確率的言語モデルによる自由発話認識に関する研究

鳥取大学　工学部　村上仁一

研究の背景および目的

文法的な言語モデル

ネットワーク文法

文脈自由文法

文脈依存文法

問題点：ルールの維持管理

（人間に大きな負荷が必要）

確率的な言語モデル

N-gram (bigram, trigram)

確率付きネットワーク文法

確率付き文脈自由文法

確率付き文脈依存文法

問題点：確率の付与方法

（基本的には大量の
テキストが必要）

最近の傾向

コンピュータの発達:

CPU、メモリー、DISKのコスト小

大量のテキストデータベース:

英語:Brown corpus, AP corpus

日本語:なし → 新聞記事のコンピュータ化

CD-ROM販売など

↓

確率的言語モデルの有効性の定量的な評価

研究の位置づけ

・確率的言語モデルの有効性の定量的な評価
  ・日本語のN-gram
  ・日本語の確率付ネットワーク文法
  ・音声認識への応用(自由発話への適用)
・自由発話の調査
  ・音響的な特徴
  ・言語的な特徴
  ・アクセントの情報量など

同類の研究(英語では古い)
  ・N-gramモデル　シャノンの情報量
  ・N-gramの音声認識への適用(IBM)
    1970年代　ただし発表は1980

年代後半
  ・N-gramの形態素解析への適用
    ATT Ken-church　1980年代後半

結論

２．連続音声認識システムに使用するアルゴリズム

- HMMの説明
- Baum-Welch アルゴリズム
- Vietbi アルゴリズム
- One-pass DP
- その他、高速化の手法

３．日本語のN-gramによるモデル化

- データ量に対するエントロピーの変化
- 新聞記事
- X線CTレポート
- ATRの対話データ

「学習データが増加した場合、全体に占める割合は少ないが、
たえず新しい種類の連鎖が出現する」

「言語モデルとしてのマルコフモデルの妥当
性:滅多に出現しない言語現象は、あえてモデルに適合させる必要がない

結論

４．N-gram を用いた音声認識

・かな、漢字、品詞のtrigramの有効性
（新聞記事・シミュレーション）

・X線CTにおけるbigramの有効性

・ATRの国際会議の申し込み文の入力におけるtrigramの有効性
「音声認識においてtrigramは有効」

５．自由発話の音声認識

・言語モデル(Perplexityの低い言語モデル）

・言い淀み、言い直しの対処
Garbage model 音素スキップ
「単語のtrigramは有効」
「音素スキップが有効」

結論

6. 自由発話音声における音響的・言語的な特徴

・音響的
　発話速度
　音素認識率
「自由発話は朗読発声とあまり差がない」

・言語的
　間投詞の出現率　　「文の約40%に出現」
　言い誤りの出現率　　「文の約10%に出現」

7. 音声におけるアクセント情報の持つ情報量の考察

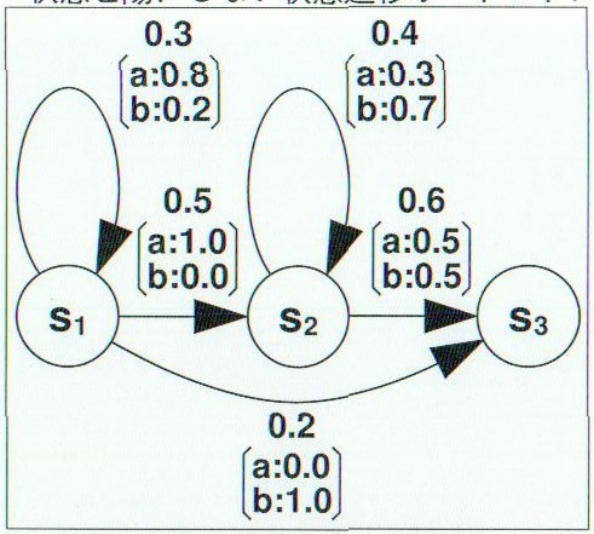
・漢字かな変換を用いた測定方法
「韻律は多くの情報量を持つ」

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

・複数話者が話されたときの話者認識
・Erogodic HMMと、その利用
「話者特徴量の抽出に長時間分析が有効」
「高い尤度を持つ初期モデルの選択が有効」

HMM　(Hidden Markov Model)

・状態を陽にしない状態遷移オートマトン



3 状態 left-to-right HMM

S1,S2,S3：状態
a,b: シンボル出力確率

S1, S2, S3 ：状態
a, b: シンボル出力確率

Baum-Welch学習

シンボル系列が与えらた時、尤度を最大にするように
パラメータ(初期状態確率、状態遷移確率、シンボル出力確率)
を学習

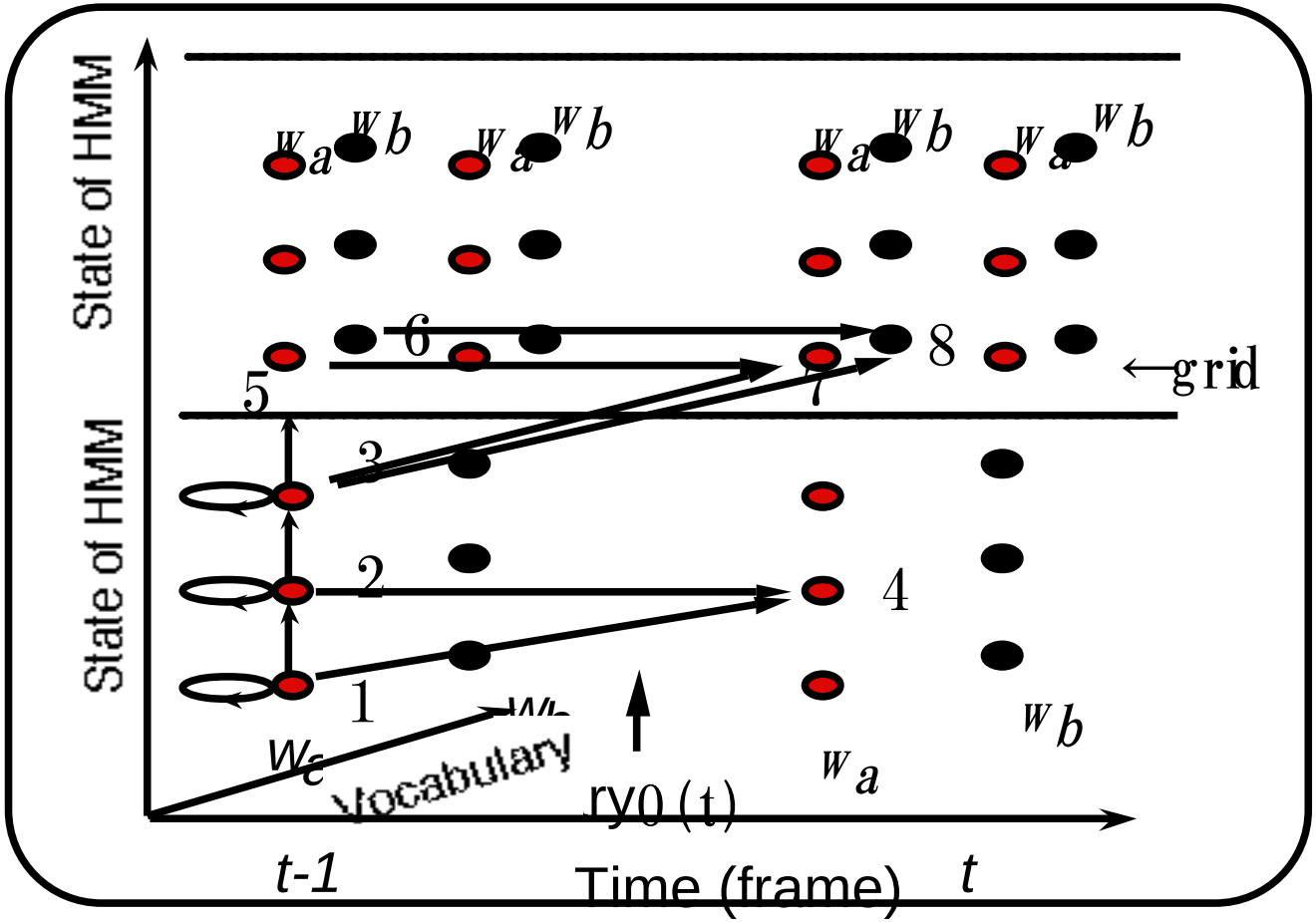
- Forwardアルゴリズム
  前から入力した時の尤度
- backwardアルゴリズム
  後ろから入力した時の尤度
- Forward-Backwardアルゴリズム

連続音声認識のアルゴリズム

（与えられたデータの尤度が最大になる状態系列を推定）

- Tree -- Trellis サーチ（フルサーチ）
- Viterbiサーチ(One Pass DP)
- （Level building）

Tree Trellis Search



Grid: score maximum likelihood from t=0 to t'

認識アルゴリズムの改良点

- N-bestサーチ
- 経路計算
- beam-search
- ビームの枝刈りの方法
- 近接したフレームにおける言語モデルの確率値の再計算
- trigramの値のindex方法(完全hash)
- log計算
- 認識単位(音素)
- 遅延言語処理

ビームの枝刈りの方法

1）尤度　　予め決めておいた値で計算を打ち切る

利点：計算速度　　欠点：不安定

2）幅　　一定の幅で計算を打ち切る

欠点：フレームごとのソーティングが必要

改良点）　　着目点：正確なビーム幅は不要
ビーム幅のスレッシュホールドをhistgramで
計算して枝刈り

→　計算量が通常のフルソーティングと比較して大幅に減少

改良点

- N-best tサーチ
- 経路計算
- Beam-search
  - ビームの枝刈りの方法
    尤度の値で枝刈り
    ビーム幅で枝刈り
  - 近接したフレームにおける言語モデルの値の再計算
- trigramの値のindex方法（完全hash）
- log計算
- 認識単位（音素）
- Look ahead処理（言語モデルを後で計算）

ビーム幅の決め方

1）尤度　予め決めておいた値で計算を打ち切る

利点：計算速度

欠点：不安定

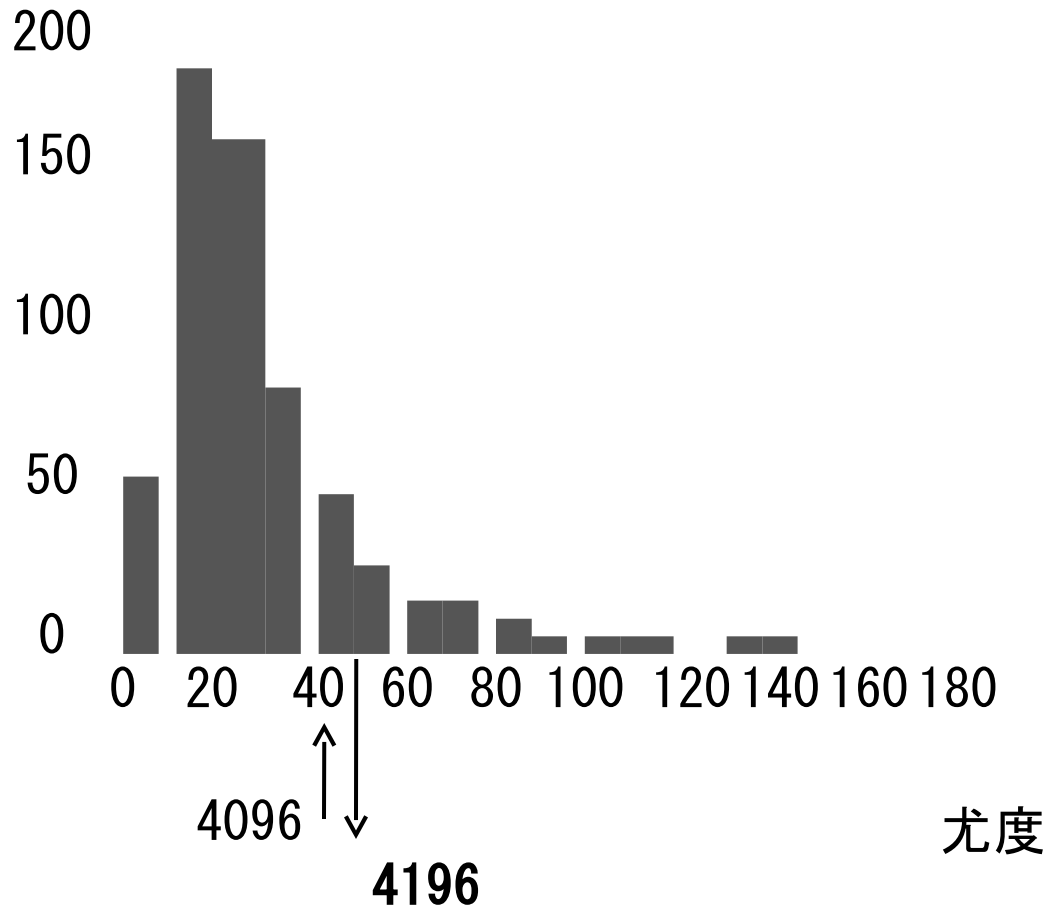
2）幅　一定の幅で計算を打ち切る

欠点：フレームごとのソーティングが必要

→ これが大きなオーバーヘッド？

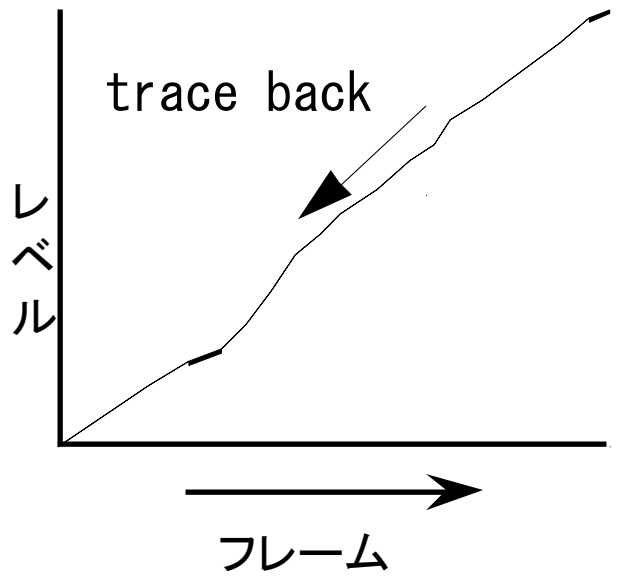
改良点：ビーム幅のスレッシュホールドを計算して枝刈り

$O(\log_2 N)$



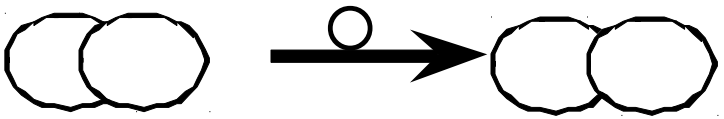
Histgramソート

Viterbiサーチの経路計算　（メモリー量の削減）



最大尤度の計算を終了後
トレースバック

従来法



尤度12.2
a b c d

尤度14.5
a c e d

X

尤度15.6
a b c d

最大尤度と
Viterbipath
を同時に次の
状態に渡す

各フレーム・各状態においてトレースバックなしで最尤の単語列がわかる

改良法

４．N-gramを用いた音声認識

- 新聞記事の入力における　かな、漢字、品詞のBigram, trigramの有効性(シュミレーション)
- X線CTにおけるbigramの有効性
- ATRの国際会議の申し込み文の入力におけるtrigramの有効性

評価関数

$$\Sigma \log(P(w)) + \alpha \Sigma \log(P(W_i \mid W_{i-2}\ W_{i-1}))$$

↑　　↑　　↑

音響尤度　結合値　　言語の連鎖確率

文(文節)音声認識

音響処理だけでは認識性能は低い

↓

言語処理による認識性能の改善

↓

単語のbigram,trigramが有効

↓

bigram,trigramは統計量

↓

どれだけの学習データ量が必要？

ここでの報告

a. HMMとtrigramを組み合わせた文音声認識実験

２　ポーズの処理　・ポーズのスキップ
・ポーズの学習

文音声認識の実験条件

基本アルゴリズム	Continuous 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	4-state3-loop 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(特定話者)	単語発声(5240単語)
(不特定話)	単語発声(12名736単語)
音素カテゴリ数	52音素
認識単語数	1567
ビーム幅	4096
言語情報	単語のtrigram
実験文数	261文
発声様式	朗読発話
発声内容	国際会議の申し込み(通称モデル会話)
trigramの計算に使用したデータ	ATRの対話データベース(国際会議の申し込み)
フロアリング	171978単語 exp(-1000.0)

ポーズ処理 1　　ポーズのスキップ

文中に存在するポーズ.

例　　住所は　pause　大阪市　pause

電話番号は　pause　３３９の　pause

対策

音響処理では，ポーズを認識

言語処理ではポーズを無視するように計算

例　東京都　港区　新橋　pause　１丁目

Ｐ（新橋 | 東京都 港区）Ｘ 1.0 Ｘ Ｐ（一丁目 | 港区 新橋）

ポーズ処理 2　　ポーズの学習

音声のテストデータの先頭のポーズを利用して

ポーズのHMMを学習

model		bigram		trigram	
		特定話者	不特定話者	特定話者	不特定話者
累積文認識率	1	42.5%	0.0%	66.7%	0.0%
	~2	47.9%	0.0%	72.4%	0.0%
	~8	51.3%	0.0%	75.1%	0.0%
word-correct		80.7%	55.8%	88.8%	74.2%
word-accuracy		63.0%	1.2%	81.1%	31.1%

認識実験の結果　文認識率（%）

model		bigram		trigram	
		特定話者	不特定話者	特定話者	不特定話者
累積文認識率	1	49.4%	31.4%	71.6%	61.7%
	~2	56.3%	41.0%	77.0%	72.0%
	~8	60.2%	44.4%	79.7%	76.6%
word-correct		81.32%	62.5%	89.4%	85.1%
word-accuracy		66.83%	43.0%	85.0%	77.9%

認識実験の結果（ポーズのスキップ）　文認識率（%）

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

認識実験の結果（ポーズのスキップ、ポーズ学習）　文認識率（%）

実験結果のまとめ

１）音素スキップ、garbageモデル、共に有効

２）認識性能　音素スキップ ＞ garbageモデル

３）認識性能　平滑化しない場合 ＞ 平滑化する場合

４）自由発話において　47.7%の文認識率
単語のtrigramの連鎖確率値の平滑化しない場合
音素スキップ

５）意味的に正しいとみなされる文を正解に含めた場合
1位文理解率で約75%、
8位までの累積文理解率は90%

まとめ

1　単語のtrigramをもちいた連続音声認識

2　ポーズの処理(ポーズのスキップおよび学習)
　朗読発話　文認識率　83.9%
　　(不特定話者認識, text-closed)

今後の研究:
　text-closed dataとtext-open dataの認識率の差を減少
　方法　　大量のテキストデータを収集

５　自由発話の音声認識

間投詞「あのー」「えーと」言い淀みや言い誤りおよび言い直し

１）言語モデル認識精度の高い音響モデルを作成することは困難？

→ perplexityの低い言語モデル（単語trigram ）

２）間投詞や言い直しの対応方法

A)　garbageモデル　　B)　音素スキップ

自由発話の音声データ

a. 朗読発話

テキストを読みあげた音声データ。
間投詞や言い淀み・言い直しなし。
言語モデルに対して text-closed データ

1 疑似自由発話

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

3 自由発話

話者はテキストを覚えて、その意図を理解してから自由に発話した音声データ。
間投詞や言い直しや未知語を含む。
言語モデルに対してtext-openのデータ？
（テキストを覚えて発話したデータであるため、
発話内容はtext-closed データに近い。）

garbageモデル（音響モデルによる対策）

認識アルゴリズム

garbageモデル=1単語

単語のtrigramの連鎖確率値

garbageモデルをスキップ

音素スキップ（言語モデルによる対策）

間投詞や言い直し　→　音素系列
言語モデル　→　音素系列をスキップ
（ペナルティ　→　音素のtrigram）

「”東京都“ ”港区“ ”新橋“ ”あのう(anou)“ “一丁目”」
「あのう」→間投詞

言語モデルの連鎖確率値

P(``新橋''| ``東京都'', ``港区'') x P(/a/|/sh/, /i/) x P (/n/|/i/,/a/) x
P(/o/|/a/,/n/) x P(/u/|/n/,/o/) x P(``1丁目'' | ``港区'' ``新橋'')

P(/a/|/sh/, /i/) ＝　ペナルティ、
P(”1丁目” | ”港区” ”新橋”) ＝音素スキップ

	累積文認識率	base-line	garbage	skip-phone
朗読発話	1	89.7%	83.2%	88.5%
	~2	97.3%	90.5%	96.2%
	~8	100.0%	97.3%	99.2%
	Word-Correct	97.5%	93.4%	96.4%
	Word-Accuracy	96.9%	93.2%	96.0%
疑似自由発話	1	41.6%	64.5%	73.3%
	~2	43.1%	70.2%	79.0%
	~8	44.3%	78.2%	82.8%
	Word-Correct	70.6%	81.5%	89.5%
	Word-Accuracy	34.2%	76.6%	82.3%
自由発話	1	10.7%	37.8%	47.7%
	~2	15.3%	46.9%	57.2%
	~8	19.5%	56.1%	66.8%
	Word-Correct	44.7%	65.7%	80.9%
	Word-Accuracy	9.1%	58.9%	73.3%

自由発話の文認識実験結果

考察　　自由発話の音声認識

全ての音素を完全に認識する必要性なし

意味的に合っている文章を出力自由発話の認識のための言語モデル、

非文を生成しないこと、

perplexityが低いことモデルがカバーできない範囲

→　　garbageモデルや音素スキップ

考察

実験結果

音素モデルのスキップ　＞　garbageモデル

広いビーム幅が必要語彙数が多い場合や
ビーム幅が小さい場合

garbageモデルの　＜　音素モデルのスキップ
となる可能性

考察

自由発話音声の認識とTEXT-open data

自由発話：もしTEXT-closed dataならば、ある程度認識可能

現実:　　TEXT-open dataになる。

今後の研究：

TEXT-closed dataとTEXT-open dataの認識率の差を減少

方法　　大量のテキストデータを収集？

テストデータへの動的な適用？

単語間の距離の測定？

まとめ

1　連続音声認識アルゴリズム
　Word trigram + Viterbiアルゴリズム
　間投詞、言い直しの対策

・garbage model
・音素モデルによるスキップ

1　自由発話の認識実験
　ビーム幅16384語彙数435において47.7%の文認識率
　意味的に正しい文を正解に含めた場合
　1位文理解率で約75%、