向があるか調べた．

式 5.1 と同様に，

$$CM_{Rank}(w_K, d_i^{w_K}) = 1 - \frac{min_{n=1}^{N}(n \cdot \delta(w_K, \mathcal{L}_{d_i^{w_K}}^{w^n})) - 1}{N}$$

および

$$CM_{Rank}(s_K, d_i^{s_K}) = 1 - \frac{min_{n=1}^{N}(n \cdot \delta(s_K, \mathcal{L}_{d_i^{s_K}}^{s^n})) - 1}{N}$$

と定義される．

$d_i^{w_K}, d_i^{s_K}$ はそれぞれ s_K を音節 N-best 中に含む区間，w_K を単語 N-best 中に含む区間である．それぞれの方法で得られる検出区間の集合を以下のように示す．

$\boldsymbol{W}_{(T)}^{Rank}$ 単語音声認識 N-best

全検出評価用データに対する単語音声認識 N-best 出力に対し，閾値を n 位に対応する $T = 1 - (n-1)/N$ とした場合に $CM_{Rank}(w_K, d^{w_K}) \geq T$ となる音声区間の集合

$\boldsymbol{S}_{(T)}^{Rank}$ 音節音声認識 N-best
全検出評価用データに対する音節音声認識 N-best 出力に対し，閾値を n 位に対応する $T = 1 - (n-1)/N$ とした場合に $CM_{Rank}(s_K, d^{s_K}) \geq T$ となる音声区間の集合

$\boldsymbol{C}_{(T)}^{Rank}$ 単語・音節音声認識組み合わせ (提案手法)
$\boldsymbol{C}_{(T)}^{Rank} = \boldsymbol{W}_{(1)}^{Rank} \cap \boldsymbol{S}_{(T)}^{Rank}$．単語認識結果がキーワード 1-best 文字列に一致し，かつ音節音声認識 N-best 出力に対し，閾値を n 位に対応する $T = 1 - (n-1)/N$ とした場合に $CM_{Rank}(s_K, d^{w_K}) \geq T$ となる音声区間の集合

表 5.5 に単語音声認識 $\boldsymbol{W}^{Rank}$，音節音声認識 $\boldsymbol{S}^{Rank}$ および提案手法 $\boldsymbol{C}^{Rank}$ において閾値を最大・最小とした場合の再現率・適合率・F 値を示す．$\boldsymbol{W}_{(1)}^{Rank}$ と $\boldsymbol{S}_{(1)}^{Rank}$ の点がそれぞれ単語音声認識 1-best 出力および音節音声認識 1-best 出力に対応する再現率・適合率である．$\boldsymbol{W}_{(1)}^{Rank}$ と $\boldsymbol{S}_{(1)}^{Rank}$ の点を比較すると音節音声認識の F 値 0.526 に対し単語音声認識の F 値 0.688 と 0.162 ポイント高くなっており，従来研究と同様，単語音声認識の単語検出性能は音節単位での音声認識によるキーワード検出性能に比べ高い．$\boldsymbol{W}^{Rank}$ に注目すると，$\boldsymbol{W}_{(0)}^{Rank}$ は $\boldsymbol{W}_{(1)}^{Rank}$ に比べて，再現率は 0.154 ポイント上昇したが適合率は 0.401 ポイント下がり，結果として F 値は 0.314 ポイント下がっている．同様に音節音声認識の F 値も閾値の変化 $T = 1 \rightharpoonup 0$ にしたがい，0.213 ポイント下がる．

図 5.3 に各モデルにおいて閾値を $T = 1 \rightharpoonup 0$ と段階的に変化させていった結果を示している．なお，図中の $\boldsymbol{W}^{Rank}$ において $T = 1$ 以外の点も図中にプロットしてあるが，$\boldsymbol{C}^{Rank}$ を求める際には 1-best にあたる $\boldsymbol{W}_{(1)}^{Rank}$ 以外の点は用いていない．音節音声認識結果の 1-best を閾値とした $\boldsymbol{C}_{(1)}^{Rank}$ は $\boldsymbol{W}_{(1)}^{Rank}$ に比べ F 値が向上し (0.688 → 0.717)，適合率の上限を 0.190 ポイント向上させることができている．

表 5.5: $CM_{Rank}(s_K, d^{w_K})$ を用いた $\boldsymbol{C}^{Rank}$ の再現率・適合率

	$T = 1$			$T = 0$		
モデル	再現率	適合率	F 値	再現率	適合率	F 値
$\boldsymbol{W}^{Rank}$	**0.746**	0.637	0.688	*0.900*	*0.236*	*0.374*
$\boldsymbol{S}^{Rank}$	0.710	0.418	0.526	0.849	0.189	0.313
$\boldsymbol{C}^{Rank}$	0.634	**0.827**	**0.717**	**0.706**	**0.735**	**0.720**

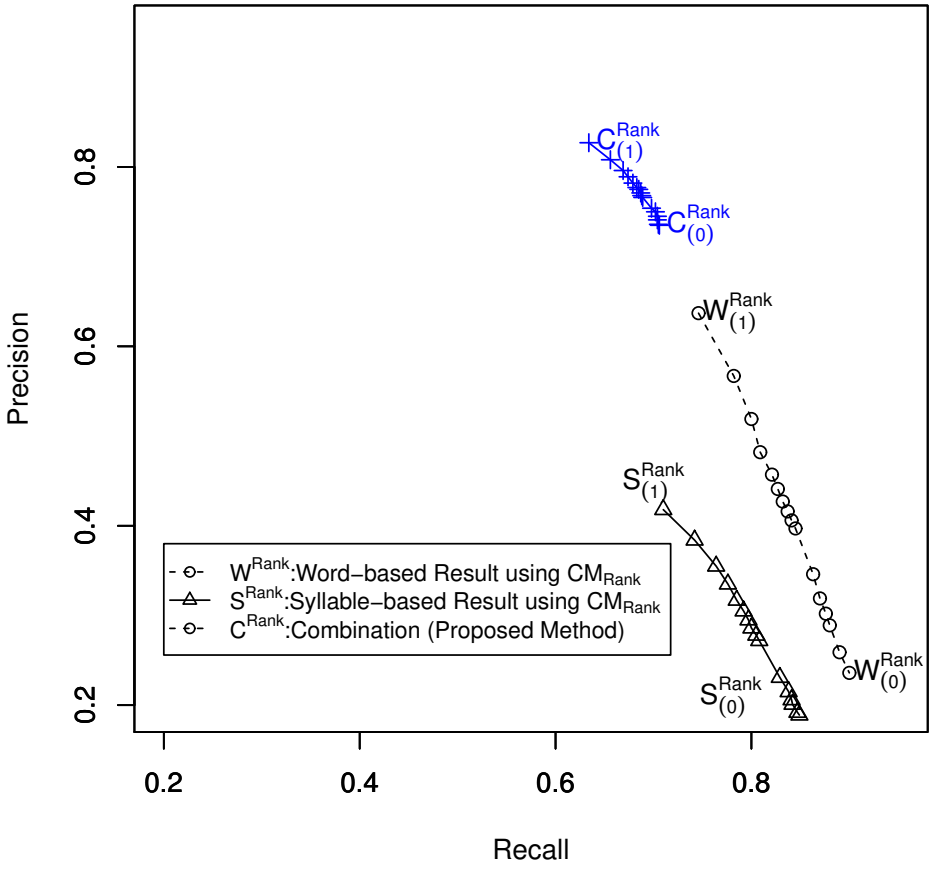

図 5.3: $\boldsymbol{C}^{Rank}$ の再現率/適合率曲線

5.3.5 事後確率を信頼度として用いた実験

次にランキング $CM_{Rank}(s_K, d^{w_K})$ の代わりに音節列事後確率 $CM_{Post}(s_K, d^{w_K})$ を信頼度として定義し，同様に実験を行った．キーワード文字列 w_K が単語音声認識の 1-best である $\boldsymbol{W}_{(1)}^{Rank}$ に対して一致した音声区間に対して，音節 N-best を用いた事後確率で信頼度付与した結果を示す．音節の事後確率 $\boldsymbol{S}_{(T)}^{Post}$，および $\boldsymbol{W}_{(1)}^{Rank}$ と $\boldsymbol{S}_{(T)}^{Post}$ から計算される検出区間の集合を以下のように示す．

音節モデルの事後確率による信頼度は

$$CM_{Post}(s_K, d_i^{s_K}) = \frac{\sum_{n=1}^{N} p(x|s^n) \cdot p(s^n) \cdot \delta(s_K, \mathcal{L}_{d_i^{s_K}}^{s^n})}{\sum_{n=1}^{N} p(x|s^n) \cdot p(s^n)}$$

として定義される．

$\boldsymbol{S}_{(T)}^{Post}$ 音節音声認識 N-best
全検出評価用データに対する各音節音声認識 N-best 出力に対し，音節列事後確率が $CM_{Post}(s_K, d^{s_K}) \geq T$ となる音声区間の集合．

$\boldsymbol{C}_{(T)}^{Post}$ 単語・音節音声認識組み合わせ (提案手法)
$\boldsymbol{C}_{(T)}^{Post} = \boldsymbol{W}_{(1)}^{Rank} \cap \boldsymbol{S}_{(T)}^{Post}$．単語認識結果がキーワード 1-best 文字列に一致し，かつ音節認識 N-best 出力から得られる音節列事後確 $CM_{Post}(s_K, d^{w_K}) \geq T$ となる音声区間の集合．

信頼度として単語事後確率 $CM_{Post}(s_K, d^{w_K})$ を用いた場合の音節音声認識 $\boldsymbol{S}^{Post}$ および組み合わせ提案手法 $\boldsymbol{C}^{Post}$ のそれぞれの閾値を $T = 1, 0$ としたときの再現率・適合率・F 値を表 5.6 の $\boldsymbol{S}^{Post}$，$\boldsymbol{C}^{Post}$ 行に示す．$\boldsymbol{C}^{Post}$ の適合率の上限は $\boldsymbol{W}^{Rank}$ に比べ 0.248 ポイント高く，改善している．また，図 5.3 と同様に図 5.4 に閾値を $T = 1 \rightharpoonup 0$ と段階的に変化させていった結果を示す．閾値 T が 1 に近づくとともに適合率は若干下がる傾向にあり，適合率が最高になったのは閾値が $T = 0.94$ の点で適合率は 0.888 であり，この点での再現率・適合率・F 値を表 5.6 の下 2 行に示す．ランキング $CM_{Rank}(s_K, d^{w_K})$ を信頼度とした $\boldsymbol{C}_{(1)}^{Rank}$ とを比較すると $\boldsymbol{C}_{(0.94)}^{Post}$ のほうが 0.061 ポイント最高適合率が高かった．

表 5.6: $CM_{Post}(s_K, d^{w_K})$ を用いた $\boldsymbol{C}^{Post}$ の再現率・適合率

	$T = 1(0.94)$			$T = 0$		
モデル	再現率	適合率	F 値	再現率	適合率	F 値
$\boldsymbol{W}^{Rank}$	**0.746**	0.637	**0.688**	*0.900*	*0.236*	*0.374*
$\boldsymbol{S}^{Post}$	0.433	0.494	0.461	**0.849**	0.189	0.309
$\boldsymbol{C}^{Post}$	0.402	**0.885**	0.553	0.706	**0.735**	**0.720**
$\boldsymbol{S}_{(0.94)}^{Post}$	0.516	0.507	0.511			
$\boldsymbol{C}_{(0.94)}^{Post}$	0.480	**0.888**	0.623			

$\boldsymbol{C}^{Rank}$，$\boldsymbol{C}^{Post}$ どちらの信頼度を用いた場合においても，適合率の上限を高めるための調整方法として有効に働いていることがわかる．いずれも単語 1-best の結果に対して音節音声認識 N-best を使って信頼度付与した結果であり，検出対象音声に対し単語 1-best と音節 N-best の認識結果を得ておけば，キーワード検出時には通常の文字列検出を行い，キーワードが検出された音声区間に対してのみ信頼度評価を行うことで，適合率の高い音声区間のみを得ることができる．

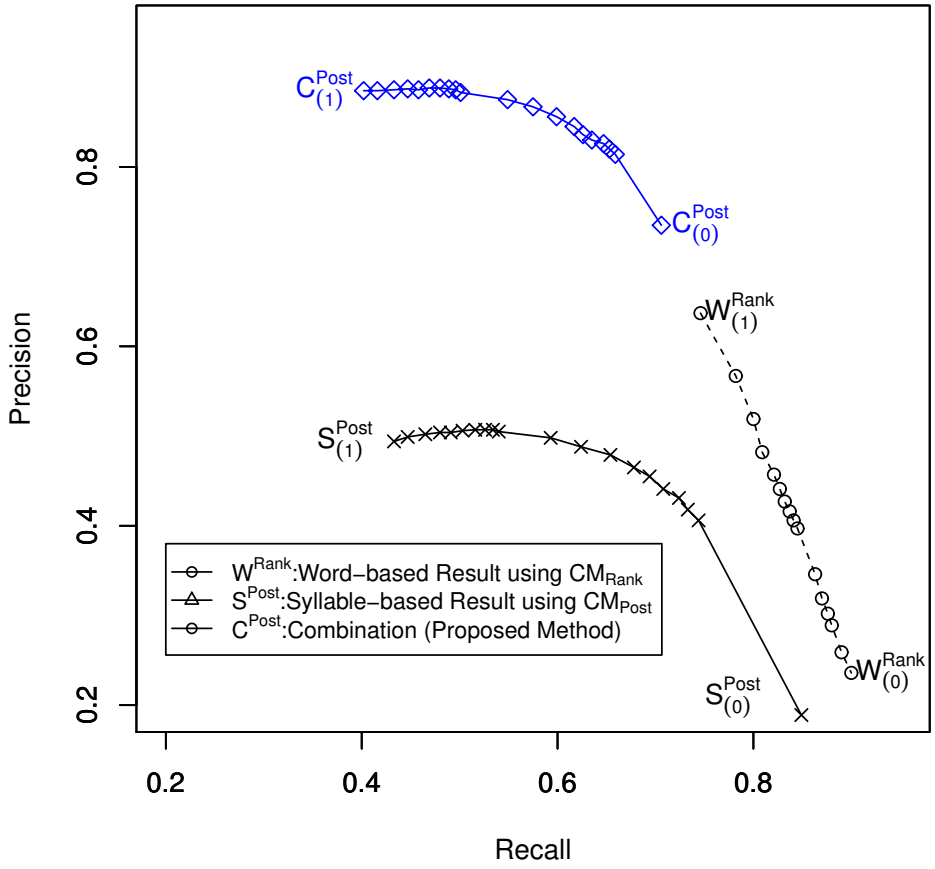

図 5.4: $CM_{Post}(s_K, d^{w_K})$ を用いた場合の再現率/適合率曲線

5.3.6 単語音声認識に対して事後確率を信頼度として用いた実験

単語 1-best の結果に対して音節音声認識 N-best を $CM_{Rank}(s_K, d^{w_K})$ および $CM_{Post}(s_K, d^{w_K})$ により計算される信頼度により信頼度付与できることがわかったが，単語認識 N-best の結果から直接 $CM_{Post}(s_K, d^{w_K})$ を用いて信頼度の評価を行った．

音節モデルの事後確率による信頼度は

$$CM_{Post}(w_K, d_i^{w_K}) = \frac{\sum_{n=1}^{N} p(x|s^n) \cdot p(s^n) \cdot \delta(w_K, \mathcal{L}_{d_i^{w_K}}^{w^n})}{\sum_{n=1}^{N} p(x|w^n) \cdot p(w^n)}$$

として定義される．

単語音声認識 N-best の結果に対して直接単語事後確率を用いた結果 $\boldsymbol{W}^{Post}$ と，

提案手法である組み合わせ $\boldsymbol{W}^{Post}$ の事後確率の最も高い $\boldsymbol{W}^{Post}_{(1)}$ に対して音節 N-best を用いた事後確率 $\boldsymbol{S}^{Post}$ を用いて信頼度付与した結果から計算される検出区間の集合を以下のように示す.

$\boldsymbol{W}^{Post}_{(T)}$ 単語音声認識 N-best
: 全検出評価用データに対する各音節音声認識 N-best 出力に対し，単語事後確率が $CM_{Post}(w_K, d^{w_K}) \geq T$ となる音声区間の集合

$\boldsymbol{G}^{Post}_{(T)}$ 単語・音節音声認識組み合わせ (提案手法)
: $\boldsymbol{G}^{Post}_{(T)} = \boldsymbol{W}^{Post}_{(1)} \cap \boldsymbol{S}^{Post}_{(T)}$. 単語認識結果がキーワード W_K が単語事後確率が最大となる単語列 $\boldsymbol{W}^{Post}_{(1)}$ に一致し，かつ音節認識結果の事後確率 $CM_{Post}(s_K, d^{w_K}) \geq T$ となる音声区間の集合.

表 5.7 に閾値を最大・最小とした際の性能を示し，図 5.5 に閾値を $T = 1 \rightharpoonup 0$ と段階的に変化させていった結果を示す．表 5.7 の $\boldsymbol{W}^{Post}$ に着目すると閾値が $T = 1$ とした場合 $\boldsymbol{W}^{Post}_{(1)}$ の適合率は $\boldsymbol{C}^{Post}_{(1)}$ の適合率より高く，0.924 であった．さらに $\boldsymbol{W}^{Post}_{(1)}$ で検出された音声区間において $\boldsymbol{S}^{Post}$ を用いて信頼度付与を行ったところ，$\boldsymbol{W}^{Post}_{(1)}$ の最大適合率 0.924 から 0.959 に 0.035 ポイント向上し，46% の適合率向上を示している．さらに，$\boldsymbol{W}^{Post}$ モデルと $\boldsymbol{C}^{Post}$ モデルのそれぞれの閾値を独立に変えて組み合わせた場合の F 値の最も高い点をつないだ結果を図 5.6 に示し $\boldsymbol{G}^{PostMax}$ とする．これは本提案手法にて得られる最も高い性能を示す.

表 5.7: $\boldsymbol{W}^{Post}$ モデルおよび $\boldsymbol{G}^{Post}$ モデルの再現率および適合率

	$T = 1$			$T = 0$		
モデル	再現率	適合率	F 値	再現率	適合率	F 値
$\boldsymbol{W}^{Post}$	0.346	0.924	0.503	**0.900**	0.236	0.374
$\boldsymbol{S}^{Post}$	**0.433**	0.494	0.461	0.849	0.189	0.309
$\boldsymbol{G}^{Post}$	0.272	**0.959**	**0.423**	0.331	**0.943**	**0.490**

また，キーワードの文字列長の違いにより再現率・適合率に違いがあるかを調べるため，40 語のキーワードを文字列長が 4 文字以下のキーワード 20 語と 5 文字以上のキーワード 20 語に分け，それぞれの再現率・適合率・F 値を調べた．表 5.8 に結果を示す．$\boldsymbol{W}^{Post}$ モデルに着目すると，文字列長の長いキーワードの F 値が高いが $T = 0$ における再現率は文字列長の短いキーワードの方が高い．また $\boldsymbol{S}^{Post}$ に関しては $T = 1, T = 0$ のどちらにおいても文字列長の短いキーワードが，再現率は高いが適合率は大幅に低い結果となっている．文字列の短いキーワードは音節長も短く，同じ音節列をもつ別の語 (文字列の異なる語) に一致しやすく，適合

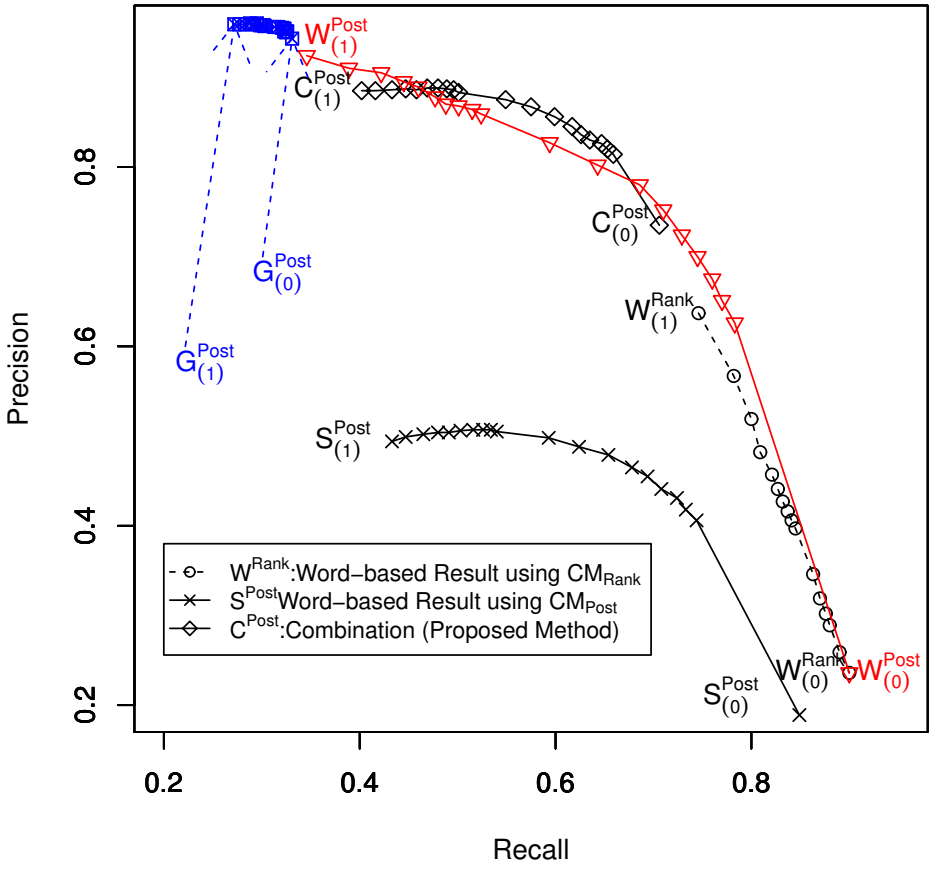

図 5.5: $\boldsymbol{W}^{Post}$ モデルおよび $\boldsymbol{G}^{Post}$ モデルの再現率/適合率曲線

率を下げる要因になっている．$\boldsymbol{G}^{Post}$ モデルでは，$T = 0$ における F 値は文字列長の長いキーワードのほうが高いが，これは $\boldsymbol{W}^{Post}$ の $T = 1$ の値に依存して高くなっている．また $\boldsymbol{G}^{Post}$ モデルでの $T = 1$ の F 値は文字列長の短いキーワードのほうが高く，$\boldsymbol{G}^{Post}$ モデルで用いている $\boldsymbol{S}^{Post}$ モデルの再現率の高さが正しい信頼度付与に寄与しているものと考えられる．

5.4 考察

大規模なコールセンター [64] の例では，通話時間が約 10 分程度の問い合わせが月に 35 万件あり，平日のみの受電と仮定すると 1 日あたり約 17,000 件．約 3,000 時間の音声がコールモニタリングの対象となる．これら 1 日数千時間の音声に対し

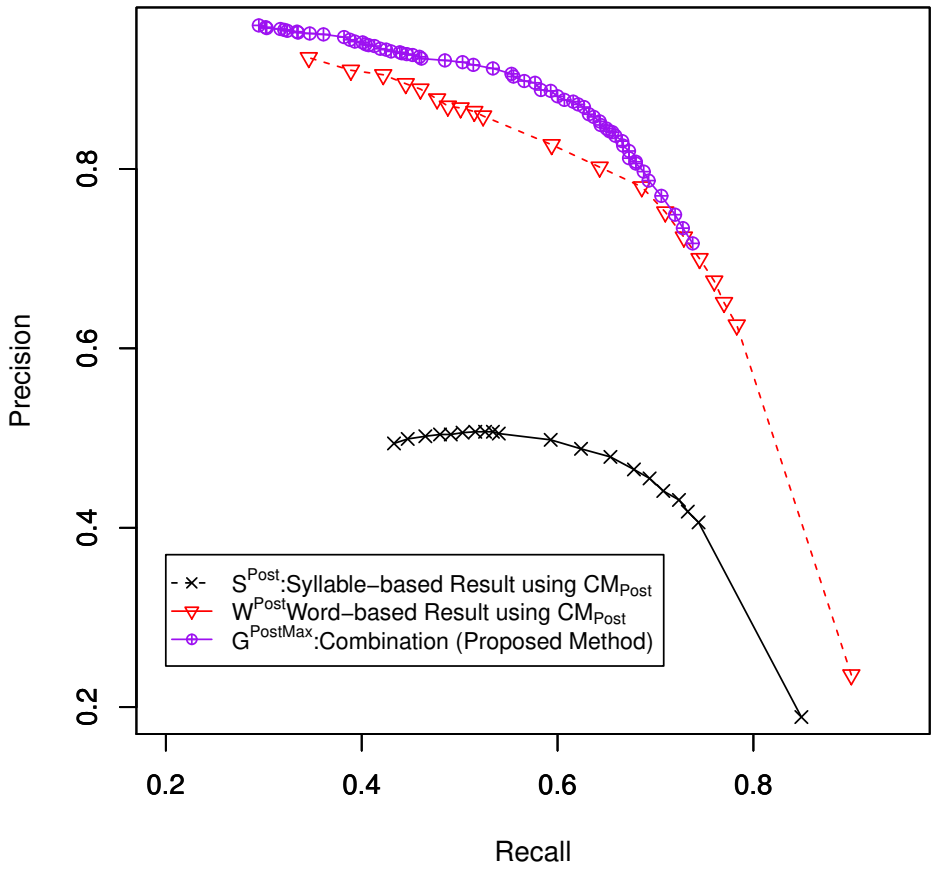

図 5.6: $\boldsymbol{W}_{(T)}^{Post}$ と $\boldsymbol{S}_{(T)}^{Post}$ を用いて得られる最も F 値の高い再現率-適合率曲線

て音声認識処理を日々行うためには，100 台規模の大きな計算資源が必要となる．本システムもこのような規模のコールセンターデータを対象としており，実験の結果をふまえ，本研究の目的とする「計算量を大幅に増やさない」「作業効率を向上させる」「高い適合率に調整が可能である」という観点から考察を行う．

5.4.1 適合率の範囲と作業効率

実験の結果，音節音声認識による信頼度の付与を行うことで検出の適合率を高めることができた．すでに認識された結果を用いるので，再認識処理することなく検出時に閾値を指定することにより適合率の高い区間の音声をチェックできる．前章の実験で得られた適合率の区間を図 5.7 にまとめる．横軸に平行な 適合率 $= 0.637$

表 5.8: 単語の文字列長の違いによる再現率および適合率

モデル	$T=1$ 再現率	$T=1$ 適合率	$T=1$ F 値	$T=0$ 再現率	$T=0$ 適合率	$T=0$ F 値
	文字列長 4 文字以下					
$\boldsymbol{W}^{Post}$	0.336	0.921	0.493	**0.910**	0.218	0.352
$\boldsymbol{S}^{Post}$	**0.465**	0.423	0.443	**0.856**	0.161	0.271
$\boldsymbol{G}^{Post}$	**0.282**	**0.961**	**0.436**	0.325	0.942	0.483
	文字列長 5 文字以上					
$\boldsymbol{W}^{Post}$	**0.365**	**0.930**	**0.524**	0.879	**0.284**	**0.430**
$\boldsymbol{S}^{Post}$	0.369	**0.844**	**0.513**	0.836	**0.294**	**0.436**
$\boldsymbol{G}^{Post}$	0.250	0.954	0.396	**0.343**	**0.944**	**0.503**

の破線は単語 1-best で得られる適合率を示す．$\boldsymbol{C}^{Rank}$, $\boldsymbol{C}^{Post}$, $\boldsymbol{G}^{Post}$ はそれぞれベースシステムとの組み合わせとして用いられるので，それぞれのベースモデルの適合率の下端から $T=0$ における適合率の区間を斜線の区間にて示す．単語 1-best である $\boldsymbol{W}^{Rank}_{(1)}$ に対して，$\boldsymbol{C}^{Rank}$ においては 0.190，$\boldsymbol{C}^{Post}$ においては 0.248 ポイント適合率を向上させることができている．同様に，$\boldsymbol{W}^{Post}$ に比べて $\boldsymbol{G}^{Post}$ では 0.035 ポイント適合率を向上させることができている．

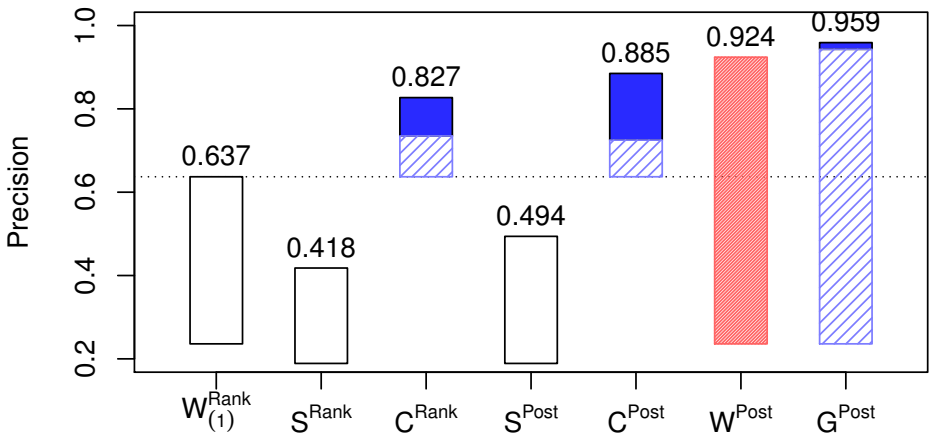

図 5.7: 適合率の範囲

また，最終的な検出結果の確認 (キーワードが実際に音声区間と一致しているか)

として人手による聴取作業を前提としている．したがって，できるだけ誤検出が少なくかつ全体を網羅できると効率的に作業が行える．実際のコールモニタリング作業を模した作業効率により評価を行った．再現率・適合率の結果を用いて，図 5.8 に信頼度上位から音声を聴取した際にどれだけ誤った音声が含まれるかを示した．横軸に検出区間数，縦軸にその検出区間に含まれる誤り数を示す．単語事後確率 $\boldsymbol{W}^{Post}$ および音節事後確率 $\boldsymbol{S}^{Post}$ の結果をベースにした $\boldsymbol{G}^{PostMax}$ の検出誤りが全区間においてもっとも少ないことが分かる．

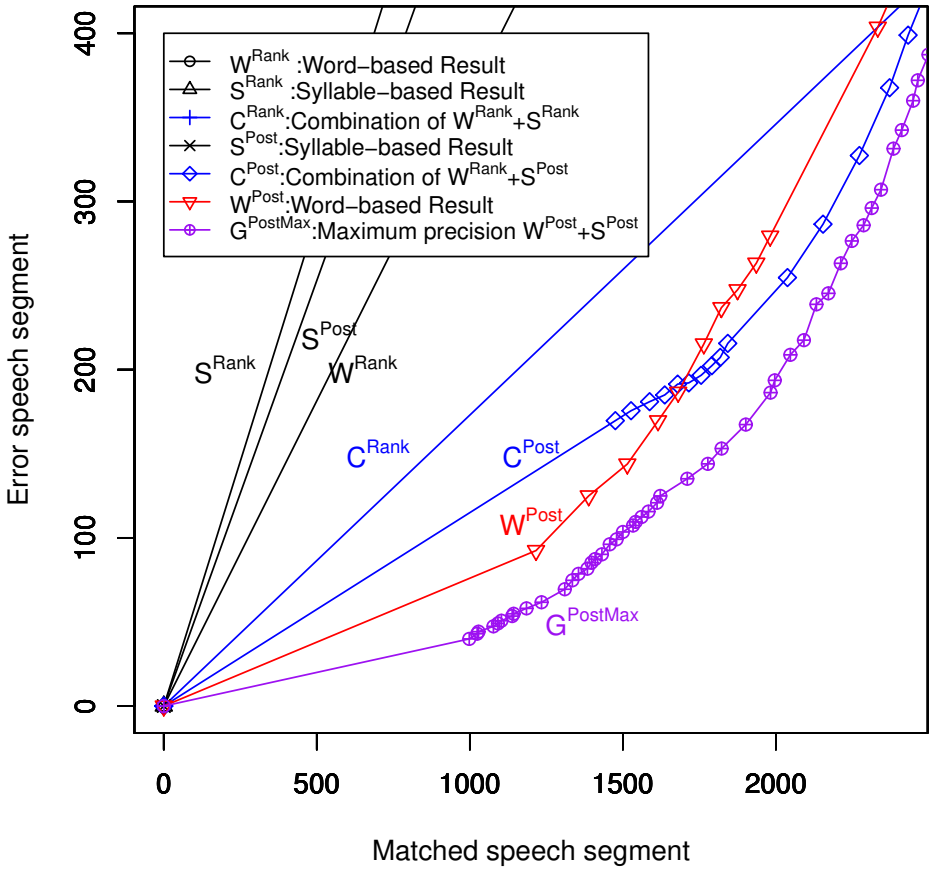

図 5.8: 検出区間に対する誤り数

正解であるのべ 3148 個のキーワードの 25%, 50% を網羅する再現率 0.250, 0.500 のポイントにおいて誤検出数を比較した結果を表 5.9 に示す．再現率 0.500 の場合，単語 1-best である $\boldsymbol{W}_{(1)}^{Rank}$ と提案法を比較すると，$\boldsymbol{W}^{Rank}$ の場合 2550 個の検出に対して 1624 個の正解と 926 個の誤りを含むのに対し，ランキングを信頼度とし

表 5.9: 再現率 0.250, 0.500 のポイントにおける誤り検出数

	再現率 0.250% (正解数 812)		再現率 0.500% (正解数 1624)	
モデル	検出数	誤検出数	検出数	誤検出数
$\boldsymbol{W}_{(1)}^{Rank}$	1274	462	2550	926
$\boldsymbol{C}^{Rank}$	970	**158**	1940	**316**
$\boldsymbol{C}^{Post}$	917	**105**	1839	**215**
$\boldsymbol{W}^{Post}$	879	**67**	1871	247
$\boldsymbol{G}^{PostMax}$	846	**34**	1767	**143**

た $\boldsymbol{C}^{Rank}$ の場合 316 個，事後確率を信頼度とした $\boldsymbol{C}^{Post}$ の場合 215 個の誤り数となり，それぞれ 65.8%, 76.7% の誤検出を減らすことができている．また単語 N-best を用いた $\boldsymbol{W}^{Rank}$ と提案法である $\boldsymbol{G}^{PostMax}$ を比較した場合，それぞれ 247 個の誤りと 143 個の誤りを含み 42.1% の誤検出を減らしている．再現率 0.250 の場合では $\boldsymbol{G}^{PostMax}$ の誤りが最も少なく次いで $\boldsymbol{W}^{Post}$ の誤り数が次に少ない．この場合でも提案法 $\boldsymbol{G}^{PostMax}$ は単語 N-best のみを用いた $\boldsymbol{W}^{Post}$ に比べて誤りを 49.2% 減らすことができている．どちらの場合においても本手法にて大幅に作業効率が削減できていることがわかる．

5.4.2 計算時間

単語音声認識と音節音声認識の組み合わせにより，未知語や認識誤りへの対応だけでなく辞書後の検出性能向上にも寄与できることがわかったが，前述したように大量の音声を限られた時間で処理するためには計算量に対して考慮する必要がある．コールモニタリングを目的とした場合，一般的な検索エンジンのように自由なキーワードを検出時に入力するのではなく，あらかじめ設定しておいたキーワードに対して検出を行う場合が多い．キーワードが既知であるため，$\boldsymbol{C}^{Rank}$，$\boldsymbol{C}^{Post}$ ではあらかじめ $\boldsymbol{W}_{(1)}^{Rank}$ により絞り込まれた検出区間のみに対して $CM_{Post}(s_K, d^{w_K})$ の計算 (音節音声認識および音節 N-best の出力) を行えばよい．また $\boldsymbol{G}^{Post}$ では単語 N-best に含まれる検出区間のみに対して $CM_{Post}(s_K, d^{w_K})$ の計算を行えばよい．CPU が Intel Xeon 3.16G Hz，メモリ 4GB の計算機を用いて実際に計測を行った結果，システムで用いた音声認識器では，単語音声認識 1-best 出力に要する時間を r とすると単語 N-best 出力に $1.37r$, 音節音声認識 1-best に $1.77r$, 音節 N-best 出力に $5.08r$ 要していた．なお，文字列検出および信頼度計算のステップは音声認識時間に比べると小さいため，これらの数字には文字列検出および信頼度計算の時間も含まれるものとする．表 5.10 に，インデックス作成およびキーワー

ド検出に要する実行時間をモデルごとに示す．表 5.10 中の a は $\boldsymbol{W}_{(1)}^{Rank}$ により絞り込まれた検出区間の全検出対象データに対する割合，b は単語 N-best にキーワードが含まれる検出区間の全検出対象データに対する割合である．本実験で用いた検出評価用データおよび検出キーワードの場合 40 語の合計で $a = 0.159, b = 0.566$ であった．この場合の実行時間を条件 1 の列に記す．一方，実際のコールセンターのモニタリングでは，検出対象の音声が非常に大量ではあるものの，検出キーワードは低頻度で出現する語が用いられる．このような場合 a, b はともに 0 に近くなり，信頼度付与のためのインデックス作成に要する時間は無視できる．$a = 0, b = 0$ とした場合の実行時間を条件 2 の列に示す．

表 5.10: モデルの違いに対する計算時間

モデル	実行時間	条件 1	条件 2
$\boldsymbol{W}_{(1)}^{Rank}$	r	r	r
$\boldsymbol{S}^{Rank}$, $\boldsymbol{S}^{Post}$	$1.77r$	$1.77r$	$1.77r$
$\boldsymbol{C}^{Rank}$, $\boldsymbol{C}^{Post}$	$r + a \times 5.08r$	$2.81r$	$1.00r$
$\boldsymbol{W}^{Post}$	$1.37r$	$1.37r$	$1.37r$
$\boldsymbol{G}^{Post}$	$1.37r + b \times 5.08r$	$4.24r$	$1.37r$

条件によって計算時間は大きく異なるが，比較的頻度の高いキーワードを検出対象とした場合，全検出対象データに対して単語 N-best を計算する $\boldsymbol{W}^{Post}$ が検出効率および計算時間の面から有利である．一方，コールモニタリング業務のような大量のデータから低頻度のキーワードを検出する場合は，計算時間にほとんど影響をおよぼさず，作業効率の良い $\boldsymbol{C}^{Post}$ が有利となる．また $\boldsymbol{G}^{Post}$ は計算資源に余裕があり同様に低頻度のキーワードを検出する場合は人手による聴取作業を最小限にすることができる．$\boldsymbol{S}^{Rank}$, $\boldsymbol{S}^{Post}$ 単体では検出効率および実行時間の面からはあまり実用的ではなかった．

5.5 本章のまとめ

「計算量を大幅に増やさない」「作業効率を向上させる」「高い適合率に調整が可能である」という観点で単語音声認識結果と音節音声認識結果を組み合わせたシステムを提案した．本論文では，既知語の適合率改善のために音節音声認識の結果を用い，ランキング，事後確率による信頼度のいずれにおいても適合率の高い音声区間の検出を試みた．この信頼度を利用することでユーザーは適合率の高い音声のみを検出漏れをできるだけ少なくチェックといった作業が可能になる．有効性の評価は，各システムで得られる適合率の範囲と，実際のコールモニタリン

グ作業を模した作業効率 (信頼度上位から音声を聴取した際に誤った音声が含まれる割合) により行った．信頼度の計算方法として，ランキングと単語事後確率とを比較すると，事後確率のほうが適合率の上限が高いが，ランキングの場合，必要な適合率の幅に応じて計算量を減らすことができる．出力を 10-best に限定して適合率の幅を減らすと，事後確率に基づく場合の 0.32 倍程度の計算量でキーワードの検出ができる．

ベースシステムとして単語音声認識 1-best を用いた場合と，単語音声認識 1-best に音節音声認識 N-best を用いて信頼度計算を行った場合とを比較すると，適合率の上限は最大 0.248 ポイント向上し，作業効率の観点からは誤検出の割合を 76.7% 減らすことができた．また，ベースシステムとして単語認識 N-best を用いて信頼度計算を行った結果に対してさらに音節音声認識 N-best を用いて信頼度計算を行うことで，適合率の上限は 0.035 ポイント向上し，作業効率の観点からは誤検出の割合を 42.1% 減らすことができた．

一方，ベースシステムにかかわらず音節認識 N-best を用いて信頼度計算を行う手法は信頼度計算にコストがかかるが，信頼度計算のコストはベースシステムにより検出されるキーワードの頻度に比例するため，一般的なコールモニタリングで用いられる低頻度のキーワードを対象にした場合，全体の計算コストにほとんど影響を与えることなく，高い作業効率を実現することができる．

第6章　結論

本論文では，**コーパスおよび辞書を入力とする統計的音声合成フロントエンドと2種類の認識単位の異なる音声認識システムを用いた効率的に音声検索語検出が可能なシステム**について論じた．これら音声合成システムおよび音声認識アプリケーションのそれぞれに，新たな音声言語処理の枠組みを導入した．

人の発話と区別のつかない音声合成を実現するためには，発話者の発話から特徴量を抽出し，モデルを学習し，再合成する必要がある．本論文での研究は，音声合成の言語処理部における読みとアクセントを高精度に推定することで，音韻・韻律特徴量を言語的な側面からモデル化し，基本的な韻律情報及び音韻情報である読み及びアクセントの付与に関しては．1つの確率的モデルを用いた枠組みで学習できることを明らかにした．提案した**コーパスおよび辞書を入力とする統計的音声合成フロントエンド**においては⟨単語境界, 品詞, 読み, アクセント⟩の4つ組，または品詞を用いない⟨単語境界, 読み, アクセント⟩の3つ組を1つの単位と捉え，N-gram モデルを用いて推定を行った．つまり，⟨単語境界, 品詞, 読み, アクセント⟩の4つの値，または⟨単語境界, 読み, アクセント⟩の3つの値を同時に推定する．約1万文のコーパスにより学習を行い，テストコーパスに対する推定結果に対し，コーパスに予め与えられた正解との単語毎の精度を計算した結果，4つ組および3つ組確率モデルに基づく手法の精度は，既存手法であるルールを用いた手法と逐次的手法との精度を上回った．また，4 つ組と3つ組のモデルを，学習コーパスの作成に要する時間と推定精度の2つの観点から考察を行った結果，品詞を用いない3 つ組のモデルが適していた．また，アクセント句・アクセント核という，従来，日本語のアクセント付与で用いられていた単位を本手法では用いないため，アクセント核が2つ以上ある語にも対応できる．このことは，コーパスのみ与えることができれば，言語固有のルールを用いることなく，日本語の標準語のみでなく方言や他言語でも同じ枠組みで読み及びアクセントが付与出来ることを示している．

さらに，学習データのスパースネスに起因して，文脈を伴って現れることの無い語彙の集合に対し正しいアクセントを付与できるように言語モデルを改善した．日本語のアクセントの特徴を用いて，辞書中には存在するが学習データ中現れな

い語に対して，文脈を推定し適切なアクセントを与えることとした．最終的に，テストコーパスに含まれる様々な分野の文章を対象とした場合，日本語の読みおよびアクセントの推定精度は，それぞれ約 99%，92% を達成した．

また，読みおよびアクセントおよびアクセントをユーザーが自由に修正，変更ができるようなチューニングツールの開発を行った．モデルの修正，変更が容易な，統計的フロントエンドを開発し，精度の面においては，言語モデルを改善することで，より良い読みやアクセントを付与することができるようになった．また，読みやアクセント誤りの修正や，任意の読みやアクセントを付与するためのチューニングツールを作成し，有効に修正，変更が行えることを確認した．推定精度という面では，特にアクセントの推定精度は実用的な観点からはまだ十分とはいえない．文字列および単語列へのラベル付与問題として考えると，より高精度は学習器が提案されており，今後はさらなる性能の向上を図っていく必要がある．

音声検索語検出技術については以前から研究が行われているが，実際に大量の音声データを扱う際には新しい問題が発生し，その問題への対処が必要になる．本論文では，音声検索語検出の実用化に際し，いかに音声聴取作業を効率化できるかという観点で研究を行った．単語音声認識結果と音節音声認識結果を組み合わせた**2 種類の認識単位の異なる音声認識システムを用いた効率的に音声検索語検出が可能なシステム**によって，音声検索作業の効率を向上できることを実験を通じて明らかにした．既知語の適合率改善のために音節音声認識の結果から，ランキングおよび事後確率による信頼度を用いて適合率の高い音声区間の検出を試みた．この信頼度を利用することでユーザーは適合率の高い音声のみを検出漏れをできるだけ少なくチェックといった作業が可能になる．

有効性の評価は，各システムで得られる適合率の範囲と，実際のコールモニタリング作業を模した作業効率（信頼度上位から音声を聴取した際に誤った音声が含まれる割合），および信頼度計算に必要な計算量の三点によって行った．信頼度の計算方法として，ランキングと単語事後確率とを比較すると，事後確率のほうが適合率の上限が高いが，ランキングの場合，必要な適合率の幅に応じて計算量を減らすことができる．出力を 10-best に限定して適合率の幅を減らすと，事後確率に基づく場合の 0.32 倍程度の計算量でキーワードの検出ができる．ベースシステムとして単語音声認識 1-best を用いた場合と，単語音声認識 1-best に音節音声認識 N-best を用いて信頼度計算を行った場合とを比較すると，適合率の上限は最大 0.248 ポイント向上し，作業効率の観点からは誤検出の割合を 76.7% 減らすことができた．また，ベースシステムとして単語認識 N-best を用いて信頼度計算を行った結果に対してさらに音節音声認識 N-best を用いて信頼度計算を行うことで，適合率の上限は 0.035 ポイント向上し，作業効率の観点からは誤検出の割合を 42.1%

減らすことができた．一方，ベースシステムにかかわらず音節認識 N-best を用いて信頼度計算を行う手法は信頼度計算にコストがかかるが，信頼度計算のコストはベースシステムにより検出されるキーワードの頻度に比例するため，一般的なコールモニタリングで用いられる低頻度のキーワードを対象にした場合，全体の計算コストにほとんど影響を与えることなく，高い作業効率を実現することができる．音声検索システムの効率化のためには，より誤りの少ない音声認識器の実現に関しても取り組む必要があるが，音声認識誤りがなくなることはない．音声認識を用いた情報抽出アプリケーションの実用性向上という観点で，認識誤りを前提とした音声アプリケーションの効率化に取り組む必要がある．

インターネットの普及につれ，テキストデータのビジネス利用がさかんになった．いわゆる検索サイトをはじめ，チャットや掲示板，トレンド分析やテキストマイニング等の入力データとして幅広く活用されている．一方，大量の音声データが容易に収集できるようになったのはこの数年であり，まだテキストデータのように十分な活用はなされていないが，モバイル端末を中心とした情報検索・対話技術が急速に進化しつつある．

本研究では，音声データの効率的な活用を目指して，音声言語処理の新たな適用可能性について論じた．収集された音声データは情報検索・音声対話アプリケーションの知識源として活用，また音声認識・音声合成の性能向上のための入力データとして用いられ，利用者にフィードバックされることになる．このような効率的なフィードバックループを構築することで，将来，人間の単純作業の代替を超えて，医師・翻訳者・歌手のような新たな役割・価値を生み出せる可能性がある．今後も，音声言語処理の新たな適用を通じて，音声データの新たな活用方法に対する研究開発に取り組んでいきたい．

参考文献

[1] G. Hinton, L. Deng, D. Yu, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and Kingsbury B. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, Vol. 29, No. 6, pp. 82–97, November 2012.

[2] The Rich Transcription 2009 Speech-To-Text (STT) and Speaker Attributed STT (SASTT) Results. Ajot, j. and fiscus, j. `http://www.itl.nist.gov/iad/mig/tests/rt/2009/workshop/RT09-STT_SASTT-v1.pdf`.

[3] IBM Watson Developer Cloud : Speech to Text. `http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/speech-to-text.html`.

[4] Microsoft Project Oxford : Speech APIs. `https://www.projectoxford.ai/speech`.

[5] docomo Developer support : 音声認識. `https://dev.smt.docomo.ne.jp/?p=docs.api.page&api_name=speech_recognition`.

[6] iOS - Siri. `http://www.apple.com/ios/siri`.

[7] ソフトバンク：ロボット. `http://www.softbank.jp/robot`.

[8] R. Moore. A comparison of the data requirements of automatic speech recognition systems and human listeners. In *2003 European Conference on Speech Communication and Technology (EUROSPEECH2003)*, pp. 2582–2584, 2003.

[9] M. Goto and J. Ogata. Podcastle: Recent advances of a spoken document retrieval service improved by anonymous user contributions. In *The 12th Interspeech Conference (INTERSPEECH2011)*, pp. 3073–3076, 2011.

[10] C. Callison-Burch and M. Dredze. Creating speech and language data with Amazon's Mechanical Turk. In *The NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pp. 1–12, 2010.

[11] S. Novotney and C. Callison-Burch. Cheap, fast and good enough: Automatic speech recognition with non-expert transcription. In *The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT2010)*, pp. 207–215, 2010.

[12] M. Marge, S. Banerjee, and A. Rudnicky. Using the Amazon Mechanical Turk for transcription of spoken language. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2010)*, pp. 5270–5273, 2010.

[13] J.G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER). In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 1997)*, pp. 347–354, 1997.

[14] F. Wessel and H. Ney. Unsupervised training of acoustic models for large vocabulary continuous speech recognition. *Speech and Audio Processing, IEEE Transactions on*, Vol. 13, No. 1, pp. 23–31, 2005.

[15] L. Wang, M. J. Gales, and P. C. Woodland. Unsupervised training for Mandarin broadcast news and conversation transcription. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2007)*, pp. 353–356, 2007.

[16] N. T. Vu, F. Kraus, and T. Schultz. Cross-language bootstrapping based on completely unsupervised training using multilingual A-stabil. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2011)*, pp. 5000–5003, 2011.

[17] 森信介, 山地治. 日本語の情報量の上限の推定. 情報処理学会論文誌, Vol. 38, No. 11, pp. 2191–2199, 1997.

[18] D. Klatt. Review of text-to-speech conversion for English. *Journal of the Acoustical Society of America*, Vol. 82, No. 3, pp. 737–793, 1987.

[19] F. Charpentier and M. Stella. Diphone synthesis using an overlap-add technique for speech waveforms concatenation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP1986)*, pp. 2015–2020, 1986.

[20] P.800 : Methods for subjective determination of transmission quality. https://www.itu.int/rec/T-REC-P.800.

[21] H. Zen and H. Sak. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2015)*, pp. 4470–4474, 2015.

[22] L. Sun, S. Kang, K. Li, and H. Meng. Voice conversion using deep bidirectional long short-term memory based recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2015)*, pp. 4689–4873, 2015.

[23] 増田恵子, 梅村恭司. 人名辞書から名前読み付与規則を抽出するアルゴリズム. 情報処理学会論文誌, Vol. 40, No. 7, pp. 2927–2936, 1999.

[24] 浅野久子, 永田昌明, 阿部匡伸. 日本語テキストにおけるアルファベット文字列の読みクラス分類. 言語処理学会 第 9 回年次大会, pp. 465–468, 2003.

[25] J. Rao, F. Peng, H. Sak, and F Beaufays. Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2015)*, pp. –, 2015.

[26] John F. Pitrelli, Ellen M. Eide Raimo Bakis, Raul Fernandez, Wael Hamza, and Michael A. Picheny. The IBM expressive text-to-speech synthesis system for American English. *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 14, No. 4, pp. 1099–1108, Jul. 2006.

[27] Tachibana R., Nagano T., Kurata G., Nishimura M., and Babaguchi N. Preliminary Experiments toward Automatic Generation of New TTS Voices from Recorded Speech Alone. In *Proceedings of the InterSpeech 2007*, pp. 1917–1920, 2007.

[28] S. Seto, M. Morita, T. Kagoshima, and M. Akamine. Automatic rule generation for linguistic features analysis using inductive learning technique: linguistic features analysis in TOS drive TTS system. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP1998)*, pp. 1059–1063, 1998.

[29] Q. Shi and V. Fischer. A comparison of statistical methods and features for the prediction of prosody prosodic structures. In *Proceedings of the 8th International Conference on Spoken Language Processing (ICSLP2004)*, pp. 1165–1169, 2004.

[30] K. Knight and J. Graehl. Machine transliteration. *Computational Linguistics*, Vol. 24, No. 4, pp. 599–612, 1998.

[31] 永田昌明. 日本語ＯＣＲのための表記と読みの同時形態素解析. 電子情報通信学会 技術研究報告, SP2002-33, pp. 55–60, 2002.

[32] 匂坂芳典, 佐藤大和. 日本語単語連鎖のアクセント規則. 電子情報通信学会 論文誌, Vol. J66-D, No. 7, pp. 849–856, 1983.

[33] R. Quinlan. *Programs for machine learning*. Morgan Kaufmann, 1993.

[34] 中嶋秀治, 永田昌明, 浅野久子, 安部匡伸. Support vector machine を使ったモーラ列からの日本語姓名のアクセント推定. 電子情報通信学会 論文誌, Vol. J88-D-II, No. 3, pp. 480–488, 2005.

[35] M. Nagata. A stochastic Japanese morphological analyzer using a forward-DP backward-A* N-best search algorithm. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING1994)*, pp. 201–207, 1994.

[36] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to algorithms*. The MIT Press, 1990.

[37] F. Jelinek. Self-organized language modeling for speech recognition. *Technical report, IBM T. J. Watson Research Center*, 1985.

[38] C. Stanley and J. Goodman. An Empirical study of smoothing techniques for language modeling. In *Proceedings of the Association for Computational Linguistics 34th Annual Meeting (ACL1996)*, pp. 310–318, 1996.

[39] J. Aoe. An Efficient Digital Search Algorithm by Using a Double-Array Structure. *IEEE Transactions on Software Engineering*, Vol. 15, No. 9, pp. 1066–1077, 1989.

[40] N. Minematsu, R. Kuroiwa, and K. Hirose. CRF-based statistical learning of Japanese accent sandhi for developing Japanese text-to-speech synthesis

systems. In *Proceedings of the 6th ISCA Workshop on Speech Synthesis*, pp. 148–153, 2007.

[41] 長野徹, 森信介, 西村雅史. 確率モデルを用いた音声合成のための読み及びアクセント推定. 情報処理学会 論文誌, Vol. 47, No. 6, pp. 1973–1981, 2006.

[42] P. Brown, V. Della Pietra, P. deSouza, J. Lai, and R. Mercer. Class-Based N-Gram Models of Natural Language. *Computational Linguistics*, Vol. 18, No. 4, pp. 466–479, 1992.

[43] S. Mori, M. Nishimura, and N. Itoh. Word Clustering for A Word Bi-gram Model. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP1998)*, pp. 310–318, 1998.

[44] D. Linares, J. Benedí, and Y. Sánchez. A hybrid language model based on a combination of N-grams and stochastic context-free grammars. *ACM Transactions on Asian Language Information Processing (TALIP)*, Vol. 3, No. 2, pp. 113–127, 2004.

[45] http://www.nuance.com/promptsculptor/. Nuance PromptSculptor.

[46] http://www.loquendo.com/en/technology/tts_director.htm. Loquendo TTS Editor.

[47] A. Amir, A. Efrat, and S. Srinivasan. Advances in Phonetic Word Spotting. In *Proceedings of the tenth international conference on Information and knowledge management (CIKM '01)*, pp. 580–582, 2001.

[48] J. Mamou and B. Ramabhadran. Vocabulary independent spoken term detection. In *The 30th Annual International ACM SIGIR Conference (SIGIR 2007)*, pp. 615–622, 2007.

[49] 坂本渚, 山本一公, 中川聖一. 距離付き音節 n グラムインデックスを用いた音声入力による音声ドキュメントの検索語検出法の評価. 第 7 回音声ドキュメント処理ワークショップ論文集, pp. 2013–05, 2013.

[50] P. Liu, Y. Tian, J. Zhou, and F. Soong. Background model based posterior probability for measuring confidence. In *The 9th European Conference on Speech Communication and Technology (INTERSPEECH2005)*, pp. 1465–1468, 2005.

[51] 西崎博光, 中川聖一. 音声認識誤りと未知語に頑健な音声文書検索手法. 電子情報通信学会論文誌. D-II, 情報・システム, II-パターン処理, Vol. J86-D-II, No. 10, pp. 1369–1381, 2003.

[52] 宇津呂武仁ほか. 複数の大語彙連続音声認識モデルの出力の共通部分を用いた高信頼度部分の推定. 電子情報通信学会論文誌. D-II, 情報・システム, II-パターン処理, Vol. J86-D-II, No. 7, pp. 974–987, 2003.

[53] J. Mamou, et al. System combination and score normalization for spoken term detection. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2013)*, pp. 8272–8276, 2013.

[54] F. Wessel, R. Schluter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transaction Speech and Audio Prcesing*, Vol. 9, No. 3, pp. 288–298, 2001.

[55] B. Rueber. Obtaining confidence measures from sentence probabilities. In *Fifth European Conference on Speech Communication and Technology (EUROSPEECH 1997)*, pp. 739–742, 1997.

[56] 中川聖一, 堀部千寿. 音響尤度と言語尤度を用いた音声認識結果の信頼度の算出. 情報処理学会研究報告 音声言語情報処理, Vol. 2001, No. 55, pp. 97–92, 2001.

[57] 緒方淳, 有木康雄. 音声認識精度向上のための信頼度尺度の比較. 情報処理学会研究報告 音声言語情報処理, Vol. 2000, No. 119, pp. 113–118, 2000.

[58] H. Jiang. Confidence measures for speech recognition: A survey. *Speech Communications*, Vol. 45, No. 4, pp. 455–470, 2005.

[59] B. Roark, M. Saraclar, and M. Collins. Corrective language modeling for large vocabulary asr with the perceptron algorithm. In *2004 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2004)*, pp. 749–752, 2004.

[60] D. Vergyri, I. Shafran, A. Stolcke, R. R. Gadde, M. Akbacak, B. Roark, and W. Wang. The sri/ogi 2006 spoken term detection system. In *The eighth conference in the annual series of INTERSPEECH (INTERSPEECH2007)*, pp. 2393–2396, 2007.

[61] M. Akbacak, D. Vergyri, and A. Stolcke. Open-vocabulary spoken term detection using graphone-based hybrid recognition systems. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2008)*, pp. 5240–5243, 2008.

[62] S. Chen. Conditional and Joint Models for Grapheme-to-Phoneme Conversion. In *2003 European Conference on Speech Communication and Technology (EUROSPEECH2003)*, pp. 2033–2036, 2003.

[63] S. Hahn, P. Lehnen, S. Wiesler, R. Schluter, and H. Ney. Improving LVCSR with Hidden Conditional Random Fields for Grapheme-to-Phoneme Conversion. In *The 14th Interspeech Conference (INTERSPEECH2013)*, pp. 495–499, 2013.

[64] 厚生労働省. (30) コールセンター事業 (年金電話相談事業). http://www.mof.go.jp/budget/topics /budget_execution_audit/fy2014/sy2607/2607d.htm1, 2014.

研究業績目録

査読付き論文雑誌

[1] 長野徹, 倉田岳人, 鈴木雅之, 立花隆輝, 西村雅史. 大語彙連続音声認識と音節 N-best 音声認識を用いたキーワード検索の高精度化. 情報処理学会論文誌, Vol. 56, No. 8, pp. 1646–1656, 2015.

[2] 田口高也, 根本清貴, 太刀川弘和, 長野徹, 立花隆輝, 西村雅史, 新井哲明, 朝田隆. OpenEAR を用いた音声による心理的ストレス検出の試み. 精神医学, Vol. 56, No. 12, pp. 1027–1034, 2014.

[3] 長野徹, 立花隆輝, 西村雅史. コーパスベース日本語音声合成フロントエンド. 電子情報通信学会論文誌, Vol. J93-D, No. 10, pp. 2096–2106, 2010.

[4] R. Tachibana, T. Nagano, G. Kurata, M. Nishimura, and N. Babaguchi. Automatic prosody labeling using multiple models for Japanese. *IEICE transactions on information and systems*, Vol. E90-D, No. 11, pp. 1805–1812, 2007.

[5] 長野徹, 森信介, 西村雅史. N-gram モデルを用いた音声合成のための読みおよびアクセントの同時推定. 情報処理学会論文誌, Vol. 47, No. 6, pp. 1793–1801, 2006.

[6] K. Takeda, H. Watanabe, N. Uramoto, H. Nomiyama, H. Matsuzawa, T. Nasukawa, T. Nagano, A. Murakami, H. Takeuchi, H. Kanayama, M. Kobayashi, M. Aono, A. Inokuchi, and M. Houle. Unstructured information management projects at IBM Tokyo Research Laboratory. *Korea Information Processing Society Review*, Vol. 11, No. 2, pp. 4–16, 2004.

査読付き国際会議論文

[1] M. Suzuki, G. Kurata, T. Nagano, and R. Tachibana. Speech recognition robust against speech overlapping in monaural recordings of telephone conversations. In *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2016)*, 2016. To appear.

[2] R. Takashima, T. Nagano, R. Tachibana, and M. Nishimura. Agglomerative hierarchical clustering of emotions in speech based on subjective relative similarity. In *The 12th conference in the annual series of INTERSPEECH (INTERSPEECH2011)*, pp. 2473–2476, 2011.

[3] M. Kobayashi, T. Nagano, K. Fukuda, and H. Takagi. Describing online videos with Text-to-Speech narration. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, pp. 1–2, 2010.

[4] T. Nagano, R. Tachibana, N. Itoh, and M. Nishimura. Improving phoneme and accent estimation by leveraging a dictionary for a stochastic TTS front-end. In *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2008)*, pp. 4689–4692, 2008.

[5] R. Tachibana, T. Nagano, G. Kurata, M. Nishimura, and N. Babaguchi. Preliminary experiments toward automatic generation of new TTS voices from recorded speech alone. In *The eighth conference in the annual series of INTERSPEECH (INTERSPEECH2007)*, pp. 1917–1920, 2007.

[6] T. Nagano, S. Mori, and M. Nishimura. A stochastic approach to phoneme and accent estimation. In *The sixth conference in the annual series of INTERSPEECH (INTERSPEECH2005)*, pp. 3293–3296, 2005.

[7] K. Takeda, H. Nomiyama, T. Nasukawa, M. Kobayashi, T. Sakairi, H. Matsuzawa, T. Nagano, A. Murakami, and H. Takeuchi. Text Mining and Site Outlining projects. In *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium (NLPRS)*, pp. 773–774, 2001.

[8] T. Nagano, K. Takeda, and Nasukawa T. Knowledge discovery using robust natural language processing. In *Proceedings of the Pacific Association for Computational Linguistics (PACLING)*, pp. 189–198, 2001.

研究会報告

[1] 立花隆輝, 福田隆, 長野徹. IBM Watson の SaaS 型音声認識・音声合成サービス. 情報処理学会研究報告. 音声言語情報処理研究会 SLP-108, 2015.

[2] 和家尚希, 鈴木雅之, 長野徹, 立花隆輝, 西村雅史, 田口高也, 根本清貴, 太刀川弘和. 精神疾患診断補助に有効な発話課題と音声特徴に関する検討. 電子情報通信学会技術研究報告. 音声研究会 SP2014-133, 2015.

[3] 長野徹, 倉田岳人, 鈴木雅之, 立花隆輝, 西村雅史. 大語彙連続音声認識と音節 N-best 音声認識を用いた Spoken Term Detection の高精度化. 情報処理学会研究報告. 音声言語情報処理研究会 SLP-102, 2014.

[4] 田口高也, 太刀川弘和, 根本清貴, 鈴木雅之, 長野徹, 立花隆輝, 西村雅史, 朝田隆. 音声を用いた日常生活におけるストレス評価の予備的検討. 第 110 回日本精神神経学会学術総会, 2014.

[5] 樋口卓哉, 鈴木雅之, 長野徹, 立花隆輝, 西村雅史, 田口高也, 根本清貴, 太刀川弘和. 携帯端末を用いて日常的に収録した音声からの抑うつ度推定. 日本音響学会 2014 年春季研究発表会, 2014.

[6] 根本清貴, 太刀川弘和, 長野徹, 立花隆輝, 朝田隆. OpenEAR を用いた音声による心理的ストレス検出の試み. 第 27 回日本ストレス学会学術総会, 2011.

[7] 小林正朋, 長妻令子, 立花隆輝, 長野徹, 高木啓伸. 合成音声を用いたオンライン動画音声ガイド提供の実現に向けて. 電子情報通信学会技術研究報告. 福祉情報工学研究会 WIT-52, 2010.

[8] 立花隆輝, 長野徹, 高木啓伸, 西村雅史. 音声合成を用いたインターネット動画用音声ガイド. 情報処理学会研究報告. 音声言語情報処理研究会 SLP-80, 2010.

[9] 長野徹, 立花隆輝, 伊東伸泰, 西村雅史. アクセントクラスを用いた統計的 TTS フロントエンドの改善. 日本音響学会 2008 年春季研究発表会, 2008.

[10] 立花隆輝, 長野徹, 倉田岳人, 西村雅史. 複数のモデルを利用した自動アクセントラベリング. 日本音響学会 2007 年春季研究発表会, 2007.

[11] 長野徹, 立花隆輝, 森信介, 西村雅史. 確率モデルを用いたテキスト音声合成用フロントエンドの改善. 日本音響学会 2007 年春季研究発表会, 2007.

[12] 立花隆輝, 長野徹, 倉田岳人, 西村雅史, 馬場口登. 音声合成のための自動アクセントラベリング. 情報処理学会研究報告. 音声言語情報処理研究会 SLP-65, 2007.

[13] 西村雅史, 立花隆輝, 長野徹, 倉田岳人. 全自動構築可能なテキスト音声合成システムの検討. 日本音響学会 2006 年秋季研究発表会, 2006.

[14] 長野徹, 森信介, 西村雅史. 確率モデルを用いた読み及びアクセント推定. 情報処理学会研究報告. 音声言語情報処理研究会 SLP-57, 2005.

[15] 松澤裕史, 長野徹, 村上明子, 浦本直彦, 武田浩一. ライフサイエンス向けテキストマイニングツール MedTAKMI. 情報処理学会研究報告. データベース・システム研究会 DBS-130, 2003.

[16] 松澤裕史, 長野徹, 村上明子, 竹内広宜, 武田浩一, 神田靖. バイオメディカル文献データベースを対象とするテキストマイニングシステム MedTAKMI. 日本ソフトウェア科学会 データマイニング研究会 第 3 回データマイニングワークショップ, 2002.

[17] 長野徹, 松澤裕史, 浦本直彦. ライフサイエンス分野における木構造を用いたキーワードの相関抽出. 情報処理学会 データベースシステム研究会データベースと Web 情報システムに関するシンポジウム（DBWeb2003）, 2003.

[18] 長野徹. テキストからの意図抽出. 日本行動計量学会 第 30 回大会, 2002.

[19] 長野徹, 武田浩一, 那須川哲哉. テキストマイニングのための情報抽出. 情報処理学会研究報告. 情報学基礎研究会 FI-60, 2000.

[20] 那須川哲哉, 長野徹, 武田浩一. 大量のテキストからの知識マイニング. 情報処理学会研究報告. 知能と複雑系研究会 ICS-118, 1999.

[21] 長野徹, 那須川哲哉. テキストマイニングのための情報抽出. 知識発見のための自然言語処理シンポジウム, 1999.

[22] 長野徹. テキストマイニングのための情報抽出 －情報レベルの最適化－. 情報処理学会 第 59 回全国大会 講演論文集, 1999.

[23] 那須川哲哉, 長野徹. 知識発見のためのテキストマイニング技術. 情報処理学会 第 59 回全国大会 講演論文集, 1999.

[24] 長野徹, 那須川哲也, 諸橋正幸. テキストマイニング －システムの概要－. 人工知能学会 第 13 回全国大会 講演論文集, 1999.

[25] 那須川哲哉, 諸橋正幸, 長野徹. テキストマイニング：膨大な文書データからの知識獲得：概要. 情報処理学会 第 57 回全国大会 講演論文集, 1998.

[26] 諸橋正幸, 那須川哲哉, 長野徹. テキストマイニング：膨大な文書データからの知識獲得：意図の認識. 情報処理学会 第 57 回全国大会 講演論文集, 1998.

解説記事ほか

[1] 長野徹, 立花隆輝. お客様の音声をビジネスに生かす音声認識－音声ビッグデータの活用の広がり. *IBM PROVISION*, No. 83, pp. 52–55, 2014.

[2] 立花隆輝, 長野徹, 西村雅史. 作業効率の高いテキスト合成音声チューニング環境. *IBM PROVISION*, No. 66, pp. 67–73, 2010.

[3] N. Uramoto, H. Matsuzawa, T. Nagano, A. Murakami, H. Takeuchi, and K. Takeda. A text-mining system for knowledge discovery from biomedical documents. *IBM Systems Journal*, Vol. 43, No. 3, pp. 516–533, 2004.

[4] T. Nasukawa and T. Nagano. Text analysis and knowledge mining system. *IBM Systems Journal*, Vol. 40, No. 4, pp. 967–984, 2001.

[5] 那須川哲哉, 諸橋正幸, 長野徹. テキスト・マイニング －膨大な文書データの自動解析による知識発見. 情報処理学会 情報処理, Vol. 40, No. 4, pp. 358–364, 1999.

特許

[1] INFORMATION PROCESSING APPARATUS, INFORMATION PROCESSING METHOD, INFORMATION PROCESSING SYSTEM, AND PROGRAM. Application number: 20120316880.

[2] INFORMATION PROCESSING APPARATUS, INFORMATION PROCESSING METHOD, INFORMATION PROCESSING SYSTEM, AND PROGRAM. Application number: 20120197644.

[3] DIALOG SERVER FOR HANDLING CONVERSATION IN VIRTUAL SPACE METHOD AND COMPUTER PROGRAM FOR HAVING CONVERSATION IN VIRTUAL SPACE. Application number: 20120158879.

[4] METHOD AND SYSTEM FOR POSITION DETECTION OF A SOUND SOURCE. Patent number: 8165317.

[5] GENERATING OBJECTIVELY EVALUATED SUFFICIENTLY NATURAL SYNTHETIC SPEECH FROM TEXT BY USING SELECTIVE PARAPHRASES. Patent number: 8015011.

[6] STOCHASTIC PHONEME AND ACCENT GENERATION USING ACCENT CLASS. Application number: 20100125459.

[7] METHOD AND SYSTEM TO ANALYZE DATA. Patent number: 7493252.

[8] TECHNIQUE OF GENERATING HIGH QUALITY SYNTHETIC SPEECH. Application number: 20080183473.

[9] STOCHASTIC SYLLABLE ACCENT RECOGNITION. Application number: 20080177543.

[10] GRAPHICS IMAGE GENERATION AND DATA ANALYSIS. Application number: 20070185904.

[11] EVALUATION INFORMATION GENERATING SYSTEM, EVALUATION INFORMATION GENERATING METHOD, AND PROGRAM PRODUCT OF THE SAME. Application number: 20050283377.