TR-IT-0278

F_0 構造からのBI情報の自動抽出

吉村 康志　　北川 敏　　ニック・キャンベル

1998.9.25

現在、日本語音声に対するJToBIによる韻律の自動ラベリングを目指している。本研究では、JToBIラベルのうちbreak indexの自動付与を試みた。音声データベースの文音声のF_0構造に対して、F_0変動の閾値を設定してbreakを検出する方法と、多層パーセプトロンを利用した方法の2通りで実験を行った。

もくじ

第 1 章

はじめに

1.1 CHATR の音声合成方法

はじめに自然音声波形接続型任意音声合成システム CHATR[1] の音声合成方法について概略を述べる。CHATR はコーパスベースの音声合成システムであり、他の多くの音声合成方式とは違って信号処理による音声波形生成は行わず、用意された音素片を接続して合成を行う。

そのために、まず音声波形のデータベースを作成することになる。録音収集された音声波形は音素ごとに音素ラベル、及び韻律ラベルが付与されてデータベースに蓄積されていく。また、合成の際に最適な音素（以下 unit）を選択するため、重みの学習を行う。

音声を合成するには、まず入力された文章を解析して韻律の予測を行う。その後予測に従い、データベース中から unit を選択していく。選択は韻律ターゲットとの距離を表すターゲットコストと、接続歪みを表す接続コストが最小となるよう行われる。選ばれた unit の接続の際に信号処理は行わず、そのまま連続音として出力される。この方法の優れている点は、波形に一切の加工を行わないため、きわめて音質が良く、データベース話者の特徴をあらわせることである。また信号処理のための計算が不要なため、短い時間での合成が可能である。欠点としては、事前にラベル付けをしたデータベースを作成しなければならず、合成音声の品質はデータベースに大きく影響をうけることがあげられる。

1.2 不自然な韻律の原因

現在 CHATR の合成音声において、特に韻律の不自然さが問題となっている。韻律が不自然になってしまう原因としては大きくわけて次の 2 つが考えられる。

まず、作成された音声データベースに起因するものである。予測した target に対応する適当な unit がデータベース中に存在しない場合、target から大きく離れた unit が選ばれることがあり、結果的に出力音声の韻律が不自然なものとなってしまう。これは通常、データベースの規模が小さ

いほど、また接続コストを重視するほど起こりやすくなる。したがって録音する音声の量をふやしたり、またその読み上げる文章が音響的、韻律的にバランスのとれたものになるよう検討することで改善されるはずである。またターゲットコストと接続コストの重みを見直すことによっても効果が期待できる。

そしてもう一つの原因としては、CHATRの韻律予測自体が誤りを含んでいるということがある。特にアクセント句の係り受けを誤って認識した場合は文全体の韻律に大きな影響をあたえる。これを改良するには、さらに大きなデータベースからの韻律構造を学習する必要がある。

1.3 本研究の目的

以上のことからCHATRの韻律改善のためには音声データベースの充実が重要であると考えられる。現在韻律ラベルの付与は人が手作業で行っているが、これを自動化することにより大幅な効率向上が期待できる。

今回は特に韻律情報のなかでもアクセント句の係り受けに関係するbreakに注目し、データベース音声にbreak indexを付与することを目的とした。

第 2 章

付与する break index の種類

2.1 JToBI について

記述法は JToBI[2] に従っている。JToBI とは日本語の韻律の記述法であり、英語のための ToBI を元にしたものである。

JToBI では、次の 5 つの層に情報を記録する。

word 層	単語の境界を記述する。
tone 層	音声の抑揚を記述する。
BI(Break Indices) 層	連続する語の韻律の分離度を記述する。
finality 層	各イントネーション句の分離度を記述する。
miscellaneous 層	その他の層で記述できない現象を記述する (言い淀みや笑い等)。

今回記述するのは break 層のみとなる。

2.2 break index について

break index は連続する 2 つの語の韻律的分離度を示すものであり、次のように 0 から 3 までの 4 段階に分けられている。

0 連続した単語の分離がほとんど無いような場合。
(「これは」 ⇒ 「こりゃ」のような単語の融合を含む場合もある)
1 連続した単語の境界。
2 連続した単語に中程度の分離がある場合。アクセント句の境界となる。
3 連続した単語に強い分離がある場合。イントネーション句の境界となる。

アクセント句境界、イントネーション句境界は全体の韻律をきめる大きな要素となっているため、今回はこのうちの 2 と 3 のラベルを自動的に付与することを試みている。

図 2.1 にラベル付けの例 (「救急車が十分に動けず」) を示す。図は上から順に F_0、 break index、

音素ラベルを表す。この例では「救急車が」「十分に」の後にそれぞれ2のラベルが、「動けず」の後には3のラベルが付与されている。

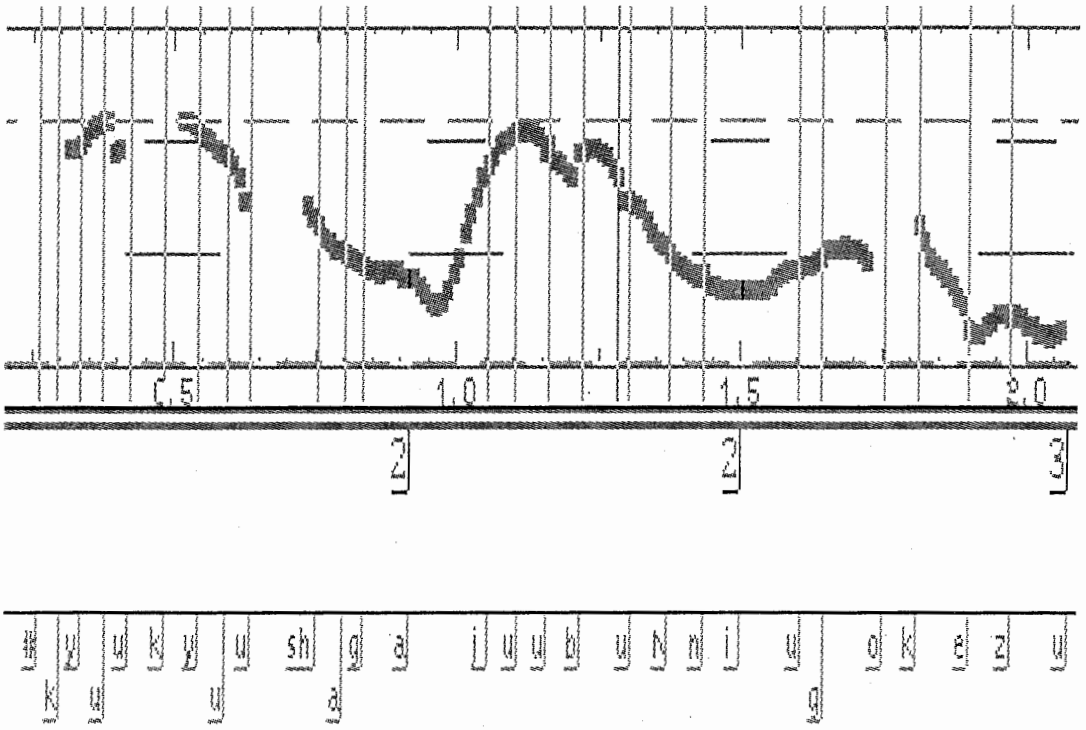

図 2.1: break index のラベリング例 (救急車が十分に動けず)

第 3 章

ラベル付与

3.1 使用データベース

今回使用したデータベースは話者 MHT による 503 文データベースである。このデータベースは A-J の 10 セットにわかれており、J セットを除いてそれぞれ 50 の文から構成されている。J セットのみ 53 の文があるが今回はそのうち 50 文のみを使用した。それぞれの文の音声にはすでに手作業で音素ラベルと韻律ラベルが付与されている。

3.2 BI ラベル付与

BI ラベル付与は各文の音声データから F_0 情報を取り出し、それを元に音素単位でラベルを付けていく。今回はラベル付与の際に語、文の区切り情報や、文法的な知識は使用しなかった。各音素位置にラベルを付けるかどうか、また、どのラベルを付けるかの識別は閾値を設定することによって行う方法と、多層パーセプトロンを利用する方法の 2 種類について試した。

第 4 章

閾値の設定による BI 識別

4.1 概要

はじめにデータベースからとりだしたデータの前処理を行う。その後 A セット 50 文を元に閾値を設定し、B-J セットの 450 文に実際にラベル付けを行い、その結果を評価する。

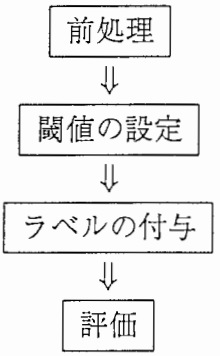

図 4.1: 処理のながれ

4.2 前処理

まず、データベース中のそれぞれの音声から F_0 を抽出する。F_0 の抽出には ESPS バージョン 5.1 の get_f0 を使用した。F_0 の値は 10 ミリ秒間隔で求めている。

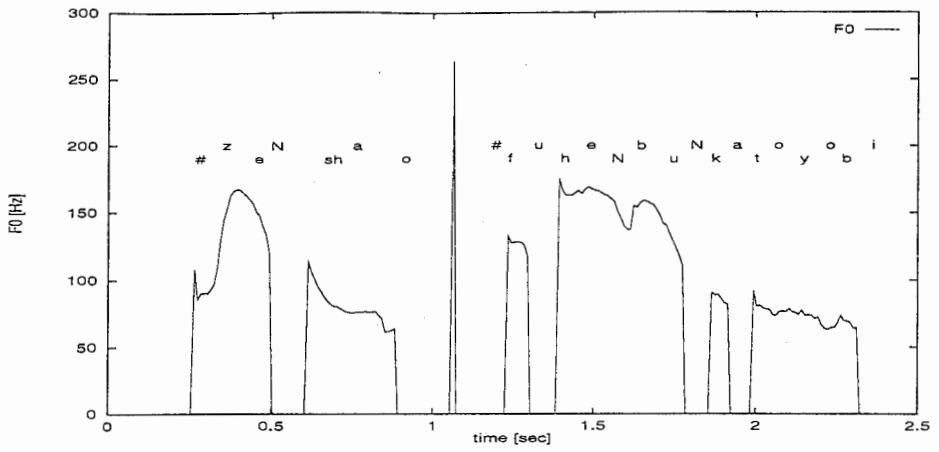

図 4.2: f0 抽出の例（「前者を普遍文化と呼び」）

図4.2でもわかるように、抽出された F_0 には明らかな失敗が含まれており、またいくつかの子音部ではその値が 0 になっている。そこでこの話者の F_0 が 200Hz を越えることはないと判断し、200Hz を越えるものを無効とした。また、子音部を F_0 を直線で補完した。
修正、補完をおこなった F_0 は、まだそのままでは break の抽出には冗長である。そこで大まか

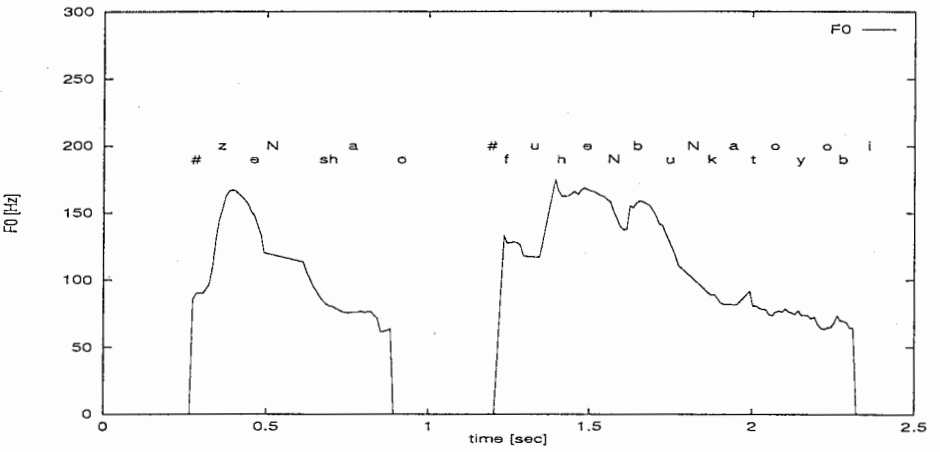

図 4.3: 修正、補完後の F_0 の例（「前者を普遍文化と呼び」）

な形をみるために各 F_0 をその次の F_0 値との平均で置き換え、これを 3 回繰り返すことでスムージングをかけた。その後、話者性を除くために全ての音声の平均 F_0 で割った。

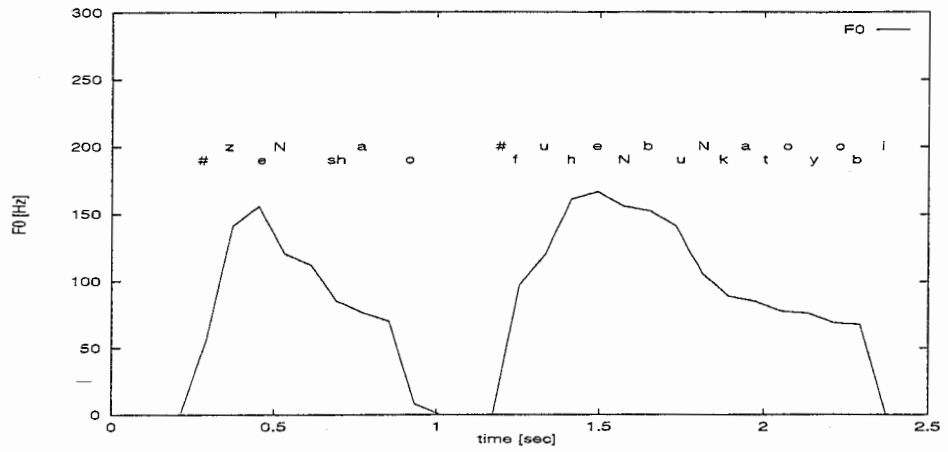

図 4.4: スムージングをかけた後の F_0 の例 — （「前者を普遍文化と呼び」）

4.3 break 抽出のアルゴリズム

はじめにスムージングをかけた F_0 の曲線から山になっている点と谷になっている点をさがしだす。その後、各点についてその次の点との F_0 の差をもとめていき、それが閾値を越えた時、その区間に break があると判断する。そして該当する区間のスムージングをかける前の F_0 を走査し、最も低い値があらわれる個所にラベルを付与する。

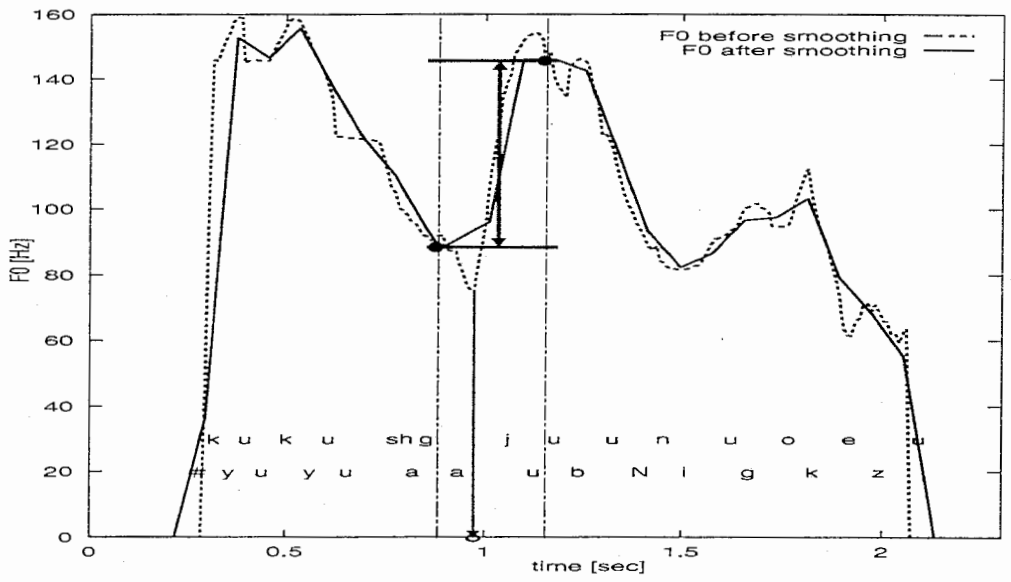

図 4.5: break 抽出の例（「救急車が十分に動けず」）

図 4.5(「救急車が十分に動けず」) の例では、'kyuukyuushaga'(救急車が) の'g' 付近にスムージング後の F_0 の谷がある。また'juubuNniugokezu'(十分に動けず) の最初の'u' 付近には F_0 の山となっ

ている部分がある。この2点のF_0の差が閾値を越えているとする。2点に挟まれた区間のスムージング前のF_0を走査すると、ちょうど'kyuukyuushaga'と'juubuNniugokezu'の間で最も低い値となっている。BIラベルはこの位置に付与される。

閾値の設定は次のようにして行う。まず3のラベルの閾値について

1. 仮の閾値の設定
2. 設定された閾値にしたがってAセットの50文にラベルを付与
3. 付与されたラベル数と実際のラベル数を比較
4. 閾値を更新
5. 2から繰り返し

という手順を付与されたラベル数と実際のラベル数の差が最小になるまでつづける。また2のラベルの閾値についても同様に行う。

4.4 評価の方法

自動ラベル付与の結果の評価は、データベースに人が付与したラベルと比較することによって行った。ただし今回はラベルが付与された位置から前後2音素以内の範囲に正しいラベルがつけられていれば正解とした。これは将来的に語、文、の区切りや文法的な知識を導入したときに十分補正できる範囲だと判断したためである。

ラベル付与の精度は次の式によって求めた。正しいラベルが過不足なく付与されたか評価している。

$$Accuracy = \frac{N - D - S - I}{N} * 100[\%]$$

ただし、

N: 本来のラベルの数の合計
D: 見落としたラベル数
S: 間違ってつけたラベル数
I: 余分につけたラベル数

4.5 結果

この方法でのラベル付与の評価結果は次のようになった。

本来のラベルの数の合計 N　2735
見落としたラベル数 D　265
間違ってつけたラベル数 S　683
余分につけたラベル数 I　257

$$\begin{aligned} Accuracy &= \frac{2735-265-683-257}{2735} * 100[\%] \\ &= 55.9[\%] \end{aligned}$$

第 5 章

多層パーセプトロンによる BI 識別

5.1 概要

はじめにデータベースからとりだしたデータの前処理を行い、学習、実験用データを用意する。その後 A セット 50 文を使用して学習を行い、B-J セットの 450 文にラベル付与を行い、その結果を評価する。

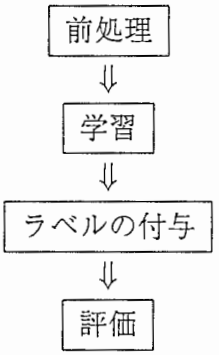

図 5.1: 処理のながれ

5.2 前処理

まず、前章と同様にデータベース中のそれぞれの音声から F_0 を抽出し、修正、補完をおこなう。

次に F_0 を各音素を中心とした 1.0 秒ごとに切り出し、それを 0.1 秒ごとの 10 のフレームに区切る。そして各フレーム中の F_0 を平均し、それをそのフレームの F_0 値とする。また、直前のフレームの F_0 値との差分である ΔF_0 を求める。

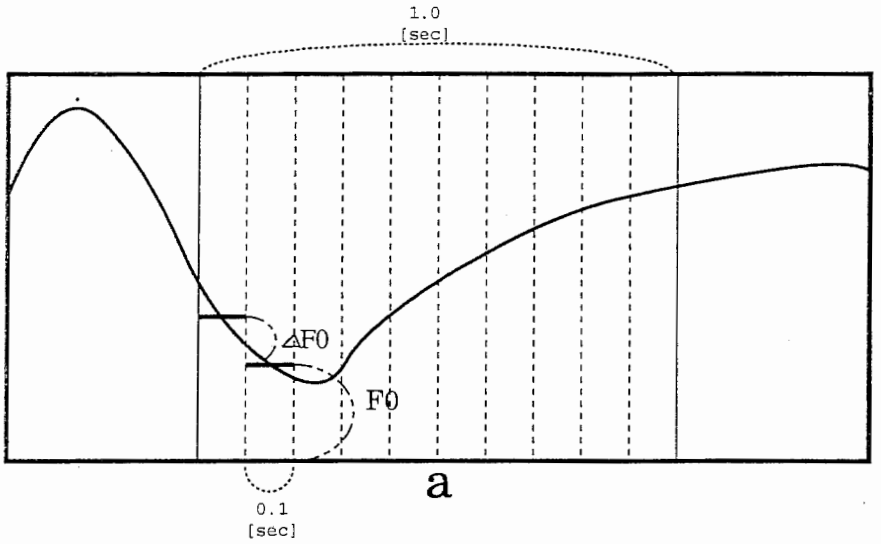

図 5.2: F_0、ΔF_0 の例

5.3 多層パーセプトロンの構成

次に学習、識別に使用する多層パーセプトロン [3] を構築する。ネットワークは入力層ユニット 20、中間層ユニット 30、出力層ユニット 2 の 3 層構造とした。出力ユニットはそれぞれ 2 と 3 のラベルに相当しており、その位置にラベルをつける場合に 1 を出力する。入力ユニットには各音素を中心とした 10 フレームの F_0 及び ΔF_0 が対応する。

教師信号として人が手作業で付与したラベルを出力ユニットに対応させたものを使用し、1 組の F_0、ΔF_0 を入力する度に出力誤差を計算し、バックプロパゲーション学習則 [4][5] に従ってユニット間の重みを変更する。これを A セットの全音素 2826 個について行い、それを 1 回と数える。学習回数は 3000 回とした。図 5.4に学習を繰り返すに従って誤差が減少していく様子を示す。横軸が学習回数、縦軸が誤差を表している。

学習終了後、実際にラベルを付与する際には、各出力ユニットが 0.5 以上の値を出力したとき対応するラベルを付与することとした。

5.4 結果

評価の基準は前章と同様とした。結果は次のようになった。

本来のラベルの数の合計 N　2735
見落としたラベルの数 D　325
間違ってつけたラベル数 S　823
余分につけたラベル数 I　611

$$\begin{aligned} Accuracy &= \frac{2735-325-823-611}{2735} * 100[\%] \\ &= 35.7[\%] \end{aligned}$$

閾値の設定による方法にくらべ悪い結果となっている。特に余分につけたラベル数は倍以上となった。

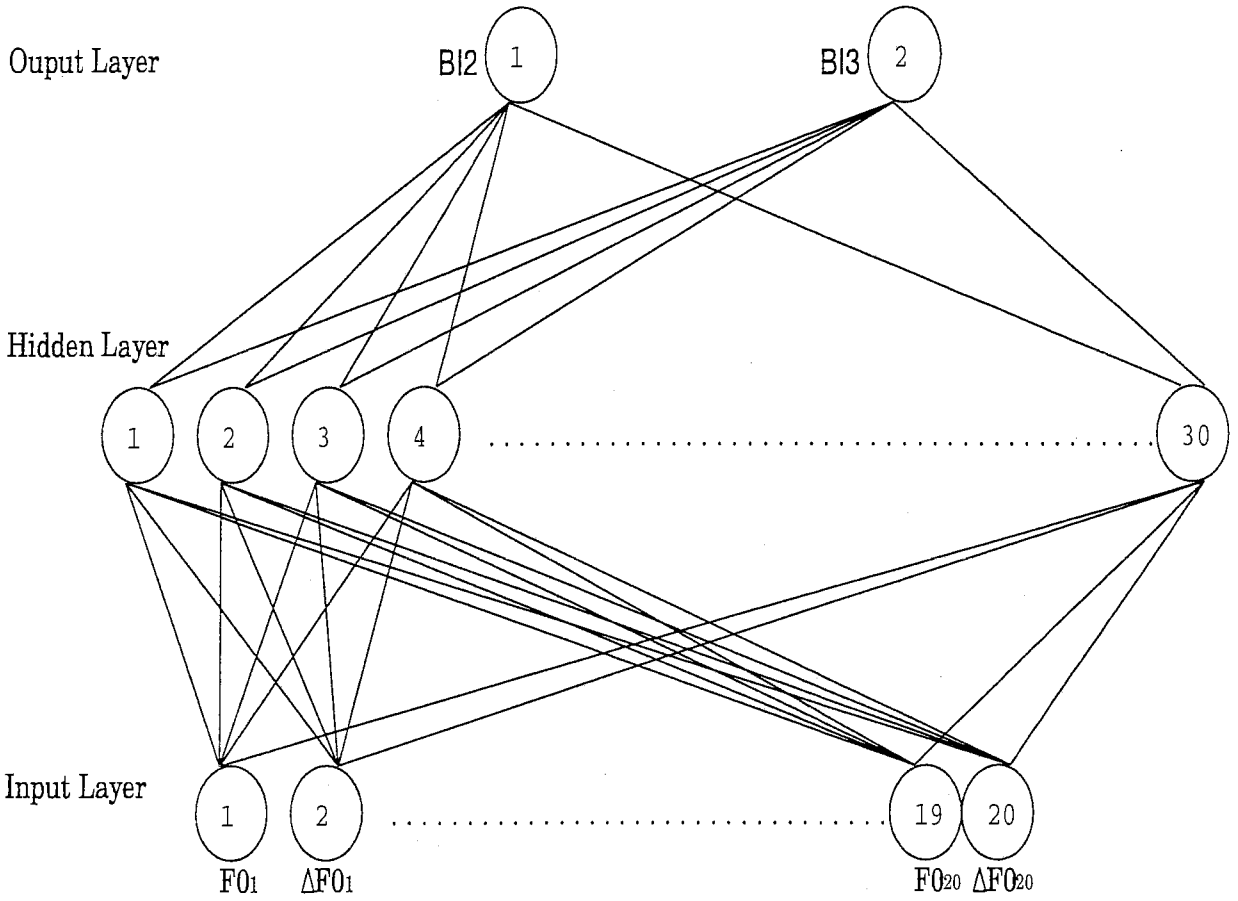

図 5.3: ネットワークの構造

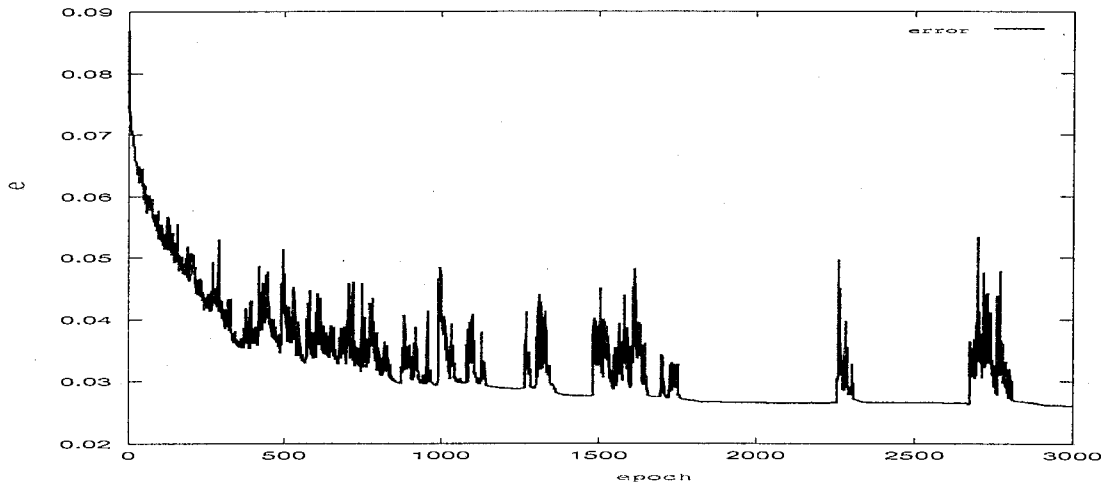

図 5.4: 学習進行の様子

第 6 章

考察

6.1 誤認識している箇所

2つの方法でラベル付与を行った結果について、誤認識の起こっている原因と考えられる事に次のようなものがあった。これらはどちらの方法にもあてはまる事である。

- F_0 の抽出失敗
- スムージングの失敗
- 後続の語による影響

6.1.1 F_0 の抽出失敗

F_0 抽出の際に、明らかに抽出の失敗であると思われる値については無視するようにしているが、修正しきれていない場合が見受けられた。このような場合、スムージング後にも影響が残り、本来付与されるべきでない箇所であるにもかかわらず、ラベルが付与されてしまう。（図 6.1「充実していく必要があろう」の末部参照）

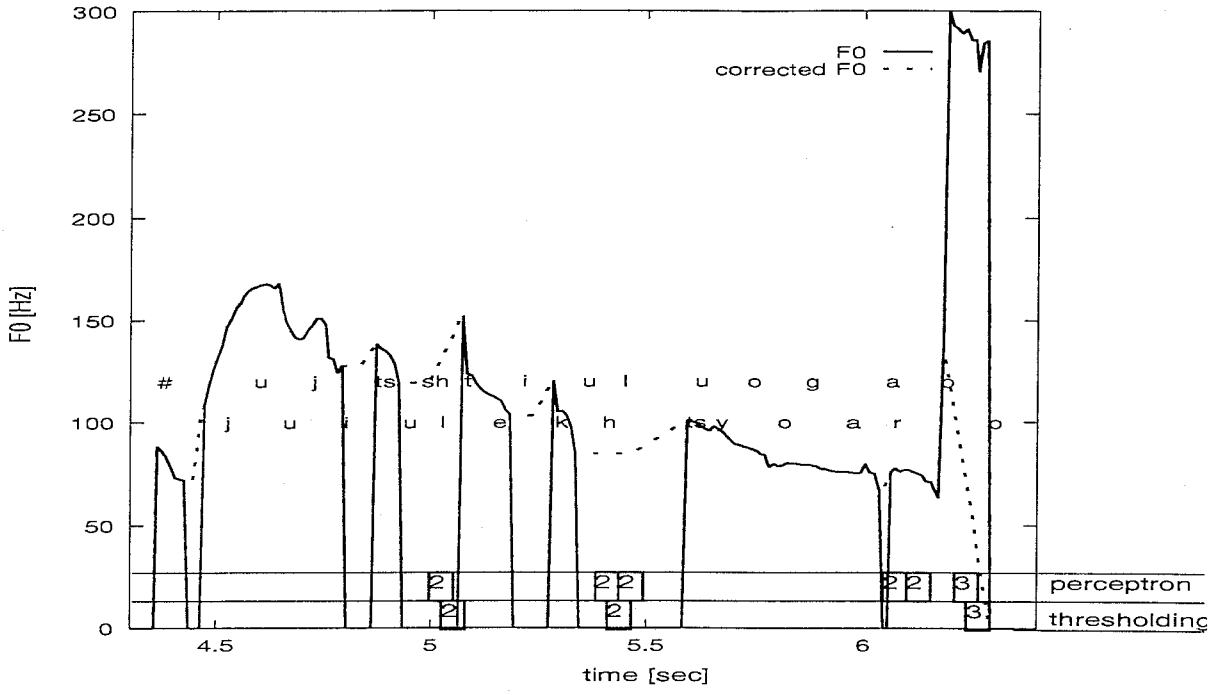

図 6.1: F_0 抽出失敗の例 — (「充実していく必要があろう」)

6.1.2 スムージングの失敗

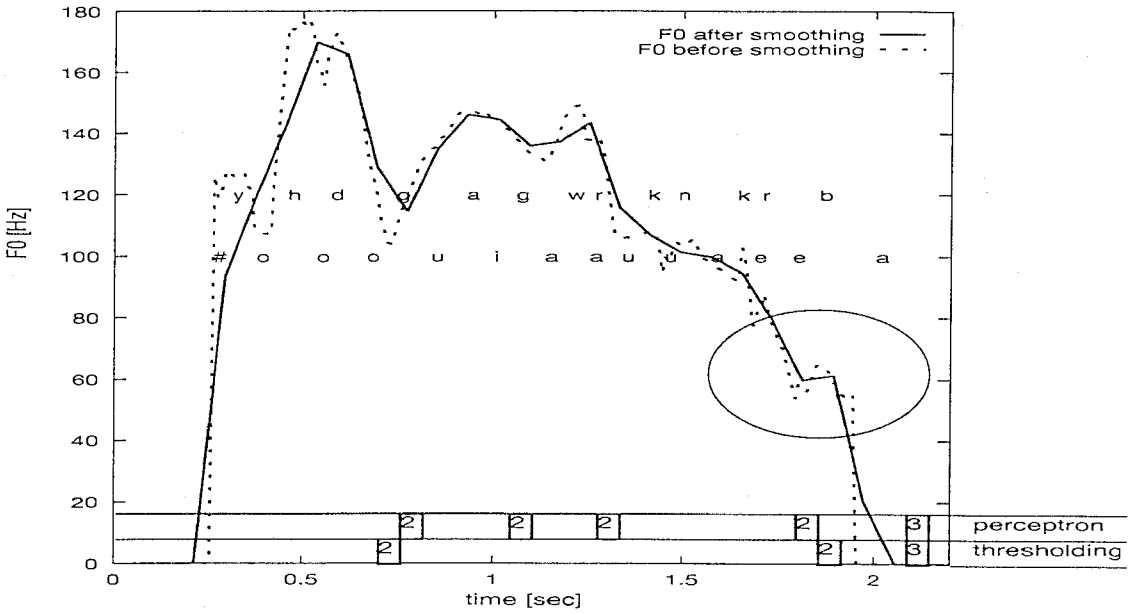

図 6.2: スムージング失敗の例 — (「よほど具合が悪くなければ」)

図6.2のように F_0 抽出はうまくいっているものの、スムージングが十分でなく、本来無視するべ

き F_0 の変動に影響を受け、F_0 が谷型となった所にラベルが付与されてしまう場合がある。

6.1.3 後続の語による影響

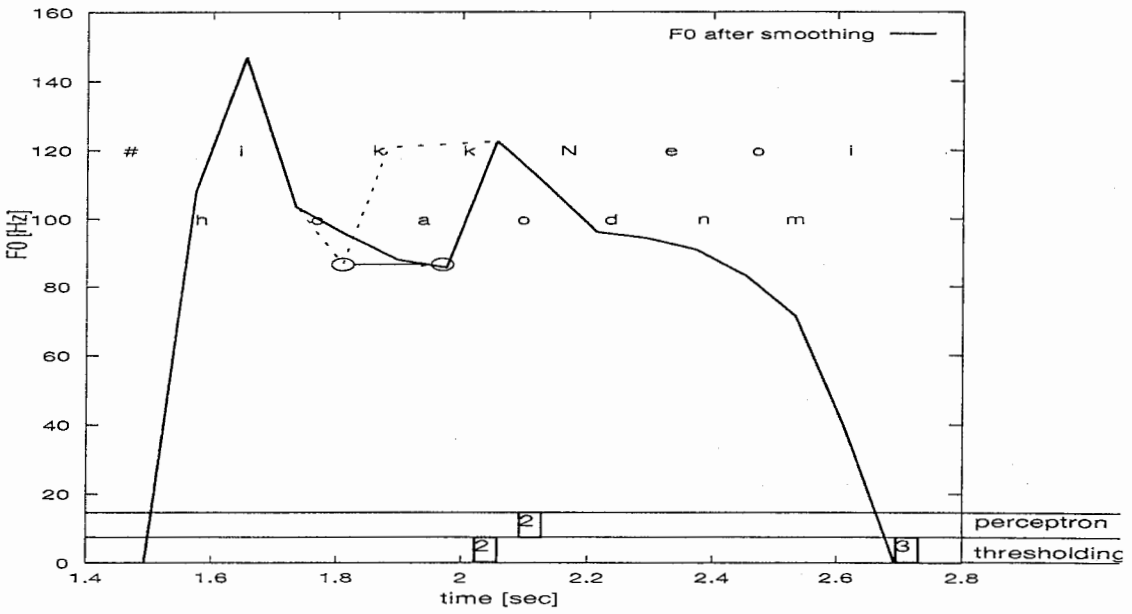

図 6.3: 後続の語の影響をうけている例（「火を囲んで飲み」）

F_0 の谷となる位置は語のつながりよって影響をうける。後続の語のアクセントが低い位置から始まる場合、ラベル位置は後ろの方へずれてしまうことになる。（図 6.3「火をかこんで飲み」の「かこ」の部分参照）

6.2 方法の違いによる評価結果への影響

評価の結果をみてみると、特に多層パーセプトロンを利用した方法の精度が悪い。この方法は、より様々な F_0 構造や F_0 の時間変化に柔軟に対応することを期待して導入したものである。今回このような結果となった原因としては、実験データの F_0 パターン数に対して、学習サンプルが不足していたことが考えられる。また、ネットワークの構成に問題があったとも考えられる。

第 7 章

まとめ

CHATR の韻律改善のためには、より一層の音声データベースの充実が重要である。本研究ではデータベース作成効率向上のため、 break index の自動付与を試みた。自動付与にあたっては閾値を設定する方法と、多層パーセプトロンを利用する方法の 2 種類を使用した。しかし評価の結果、精度は閾値を設定する方法では 55.9%、多層パーセプトロンを利用する方法では 35.7% と、どちらも満足のいくものではなかった。

今後の課題

- 語の区切りや文法知識の導入
  語の区切りや形態素などの文法情報を導入することによって、 F_0 を元に付与したラベルを修正し、より正確で実用的なラベル付与がおこなえると考えられる。

- F_0 の修正、補完、スムージングの方法の改善
  現状では F_0 の修正やスムージングがうまくいかない場合があり、結果に影響をあたえている。これらをより信頼のできるものに改善する必要がある。

- 自然会話文への応用
  日常会話のような、より柔軟性が求められるような場合への応用が考えられる。

- BI ラベル付与基準の明確化
  手作業でラベル付与をおこなう場合であっても、現在はラベル付与の基準があいまいであるため、同じ文音声であっても人によってラベルの付け方が違うことがある。何か明確な基準を設ける必要があると思われる。

謝辞

本研究に取り組むにあたり、御指導を頂いたNick Campbell第2研究室室長、北川 敏研究員に厚くお礼申し上げます。また初歩的な質問にも気軽に答えて下さいました第2研究室の皆様に感謝致します。最後に今回の実習の機会を与えてくださいました山本誠一ATR音声翻訳通信研究所社長に深く感謝いたします。

参考文献

[1] http://www.itl.atr.co.jp/chatr CHATR (Generic Speech Synthesis System). ATR Interpreting Telecommunications Research Labs 1997.

[2] Jennifer J.Venditti. Japanese ToBI Labeling Guidelines. Ohio State University,October 1995.

[3] 八名和夫, 鈴木義武. ニューロ情報処理技術 - 基礎と応用 -. 海文堂 1992

[4] Russell Beale,Tom Jackson 著, 八名和夫 監訳. ニューラルコンピューティング入門. 海文堂 1993

[5] 中野馨 監修, 飯沼一元 編, ニューロンネットグループ 桐谷滋 著. 入門と実習ニューロコンピュータ. 技術評論社 1989

付録 A

研究資料の格納場所

A.1　4、5 章共通のファイル

- 503 文のテキストファイル
  /home/as64/xyyoshi/503/*_set
- データベース音声から抽出した F_0 のファイル
  /home/as64/xyyoshi/Work/MHTc/NEW0916/*.f0l
- 修正、補完後の F_0 のファイル
  /home/as64/xyyoshi/Work/MHTc/NEW0916/*.l
- F_0 の平均を求める perl スクリプト
  /home/as64/xyyoshi/Work/MHTc/NEW0916/0918getf0mean.pl
- F_0 修正、補完 perl スクリプト
  /home/as64/xyyoshi/Work/prog/ezinterp.pl

A.2　4 章で使用したファイル

- スムージングをかけた F_0 のファイル
  /home/as64/xyyoshi/Work/MHTc/NEW0916/*.lb3
- 閾値設定 perl スクリプト
  /home/as64/xyyoshi/Work/MHTc/NEW0916/0916calib.pl
- ラベル付与 perl スクリプト
  /home/as64/xyyoshi/Work/MHTc/NEW0916/0917chunkig.pl

- ラベリング結果評価 perl スクリプト
  /home/as64/xyyoshi/Work/MHTc/NEW0916/0922score.pl
- ラベリング結果集計 perl スクリプト
  /home/as64/xyyoshi/Work/MHTc/NEW0916/0922score_all.pl

A.3 5章で使用したファイル

- 学習用 F_0 と教師信号のファイル
  /home/as64/xyyoshi/Work/MHTc/NEW0916/*.train
- 学習用データ作成 perl スクリプト
  /home/as64/xyyoshi/Work/MHTc/NEW0916/0920make_traindata.pl
- ラベリング結果評価 perl スクリプト
  /home/as64/xyyoshi/Work/MHTc/NEW0916/0922scoreBP.pl
- ラベリング結果集計 perl スクリプト
  /home/as64/xyyoshi/Work/MHTc/NEW0916/0922scoreBP_all.pl
- バックプロパゲーション学習 C++ プログラム
  /home/as64/xyyoshi/Work/MHTc/0922/learning.C
- ラベル付与 C++ プログラム
  /home/as64/xyyoshi/Work/MHTc/0922/labeling.C
- 学習、ラベル付与プログラムに必要な C++ ヘッダファイル
  /home/as64/xyyoshi/Work/MHTc/0922/bpnet.H
  /home/as64/xyyoshi/Work/MHTc/0922/dataIO.H
  /home/as64/xyyoshi/Work/MHTc/0922/local.H