TR-I-0356

単語 trigram を利用した文音声認識アルゴリズムユーザーズマニュアル
trigram sentence recognition system User's Manual

村上仁一
Jin'ichi Murakami

1993.3

概要

本レポートでは単語の trigram を使用した連続文音声認識システムについて、その使用方法を説明する。

目次

第 1 章

はじめに

このプログラムは単語の trigram を使用した連続文音声認識システムで、不特定話者の文認識が可能です。ここでは、その使用方法を説明します。単語 trigram モデルは bigram モデルに比べ、計算量は多いものの非常に高い音声認識性能をもっています。

第 2 章

デモシステムの概要

この認識プログラムは基本的には１パスＤＰ（Viterbi サーチ）の構造をとっています。ただし音響処理としては音素認識をおこない、音素 HMM を接続して単語 HMM を作っています。そしてこの単語を連続的に認識している形態になっています。詳しくは文献 [3],[4] を見て下さい。

音素認識としては連続分布型 HMM を利用しています。ただし、ガウス分布の対角成分のみを利用しています。 HMM については文献 [1, 2] を参考にしてください。

単語の trigram は、１パスＤＰにおける単語接続のところで音声の尤度と共に言語の尤度を加える形で利用しています。

出力は漢字仮名交じり文で出力されます。

第 3 章

デモの動かし方

バッチになっています。 trigram (return) で動きます。

そうしますと */recog/trigram/main* が実行されて認識プログラムが実行されます。同時に画面に emacs が立ち上がり、認識できる文が表示されます。具体的には */language/org/demo.txt* に書いてある約 500 文が認識できます。なお、表示に関しては基本的には EUC コードになっていますが、マシンに依存する場合がありますので、適宜 kterm での漢字コードを設定して下さい。デモは認識語彙数 435 単語で、実行空間は約 24M が必要です。また、 HP9000model700 と dec station5000 内部メモリ 64 Mで実行が確認されています。

画面上で音声認識ができる用になりましたら、マウスの右ボタンをクリックして話して下さい。話し終えたら自動的に音声入力ができなくなります。左ボタンを押すと認識プログラムが実行されます。このとき入力された音声のフレーム数が出力されます。次に 0,10,20,... と数字が出力されます。この値が入力された音声のフレーム数になったとき認識結果が漢字かな交じり文で出力されます。

認識できる文を以下に示します。

認識可能な文章

もしもし。
そちらは会議事務局ですか。
はい。
そうです。
どのようなご用件でしょうか。
会議に申込たいのですが。
どのような手続きをすればよろしいのでしょうか。
登録用紙で手続きをしてください。
登録用紙はすでにお持ちでしょうか。
いいえ。
まだです。
わかりました。
それでは登録用紙をお送りいたします。
ご住所とお名前をお願いします。
住所は大阪市北区茶屋町二十三です。
名前は鈴木真弓です。
わかりました。
登録用紙を至急送らせていただきます。
よろしくお願いします。
それでは失礼します。
はい。
こちらは会議事務局です。
会議の参加料について教えていただきたいのですが。
今会議に申し込めば参加料はいくらですか。
はい。
参加料は現在お一人3万5千円です。
来月お申込になりますと4万円です。
参加料には予稿集代と歓迎会費が含まれています。
私は情報処理学会の会員なのですが。
参加料の割引はないのですか。
今回は割引を行なっておりません。
そうですか。
参加料はどのようにお支払いしたらよいのですか。
参加料は銀行振込です。
案内書に記載されている口座番号に振り込んでください。
また期限は今年いっぱいです。
わかりました。
どうもありがとうございました。
どういたしまして。
わからない点がございましたらいつでもお聞きください。
失礼いたします。
はい。
こちらは会議事務局です。
会議に論文を発表したいと思っているのですが。
会議の内容について教えてください。
今回の会議は通訳電話に関連する広範な研究分野を含んでいます。
言語学や心理学を専攻する方にも参加していただく予定です。
わかりました。
ところで会議での公式言語はなんですか。
英語と日本語です。

私は日本語が全然わからないのですが。
発表が日本語で行われる場合英語への同時通訳はあるのですか。
はい。
英語への同時通訳を用意しております。
わかりました。
どうもありがとうございました。
さようなら。
こちらは会議事務局です。
会議について詳しいことを教えてください。
会議の案内書はお持ちですか。
いいえ。
持っていません。
そうですか。
会議は８月２２日から２５日まで京都国際会議場で開催されます。
参加料は４万円です。
発表を希望されるのでしたら３月２０日までに要約を提出してください。
会議の案内書をお送りいたしますのでそれをご覧ください。
失礼ですがお名前とご住所をお願いいたします。
アダム・スミスです。
住所は大阪市東区玉造２丁目２７の７です。
わかりました。
電話番号もお聞きしたいのですが。
はい。
３７２の８０１８です。
３７２の８０１８でございますね。
はい。
そうです。
それではよろしくお願いします。
失礼します。
はい。
こちらは会議事務局でございます。
ちょっとお願いがあるのですが。
私は会議に申し込みをした者です。
参加を取り消したいのですが。
お名前をお伺いできますでしょうか。
はい。
ベル研のジム・ワイベルです。
すでに登録料の８万５千円を振り込まれておられますね。
はい。
そうです。
登録料を払い戻していただけますか。
お気の毒ですができません。
案内書にも書いていますが。
９月２７日以後の取り消しに対する払い戻しはできません。
後日プログラムと予稿集をお送りいたします。
では誰かが私の代わりに参加することはできますか。
それは別に問題ありません。
代理人が参加する場合はあらかじめこちらまでお知らせください。
わかりました。
代理人が決まりましたらお知らせいたします。
では失礼します。
はい。
こちらは会議事務局ですが。

会議の間に市内観光があるそうですが。
まだ参加できますか。
はい。
まだ参加可能です。
８月５日の午後に清水寺金閣寺龍安寺等を見学します。
参加なさいますか。
参加料はいくらですか。
８千円です。
参加料には夕食代も含まれています。
講演者も参加されるのですか。
講演者の何人かは参加する予定になっています。
そうですか。
それでは参加したいと思います。
ではお名前と人数をお願いいたします。
ケン・ブラウンと申します。
家内と参加します。
集合場所は会議場の受付の前になっております。
参加料は当日集合場所でお支払いください。
わかりました。
ありがとうございました。
ではお待ちしております。
はい。
こちらは会議事務局です。
会議で扱う話題に関して質問したいんですが。
はい。
なんでしょうか。
機械翻訳という話題が案内書に載っていますが。
具体的にこれはどういう内容のものなんですか。
申し訳ありませんがこちらでは専門的な質問にお答えできません。
第二版の案内書に会議で発表される論文の題目が載っております。
そちらを見ていただけないでしょうか。
いいですよ。
それでは早急にその案内書を送ってください。
送り先は大阪市東区城見２の１の６１渡辺明です。
大阪市東区城見２の１の６１渡辺明様ですね。
はい。
早速送らせていただきます。
他に何かございますか。
いいえ。
ありません。
ありがとうございました。
失礼します。
はい。
会議事務局です。
ちょっとお聞きしたいことがあるんですが。
私は今度の会議に発表したいと思っているんですが。
どのような手続きをすればよろしいでしょうか。
まず２百字の要約を３月２０日までにこちらまでお送りください。
こちらで審査を行い５月２０日までに結果をお送りします。
投稿が受理された場合原稿用紙を同封いたします。
６月３０日までに原稿の送付をお願いします。
わかりました。
要約はどのような書式で書けばいいんですか。

所定の申し込み用紙がありますのでそれに記入してください。
それでは申し込み用紙を送りますので送り先をお願いします。
わかりました。
人工知能研究所のジョージ・オハラです。
住所は東京都豊島区東池袋３丁目２番５号です。
人工知能研究所のジョージ・オハラ様ですね。
ご住所は東京都豊島区東池袋３丁目２番５号でよろしいですね。
はい。
そうです。
それでは申し込み用紙の送付をよろしくお願いします。
はい。
わかりました。
では早速お送りいたします。
失礼いたします。
そちら会議事務局ですか。
はい。
会議事務局です。
なんのご用件でしょうか。
会議場へどうやって行ったらいいか教えて欲しいんですが。
今京都駅にいるんです。
地下鉄で北大路駅まで行ってください。
そこから国際会議場へ行くバスが利用できます。
北大路駅ではタクシーも利用できます。
京都駅からタクシーで会議場まで行くにはいくらぐらいかかりますか。
京都駅からですとおよそ６千円かかります。
では北大路駅からですといくらぐらいかかりますか。
北大路駅からですとおよそ９百円です。
わかりました。
どうもありがとうございました。
いいえ。
どういたしまして。
もしもし。
はい。
会議事務局でございます。
会議の宿泊施設についてお尋ねしたいのですが。
そちらでどこか紹介していただけますか。
はい。
私共でご紹介できるホテルは京都ホテルと京都プリンスホテルです。
一人部屋の値段は一晩７千円から１万円です。
二人部屋の値段は９千５百円から６万円です。
そうですか。
どちらのホテルが会議場に近いですか。
京都プリンスホテルが会議場には近いんですが。
それでは京都プリンスホテルを予約したいのですが。
ホテルの手配もしていただけるのですか。
はい。
京都ホテルと京都プリンスホテルは予約できます。
そうですか。
では京都プリンスホテルの７千円の一人部屋をお願いします。
はい。
京都プリンスホテルの７千円の一人部屋ですね。
はい。
そうです。

いつからお泊まりになりますか。
８月４日の夜からです。
８日の朝までお願いします。
わかりました。
少々お待ちください。
お部屋が取れるかどうか調べます。
お部屋をお取りできます。
ではお名前とご住所をお願いします。
中村一雄です。
住所は東京都港区新橋１丁目１番３号です。
電話番号もお願いします。
電話番号は３３１の２５２１です。
わかりました。
京都プリンスホテルに８月４日から８日まで一人部屋をお取りしました。
どうもありがとうございました。
失礼します。
もしもし。
そちらは会議事務局ですか。
はい。
そうです。
会議に申込たいのですが。
登録用紙はすでにお持ちでしょうか。
いいえ。
まだです。
わかりました。
それでは登録用紙をお送りいたします。
ご住所とお名前をお願いします。
住所は大阪市北区茶屋町二十三です。
名前は鈴木真弓です。
わかりました。
登録用紙は至急送らせていただきます。
わからない点がございましたらいつでもお聞きください。
ありがとうございます。
それでは失礼します。
どうも失礼いたします。
もしもし。
こちらは会議事務局です。
会議に参加したいのですが。
どうすればよろしいですか。
まず登録用紙で手続きをしていただかなくてはなりませんが。
もう登録用紙はお持ちでしょうか。
まだです。
用紙を送ってください。
ではご住所とお名前をお願いします。
住所は大阪市東区徳井町一の二です。
名前は清水太郎です。
わかりました。
参加料は要るのでしょうか。
はい。
登録費としてお一人三万五千円が必要です。
そうですか。
どうもありがとうございました。
失礼いたします。

もしもし。
そちらは会議事務局ですか。
はい。
そうです。
会議に申込たいのですが。
登録用紙はすでにお持ちでしょうか。
いいえ。
まだです。
わかりました。
それでは登録用紙をお送りいたします。
ご住所とお名前をお願いします。
住所は大阪市北区茶屋町二十三です。
名前は鈴木真弓です。
わかりました。
登録用紙は至急送らせていただきます。
わからない点がございましたらいつでもお聞きください。
ありがとうございます。
それでは失礼します。
どうも失礼いたします。
もしもし。
こちらは会議事務局です。
会議に参加したいのですが。
どうすればよろしいですか。
まず登録用紙で手続きをしていただかなくてはなりませんが。
もう登録用紙はお持ちでしょうか。
まだです。
用紙を送ってください。
ではご住所とお名前をお願いします。
住所は大阪市東区徳井町一の二です。
名前は清水太郎です。
わかりました。
参加料は要るのでしょうか。
はい。
登録費としてお一人三万五千円が必要です。
そうですか。
どうもありがとうございました。
失礼いたします。
もしもし。
そちらは会議事務局ですか。
はい。
そうです。
どのようなご用件でしょうか。
会議に申込たいのですが。
どのような手続きをすればよろしいのでしょうか。
登録用紙で手続きをしてください。
登録用紙はすでにお持ちでしょうか。
いいえ。
まだです。
わかりました。
それでは登録用紙をお送りいたします。
ご住所とお名前をお願いします。
住所は大阪市北区茶屋町二十三です。
名前は鈴木真弓です。

わかりました。
登録用紙を至急送らせていただきます。
よろしくお願いします。
それでは失礼します。
はい。
こちらは会議事務局です。
会議の参加料について教えていただきたいのですが。
今会議に申し込めば参加料はいくらですか。
はい。
参加料は現在お一人3万5千円です。
来月お申込になりますと4万円です。
参加料には予稿集代と歓迎会費が含まれています。
私は情報処理学会の会員なのですが。
参加料の割引はないのですか。
今回は割引を行なっておりません。
そうですか。
参加料はどのようにお支払いしたらよいのですか。
参加料は銀行振込です。
案内書に記載されている口座番号に振り込んでください。
また期限は今年いっぱいです。
わかりました。
どうもありがとうございました。
どういたしまして。
わからない点がございましたらいつでもお聞きください。
失礼いたします。
はい。
こちらは会議事務局です。
会議に論文を発表したいと思っているのですが。
会議の内容について教えてください。
今回の会議は通訳電話に関連する広範な研究分野を含んでいます。
言語学や心理学を専攻する方にも参加していただく予定です。
わかりました。
ところで会議での公式言語は何ですか。
英語と日本語です。
私は日本語が全然わからないのですが。
発表が日本語で行われる場合英語への同時通訳はあるのですか。
はい。
英語への同時通訳を用意しております。
わかりました。
どうもありがとうございました。
さようなら。
こちらは会議事務局です。
会議について詳しいことを教えてください。
会議の案内書はお持ちですか。
いいえ。
持っていません。
そうですか。
会議は8月22日から25日まで京都国際会議場で開催されます。
参加料は4万円です。
発表を希望されるのでしたら3月20日までに要約を提出してください。
会議の案内書をお送りいたしますのでそれをご覧ください。
失礼ですがお名前とご住所をお願いいたします。
アダム・スミスです。

住所は大阪市東区玉造２丁目２７の７です。
わかりました。
電話番号もお聞きしたいのですが。
はい。
３７２の８０１８です。
３７２の８０１８でございますね。
はい。
そうです。
それではよろしくお願いします。
失礼します。
はい。
こちらは会議事務局でございます。
ちょっとお願いがあるのですが。
私は会議に申し込みをした者です。
参加を取り消したいのですが。
お名前をお伺いできますでしょうか。
はい。
ベル研のジム・ワイベルです。
すでに登録料の８万５千円を振り込まれておられますね。
はい。
そうです。
登録料を払い戻していただけますか。
お気の毒ですができません。
案内書にも書いていますが。
９月２７日以後の取り消しに対する払い戻しはできません。
後日プログラムと予稿集をお送りいたします。
では誰かが私の代わりに参加することはできますか。
それは別に問題ありません。
代理人が参加する場合はあらかじめこちらまでお知らせください。
わかりました。
代理人が決まりましたらお知らせいたします。
では失礼します。
はい。
こちらは会議事務局ですが。
会議の間に市内観光があるそうですが。
まだ参加できますか。
はい。
まだ参加可能です。
８月５日の午後に清水寺金閣寺龍安寺等を見学します。
参加なさいますか。
参加料はいくらですか。
８千円です。
参加料には夕食代も含まれています。
講演者も参加されるのですか。
講演者の何人かは参加する予定になっています。
そうですか。
それでは参加したいと思います。
ではお名前と人数をお願いいたします。
ケン・ブラウンと申します。
家内と参加します。
集合場所は会議場の受付の前になっております。
参加料は当日集合場所でお支払いください。
わかりました。

ありがとうございました。
ではお待ちしております。
はい。
こちらは会議事務局です。
会議で扱う話題に関して質問したいんですが。
はい。
何でしょうか。
機械翻訳という話題が案内書に載っていますが。
具体的にこれはどういう内容のものなんですか。
申し訳ありませんがこちらでは専門的な質問にお答えできません。
第二版の案内書に会議で発表される論文の題目が載っております。
そちらを見ていただけないでしょうか。
いいですよ。
それでは早急にその案内書を送ってください。
送り先は大阪市東区城見２の１の６１渡辺明です。
大阪市東区城見２の１の６１渡辺明様ですね。
はい。
早速送らせていただきます。
他に何かございますか。
いいえ。
ありません。
ありがとうございました。
失礼します。
はい。
会議事務局です。
ちょっとお聞きしたいことがあるんですが。
私は今度の会議に発表したいと思っているんですが。
どのような手続きをすればよろしいでしょうか。
まず２百字の要約を３月２０日までにこちらまでお送りください。
こちらで審査を行い５月２０日までに結果をお送りします。
投稿が受理された場合原稿用紙を同封いたします。
６月３０日までに原稿の送付をお願いします。
わかりました。
要約はどのような書式で書けばいいんですか。
所定の申し込み用紙がありますのでそれに記入してください。
それでは申し込み用紙を送りますので送り先をお願いします。
わかりました。
人工知能研究所のジョージ・オハラです。
住所は東京都豊島区東池袋３丁目２番５号です。
人工知能研究所のジョージ・オハラ様ですね。
ご住所は東京都豊島区東池袋３丁目２番５号でよろしいですね。
はい。
そうです。
それでは申し込み用紙の送付をよろしくお願いします。
はい。
わかりました。
では早速お送りいたします。
失礼いたします。
そちら会議事務局ですか。
はい。
会議事務局です。
何のご用件でしょうか。
会議場へどうやって行ったらいいか教えてほしいんですが。

今京都駅にいるんです。
地下鉄で北大路駅まで行ってください。
そこから国際会議場へ行くバスが利用できます。
北大路駅ではタクシーも利用できます。
京都駅からタクシーで会議場まで行くにはいくらぐらいかかりますか。
京都駅からですとおよそ６千円かかります。
では北大路駅からですといくらぐらいかかりますか。
北大路駅からですとおよそ９百円です。
わかりました。
どうもありがとうございました。
いいえ。
どういたしまして。
もしもし。
はい。
会議事務局でございます。
会議の宿泊施設についてお尋ねしたいのですが。
そちらでどこか紹介していただけますか。
はい。
私共でご紹介できるホテルは京都ホテルと京都プリンスホテルです。
一人部屋の値段は一晩７千円から１万円です。
二人部屋の値段は９千５百円から６万円です。
そうですか。
どちらのホテルが会議場に近いですか。
京都プリンスホテルが会議場には近いんですが。
それでは京都プリンスホテルを予約したいのですが。
ホテルの手配もしていただけるのですか。
はい。
京都ホテルと京都プリンスホテルは予約できます。
そうですか。
では京都プリンスホテルの７千円の一人部屋をお願いします。
はい。
京都プリンスホテルの７千円の一人部屋ですね。
はい。
そうです。
いつからお泊まりになりますか。
８月４日の夜からです。
８日の朝までお願いします。
わかりました。
少々お待ちください。
お部屋が取れるかどうか調べます。
お部屋をお取りできます。
ではお名前とご住所をお願いします。
中村一雄です。
住所は東京都港区新橋１丁目１番３号です。
電話番号もお願いします。
電話番号は３３１の２５２１です。
わかりました。
京都プリンスホテルに８月４日から８日まで一人部屋をお取りしました。
どうもありがとうございました。
失礼します。

第 4 章

アルゴリズムの概要

4.1 まえがき

人とコンピュータのインターフェースとして、あるいは異なる母国語を話す人と人とのインターフェースとして、自然言語を認識できる連続音声認識システムが研究され、そして報告されている。これらのシステムの多くは、音響パラメータからのパターンマッチングの方法では高い認識性能が得られないため、言語情報を利用することで認識性能を向上させている。

基本的に音声認識において使用される言語情報としては、高いカバー率と低い perplexity を持つ言語情報が、相応しいと考えられる。そして、経験的には、音節認識率と、言語 perplexity と文認識率には相関があることが知られて、中川等は、この関係を推定している。この結果を見ると、文認識率を向上させるためには、音節認識率を向上させるよりも、言語の perplexity を下げるほうが、全体の認識性能が向上するように思われる。

言語の perplexity を下げる方法には、大きくわけて 2 つの方法がある。1 つは、より複雑なルールを書いていくことで、もう 1 つは、確率を採用することである。確率を採用した文法としては、CFG のルールに確率をいれた SCFG、や、bigram,trigram そして、HMM などがある。これらの確率文法の perplexity をみると、確率を計算する方法や、学習データ量、そしてオープンデータとクローズデータの差などの問題があるが、trigram がもっとも小さい値をしめしているようである。

しかし、trigram モデルは、前の前の単語と前の単語が存在したときに、現在の単語に遷移する確率を逐次計算する必要があるため、非常に多くの空間と計算量が必要である。そのため、連続単語音声認識に、直接この形では計算されていないと思われる。例えば trigram モデルをもちいた連続音声認識の報告として、IBM が初めてであると思われるが、これは単語孤立発声の音声認識である。鹿野、や南他の論文では、品詞と組み合わた、疑似的な trigram を用いている。松永らは、LR パーザーと組み合わせることによって、trigram のサーチ空間を減少させている。村上らの研究は、音節のラティス構造を仮定したシュミレーション結果を報告している。

ところで連続単語認識における計算方法として、1 パスＤＰが良く知られている。このアルゴリズムにおいて、さらに計算量を減らすために、ビームサーチがある。ここでは、ビームサーチを使用することにより、trigram モデルのサーチ空間が大幅に減らせることを示し、この結果、計算量の大幅な減少を示す。

ここでは、まず初めに trigram を利用した 1 パスＤＰの連続音声認識のアルゴリズムについて述べる。次にビームサーチを用いた trigram のサーチアルゴリズムについて述べ、メモリー空間および、計算量の減少量について報告する。最後に、このアルゴリズムを用いた連続音声認識実験の実験結果について報告する。この実験の結果、単語の trigram モデルは bigram モデルと比較して、大幅に認識性能が改良されることが示された。

4.2 単語の trigram model を使用した連続音声認識システム

この章では単語の trigram model を利用した連続音声認識システムのアルゴリズムについて述べる。

言語情報として単語の trigram を用いた場合、求める解は、文候補 $l(w1, w2, .., wN)$ を以下のように定式化して、これを最大にする文 $w1, w2, ..., wN$ を選択することである。

$$\Sigma \log(Pa(wi)) + \alpha \cdot \Sigma p(wi+2|wi+1, wi)$$

ここで $Pa(wi)$ は単語 wi の音響的尤度、$p(wi+2|wi+1, wi)$ は単語 wi の次に単語 $wi+1$ が継ったときに $wi+2$ に遷移する確率、α は音響尤度と言語の連鎖確率を結びつける結合定数である。なお、通常 $p(wi+2|wi+1, wi)$ の値は、大量の文から予め計算しておく。

ところで、従来から連続音声認識のためのアルゴリズムとして２段ＤＰや、１パスＤＰ、level building などのアルゴリズムが提案されている。このうち１パスＤＰは入力された音響パラメータに対して、データを次々に計算していくため、基本的に言語モデルにマルコフモデルを使用した場合、非常に整合が取りやすい。具体的には、１パスＤＰにおいて、各認識単語の最初の状態は、前の単語の状態と、各認識単語の最後の状態の最大値をサーチすることによって計算が進んで行くが、各認識単語の最後の状態と単語の最初の状態の trigram の確率を掛けることによって１パスＤＰと trigram が結合できる。ただし、trigram は２つ前の単語が決定されて初めて現在の単語の出現確率が計算できるため、１パスＤＰの内部状態においては、現在の単語と以前の単語の２つの状態の値を、つねに保持する必要がある。したがって認識単語数の２乗の空間が常に必要になる。各単語それぞれにモデルを持っているとして、このアルゴリズムを表 4.1 に示す。

表 4.1: 単語の trigram を用いた Viterbi サーチのアルゴリズム

[定義]
l_w：単語 w における状態数 a_{ij}^w：単語 w における状態 s_i から状態 s_j への遷移確率 $b_j^w(v)$：単語 w の状態 s_j におけるベクトル v の出力確率 $Tr(w_2, w_1, w_0)$： 単語 w_2, w_1 が出現したときに w_0 に遷移する確率 $P(w_0\|(w_2, w_1))$ Q：語彙数 T：入力フレーム数 $O(t)$：フレーム t における観測ベクトル $G_t(w_1, w_0, i)$：前単語 w_1, 単語 w_0, 状態 i での フレーム t までの最大累積尤度
[初期化]
$w_0 = 0, ..., Q-1$ において step1 を実行 1) $G_0(start, w_0, 0) = Tr(start, start, w_0)$ $start$ は文頭を意味
[Viterbi サーチ]
$t = 0, 1, .., T-1$ において step2,step6 を実行 2) $w_1 = 0, ..., Q-1$ において step3 を実行 3) $w_0 = 0, ..., Q-1$ において step4 を実行 4) $i = 0, 1, ..., l_{w_0} - 2$ において step5 を実行 5) $G_t(w_1, w_0, i) =$ $\max(G_{t-1}(w_1, w_0, i) \times a_{i,i}^{w_0} \times b_i^{w_0}(O_t),$ $G_{t-1}(w_1, w_0, i-1) \times a_{i-1,i}^{w_0} \times b_{i-1}^{w_0}(O_t))$
[単語境界の計算]
6) $w_1 = 0, 1, ..., Q-1$ において step7 を実行 7) $w_0 = 0, 1, ..., Q-1$ において step8 を実行 8) $\Delta = \max_{0 \le w_2 \le Q-1}(G_{t-1}(w_2, w_1, l_{w_1} - 2)$ $\times a_{l_{w_1}-2, l_{w_1}-1}^{w_1} \times b_{l_{w_1}-1}^{w_1}(O_t) \times Tr(w_2, w_1, w_0))$ もし $\Delta \ge G_t(w_1, w_0, 0)$ ならば $G_t(w_1, w_0, 0) = \Delta$

4.3 beam search を使用した連続音声認識システム

1パスDPにおいては、フレームごとに全ての単語の状態の遷移および単語の接続の計算がされる。しかし、最終的には確率が最大の文候補のみが出力されるため、最適経路として可能性の低いものは、以後の探索から除外できる。つまり、フレームごとに全ての計算を行なう必要がなく、最尤なものから、ある個数のみを計算しておけばよいと考えられる。このビームサーチを行なうことにより、計算量およびメモリー空間が大幅に減らせる。

具体的には、すべての w_1, w_0, i に対して6）の式

6） $G_t(w_1, w_0, i) = \max(G_{t-1}(w_1, w_0, i) + a_{i,i}^{w_0} \times b_j^{w_0}(O_t),$
$G_{t-1}(w_1, w_0, i-1) + a_{i-1,i}^{w_0} \times b_i^{w_0}(O_t))$

の計算を行なわずに、高い尤度から、ある程度の個数（ビーム幅）のみを計算する。ただし計算アルゴリズムでは、この式にしたがわず、Gを w_1, w_0, i の3次元配列にしなくてビーム幅の1次元配列にして w_i, w_0, はそれぞれ、ビームの1次元配列で記憶しておく必要がある。

このビームサーチを行なうことにより、以下のメリットが生じる

1）メモリー量の削減

フルサーチが、$G_t(w_1, w_0, i)$ を記憶するために、認識語彙数 × 認識語彙数 × 単語の状態数が必要であるのに対し、ビーム幅のみしか必要なくなるため、必要なメモリー量が大幅に減少する。

2）計算量の削減

ビーム幅に入っている各単語の状態が単語の最後の状態にないばあい、ビーム幅は、最大でも初めのビームの2倍にしか広がらない。（つまり、現在の状態を継続する形になるか、次の状態を継続するかの2状態）したがって、この状態が続くと、計算量が大幅に削減される。

3） trigram の連鎖確率の計算量の削減

trigram の連鎖確率は、展開して記憶すると、認識単語数の3乗の空間が必要なため、通常はリスト形式で計算機上に記憶される。したがって1パスＤＰの計算上、連鎖確率を調べるためには、計算コストが必要になる。たとえば N 種類の trigram の連鎖確率があったとき log2N の比較をして、値が得られる。ビームサーチを使用した場合、ビーム幅にはいる単語のみの trigram の連鎖確率の値の検索ですむため、計算量が大幅に減る。

インプリメンデーション上でのメモリー量、および計算量の削減

なお、実際のインプリメンテーションでは、コンピュータのメモリー容量と計算量を減らすために、さらにつぎのような方法をとった。これらの工夫をすることにより、認識単語数が1500でも実行空間が 15Mbyte で実行可能になった。

Viterbi サーチの経路計算　今回使用した認識アルゴリズムでは最終的に最尤度のワード列を知るために、計算の途中において選択した経路を残す必要がある。このために［最大認識単語数^2× 最大の単語の状態数 × 最大認識時間のフレーム長］の空間が必要である。しかし、どの経路を選択したかを最大シンボル出力確率とともに、次の状態に渡すことによって［最大認識単語数^2× 最大の単語の状態数 × 文を構成する最大単語数］の空間で計算可能である。後者を選択したとき計算時間は少し増加するが、一般的には文を構成する最大単語数は最大認識時間のフレーム長より小さいため、必要なメモリ量は大幅に削減できる。

認識単位・音素　このアルゴリズムでは、各単語ごとにＨＭＭのモデルを持っているとしたが、認識単位を音素とすることによって、単語の尤度計算において共通なモデルが利用できる。つまり、単語のＨＭＭは、音素のＨＭＭが、連続的に接続されて構成されていると考える。これによってＨＭＭの尤度計算が、各単語の状態全てにおこなうかわりに、１フレームごとに各音素のＨＭＭの尤度を計算し、単語の尤度は、この値をコピーするとによって、同様な計算が可能である。これにより大幅な計算量が削減できる。

トライグラムの値の記憶　トライグラムの値を記憶しておくには［最大認識単語数^3］の空間が必要である。しかし、サンプリングデータ中に存在する組み合わせのみをリスト構造で記憶することにより、大幅にメモリを節約できる。

ビーム幅の決定　ビーム幅の決定には、各フレームにおいて

1）尤度のしきい値をもって、計算を打ち切る方法 2）一定の個数を残す方法

の２点がある。尤度の閾値で計算を打ち切る方法では、計算は早いが、その閾値を、前持って決定しておく必要がある。したがって、ある状態では全ての経路が打ち切られる可能性がある。

一方、一定の個数を残す方法では、フレームごとのソーティングが必要になるため、これが大きなオーバーヘッドになると考えられてきた。しかし、この部分では、一定の個数を残す尤度を計算するためでよく、全てをフルソートする必要がないため、計算量が通常のソーティングに比

表 4.2: 自由発話における連続音声認識の実験条件

基本アルゴリズム	Continuous mixture HMM + Beam search + word trigram
Mixture 数	最大 14 (各音素によって変化)
1 音素あたりの状態数	3state4loop left-right model
使用パラメータ	LPC ケプストラム 16 次 + パワー + デルタパワー + デルタケプストラム 16 次
フレーム間隔	10ms
フレーム周期	5ms
HMM の学習音声	男性 12 人分の単語発声 (216 単語)
音素カテゴリ数	31 音素
認識単語数	435
ビーム幅	4096
duration control	なし
言語情報	単語のトライグラム

較すると遥かに少なくて済む。具体的には N 個のデータがあったとき、ソーティングには Nlog2N の比較が必要であるが、ここでは、ある幅の尤度のみを計算すればよいため、log2N ですむ。そのため、あまり大きなオーバーヘッドにならずにすむと予想される。

4.4 trigram model を使用した連続日本文認識システム

この章では、単語の trigram モデルをもちいた連続文認識実験の結果をしめす。認識対象は、国際会議のモデル文で、文認識率を計算した。発声はアナウンサーで、不特定話者の音素モデルを使用した。音素モデルは小坂氏が作成したものである。その他の実験条件は表 3 にしめす。

なお、使用した HMM の音素の平均音素認識率は 80% であった。

以上に示した方法でデモシステムは作成されている。

第 5 章

移植に関して

このプログラムは SpeechIn と Wavepara34 の 2 つのプログラムが必要です。前者は音声の波形をコンピュータに取り込むプログラムで、後者は波形を $logpower+16ordercepstrum+\delta logpower+\delta 16ordercepstrum$ に $float$ の形式で変換するものです。このプログラムは main.c の中で shell の形式で絶対 path の形で呼ばれています。したがって、directory を変更するときには注意してください。その他は全て相対 path でプログラムされています。

他の文章を認識したい場合は多くの変更点が必要です。

trigram を使用した連続音声認識では、始めに trigram の値を決定する必要があります。*/language/org* が trigram を計算するのに必要な text ファイルです。*/language/org/makedata.batch* を動かすことにより */language/wmarkov1/number_1.dat*, */language/wmarkov1/number_2.dat*, */language/wma* が作成されます。これらのファイルはテキストデータにおいて各々 unigram,bigram,trigram が何回起きたかを示すデータになります。

認識アルゴリズムは */recog/trigram/main* を動かすことによって動きます。しかし、認識アルゴリズムの本体は */recog/trigram/ngram_main.c* です。このプログラムの設定は */recog/trigram/par* で全て定義されます。このパラメータの意味は付録に示します。

```
/***************(/recog/trigram/parameter.h)******************/

# include <stdio.h>
# include <math.h>

char  DATA_FILE_NAME[256];
int   data_start, data_end;

# define SYLLABLE_FILE_DIR    "../bigram/INDEPENDENT/"   不特定話者音素モデルの場所
# define LABEL_FILE           "../bigram/label"          音素ラベルの一覧表の位置
# define DIC_FILE_NAME        "../bigram/MAU/dic1"       辞書の位置

# define  MAX_UNIGRAM_NUMBER  10000                                    unigram の最大の数
# define  UNIGRAM_FILE_NAME  "../../language/wmarkov1/number_1.dat"    unigram の位置
# define  MAX_BIGRAM_NUMBER   50000                                    bigram の最大の数
# define  BIGRAM_FILE_NAME   "../../language/wmarkov2/number_2.dat"    bigram の位置
# define  MAX_TRIGRAM_NUMBER  100000                                   trigram の最大の数
# define  TRIGRAM_FILE_NAME  "../../language/wmarkov3/number_3.dat"    trigram の位置

# define  MIN_VALUE    0.0000000001    /* 2.2e-308  */          計算上の最小値
# define  MAX_VALUE    1000000000.0    /* 1.8e308   */          計算上の最大値
# define  MAX_2_VALUE 100000.0         /* 1.8e308   */          計算上の最大値 /1000
```

```
# define  MAX_LABEL_NUMBER            31                          音素ラベルの数
# define  MAX_PHONE_STATE_NUMBER       4                          HMM の状態数
# define  MAX_MIXTURE_NUMBER          32                          HMM の最大の混合数
# define  MAX_PARAMETER_NUMBER        34                          音響パラメータの数
# define  MAX_WORD_STATE_NUMBER       64                          単語の最大音素構成数
# define  MAX_DATA_LENGTH           4096                          音声認識できる最大音響フ
レーム数
# define  WORD_NUMBER                435                          認識語彙数
# define  BEAM_SEARCH               4096 /* >= WORD_NUMBER */     ビーム幅
# define  SORT_BUFFER_NUMBER       10000 /* MAX WORD_NUMBER*WORD_NUMBER+2*BEAM_SEARCH */
                                                                  ソート用バッファー領域
# define  LEVEL_NUMBER       64                                   1 文における最大単語構成
数
# define  NGRAM_WEIGHT      1.0                                   音響尤度と言語尤度の重み
# define  DURATION_WEIGHT     0.0
```

```
/***************(/recog/trigram/parameter.h)******************/
```

ソースの大まかな構造

```
|- language (4)
|   |- org (7)
|   |- wmarkov1 (5)
|   |- wmarkov2 (7)
|   |- wmarkov3 (6)
|- recog (3)
|   |- bigram (44)
|   |   |- INDEPENDENT (53)
|   |   |- MAU (67)
|   |- ngram (47)
```

ソースの説明

```
|- language (4)            言語モデル
|   |- org (7)             テキストデータ
|   |- wmarkov1 (5)        unigram
|   |- wmarkov2 (7)        bigram
|   |- wmarkov3 (6)        trigram
|- recog (3)               認識アルゴリズム
|   |- bigram (44)         bigram を使用した文音声認識アルゴリズム
|   |   |- INDEPENDENT (53) 不特定話者用音素モデル
|   |   |- MAU (67)        特定話者 (MAU) 用音素モデル
|   |- ngram (47)          trigram を使用した文音声認識アルゴリズム
```

細かいソースの説明

```
|- language (4)
|   |- org (7)
|   |   |- demo.txt                テキストデータ (漢字仮名交じり文)
|   |   |- makedata.batch          trigram の値を得るための batch ファイル
```

```
|   |   |- sentence_number          テキストデータ（単語番号）
|   |- wmarkov1 (5)
|   |   |- a.out
|   |   |- devide_1.c               sentence_number を unigram に分割するプログラム
|   |   |- makedata.sh              unigram の値を得るための batch ファイル
|   |   |- number_1.dat             unigram のカウント数
|   |- wmarkov2 (7)
|   |   |- a.out
|   |   |- devide_2.c               sentence_number を bigram に分割するプログラム
|   |   |- makedata.sh              bigram の値を得るための batch ファイル
|   |   |- number_2.dat             bigram のカウント数
|   |- wmarkov3 (6)
|   |   |- a.out
|   |   |- devide_3.c               sentence_number を trigram に分割するプログラム
|   |   |- makedata.sh              trigram の値を得るための batch ファイル
|   |   |- number_3.dat             trigram のカウント数
|- recog (3)
|   |- bigram (44)                  bigram による文音声認識アルゴリズム
|   |   |- INDEPENDENT (53)         不特定話者音素モデル
|   |   |   |- hmm_-
|   |   |   |- hmm_CH1
|   |   |   |- hmm_CH2
|   |   |   |- hmm_H
|   |   |   |- hmm_K1
|   |   |   |- hmm_K2
|   |   |   |- hmm_N
|   |   |   |- hmm_P1
|   |   |   |- hmm_P2
|   |   |   |- hmm_Q1
|   |   |   |- hmm_S
|   |   |   |- hmm_SH
|   |   |   |- hmm_TS1
|   |   |   |- hmm_TS2
|   |   |   |- hmm_Ui
|   |   |   |- hmm_Uu
|   |   |   |- hmm_a
|   |   |   |- hmm_a1
|   |   |   |- hmm_b1
|   |   |   |- hmm_b2
|   |   |   |- hmm_by
|   |   |   |- hmm_cy
|   |   |   |- hmm_d1
|   |   |   |- hmm_d2
|   |   |   |- hmm_e
|   |   |   |- hmm_e1
|   |   |   |- hmm_g1
|   |   |   |- hmm_gy1
|   |   |   |- hmm_hy
|   |   |   |- hmm_i
|   |   |   |- hmm_i1
|   |   |   |- hmm_ky
|   |   |   |- hmm_m
|   |   |   |- hmm_my
|   |   |   |- hmm_n
```

```
|   |   |   |- hmm_ng
|   |   |   |- hmm_ngy
|   |   |   |- hmm_ny
|   |   |   |- hmm_o
|   |   |   |- hmm_ol
|   |   |   |- hmm_py
|   |   |   |- hmm_r
|   |   |   |- hmm_ry
|   |   |   |- hmm_sy
|   |   |   |- hmm_t1
|   |   |   |- hmm_t2
|   |   |   |- hmm_u
|   |   |   |- hmm_ul
|   |   |   |- hmm_w
|   |   |   |- hmm_y
|   |   |   |- hmm_z
|   |   |   |- hmm_zy
|   |   |- MAU (67)          特定話者 (MAU) 音素モデル
|   |   |   |- dic1          単語辞書 (単語と単語番号の対応)
|   |   |   |- hmm_-
|   |   |   |- hmm_CH1
|   |   |   |- hmm_CH2
|   |   |   |- hmm_H
|   |   |   |- hmm_K1
|   |   |   |- hmm_K2
|   |   |   |- hmm_N
|   |   |   |- hmm_P1
|   |   |   |- hmm_P2
|   |   |   |- hmm_Q1
|   |   |   |- hmm_S
|   |   |   |- hmm_SH
|   |   |   |- hmm_TS1
|   |   |   |- hmm_TS2
|   |   |   |- hmm_Ui
|   |   |   |- hmm_Uu
|   |   |   |- hmm_X
|   |   |   |- hmm_a
|   |   |   |- hmm_al
|   |   |   |- hmm_b1
|   |   |   |- hmm_b2
|   |   |   |- hmm_by
|   |   |   |- hmm_cy
|   |   |   |- hmm_d1
|   |   |   |- hmm_d2
|   |   |   |- hmm_e
|   |   |   |- hmm_el
|   |   |   |- hmm_g1
|   |   |   |- hmm_gy1
|   |   |   |- hmm_hy
|   |   |   |- hmm_i
|   |   |   |- hmm_il
|   |   |   |- hmm_ky
|   |   |   |- hmm_m
|   |   |   |- hmm_my
```

```
|   |   |   |- hmm_n
|   |   |   |- hmm_ng
|   |   |   |- hmm_ngy
|   |   |   |- hmm_ny
|   |   |   |- hmm_o
|   |   |   |- hmm_ol
|   |   |   |- hmm_py
|   |   |   |- hmm_r
|   |   |   |- hmm_ry
|   |   |   |- hmm_sy
|   |   |   |- hmm_t1
|   |   |   |- hmm_t2
|   |   |   |- hmm_u
|   |   |   |- hmm_ul
|   |   |   |- hmm_w
|   |   |   |- hmm_y
|   |   |   |- hmm_z
|   |   |   |- hmm_zy
|   |   |- a.c                    HMM の状態遷移確率の計算
|   |   |- a.o
|   |   |- b.c                    HMM のシンボル出力確率の計算
|   |   |- b.o
|   |   |- bigram.c               単語の bigram の計算
|   |   |- bigram.h               bigram のバッファー
|   |   |- bigram.o
|   |   |- bigram_main.c          bigram による文音声認識アルゴリズムの本体
|   |   |- bigram_main.o
|   |   |- change_table.c         単語番号と計算上の累積認識番号のテーブル
|   |   |- change_table.o
|   |   |- continuous.h           認識に必要なバッファーのテーブル
|   |   |- init.c                 初期化
|   |   |- init.o
|   |   |- label                  音素ラベル表
|   |   |- main                   bigram の実行ファイル
|   |   |- main.c                 認識アルゴリズムの集合プログラム
|   |   |- main.o
|   |   |- makefile               メイクファイル
|   |   |- parameter.h            認識アルゴリズムの各種パラメータの設定ファイル
|   |   |- read_data.c            音声データ読み込み
|   |   |- read_data.o
|   |   |- read_dic.c             単語辞書データ読み込み
|   |   |- read_dic.o
|   |   |- read_hmm.c             音素データ読み込み
|   |   |- read_hmm.o
|   |   |- read_label.c           ラベルデータ読み込み
|   |   |- read_label.o
|   |   |- sort.c                 ソートプログラム
|   |   |- sort.h                 ソート用バッファ
|   |   |- sort.o
|   |- trigram (47)
|   |   |- a.c                    HMM の状態遷移確率の計算
|   |   |- a.o
|   |   |- b.c                    HMM のシンボル出力確率の計算
|   |   |- b.o
```

		|- buffer.h	認識に必要なバッファーのテーブル
		|- change_table.c	単語番号と計算上の累積認識番号のテーブル
		|- change_table.o	
		|- continuous.h	認識に必要なバッファーのテーブル
		|- init.c	初期化
		|- init.o	
		|- main	trigram の実行ファイル
		|- main.c	認識アルゴリズムの集合プログラム
		|- main.o	
		|- makefile	
		|- ngram.c	trigram の計算
		|- ngram.h	trigram のバッファー
		|- ngram.o	
		|- ngram_main.c	bigram による文音声認識アルゴリズムの本体
		|- ngram_main.o	
		|- parameter.h	認識アルゴリズムの各種パラメータの設定ファイル
		|- read_data.c	音声データ読み込み
		|- read_data.o	
		|- read_dic.c	単語辞書データ読み込み
		|- read_dic.o	
		|- read_hmm.c	音素データ読み込み
		|- read_hmm.o	
		|- read_label.c	ラベルデータ読み込み
		|- read_label.o	
		|- sort.c	ソートプログラム
		|- sort.h	ソート用バッファ
		|- sort.o	

第 6 章

最後に

trigram を bigram と読みかえることによって bigram を用いた文音声認識アルゴリズムが動きます。

参考文献

[1] 中川聖一：“確率モデルによる音声認識”, 第3章, 電子情報通信学会 (1988).

[2] X.D. Huang, Y. Ariki and M.A. Jack: “Hidden Markov Models for Speech Recognition”, Edinburgh University Press (1990).

[3] 村上, 嵯峨山: “単語の trigram を用いた連続音声認識の一アルゴリズム,” 音学講論, 2-Q-7, pp. 185-186 (1992-10).

[4] Jin'ichi Murakami and Shigeki Sagayama: “An Efficient Algorithm For Using Word Trigram Models For Continuous Speech Recognition,” Proc. SST92, pp. 330-335(1992-12).