Vol.2011-NL-204 No.1
2011/11/21

発音照合アルゴリズムを用いた早口言葉の検索

安川　美智子[†1]　横尾　英俊[†1]

日本語文書を読み上げる際に，発音が困難となる箇所の検索を目的として，日本語の発音表記の近似照合を行うアルゴリズムを提案する．片仮名の読みの記号列を，日本語の音声に基づく近似照合用の記号列に変換し，変換された記号列に対して文字 n-gram を素性とする類似文字列検索を行う．本稿では，まず，英語の発音照合アルゴリズムを，日本語のローマ字表記に適用した場合の問題点を明らかにする．そして，日本語の五十音に基づく 2 種類の発音照合アルゴリズムを提案する．また，提案法を用いて，日本語の文書データから読み上げ困難な箇所を検索し，検索された日本語文字列の読み上げを行うユーザ評価実験を行ったので，その結果についても報告する．

Applying Phonetic Matching Algorithm to Tongue Twister Retrieval in Japanese

Michiko Yasukawa[†1] and Hidetoshi Yokoo[†1]

In this paper, we propose a Japanese phonetic matching algorithm for tongue twister retrieval. We first apply a standard English phonetic algorithm to Japanese texts written in Romaji. Since a transliterated text in Romaji does not retain its original phonetic sound of Japanese, some search results are poor from the viewpoint of approximate phonetic matching in Japanese. To solve the problem, we present a new approach to phonetic matching, which is based on the Japanese syllabary and on the n-gram based inverted index search. We also report user evaluation of searched results with our approach.

1. はじめに

公式な場面におけるスピーチや学会での研究発表において，また，携帯電話等における音声認識機能を利用する際などに，言い間違いのない明瞭な発音をすることが求められる．正確かつ明確な発音を行うためには，発音が困難な箇所を何度も練習して，口の動きを滑らかにすること，すなわち，滑舌[1] を良くする訓練を行うことが効果的である．著者らはこれまでに言い間違い[2)–5)] の調査と，発音が困難な文字列[4] を用いた言い間違いの実験を行い，滑舌訓練用の例文集[6] を検索対象として，日本語の子音に注目した早口言葉検索の手法を提案した[7]．本稿では著者らの従来法と英語における標準的な手法をもとに，日本語の音声の類似性の観点から文字列検索を行う手法を検討する．

類似文字列検索を行うさまざまなアルゴリズムがこれまでに研究されており[8),9)]，情報検索における単語の表記揺れの問題を解決する手法などで発音の類似性に基づく類似文字列検索が用いられている[10),11)]．また，単語が発音されたときの音声の類似性を利用した「ことば遊び (wordplay)」[12] や「なぞなぞ」の生成[13]，ある楽曲の中で繰り返される類似のメロディを特定する手法[14] が研究されている．しかし，本研究が扱う早口言葉を目的とした類似文字列検索や日本語の音声を対象とした発音照合アルゴリズムはこれまでに検討されていない．本研究では，日本語の言語の音声という観点から，あらかじめ用意された例文を読むという状況を想定して，滑舌訓練のための例文検索を目的として，類似文字列検索を行うということに焦点を当てる．以下，本稿では，まず，広く利用されている英語用の発音アルゴリズムを，日本語のローマ字表記に直接適用した場合に生じる問題を予備実験により確認する．次に，日本語の五十音に基づく発音照合アルゴリズムを提案する．また，提案法を用いて，日本語の文書群[15] から読み上げの困難な箇所を検索する早口言葉検索について説明し，検索された結果の読み上げを行うユーザ評価実験により提案法の有用性を検証する．

2. 関連研究

発音された音声の類似性に基づく文字列検索を行うアルゴリズムのことを発音照合アルゴリズム (phonetic matching algorithm) という．英語の発音照合アルゴリズムとしてよく知られているものに Soundex[16] と Metaphone[17] がある．

Soundex は Russell らによって開発され，1918 年と 1922 年に米国特許として認定された技術であり，米国の国勢調査において，異なる綴りで同じ発音を持つ「名字 (surname)」の検索に利用された[16]．たとえば，Soundex では，異なる綴りで同じ発音を持つ **SMITH** と **SMYTH** に同一のコード **S-530** を割り当てる．このような文字列変換により，同じ発音で多様な綴りが用いられる名字を検索できるようになる．このようなアルゴリズムの目的は，1 つの名前のさまざまな変種すべてを 1 つのコードに変換することである[18]．Soundex と同様の方法は，手書き文字の読み間違いや，聞き間違いによって人名の綴り間違いが起きる

†1 群馬大学
Gunma University

Vol.2011-NL-204 No.1
2011/11/21

Step-1 最初の文字を残し，残りの文字のうち a, e, h, i, o, u, w, y を消す．

Step-2 最初の文字以外に残った文字に対して次の**数字**を割り当てる．

b, f, p, v	→	**1**	l	→	**4**
c, g, j, k, q, s, x, z	→	**2**	m, n	→	**5**
d, t	→	**3**	r	→	**6**

Step-3 コード化前の名前で同じコードを持つ文字が 2 文字以上隣接しているか，または h と w をはさんで隣接していれば最初以外をすべて消す．

Step-4 数字が 3 つ未満ならば 0 を後に付け加え，数字が 3 つを超えるならば超えた分だけ右側の数字を消すことによって「文字，数字，数字，数字」の 4 文字になるように変換する．

図 1 Soundex のコード化の規則と手順
Fig. 1 Basic coding rules and processing of the Soundex Indexing System.

航空便の予約システム[19] にも応用されている．Soundex のアルゴリズムは，英語の名字のコード化を図 1 に示す規則と手順で行う[18]．Soundex の実装が Soundex Calculator[⋆1]で公開されており，英語のアルファベットで表記された任意の文字列に対する Soundex のコードを得ることができる．Soundex の問題点は，変換後のコードの長さが 4 文字 (先頭文字に続くハイフンを含めて 5 文字) に固定されているため，元の文字列が長い場合は先頭部分しか変換されないということである．また，変換規則が非常に単純であるため，例外的な発音を持つ英語の綴りに対応できないという問題もある．

Soundex の問題点を解決する手法として，Philips[17] によって Metaphone と Double Metaphone が提案された．Metaphone は，Soundex が考慮していない例外的な発音も考慮に入れ，可変長の変換を行うことを可能としたアルゴリズムである．Metaphone では，Soundex と同様に先頭の文字を保持し，途中の母音を捨て，子音に対しては英語の発音を考慮した変換規則に従って変換することで同じ発音が同じ文字にコード化されるようにするが，Soundex では先頭以外は 6 つの数字 (123456) にコード化されるのに対して，Metaphone では 16 文字の子音を表す文字 (0BFHJKLMNPRSTWXY) でコード化される．数字の 0 は “th” を，文字 X は “sh” または “ch” の発音表記に用いられる．Metaphone では，英語の発音を Soundex よりも多くの文字を使い，詳細に表現できる．また，Metaphone では，元の文字列の先頭から何文字目までをコード化するかを指定することができる．これにより，先頭以外が無視されるという Soundex の問題を解決している．しかし，Metaphone は Soundex と同様に，ある入力文字列に対して 1 つのコードしか出力しない．同じ綴りに対して複数の発音があり得る場合は，複数の出力をすることが望ましい．そこで，Double Metaphone では，ある綴りに対して代表的な発音 (primary) に加えて，他の異なる発音 (secondary) も計算できるようにしている．これにより，発音が 1 つに限定されない外国語の名前など，曖昧性のある綴りには 2 つのコードが出力される．たとえば，Metaphone では SMITH に対して 1 つのコード SM0 だけが出力されるが，Double Metaphone では，primary として SM0，secondary として XMT というように異なる 2 つのコードを出力できる．Double Metaphone 曖昧性のない名前に対しては，secondary のコードは primary のコードと同じである．また，Double Metaphone では，Metaphone よりもさらに英語の発音の類似性を考慮した異なる綴りのマッチングが行えるような改良が加えられている．たとえば，Metaphone では，AUTO は “AT” に，また，OTTO は “OT” に変換され，変換後のコードが互いに一致しないが，Double Metaphone では，OTTO が “AT” に変換されるため，AUTO と OTTO の変換後のコードが一致する．

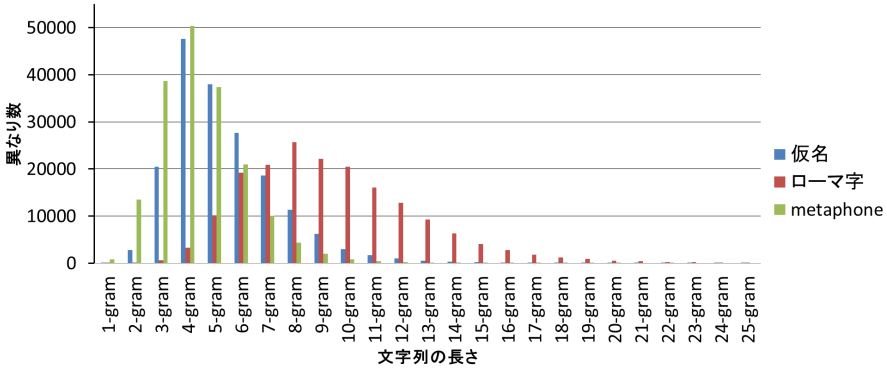

図 2 広辞苑 (第 6 版)[20] の仮名，ローマ字，metaphone の文字列長ごとの異なり数
Fig. 2 Number of unique n-grams of Hiragana/Katakana, Romaji, metaphone in a Japanese-language dictionary[20]

⋆1 http://www.eogn.com/soundex/

Vol.2011-NL-204 No.1
2011/11/21

表 1 Metaphone を用いた日本語の類似文字列検索
Table 1 Approximate string matching in Japanese using metaphone.

A	摘出手術 (てきしゅつしゅじゅつ)	判定
	構造的失業 (こうぞうてきしつぎょう)	L5
	古今銘尽大全 (ここんめいづくしたいぜん)	L0
	自発的失業 (じはつてきしつぎょう)	L5
	卓出 (たくしゅつ)	L7
	多孔質 (たこうしつ)	L1
	手形貸付 (てがたかしつけ)	L0
	特出 (とくしゅつ)	L6
	何方付かず (どっちつかず)	L0
	非自発的失業 (ひじはつてきしつぎょう)	L5
	摩擦的失業 (まさつてきしつぎょう)	L5
	利害得失 (りがいとくしつ)	L3
B	腹腔鏡手術 (ふくこうきょうしゅじゅつ)	判定
	新複合競技 (しんふくごうきょうぎ)	L4
	複合競技 (ふくごうきょうぎ)	L6
	富国強兵 (ふこくきょうへい)	L2
C	六ヵ国協議 (ろっかこくぎょうぎ)	判定
	折懸垣 (おりかけがき)	L0
	特殊法人等整理合理化計画 (とくしゅほうじんとうせいりごうりかけいかく)	L1
	理化学研究所 (りかがくけんきゅうじょ)	L3
D	貨客船万景峰号 (かきゃくせんまんぎょんぼんごう)	判定
	輝かす (かがやかす)	L1
	河況係数 (かきょうけいすう)	L4
	桔梗笠 (ききょうがさ)	L0
	公共経済学 (こうきょうけいざいがく)	L0
	呼吸酵素 (こきゅうこうそ)	L0
	国家契約説 (こっかけいやくせつ)	L3
	国共合作 (こっきょうがっさく)	L1
	後五百歳 (ごごひゃくさい)	L1
	西鶴大矢数 (さいかくおおやかず)	L0
	彭城百川 (さかきひゃくせん)	L7
	社会契約説 (しゃかいけいやくせつ)	L1
	磁化曲線 (じかきょくせん)	L6
	垂足曲線 (すいそくきょくせん)	L4
	正規曲線 (せいききょくせん)	L4
	正則曲線 (せいそくきょくせん)	L3
	大気境界層 (たいききょうかいそう)	L1
	標準正規曲線 (ひょうじゅんせいききょくせん)	L6
	忘却曲線 (ぼうきゃくきょくせん)	L3
E	マサチューセッツ州 (まさちゅーせっつしゅう)	判定
	起立性調節障害 (きりつせいちょうせつしょうがい)	L5
	公共職業能力開発施設 (こうきょうしょくぎょうのうりょくかいはつしせつ)	L0
	色彩調節 (しきさいちょうせつ)	L5

表 2 片仮名の読みからローマ字，および，Metaphone への変換表 (1)
Table 2 Encoding table to convert katakana readings to Romaji and Metaphone (1).

記号	五十音	記号	濁音・半濁音・特殊音
★	アイウエオ (母音)	★	ー (長音) ァィゥェォ(外国語音)
K	カ (ka) キ (ki) ク (ku) ケ (ke) コ (ko)	K	ガ (ga) ギ (gi) グ (gu) ゲ (ge) ゴ (go)
S	サ (sa) ス (su) セ (se) ソ (so)	S	ザ (za) ズ (zu) ゼ (ze) ゾ (zo)
X	シ (shi) チ (chi)	J	ジ (ji)
T	タ (ta) テ (te) ト (to)	T	ダ (da) ヂ (di) ヅ (du) デ (de) ド (do)
TS	ツ (tsu)	★	ッ(促音)
N	ナ (na) ニ (ni) ヌ (nu) ネ (ne) ノ (no)		
H	ハ (ha) ヘ (he) ホ (ho)	B	バ (ba) ビ (bi) ブ (bu) ベ (be) ボ (bo)
★	ヒ (hi)		
F	フ (fu)	P	パ (pa) ピ (pi) プ (pu) ペ (pe) ポ (po)
M	マ (ma) ミ (mi) ム (mu) メ (me) モ (mo)		
Y	ヤ (ya) ユ (yu) ヨ (yo)	★	ャュョ(拗音)
R	ラ (ra) リ (ri) ル (ru) レ (re) ロ (ro)		
W	ワ (wa) ヮ(wa) ヲ (wo)	★	ン (撥音)

表 3 片仮名の読みからローマ字，および，Metaphone への変換表 (2)
Table 3 Encoding table to convert katakana readings to Romaji and Metaphone (2).

記号	保持する音	記号	削除する音
A	1 文字目のア (a) とァ(a)	∅	2 文字目以降のアとァ
I	1 文字目のイ (i) とィ(i)	∅	2 文字目以降のイとィ
U	1 文字目のウ (u) とゥ(u)	∅	2 文字目以降のウとゥ
E	1 文字目のエ (e) とェ(e)	∅	2 文字目以降のエとェ
O	1 文字目のオ (o) とォ(o)	∅	2 文字目以降のオとォ
TS	ッ(促音) の後に表 4 のグループ 2 の音	∅	ッ(促音) の後に表 4 のグループ 1 の音
Y	シ，チ，ジ以外の後のャュョ(拗音)	∅	シ，チ，ジの後のャュョ(拗音)
H	ヒの後にャュョ(拗音) 以外	∅	ヒの後にャュョ(拗音)

表 4 促音に続く音の分類
Table 4 Classification of sounds behind the Japanese germinate consonant.

分類	促音に続く音
グループ 1	カ行，サ行，タ行，ハ行，ヤ行，ラ行，ザ行，ダ行，バ行，パ行の音，ヴ
グループ 2	上記以外の音 (ア行，ナ行，マ行，ワ行の音，ン)

Vol.2011-NL-204 No.1
2011/11/21

3. 予備実験

英語の発音照合アルゴリズムは，日本語の漢字かな混じり文に直接適用できないが，日本語のローマ字表記であれば適用できる．そこで，英語の発音照合アルゴリズムを用いて日本語の類似文字列検索が行えるかを確認するため，以下のような予備的な実験を行った．まず，日本語の単語を 23 万語収録している国語辞典[20] から平仮名・片仮名で表記された単語の読みを採取し,「・」「ー」などの記号を除去し，重複を省いた．さらに，KAKASI[⋆1]を用いて，片仮名・平仮名表記の文字列をローマ字表記に変換し，PostgreSQL[⋆2]の fuzzystrmatch モジュールで実装されている Metaphone 関数を用いて発音照合用の metaphone コードに変換した．得られた平仮名・片仮名，ローマ字，metaphone の文字列の異なり数を，図 2 に示す．本研究の目的は発音が困難となる文字列の検索であり，長い文字列の方が発音が困難であるため，元の仮名表記で 4 文字以下の文字列は除去することとし，得られた約 10 万個の metaphone コードを検索対象とした．次に，アナウンサーの発音困難な言葉を文献 5) から採取し，国語辞典の単語と同様にローマ字表記への変換と metaphone コードへの変換の前処理を行った．アナウンサーの発音困難な言葉の metaphone コードの先頭から 5 文字とマッチする metaphone コードを持つ日本語の単語を検索結果とし，紙に印刷したものを評価者 8 名に配付して，検索された単語が検索クエリの文字列と類似性する音声を持つかについて判定を行ってもらった．評価者は 20 代の大学生であり，7 名が日本語ネイティブ，1 名が日本語非ネイティブであった．実験結果を表 1 に示す．8 名の評価者全員が A〜E の検索クエリすべてに対して検索結果が検索クエリと類似しているかを○か×で採点した．判定レベルは，○をつけた評価者の数により，L0〜L8 の 9 段階となり，L0 は 0 人，L8 は 8 人に対応する．

実験によって明らかとなったことは，まず，人間が単語の意味を無視して，機械的に音声による類似性を判定する，ということが予想外に難しいということである．また，何が正解であるかの明確な根拠がなく，人によって音声が似ているかどうかの判断は異なり，結果として，類似性があると多くの評価者が判定した文字列というのは，検索クエリとなった文字列と平仮名の表記で共通する文字が多いもの，という結果となった．また，共通する文字があっても，長音/促音/撥音，拗音/直音，濁音/清音で不一致があるものは適合判定レベルが低くなっている．たとえば，D の「貨客船万景峰号」の先頭から 5 文字の metaphone コードは KKYKS となり，これに対応する元の平仮名の読みは「かきゃくせ (kakyakuse)」である．これと同じ metaphone を持つ「輝かす」の元の平仮名の読みは「かがやかす (kagayakasu)」であるが，判定レベルは L1 であり，類似性があると判定した評価者は 1 人だけであった．Metaphone では英語の発音の類似性に基づき kya と gaya を同じコード KY に変換しているが，日本語では「きゃ」と「がや」を同一視すると多くの人が直観的に類似しないと感じる文字列が関連付けられることになる．

ローマ字を経由して metaphone コードに変換する処理を詳しく調べるため，日本語の五十音の読みからローマ字，および，metaphone コードへの変換表を表 2 に示す．表中において ★ を付したもののうち,「ー (長音)」の記号は metaphone では処理されないため，結果として，日本語の長音は metaphone による変換では無視されることになる．また,「ン (撥音)」の後に母音が続く場合 KAKASI によるローマ字変換の際に区切りを表すシングルクォート (') が追加されるが，metaphone では処理されないため,「ナ行」の音と同じコード化がされることになる．表 2 の ★ の他の片仮名は，表 3 の変換ルールで，記号に置き換え，または，削除される．シャ(sha)，シュ(shu)，ショ(sho)，チャ(cha)，チュ(chu)，チョ(cho)，ジャ(ja)，ジュ(ju)，ジョ(jo) はローマ字変換の際にャュョの音が保持されない．また，シチジ以外にャュョが付く組み合わせでは，ローマ字変換の際に，y に変換され，Metaphone で変換される際に，拗音ャュョが，直音ヤユヨと同じ Y に変換される．たとえば，キャ(kya)，キュ(kyu)，キョ(kyo) は，Metaphone で変換されて KY となり，カ行の音にヤ行の音が続く組み合わせも変換されると KY となるため，拗音が直音と同じ音として扱われれることになる．

また，Metaphone の変換規則では，ヒャ(hya)，ヒュ(hyu)，ヒョ(hyo) のように y が後に続く h は削除され，濁音 (g) は清音 (k) と同じコード K に変換されるため，後五百歳 (ごごひゃくさい) の「ごごひゃくさ (gogohyakusa)」の metaphone コードが KKYKS となり，貨客船万景峰号 (かきゃくせんまんぎょんぼんごう) の「かきゃくせ (kakyakuse)」の metaphone コード KKYKS とマッチした．このような英語の発音に基づく類似性の判定は，日本語の話者の直感には合致せず，結果として判定レベルは L1 であった．

予備実験では，英語の発音照合アルゴリズムとして Metaphone を使用したが，Double Metaphone は英語の音声に特化したさらなる改良が加えられており，日本語に適用するとさらに類似しない文字が過度に関連付けられてしまう．たとえば,「赤魚 (あかうお)」と「英会話 (えいかいわ)」は Metaphone では，それぞれ “AK” と “EKW” に変換されるが，Double Metaphone では，同じコード “AK” に変換され，同じ発音であるとみなされる．

⋆1 http://kakasi.namazu.org/
⋆2 http://www.postgresql.org/

以上のことから，日本語文字列に対して英語の発音照合を適用した場合，日本語独特の音の類似性が考慮されず，結果として類似しない文字列が多く検索されてしまうことがわかった．特に表 2 の ★ の変換は，英語と日本語で発音音声の特徴が大きく異なるため，日本語の発音照合を検討する際には，日本語の独特の音の類似性を考慮する必要がある．

表 5 片仮名表記の読みから近似照合用の平仮名の記号への変換表
Table 5 Encoding table to convert katakana readings to hiragana phonetic symbols.

記号	五十音	記号	濁音	記号	半濁音	記号	特殊音
あ	アイウエオ					あ	ー (長音)
あ	キ ヱヲ					あ	ァィゥェォヮ
か	カキクケコ	が	ガギグゲゴ				
さ	サシスセソ	ざ	ザジズゼゾ				
		ざ	ヂヅ			っ	ッ(促音)
た	タチツテト	だ	ダ デド				
な	ナニヌネノ						
は	ハヒフヘホ	ば	バビブベボ	ぱ	パピプペポ	ば	
		ば	ヴ				
ま	マミムメモ						
や	ヤ ユ ヨ					ゃ	ャュョ(拗音)
ら	ラリルレロ						
わ	ワ					ん	ン (撥音)

4. 提 案 法

予備実験で得た知見をもとに，日本語の発音照合の 2 つの手法を提案する．著者らが 7) で検討した早口言葉検索の手法では，平仮名から発音表記のための英語のアルファベットに変換していたため，片仮名「ヴ」で表記される外国語音の中に変換できないもの（たとえば「ヴィヴァルディ」) があった．また，小文字「ヮ」の変換ルールが未定義であったため，変換できない外国語音 (たとえば「マリー・クヮント」) があった．これの問題に対処するため，本稿で提案する手法では，変換前の読みは片仮名で表記し，変換後のコードは平仮名で表記する．そして,「ヴ」と「ヮ」の変換規則を新たに追加し，表 5 に示すような変換表を用いて近似照合用の文字列への変換を行う．

まず，1 つめの発音照合アルゴリズムとして，表 5 に従い，片仮名の読みに含まれるすべての音を保持する (母音と特殊音を消去しない) 変換を行う日本語の発音照合アルゴリズム jpma-1 (Japanese Phonetic Matching Algorithm-1) を提案する．たとえば，jpma-1 では，文字列「摘出手術」の片仮名表記の読み「テキシュツシュジュツ」を表 5 の変換規則に従い，近似照合用の平仮名の記号列「たかさゃたさゃざゃた」に変換する．次に，変換された平仮名の記号列を長さ n の文字列となるように先頭から，1 文字ずつずらしながら部分文字列を切り出し，任意の n の値で先頭と終端を考慮に入れた文字 n-gram の集合を生成する．具体的には，n=3 (3-gram) の場合，平仮名の記号列「たかさゃたさゃざゃた」に先頭と終端を表す平仮名以外の記号 ($) を追加して,「$たかさゃたさゃざゃた$」とし，先頭から 1 文字ずつずらしながら 3 文字の部分文字列の集合 {$たか，たかさ，かさゃ，さゃた，ゃたさ，たさゃ，さゃざ，ゃざゃ，ざゃた，ゃた$} を作成する．同様の手順で，検索対象となるすべての文字列に対して，それぞれ n-gram の集合を生成する．そして，あらかじめ n-gram を索引語とする転置インデックス (inverted index) を作成しておき，検索クエリとなる文字列が入力された段階で，検索クエリに対して検索対象と同様の前処理を行い，作成しておいた転置インデックスを用いて全文検索を行う．検索対象となる文字列の集合が小規模であるとき，n の値が大きいと，検索結果がゼロとなる．逆に n の値が小さいと，類似性の低い文字列が検索されるようになるので大規模な文書集合に対する検索を行うときは n の値を大きくするのが良い．後述の評価実験では，5-gram の部分文字列集合を生成し，類似する文字列の検索を行っている．

次に，2 つ目の提案法として，jpma-1 と同様に日本語の発音音声を考慮した変換を行いつつ，英語の発音照合アルゴリズム Metaphone と同様に，子音以外の文字の消去を行う日本語の発音照合アルゴリズム jpma-2 (Japanese Phonetic Matching Algorithm-2) を提案する．Metaphone と同様に，jpma-2 では元の文字列の 1 文字めの母音は保持し，2 文字め以降の母音はすべて削除する．また,「ー (長音)」,「ッ(促音)」,「ャュョ(拗音)」,「ン (撥音)」などの特殊音はすべて削除する．たとえば，jpma-2 では，文字列「大きな金貨」の片仮名の読み「オオキナキンカ」を表 5 の変換規則に従い，近似照合用の平仮名の記号列「ああかなかんか」に変換した後，先頭以外の母音と特殊音を削除し，記号列「あかなかか」を得る．そして，jpma-1 と同様に，任意の n の値で先頭と終端を考慮に入れた文字 n-gram の集合を生成する．具体的には，n=3 (3-gram) の場合，平仮名の記号列「あかなかか」に先頭と終端を表す記号を追加して「$あかなかか$」とし，先頭から 1 文字ずつずらしながら 3 文字の部分文字列の集合 {$あか，あかな，かなか，なかか，かか$} を作成する．以降，jpma-1 と同様に文字 n-gram を索引語とする転置インデックスの作成と，転置インデックスを用いた全文検索を行う．

Vol.2011-NL-204 No.1
2011/11/21

表 6 日本語の発音照合アルゴリズムによる類似文字列検索 (1)
Table 6 Tongue twister retrieval with Japanese phonetic matching algorithm (1).

近接音	ちょっと立 (た)って手 (て) 伝 (つだ)ってってちょうだいと言 (い)ったんで　立 (た)ったとたん 立 (た)ったついでだ炭団 (たどん) 取 (と)ってちょうだいと　また　言 (い) われた.
（1）	受 (う) け取 (と)って頂戴 (ちょうだい)
（2）	幹部 (かんぶ) の人 (ひと) 達 (たち) と茶 (ちゃ) を喫 (の) んでいた
（3）	連 (つ) れてって頂戴 (ちょうだい)
（4）	心臓 (しんぞう) が高 (たか) い音 (おと) を立 (た) てて踊 (おど)っていた
（5）	手伝 (てつだ)ってちょうだい
（6）	ちょっと立合 (たちあ)って頂 (いただ) きたいんですが

表 7 日本語の発音照合アルゴリズムによる類似文字列検索 (2)
Table 7 Tongue twister retrieval with Japanese phonetic matching algorithm (2).

異種音	この杭 (くい) の釘 (くぎ) は　引 (ひ) き抜 (ぬ) きにくい釘 (くぎ)　引 (ひ) きにくい釘 (くぎ) くぎ抜 (ぬ) きで抜 (ぬ) く.
（1）	この記憶 (きおく) が消 (き) えてしまって
（2）	こんな大 (おお) きな金貨 (きんか) があるんだろうか
（3）	沈 (しず) んだ家 (いえ) のなかの空気 (くうき) が
（4）	過去 (かこ) 一年間 (いちねんかん) の大 (おお) きな記憶 (きおく) が
（5）	所謂 (いわゆる) 家庭的 (かていてき) な空気 (くうき) が負担 (ふたん) で
（6）	蒼白 (そうはく) な顔 (かお) に覚悟 (かくご) の瞳 (ひとみ) を輝 (かがや) かしながら

表 8 読み上げ困難な箇所
Table 8 Difficult parts to pronounce.

頻度	文字列
6	輝 (かがや) かしながら
5	人 (ひと) 達 (たち) と茶 (ちゃ)
4	立 (た) てて踊 (おど)って，頂 (いただ) きたいんですが，覚悟 (かくご) の瞳 (ひとみ)
3	受 (う) け取 (と)って，立合 (たちあ)って
2	高 (たか) い音 (おと)，手伝 (てつだ)って，沈 (しず) んだ家 (いえ)
1	喫 (の) んでいた，連 (つ) れてって，この記憶 (きおく) が，大 (おお) きな金貨 (きんか)，大 (おお) きな記憶 (きおく) が，家庭的 (かていてき) な空気 (くうき)，蒼白 (そうはく) な顔 (かお)

5. 評価実験

提案アルゴリズムは検索クエリとして入力された文字列と，類似の発音を持つ文字列を検索するため，発音が困難な文字列を検索結果として得たい場合は，検索クエリとして発音が困難である文字列を入力する必要がある．発音が困難である文字列は 2 つのタイプに分けることができ，1 つは子音の調音位置が微妙に変化する近接音型，もう 1 つは子音の調音位置が大きく変化する異種音型である．評価実験では，発音が困難であることが知られ，滑舌訓練に用いられている伝承の早口言葉から，近接音型と異種音型の例文を 1 つずつ選び，提案法の有効性をユーザ評価実験により確認する．

日本語の読み上げ困難な箇所が提案アルゴリズムにより，検索できているかを確認するユーザ評価実験を以下のように行った．まず，提案アルゴリズムを用いて，『青空文庫 全』[15] の付録 DVD-ROM に収録された日本語の文書から，読み上げの困難な箇所を検索するため，文書の前処理を行った．文書の表記には旧字と新字，旧仮名と新仮名が使用されている．実験には新字新仮名の文書を使用した．また，図 5 のような備考が含まれている文書があるが，実験では，備考を含まない文書を使用することとした．

> 「備考： この作品には，今日からみれば，不適切と受け取られる可能性のある表現がみられます．その旨をここに記載した上で，そのままの形で作品を公開します.（青空文庫）」

図 3 青空文庫の備考
Fig. 3 The remarks column of Aozorabunko.

前処理として，すべての文書に共通して記述されているヘッダとフッターの部分を除去し，本文を句読点と台詞の鍵括弧の終わりで，分割した．次に，ChaSen (chasen-2.4.2 + ipadic-2.7.0)[⋆1] を使用して漢字かな混じり文から片仮名の読みに変換した．変換後の片仮名の長さが 10 文字以上，かつ，100 文字未満の文字列を実験に使用することとした．以上の処理により，2,973 件の日本語の文書から 1,711,453 個の文字列を採取し，各文字列ごとに識別用の番号を割り当て，実験用のデータコレクションとした．

⋆1 http://chasen-legacy.sourceforge.jp/

実験用のデータコレクションの片仮名の読みの文字列を，日本語の文字列照合アルゴリズムを用いて，近似照合用の平仮名の文字列に変換し，変換後の平仮名の先頭から連続 5 文字の部分文字列を 1 文字ずつずらしながら，文字列の末尾に達するまで部分文字列を切り出し，文字 5-gram を索引語 (index term) とする仮想的な検索用文書を作成した．Indri[⋆1]で索引処理をするため，TREC SGML フォーマットと呼ばれる図 5 の書式に変換した．図 5 の例は,「大きな金貨 (おおきなきんか)」を jpma-2 で変換した記号列「あかなかか」から作成した 3-gram の集合＄あか，あかな，かなか，なかか，かか＄の各要素を索引語としている．

```
<DOC>
<DOCNO>EX-00001</DOCNO>
<TEXT>
＄あか　あかな　かなか　なかか　かか＄
</TEXT>
</DOC>
```

図 4 TREC SGML 文書形式 (文字列照合用にコード化された文字列の文字 3-gram の例)
Fig. 4 The TREC SGML document format (An example of character based 3-grams of phonetic syllables).

次に滑舌訓練用の例文集[6] から，近接音型[⋆2]と異種音型[⋆3]の例文をそれぞれ 1 件ずつ採取し，文書群と同様に，ChaSen を使用して片仮名の読みに変換した後，日本語の文字列照合アルゴリズムを用いて，近似照合用の平仮名の文字列に変換し，先頭から 1 文字ずつずらしながら文字列の末尾に達するまで連続する 5 文字を切り出して文字 5-gram を生成し，検索クエリとして使用した．索引処理と検索処理を検索エンジン Indri を用いて行った．検索時の条件は，検索結果のスコア付けに Okapi BM25 の計算式を使用するオプションを設定し，パラメータはデフォルト値 ($k1 = 1.2, b = 0.75, k3 = 7$) とした．jpma-1 を適用した検索結果と jpma-2 を適用した検索結果の上位から 3 件ずつ交互に取得して，A3 用紙にルビ付きで印刷をして，被験者の協力を得て，読み上げ困難であるかの評価を行った．実験に参加した被験者 8 名は 20 代の大学生であり，7 名が日本語ネイティブ，1 名が日本語非ネイティブであった．用紙に印刷した文字列 (検索クエリとして使用した文字列と検索結果として得られた文字列) を表 6 と表 7 に示す．表中の奇数番号が jpma-1，偶数番号が jpma-2 の検索結果である．被験者は検索結果ごとに用紙に記載された文字列を 1 分間，黙読し，読み上げる内容をあらかじめ確認した．実験者が時間を計測し，1 分間が経過した後，被験者は用紙に印刷された文字列 (検索クエリとして使用した文字列も含めてすべての文字列) を可能な限り速いスピードで読み上げを行った．被験者が文字列を読み上げる際に，言い間違い，不明瞭な発音，読む速度の急激な低下が生じた箇所を，読み上げ困難な箇所として，実験者が記録した．読み上げ困難な箇所と，言い誤り等が生じた頻度を表 8 に示す．

被験者全員に共通して読み上げ困難であった箇所は，検索クエリの「ちょっと立 (た)って手 (て) 伝 (つだ)ってって」「ついでだ炭団 (たどん) 取 (と)って」「引 (ひ) き抜 (ぬ) きにくい釘 (くぎ)　引 (ひ) きにくい釘 (くぎ)」の部分であった．jpma-1 は検索対象に検索クエリと共通する部分文字列を含むものがあれば検索クエリと高い類似性を持つ文字列の検索を行える．jpma-2 は読みで見比べると，検索クエリと検索結果の文字の共通性が，人間が直観的が理解しにくいが，検索クエリの「子音の読みにくさ」の特徴と一致するものが検索されており，また，jpma-2 の検索結果文字列を言い間違えた人も多かったので，一定の有効性があると考えられる．被験者が読み上げる例文は jpma-1 と jpma-2 が交互に印刷されたものであったため，互いの検索結果文字列が影響し合った可能性もある．たとえば，検索クエリと類似性する文字列を含む jpma-1 の検索結果を，言い間違いをせずに慎重に読んだあと，検索クエリと字面上は類似していないように見える jpma-2 の検索結果を不用意に読んで言い間違うといったように，心理的な影響が生じていた可能性もある．読み上げる例文について，日本語ネイティブからは古い言葉 (たとえば，炭団) は意味がよく分からない，というコメントがあった．また，日本語非ネイティブからは日常的に使用しない言葉（たとえば，杭）がよく分からないというコメントがあった．古くから受け継がれてきた早口言葉は，現代ではあまり使われない言葉を含んでいる．現代の身近な言葉を含むさまざまな文書集合を検索対象として早口言葉を検索することと，既存の早口言葉と類似する別の新規の早口言葉を生成することを今後，検討していく予定である．

6. おわりに

日本語の発音に基づく音声の近似照合を行うアルゴリズムを提案し，滑舌訓練用の早口言葉の検索に提案法を適用した．提案アルゴリズム 2 種を用いて，発音が困難な文字列と類似する発音を持つ文字列の検索を行った．また，検索結果を速いスピードで音読し，読み上げ

⋆1 http://sourceforge.net/apps/trac/lemur/
⋆2 子音の調音位置の近接する音が続くことにより，発音が困難となる例文.
⋆3 子音の調音位置の離れた音が続くことにより，発音が困難となる例文.

困難な文字列が検索されているかどうかを確認するユーザ評価実験を行い，提案法の有効性を確認した．検索対象に様々な文書を用いることで，伝承の日本語の早口言葉と類似する多様な例文を滑舌訓練に利用できると期待できる．存在する文書群から検索手法を検討してきたが，発音困難な文字の組み合わせは特殊であるため，大量の文書群を検索対象とした場合でも，類似文字列がほとんど検索されないことがある．既存の早口言葉と類似する新たな早口言葉を生成する手法などを今後検討していく予定である．

謝辞 本研究は科研費 (課題番号: 21700273) の助成を受けたものである．ユーザ評価実験に御協力いただいた群馬大学横尾研究室の学生の皆様に感謝申し上げます．

参 考 文 献

1) 橋本行洋：「カツゼツ (滑舌・活舌)」の語誌–近代の漢語受容と辞書，国語と国文学，Vol.82, No.12, pp.50–65 (2005).
2) Fromkin, V.: Slips of the tongue, pp.181–187 (1973).
3) 寺尾　康：音韻性錯語と健常者の言い誤りとの比較分析，失語症研究， Vol.19, No.3, pp.193–198（オンライン），DOI:http://dx.doi.org/10.2496/apr.19.193 (1999).
4) フジテレビトリビア普及委員会：トリビアの泉 (6)，pp.131–132, 講談社 (2004).
5) フジテレビトリビア普及委員会：トリビアの泉 (19)，pp.83–86, 講談社 (2007).
6) 塩原慎次郎：声を出して読む日本語の本，創拓社 (1987).
7) 鶴巻有香，安川美智子，横尾英俊：子音に注目した早口言葉の検索，情報処理学会研究報告 自然言語処理（NL）， Vol. 2011-NL-201, No. 14, pp. 1–6（オンライン），入手先⟨ http://id.nii.ac.jp/1001/00074041/ ⟩ (2011).
8) Navarro, G.: A guided tour to approximate string matching, *Computing Surveys (CSUR)*, Vol.33, No.1, pp.31–88 (online), DOI:http://dx.doi.org/10.1145/375360.375365 (2001).
9) Elmagarmid, A. K., Ipeirotis, P. G. and Verykios, V. S.: Duplicate record detection: A survey, *IEEE Trans. Knowl. Data Eng.*, Vol. 19, No. 1, pp. 1–16 (online), available from ⟨ http://doi.ieeecomputersociety.org/10.1109/TKDE.2007.9 ⟩ (2007).
10) Zobel, J. and Dart, P.W.: Phonetic string matching: lessons from information retrieval, *SIGIR '96 Proceedings*, ACM, pp.166–172 (online), DOI:http://dx.doi.org/10.1145/243199.243258 (1996).
11) French, J. C., Powell, L. and Schulman, E.: Applications of approximate word matching in information retrieval, *CIKM '97 Proceedings*, ACM, pp. 9–15 (online), DOI:http://dx.doi.org/10.1145/266714.266721 (1997).
12) Taylor, J.M. and Mazlack, L.J.: Humorous wordplay recognition, *SMC (4)*, pp.3306–3311 (online), DOI:http://dx.doi.org/10.1109/ICSMC.2004.1400851 (2004).
13) 濱田真樹，鬼沢武久：同音異義語の意味の多様性を構造にもつなぞなぞの生成，知能と情報， Vol.20, No.5, pp.696–708（オンライン），DOI:http://dx.doi.org/10.3156/jsoft.20.696 (2008).
14) Cambouropoulos, E., Crochemore, M., Iliopoulos, C.S., Mohamed, M. and Sagot, M.-F.: All maximal-pairs in step-leap representation of melodic sequence, *Inf. Sci.*, Vol.177, No.9, pp.1954–1962 (online), DOI:http://dx.doi.org/10.1016/j.ins.2006.11.012 (2007).
15) 青空文庫：『青空文庫　全　もう一つの読む自由』，http://www.aozora.gr.jp/kizokeikaku/ から冊子の本文をダウンロード可能．2007 年から 2010 年にかけて公共図書館，大学・短大・高専付属図書館，高校図書館に DVD-ROM 付き冊子を配送 (2007).
16) The U.S. National Archives and Records Administration: *The Soundex Indexing System*, (online), available from ⟨ http://www.archives.gov/research/census/soundex.html ⟩ (2007).
17) Philips, L.: The Double Metaphone Search Algorithm, *C/C++ Users Journal*, (online), available from ⟨ http://drdobbs.com/cpp/184401251 ⟩ (2000).
18) ドナルド・E. クヌース：*The Art of Computer Programming Volume 3 Sorting and Searching Second Edition* 日本語版，pp.375–376, アスキー (2004).
19) Davidson, L.: Retrieval of misspelled names in an airlines passenger record system, *Commun. ACM*, Vol.5, pp.169–171 (online), DOI:http://doi.acm.org/10.1145/366862.366913 (1962).
20) 新村　出：広辞苑 第六版，岩波書店 (2008).