BULLETIN OF INSTITUTE OF TECHNOLOGY AND SCIENCE THE UNIVERSITY OF TOKUSHIMA

サフィックス・アレイに基づく言語モデルを用いた音声認識に関する研究

柘植 覚[1]，獅々堀 正幹[1]，北 研二[2]

Study of Speech Recognition using Suffix Array

by
Satoru TSUGE, Masami SHISHIBORI, Kenji KITA

For obtaining high speech recognition performance, we need high quality acoustic model and language model of speech recognition. In this study, we focus on the language model. The conventional language models, which are CFG, *N*-gram model, and so on, have some problems which are outputted the non-language characters and words sequence. Therefore, in this paper, we proposed the language model which was used the suffix array for speech recognition. The suffix array was proposed for the information retrieval. The advantages of the suffix array were that "予測可能" "無駄な仮説が生成されない" For evaluating the proposed language model, we conducted the similarly music information retrieval experiment using MIDI database. The experimental results showed that the proposed method was useful for the music information retrieval.

Key words: Suffix Array, Speech Recognition, Language Model

1　はじめに

近年，音声認識は音声による自動応答やカーナビゲーションなどに使用されつある．しかし，まだ現状では十分な認識精度が得られず，広く一般で使用されているとは言い難い．現在の音声認識システムを高精度かつ効率的なするためには，音声認識に用いられる音響モデルや言語モデルなどの高精度化が必要となる．特に言語モデルは，音声認識の過程において，探索すべき仮説 (認識結果) の探索空間を効率的に規定するという役割を担っており，高精度な音声認識を実現するために重要なモデルと言える．そのため，従来より，構文情報に基づくモデル (例：有限オートマトン，文脈自由文法など) や統計情報に基づくモデル (例：bi-gram, tri-gram 等のマルコフモデルなど) が盛んに研究されている．しかし，従来用いられているこれらの言語モデルは，仮説の過剰生成の問題があり，認識結果としてしばしば非言語的な文字列や単語列を出力するという欠点がある．

[1] 徳島大学大学院ソシオテクノサイエンス研究部
情報ソリューション部門
[2] 徳島大学高度情報化基盤センター

そこで，本研究では，音声認識のためのより強力な言語モデルとして，サフィックス・アレイ (suffix array) と呼ばれるデータ構造に基づく言語モデルと認識仮説の探索手法を提案する．サフィックス・アレイは，元来，情報検索のために考案されたものであり，与えられた任意の文字列を高速に検索するためのデータ構造であるが，本研究では，サフィックス・アレイを拡張することにより，これを単なる検索のためのモデルとしてではなく，音声認識時の部分的な認識結果から後続する音素/文字/単語を予測/生成するためのモデルとしても用いる．このサフィックス・アレイに基づく言語モデルを音声認識対象となるコーパスや文書集合から構成することにより，対象分野に属する文 (あるいは部分文) だけを生成することのできる言語モデルを得ることができる．この特徴は，従来のマルコフモデル系統の言語モデルにはない大きな特徴である．この言語モデルは，元のコーパスや文書集合に属さない文は生成されないので，自然発話 (spontaneous speech) のような認識には適さない．しかし，制限された文だけを高精度に認識したい

という音声認識の利用分野おいては，きわめて適しているといえる．

さらに，サフィックス・アレイに基づく言語モデルでは，長範囲に渡る予測が可能である．場合によっては，1 つの単語を認識した部分結果から，残りの単語すべてを予測することさえもできる．このような長期的な予測能力を利用することにより，部分的な認識仮説を自動補完する音声コンプリーション機能や，雑音等の影響で認識不可能な部分を推測するということも可能である．この特徴を持つ本言語モデルを用いることにより，情報検索分野において音声認識を用いる場合の 2 段階の処理 (音声認識を行い，次に認識した結果を用いて検索する) 処理が必要でなくなり，音声認識が終了した時点で同時に情報検索の結果が得られるという，非常に効率的な処理を行うことが可能となる．さらに，サフィックス・アレイは，部分認識仮説に後続する音素/文字/単語等の予測ばかりでなく，認識仮説の前にくる音素/文字/単語等をも予測することが可能である．これを利用して，前向き方向の探索と後向き方向の両方向による前向き・後向き探索に基づく音声認識方式の言語モデルに適している．

本稿では，この言語モデルを音声認識ではなく，MIDI を用いた音楽検索に用いることにより，本モデルの有効性を検証する．音楽検索では，音声認識における音響モデルが担う部分を音階認識として考え，言語モデルが担う探索空間の絞りこみなどの言語モデルが担う部分に着目ができるとして，本稿では本モデルを音楽検索により検証を行う．

以下，本稿では，**2** でサフィックス・アレイの簡単な説明を行い，**3** では，本稿で提案する拡張サフィックス・アレイを提案し，それを類似音楽検索に用いることを説明する．**4** では，提案手法を MIDI 音楽データベースを用い，有効性を検証し，最後に **5** において，本稿のまとめと今後の課題を述べる．

2 サフィックス・アレイ

本節では，サフィックス・アレイについて述べる．サフィックス・アレイとは，高速な文字列検索を可能にするデータ構造であり，以下のような特徴を持つ．

- 索引のコンパクト性
  インデキシングをポインタで表現するために索引がコンパクトになる．
- 字彙のサイズに依存しない計算効率
  サフィックス・アレイは，索引の構築および文字列照合時の記憶および時間効率がアルファベットサイズに依存しない．
- 索引の更新のコストが大きい
- 索引構築の計算コストが大きい

サフィックス・アレイは，テキスト中の文字位置を要素とする 1 次元配列であり，それら文字位置は対応するサフィックスが辞書式順になるように並んでいる．よってサフィックス・アレイの構築はサフィックスの辞書順ソートに相当する．

サフィックス・アレイの構築は，その提案者である Manber と Myers による方法 [1] の他に sadakane によって，より高速な構築法が提案されている [2]．これら 2 つの構築法は，テキスト長を $N(N < 2^{32})$ として $8N$ バイトの内部記憶量を要するが，最悪時の計算効率は $O(N \log N)$ となる．

一方，radix sort などの文字列ソート法を用いてサフィックス・アレイを構築することも可能である．テキストが 1 バイトの文字列であれば，構築に $5N$ バイトの内部記憶量を要する点で上記の構築法よりもコンパクトである．しかしその最悪時の計算効率は $O(N^2)$ となる．

以下，サフィックス・アレイの構築法，検索法について説明する．

2.1 構築法

サフィックス・アレイはテキスト中の文字位置を要素とする 1 次元配列であり，それらの文字位置は，対応するサフィックスが辞書順序になるように並んでいる．よって，サフィックス・アレイの構築はサフィックスの辞書順ソートに相当する．

長さ N の文字列を $a_0, a_1, ..., a_{N-1}$ で表す．ここで a_i は，記号の有限集合 Σ の要素であり文字と呼ぶ．$|\Sigma|$ により記号の総数を表す．各文字には非負の文字値が定義されており，この文字値に基づいて文字列間にいわゆる辞書順 $<, =, >$ が定義されている．テキスト $T = a_0, a_1, ..., a_{N-1}$ に対し文字列 $S_i = a_i, a_{i+1}, ..., a_{N-1}$ をテキスト T の先頭から i 番目の文字位置から始まるサ

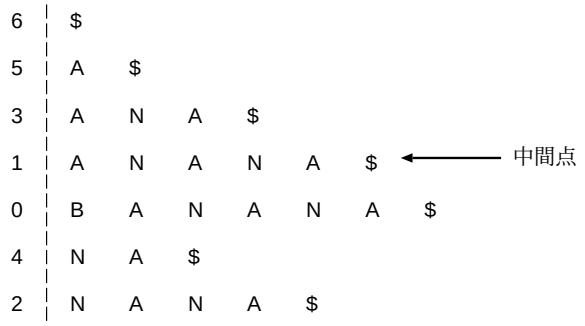

図 1: 中間点

図 2: 絞り込まれた範囲

フィックスと呼ぶ．サフィックス・アレイは全てのサフィックスを辞書順に並べて得られる長さ N のポインタ列 $A = p_0, p_1, ..., p_{N-1}$ である．すなわちサフィックス間の辞書順は $S_{p_0} < S_{p_1} < ... < S_{p_{N-1}}$ となる．

任意のサフィックス間の辞書順を確定するために Σ に属さない文字 (“$”) をテキストの末尾に加える．“$” には文字値の最小値 0 を与える．また，文字列およびポインタ列を表現するデータ構造として配列を用いる．配列 X の添字 i で指定する配列要素を $X[i]$ で表し，添字 i から $j (i \leq j)$ の範囲に対応する X の部分を $X[i, j]$ で表す．サフィックス・アレイに関する詳細は文献 [3] などを参考にされたい．

2.2 検索法

サフィックス・アレイでの検索は二分探索を用いて検索を行う．以下に構築法の説明で構築したサフィックス・アレイを用いて検索法について説明する．

図 1 に示すように，検索文字列 $T =$ “NA” としたとき，まずサフィックス・アレイの中間点をとる．図 2 参照に示すように，中間点のサフィックス “ANANA” と “NA” を比較すると “ANANA”＜“NA” となることから図 1 の中間点よりも上の部分には検索文字列 T は含まれず下の部分に検索文字列 T が含まれることがわかる．次に，図 3 に示すように，比較を行った結果絞り込まれた検索文字列 T が含まれる範囲から中間点を求める．そして求めた中間点のサフィックス “NA” と検索文字列 T の比較を行い同値となるので検索文字列 T が検索され検索終了となる．

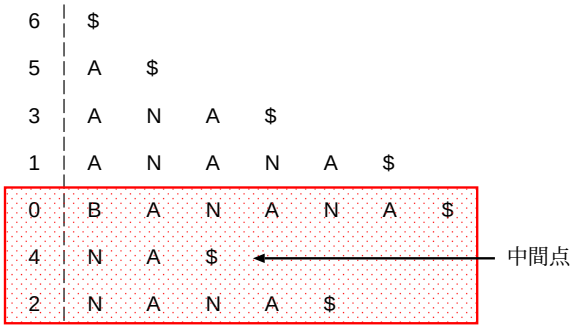

図 3: 中間点

3 拡張サフィックス・アレイ

前節において，サフィックス・アレイについて述べた．本節では，音声認識などの様々な分野に応用できるようにサフィックス・アレイを拡張する手法を提案する．従来のサフィックス・アレイを用いた検索は，完全一致で行うために類似したものを検索することができない．そのため，音声認識に用いる言語モデルとしては不十分である．そこで，サフィックス・アレイを拡張し，曖昧性を持つ入力に対しても検索が可能であるように拡張をする．本節では，この拡張サフィックス・アレイを類似検索を例にし，説明を行う．

3.1 構築法

サフィックス・アレイの構築は以下の手順で行う．

1. 与えられた特徴量を用いサフィックスを作成する
2. サフィックスをソート

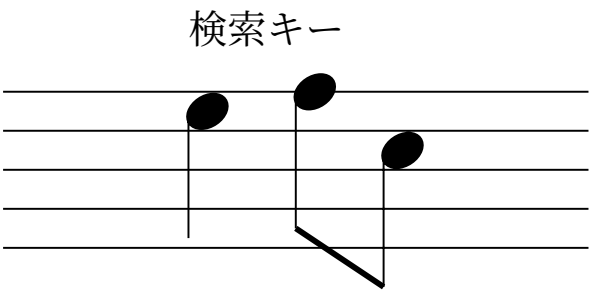

図 4: 検索キー

音高推移
(0, 1, -4)

音長
(1, 0.5, 0.5)

図 5: 検索キーの特徴量

本提案手法は，類似音楽検索により評価をするため，本節では，例として MIDI データを用いて説明をする．MIDI データより，音高，音長，音量などを特徴量として使用するためテキストデータとして抽出する．類似音楽検索を行うため，音高からさらに特徴量として各々の音高の推移を特徴量として抽出する．各特徴量ごと (音高推移，音長) にサフィックスを作成し，ソートし，各特徴量ごとのサフィックス・アレイを作成する．

3.2 検索法

検索は以下の手順で行う．

1. 検索キーから特徴量抽出
2. 特徴量ごとに範囲の絞り込みを行う
3. 特徴量ごとに求まった範囲で重複する部分を最終検索結果として出力する

作成したサフィックス・アレイと図 4 のような検索キーで検索を行う場合を説明する．まず，サフィックス・アレイ構築と同じように検索キーから特徴量を抽出する (図 5)．そして特徴量ごとに範囲の絞り込みを行う．しかし，サフィックス・アレイではマージンをとった検索を行うことができないためにマージンを取りながら検索する方法について以下に説明する．

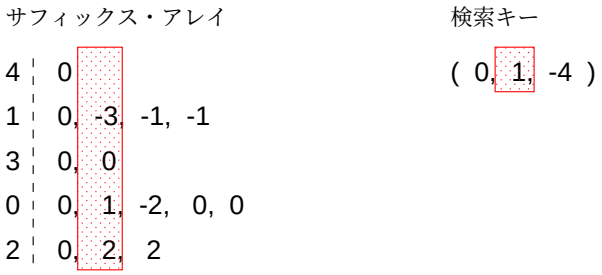

図 6: 比較部分

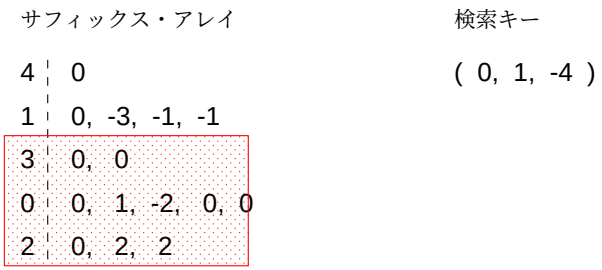

図 7: 絞り込んだ範囲

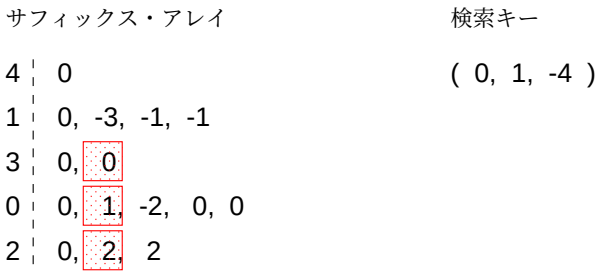

図 8: クラス分け

構築したサフィックス・アレイを $S_0, S_1, ..., S_4$，検索キーを N_0, N_1, N_2 とし，1 つの特徴量音高推移を用いて実際の検索手順の以下に述べる．

特徴量音高推移の場合は最初の音符は基準音になるために全てのサフィックス，検索キーにおいて 0 になるために 2 個目の音符から絞り込みを行う．図 6 の枠で囲った $S_1[1], S_2[1], ..., S_4[1]$ と N_1 の音符のみを用いて範囲の絞込みを行う．マージンを ± 2 とするとこの場合図 7 の枠で囲んだ範囲に絞り込める．

次に絞り込んだ範囲の中で $S_i[1]$ が同じ値の部分を 1 つのクラスとしてクラス分けを行う (図 8)．そして，クラス毎に N_2 を用いて範囲の絞込みを行い図 9 の囲んだ部分まで範囲が絞り込まれる．ここで検索キーが終了し

サフィックス・アレイ		検索キー
4	0	(0, 1, -4)
1	0, -3, -1, -1	
3	0, 0	
0	0, 1, -2, 0, 0	
2	0, 2, 2	

図 9: 特徴量音高推移での検索結果

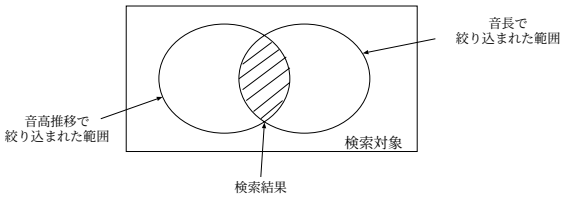

図 10: 最終検索結果範囲

ているので図 9 で囲んだ部分が特徴量音高推移での検索結果となる．またクラス分けを行う時点で検索キーとの差をそれぞれのクラスに持たせることで尤度を計算させ，検索結果が複数検出された場合順位をつけることができる．

上記の手順を特徴量ごとのサフィックス・アレイで行い，それぞれで検索範囲を絞り込む．そして，図 10 のように絞り込んだ範囲の重なる部分を最終の検索結果として出力する．

4 類似音楽検索実験

4.1 類似検索実験

提案した拡張サフィックス・アレイの有効性を検証するため，類似音楽検索実験を行った．入力される検索キーにゆらぎがある場合には，従来のサフィックス・アレイでは検索ができない．そこで，提案した拡張サフィックス・アレイにより入力にゆらぎを持つ場合の検索に関し，検証を行った．本実験では，類似検索において絞り込みに必要な音符数，検索速度を調べた．

4.1.1 実験条件

- MIDI データベース
  J-POP，演歌，童謡などのジャンルを含む 483 曲
  主旋律のみを抽出したもの
- 特徴量
  - 音高推移
  - 音長
- 検索キー
  MIDI データベースの中から 20 音符で切り出したデータをそのまま検索キーに使用.
- 閾値
  音高推移 ±3
  音長 ±8 分音符
- 検索キー
  MIDI データベースの中から 20 音符で切り出したデータを以下の条件で編集を行った 8 個のデータ.

  1. 切り出したデータ
  2. 音高を閾値内で変化　音長は変更なし
  3. 音長を閾値内で変化　音高は変更なし
  4. 音高，音長を閾値内で変化
  5. 14 音符目で閾値より大きく音高を変化
  6. 13 音符目を閾値より大きく音長を変化
  7. 前半部分の音長音高を閾値内で変化
     後半部分は変更なし
  8. 前半部分変更なし
     後半部分を音長音高を閾値内で変化

  検索キー 2〜4 は小さなズレを想定し検索キー 5, 6 は大きなズレがあったときを想定，そして検索キー 7，8 は検索キーの前半部分と後半部分でズレがあった場合ズレの位置が検索結果に影響するかどうかを調べるための検索キーである．

4.1.2 実験結果

表 1〜4 に各条件における実験結果を示す．この結果より，検出件数から十分に絞り込むためにはどの入力キーも長さが 9 音符〜10 音符と完全一致検索に比べて

表 1: 実験 2 検出件数と正解率

入力キーの長さ	条件 1			条件 2		
	検出件数	正解数	正解率(%)	検出件数	正解数	正解(%)
5	16586	50	0.30	5182	13	0.25
6	11082	44	0.40	3273	10	0.31
7	7041	22	0.31	1771	7	0.40
8	76	4	5.26	30	4	13.33
9	13	4	30.77	15	4	26.67
10	12	4	33.33	14	4	28.57
11	12	4	33.33	10	4	40.00
12	4	4	100.00	5	4	80.00
13	4	4	100.00	5	4	80.00
14	4	4	100.00	4	4	100.00
15	4	4	100.00	4	4	100.00
16	4	4	100.00	4	4	100.00
17	4	4	100.00	4	4	100.00
18	4	4	100.00	4	4	100.00
19	4	4	100.00	4	4	100.00
20	4	4	100.00	4	4	100.00

表 2: 実験 2 検出件数と正解率

入力キーの長さ	条件 3			条件 4		
	検出件数	正解数	正解率(%)	検出件数	正解数	正解(%)
5	19250	50	0.26	8886	19	0.21
6	12520	44	0.35	5428	13	0.24
7	7967	22	0.28	2979	7	0.23
8	95	4	4.21	21	4	19.05
9	17	4	23.53	6	4	66.67
10	16	4	25.00	6	4	66.67
11	12	4	33.33	6	4	66.67
12	7	4	57.14	4	4	100.00
13	7	4	57.14	4	4	100.00
14	4	4	100.00	4	4	100.00
15	4	4	100.00	4	4	100.00
16	4	4	100.00	4	4	100.00
17	4	4	100.00	4	4	100.00
18	4	4	100.00	4	4	100.00
19	4	4	100.00	4	4	100.00
20	4	4	100.00	4	4	100.00

表 3: 実験 2 検出件数と正解率

入力キーの長さ	条件 5			条件 6		
	検出件数	正解数	正解率(%)	検出件数	正解数	正解率(%)
5	7799	24	0.31	18142	50	0.28
6	4641	14	0.30	13135	47	0.36
7	2818	11	0.39	8186	22	0.27
8	42	4	9.52	83	4	4.82
9	9	4	44.44	13	4	30.77
10	8	4	50.00	12	4	33.33
11	8	4	50.00	12	4	33.33
12	5	4	80.00	4	4	100.00
13	4	4	100.00	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0

表 4: 実験 2 検出件数と正解率

入力キーの長さ	条件 7			条件 8		
	検出件数	正解数	正解率(%)	検出件数	正解数	正解率(%)
5	9381	28	0.30	16586	50	0.30
6	5791	22	0.38	11082	44	0.40
7	3543	15	0.42	7041	22	0.31
8	47	4	8.51	76	4	5.26
9	10	4	40.00	13	4	30.77
10	9	4	44.44	12	4	33.33
11	9	4	44.44	8	4	50.00
12	6	4	66.67	4	4	100.00
13	4	4	100.00	4	4	100.00
14	4	4	100.00	4	4	100.00
15	4	4	100.00	4	4	100.00
16	4	4	100.00	2	2	100.00
17	4	4	100.00	2	2	100.00
18	4	4	100.00	2	2	100.00
19	4	4	100.00	2	2	100.00
20	4	4	100.00	2	2	100.00

入力キーが長くなっていることがわかる．CPU 時間も十分に絞り込める音符数である 9 音符〜10 音符の長さで 0.18sec〜0.19sec であり，高速に検索できることがわかる．十分に絞り込めたあとは入力キーが長くなってもほとんど CPU 時間は変わっていないことから，約 500 曲の MIDI データベースからマージンを音高差 ±3，音長差 ±8 分音符では 0.18sec〜0.19sec で検索が終わるということがわかる．これらの実験結果より，入力にゆらぎがある場合においても，ある程度の長さの入力がされた場合，高速に検索が可能であることがわかる．

5 まとめ

本稿では，サフィックス・アレイを音声認識における言語モデルとして用いることを提案した．サフィックス・アレイは，情報検索のために提案された高速な探索手法

であり，この手法を音声認識の言語モデルとして用いることにより，過剰な仮説生成の抑制や対象言語外仮説の抑制などに有効である．さらに，音声認識を行うと同時に情報検索も可能であるという利点を持つ．これを実現するため，サフィックス・アレイを完全一致ではなくとも検索可能とするように拡張した，拡張サフィックス・アレイを提案した．

提案手法の有効性を検証するため，類似音楽検索実験を行った．類似音楽検索実験は，音声認識における音響モデルが担う音響計算を取り除き言語モデルの評価に適すると考えられる．実験結果より，入力にゆらぎが生じ，検索対象と完全一致ではなくとも検索可能であることを示した．また，入力長がある程度になった場合，検索をすることが可能であることを示し，高速な検索ができることがわかった．

今後は，実際に音声認識システムに本提案手法を組込み，情報検索システムを構築する予定である．

6　謝辞

本研究は，徳島大学工学部若手教員プロジェクトの資金的な支援を受けて行われた．プロジェクト関係各位の方々に深く感謝致します．

参考文献

[1] U.Manber and G.Myears. Suffix arrays: a new method for on-line string searches. *SIAM journal of Computing,*Vol.22No.5,pp935-948,1993.

[2] K.Sadakane. A fast algorithm for making suffix arrays and for burrowswheeler transformation. In *Proc. IEEE Data Compression Conference,* pp.129-138,1998.

[3] 北 研二， 津田 和彦， 獅々堀 正幹：情報検索アルゴリズム