

音声中の検索語検出における 検出誤り抑制パラメータの検討

古 屋 裕 斗^{†1} 名 取 賢^{†2}
西 崎 博 光^{†3} 関 口 芳 廣^{†3}

我々は、これまでに、音声中の検索語検出のために、複数の音声認識システムの出力結果を用いた音素ネットワーク型インデキシングを提案し、高い検索精度を実現した。また、複数の音声認識システムを利用することから、それらの多数決による信頼度情報の導入により、誤検出を抑制する方法を検討した。しかしながら、特に音素数が小さいクエリを検索する場合には、検索精度が著しく悪化することが明らかとなった。本稿では、入力するクエリの音素数に応じて探索パラメータを変更することで、さらなる検索精度改善を目指す。STD テストコレクション未知語セットを用いて評価した結果、提案手法により検索精度を改善できることが分かった。

False Detection Control Parameters for Spoken Term Detection

YUTO FURUYA,^{†1} SATOSHI NATORI,^{†2}
HIROMITSU NISHIZAKI^{†3} and YOSHIHIRO SEKIGUCHI^{†3}

We have already proposed the phoneme transition network indexing using multiple speech recognition systems' outputs for the spoken term detection (STD) task. In addition, we have shown that using a confidence measure gotten by voting the multiple speech recognizers' outputs drastically reduced the false detection errors. However, the STD performance was very low for system-inputted queries consist of low number of phonemes. This paper aims to improve the STD performance by tuning the search parameters in the STD engine. The experimental results showed our proposed technique make the better STD performance for shorter number of phonemes.

1. はじめに

これまでに、我々は、複数の音声認識システムの出力からコンフュージョンネットワーク (Confusion Network : CN) を基とした、音声中の検索語検出 (Spoken Term Detection : STD) 用のインデックスを構築する方法を提案してきた¹⁾²⁾。

これらの STD 手法において、ネットワーク型のインデックスに対する検索語の検出方法は動的計画法 (Dynamic Programming : DP) を用いた単純な検索方法であった。単純な検索アルゴリズムを用いた理由は、研究の主旨が複数の音声認識システムを利用した STD 用インデックスの構築にあったためである。

さらに、我々が提案してきたネットワーク型インデックスに対して、複数の音声認識システムから得られる情報を利用した、誤検出を抑制する検索手法を提案した³⁾。これにより、誤検出を抑制することができ、さらに高い検索精度を実現することができた。

しかしながら、音素数が小さいクエリの場合、誤検出が増加してしまい、高い検索精度が得られないことが分かった。そこで、本研究では、クエリの音素数に依存して探索パラメータを調整することで、検索精度の改善を図る。

本稿では、日本語話し言葉コーパスを利用した日本語 STD テストコレクション⁴⁾を用いて、クエリの音素数に応じて探索パラメータを変更することで、誤検出を抑制する試みについて報告する。

2. 複数音声認識システムを用いた STD

本稿では、複数の音声認識システムの出力を CN を利用してネットワーク型インデックスとして統合した音素遷移ネットワーク (Phoneme Transition Network : PTN)²⁾ に対して、PTN を構築する際に得られる情報を利用した検索方法を数種類用意し、これらの検索方法の性能比較を行う。

PTN を利用した STD の流れを図 1 に示す。

^{†1} 山梨大学工学部コンピュータ・メディア工学科

Department of Computer Science and Media Engineering, Faculty of Engineering University of Yamanashi

^{†2} 山梨大学大学院医学工学総合教育部

Department of Educational Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi

^{†3} 山梨大学大学院医学工学総合研究部

Department of Research Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi

2.1 複数の音声認識システム

本研究では、音声データは 10 種類の音声認識システムによって認識される。デコーダには Julius rev. 4.1.3 (LVCSR のためのオープンソースデコーダ) を使用した。このデコーダに対し、2 種類の音響モデル (AM) と 5 種類の言語モデル (LM) を用意し、AM と LM の組み合わせによって 10 種類の音声認識システムを構築した。

AM は HMM で、日本語話し言葉コーパス (CSJ) のコア講演以外の講演音声から学習した **Syl** (syllable モデル：音節 124 種) と **Tri** (triphone モデル：音素 43 種) の 2 種類である。

すべての LM は、単語もしくは文字ベースの trigram モデルである。以下に、LM の詳細を示す。

WBC：単語ベースの trigram モデル。単語は、漢字と英数字、平仮名、片仮名で構成されている。

例：今回 / の / 実験 / の / 目的

WBH：単語ベースの trigram モデル。単語はすべて平仮名で構成され、元の単語に漢字や英数字、片仮名が含まれている場合には、すべて平仮名系列に変換される。

例：こんかい / の / じっけん / の / もくてき

CB：文字ベースの trigram モデル。文字はすべて平仮名によって構成されている。

例：こ / ん / か / い / の / じ / っ / け / ん / の / も / く / て / き

BM：文字系列ベースの trigram モデル。文字系列は 2 文字の平仮名によって構成されている。

例：こん / かい / のじ / っけ / んの / もく / てき

Non：LM を使用しない。連続音素 (音節) 認識と等価となる。

Non 以外のすべての LM は、CSJ のコア講演以外の講演音声を書き起こしたテキストから学習している。

なお、学習した AM と LM、認識用単語辞書、音声認識は、2010 年 5 月に公開された CSJ の STD 用テストコレクション⁴⁾の音声認識条件に基づき学習、認識を行った。

LM に WBC を用いた場合の平均単語認識率を表 1 に示す。また、日本語話し言葉コーパス (CSJ) のコア講演音声 (約 44 時間) を用いた場合の、10 種類の音声認識システムの音節認識率と、10 種類の音声認識システムの出力を時間同期で連結させたときの音節認識率を表 2 に示す。

表 2 では、10 種類の音声認識システムを用意することで、94%という高い Corr. を達成することができている。また、単一の認識システムの 10-best 出力を組み合わせたとの結果と、10 種類の音声認識システムの出力を組み合わせたとの結果を比較すると、10 種類の音声認識シ

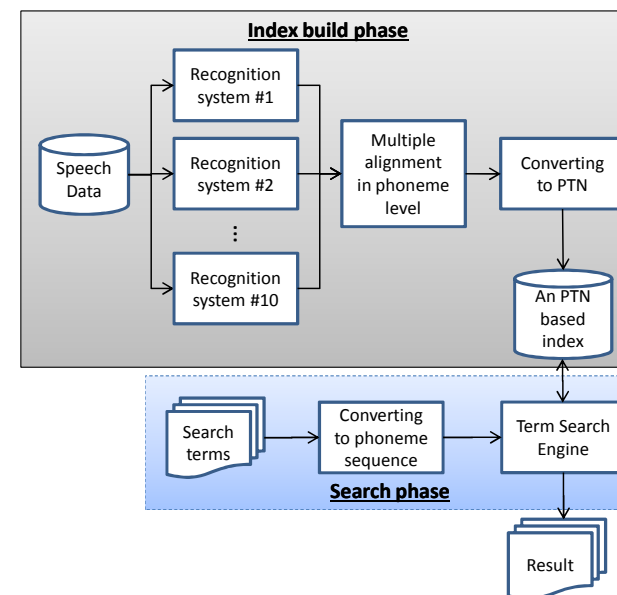


図 1 PTN 型インデックスを用いた STD 手法の概要図

表 1 CSJ 講演の平均単語認識率 [%]

LM / AM	Corr.	Acc.
WBC/Tri	76.68	71.93
WBC/Syl	67.54	64.10

テムの出力を組み合わせたとの結果の方が Corr. が良い。つまり、単一の音声認識システムの出力より、複数の音声認識システムの出力を組み合わせたとの方が、特定のキーワードを見つけられる可能性が高くなる。しかし、大量の挿入誤りが発生しているため、キーワードの検索において多くの誤検出が発生する可能性が高くなる。

2.2 ネットワーク型インデックスの構築

ネットワーク型インデックスの構築は図 2 に示すように行われる。図 2 は PTN の構築例を表しており、複数の出力の例と、認識結果全ての出力を音素ベースで整合を取った結果、整合が取られた音素系列を CN に変換したイメージを表す。

表 2 CSJ 講演の平均音節認識率 [%]

LM / AM	1-best		10-best	
	Corr.	Acc.	Corr.	Acc.
WBC/Tri	86.46	83.01	89.96	44.88
WBH/Tri	86.27	81.42	89.95	35.06
CB/Tri	81.83	77.42	85.99	41.74
BM/Tri	83.60	78.64	88.35	39.47
Non/Tri	71.00	51.20	74.56	21.06
WBC/Syl	79.11	76.35	84.19	35.73
WBH/Syl	79.32	75.83	84.29	29.90
CB/Syl	73.84	71.18	79.47	42.10
BM/Syl	77.89	74.42	84.60	37.26
Non/Syl	63.68	45.43	67.96	21.57
10Systems	94.28	-13.78	96.47	-243.51

本稿では、各 Arc に存在する音素に対する遷移確率などの重み付けは一切考慮せず、ネットワーク型インデックスに変換している。図 2 の “@” は NULL 遷移を示す。

本研究では、PTN は発話単位で構築しておく。

2.3 用語検索エンジン

2.3.1 用語検索アルゴリズム

検索語は音声認識辞書を用いて音素単位に変換され、用語検索エンジンに入力される。

検索アルゴリズムは DP を用いた単純な検索方法である。DP の傾斜制限は図 3 のように行っており、**X** が PTN、**Y** が検索語に対応する。

DP の各遷移コストは編集距離 (Edit Distance) に基づいており、一致の場合は 0、誤りの場合は置換・挿入・脱落に関わらず全て 1 としている。また、PTN には NULL 遷移が存在しており、この NULL 遷移に対するコストとして 0.1 を設定している。

図 4 は PTN から DP によって “k o s a i N” を見つける例である。PTN は 2 Node 間に複数の Arc を持っており、検索エンジンはこの複数の Arc を考慮した距離（スコア）計算を行う。

2.3.2 誤検出抑制パラメータ

本稿では、誤検出を抑制するために、PTN を構築する際に得られる複数の情報を利用する。今回利用したパラメータは次に示す 2 種類である。

Voting : 同じ音素を認識していた音声認識システムの数。

多くの音声認識システムで認識されているほど、その音素の信頼性が高くなる可能性がある

Input voice data : Cosine (/k o s a i N/)

LM/AM	Outputs of 10 recognition systems (all outputs are converted into phoneme sequence)								
	1	2	3	4	5	6	7	8	9
WBC/Tri	k	o	s	@	a	@	@	i	@
WBH/Tri	q	o	s	u	a	@	a	@	N
CB/Tri	k	o	s	@	a	m	a	i	@
CSB/Tri	k	o	s	@	a	@	@	@	N
Non/Tri	k	o	s	@	a	@	@	@	N
WBC/Syl	@	@	s	@	a	@	a	@	N
WBH/Syl	b	o	s	@	a	a	a	@	@
CB/Syl	@	@	s	@	a	b	@	i	@
CSB/Syl	@	@	s	@	a	@	@	@	N
Non/Syl	@	@	s	@	a	@	@	@	N

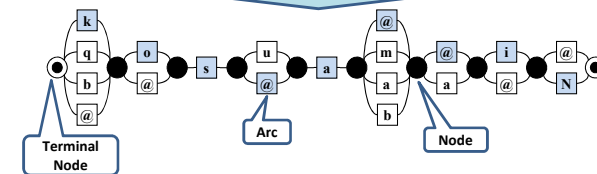


図 2 10 種類の音声認識システムの出力結果から PTN へ変換する例

ArcWidth : 2 Node 間に存在する Arc の数。

Arc の数が少なくなるほど、その Node 間の認識結果の信頼性が高くなる可能性がある

2.3.3 誤検出抑制パラメータの DP コストへの適用

編集距離に基づく距離の計算に、抑制パラメータに基づくスコアを加味することで誤検出を考慮した検索エンジンを実現する。各スコアは、式 (2) から (5) に示すように算出され、式 (1) に示すように適用される。なお、各スコアは単体または組み合わせて使用することが可能であり、その際は式 (1) の該当項が適用されなくなる。

なお、あるクエリとある発話から作成した PTN が与えられたとき、DP スコアが設定した閾値以下の場合に、そのクエリが含まれていたと判断する。閾値は、クエリの音素数で正規化しておく。

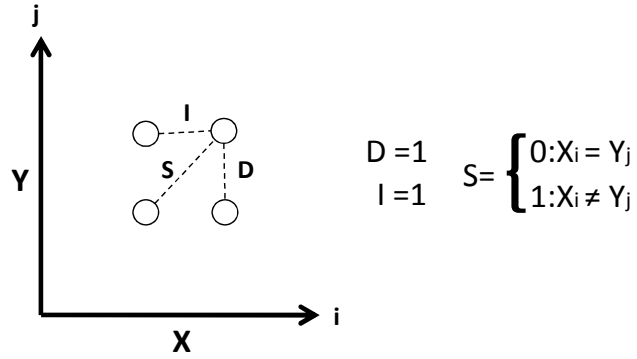


図3 DPパスの遷移コストの定義

$$D(i, j) = \min \begin{cases} D(i, j-1) + 1.0 \\ D(i-1, j) + \text{Null}(i) \\ D(i-1, j-1) + \text{Match}(i, j) + \text{Vot}(i, j) + \text{Acw}(i) \end{cases} \quad (1)$$

$$\text{Match}(i, j) = \begin{cases} 0.0 : \text{Query}(j) \in \text{PTN}(i) \\ 1.0 : \text{Query}(j) \notin \text{PTN}(i) \end{cases} \quad (2)$$

$$\text{Null}(i) = \begin{cases} 0.1 : \text{NULL} \in \text{PTN}(i) \\ 1.0 : \text{NULL} \notin \text{PTN}(i) \end{cases} \quad (3)$$

$$\text{Vot}(i, j) = \begin{cases} \alpha \div \text{Voting}(p) : \exists p \in \text{PTN}(i), p = \text{Query}(j) \\ 0.0 : \text{Query}(j) \notin \text{PTN}(i) \end{cases} \quad (4)$$

$$\text{Acw}(i) = \beta \times \text{ArcWidth}(i) \quad (5)$$

$D(i, j)$ は DP 格子上の (i, j) の位置に至るまでの距離である。

$\text{Query}(j)$ は検索語の j 番目の音素を表し、 $\text{PTN}(i)$ は PTN の i 番目の Node が持つ Arc の集合を表す。また、 p は PTN の i 番目の Node が持つ、ある Arc の音素を表す。

式 (4) の α は NULL 遷移よりコストを低くするために 0.5 を設定した。 $\text{Voting}(p)$ は $\text{Query}(j)$ と一致する p を認識していた音声認識システムの数を表す。

式 (5) の β も NULL 遷移よりコストを低くするために 0.01 を設定した。 $\text{ArcWidth}(i)$ は $\text{PTN}(i)$ の Arc の数を表す。

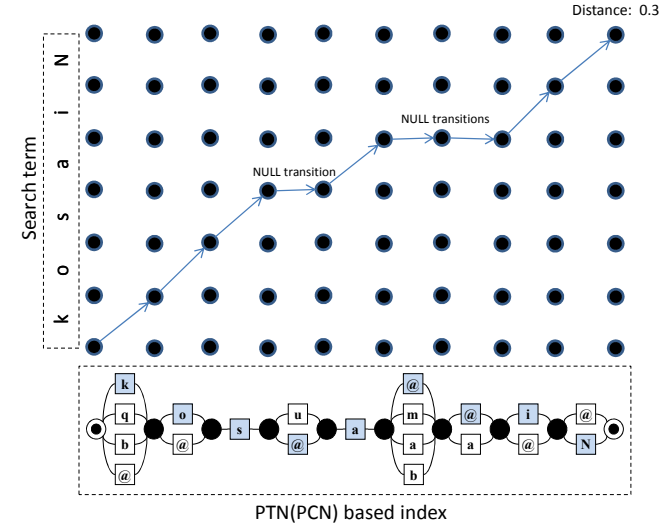


図4 クエリと PTN 型インデックスとのマッチング処理の例

3. STD 評価実験

3.1 実験条件

3.1.1 STD タスク

本稿での STD は、CSJ のコア講演音声 (約 44 時間) から検索語を検出するというタスクである。

検索語には、STD のためのテストコレクション⁴⁾のうち、コア講演用未知語セットを用いた。コア講演用未知語セットは 50 種の検索語、総出現箇所 233 のテストセットとなっている。

本稿で用意した検索性能の比較のためのインデックスは 10 種類の音声認識システムの 1-best 出力を音素単位で CN 化した PTN ベースのインデックスである。

誤検出抑制パラメータを適用させる組み合わせは、表 3 に示すものとした。

3.1.2 評価尺度

検索性能の評価には、Recall-Precision カーブ、F-measure、MAP⁵⁾を用いた。

Recall-Precision カーブは全検索語の合計検索結果から算出した。

表 3 誤検出抑制パラメータの組み合わせ

組み合わせ	誤検出抑制パラメータ
Only EditDist	—————
+ Voting Cost	“Only EditDist” + Voting
+ ArcWidth Cost	“Only EditDist” + ArcWidth
+ Vot+Acw Cost	“Only EditDist” + Voting + ArcWidth

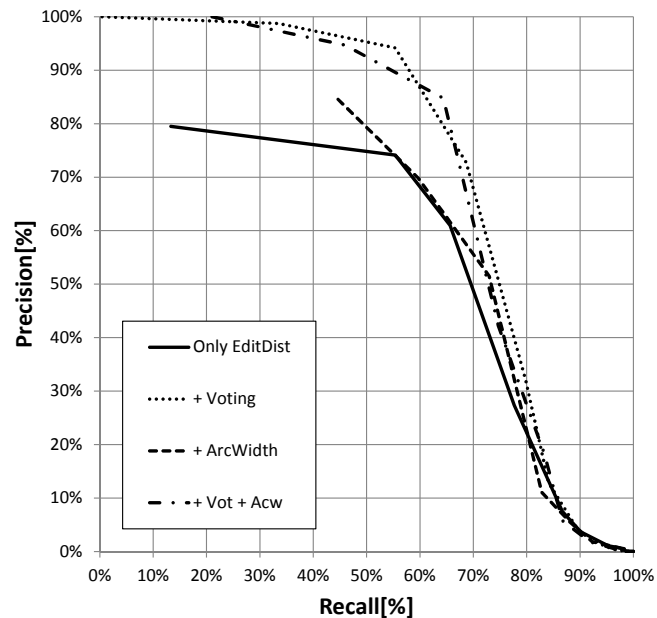


図 5 抑制パラメータを導入した際の Recall-Precision カーブの比較

MAP は Average Precision(AP) を全検索語で平均したものであり、AP は正解出現時の適合率を平均したものである。

3.2 実験結果

誤検出抑制パラメータの組み合わせごとの検索性能を図 5 と表 4 に示す。

図 5 に Recall-Precision カーブを、表 4 に各カーブにおける最大の F-measure と MAP 値を示す。

表 4 図 5 の Recall-Precision カーブ上の最大の F-measure と MAP 値

組み合わせ	F-measure	MAP 値
Only EditDist	0.63	0.80
+ Voting Cost	0.71	0.87
+ ArcWidth Cost	0.64	0.82
+ Vot+Acw Cost	0.73	0.86

表 5 “Only EditDist” における音素数別の STD 性能

閾値	音素数	正解出現数	検出数	正解数	誤検出数	Recall	Precision	F-measure
0.10	10 以上	123	93	83	10	0.68	0.89	0.77
	10 未満	110	158	70	88	0.64	0.44	0.52
	合計	233	251	153	98	0.66	0.61	0.63
0.15	10 以上	123	148	98	50	0.80	0.66	0.72
	10 未満	110	514	83	431	0.76	0.16	0.27
	合計	233	662	181	481	0.78	0.27	0.40

実験結果から示されるように、Voting と Acw を導入したときに最大の F-measure が得られる結果となった。Recall が 85%以上の区間では、抑制パラメータの導入は効果が見られなかったが、それ以下の領域では、同じ Recall を維持しながら 20%以上 Precision を改善することができている。複数の音声認識結果から同じ音素が出力されている場合、その音素は信頼できると考え、その音素の遷移コストを下げることによって、誤検出を抑えられたと考えられる。

一方、表 5 は、編集距離のみを用いた検索エンジンにおいて、クエリを構成する音素数が 10 以上／10 未満でクエリを分類したときの、検索性能を示している。式 (1) を用いて DP スコアを計算したときに、正規化 DP スコアが 0.10、0.15 以下の場合（これは Recall-Precision カーブでの Recall が 0.6～0.8 の F-measure が最も良くなる領域のところの閾値となる）に、クエリが検出されたと判断している。

表 5 より、クエリの音素数が 10 未満になると、誤検出数が明らかに増加していることが分かる。特に検出の閾値を緩くすることで (0.10 → 0.15)、音素数が短いクエリの誤検出が大幅に増加している (88 → 431)。

PTN 型インデックスを用いた場合は、その表現能力の高さが悪影響を及ぼしていると考えられる。音素数が小さいクエリを検索することを考えた場合、そのクエリと全く同じ、もしくは数音素だけが異なる音素列パターンがネットワーク上に存在していると、その部分が誤検出されてしまう。

そこで、音素数に応じて、以下のように DP の遷移コストを増やすことで、誤検出を抑制

することを考えた．

- 挿入・脱落・置換コストを変動させる
- NULL 遷移に対する遷移コストを変動させる

音素数が小さいクエリを検索する場合は，より高い遷移コストを与えることで，完全一致に近い場合に限り検出を許可する．また，NULL 遷移についても，短い音素列のクエリにおいては悪影響が大きいことから，これに対する遷移コストを変動させる．これについては次節で説明する．

STD テストコレクションの未知語セットを用いて予備実験を行った結果，音素数が10未満のクエリの場合に各遷移コストを1.5にセットするのが最も良い結果が得られた．逆に10以上のクエリについては，1.0が最適な遷移コストであることが分かった．

したがって本稿では，音素数が10未満のクエリに対しては，挿入・脱落・置換の遷移コストを1.0から1.5に変更する^{*1}．

4. クエリの音素数に基づいたパラメータ調整

4.1 パラメータの調整

前節で，クエリの音素数によって DP の遷移コストを修正することについて述べた．さらに，誤検出理由について検討してみると，NULL 遷移にも原因があることが分かった．NULL 遷移を低コストで許すことで，PTN はより高い音素列パターンを表現できるが，クエリの音素数が小さい場合には，それが逆に誤検出の原因となりうる例が多い．

そこで，新たに NULL 遷移のコストを設定し直す．具体的には，NULL 遷移にも多数決の要素を含める．すなわち，より多くの認識システムが NULL 判定をすればするほど，その NULL 遷移は信頼性が高いと判定し，NULL 遷移コストを低く設定する．逆に，NULL 判定の認識システムが少なければ，NULL 遷移は信頼できないとし，コストを大きく設定する．

以上をまとめると，クエリの音素数（10 以上／10 未満）に応じて，DP の脱落・挿入・置換の遷移コスト，NULL 遷移を変化させることで，特に音素数が小さいクエリについての検索性能を改善させる．

この処理を導入した用語検索エンジンにおける DP コストの計算式を式 (6) ～式 (12) に示す．

$$D(i, j) = \min \begin{cases} D(i, j-1) + Del \\ D(i-1, j) + Null(i) + Null_V(i) \\ D(i-1, j-1) + Match(i, j) + Vot(i, j) + Acw(i) \end{cases} \quad (6)$$

$$Match(i, j) = \begin{cases} 0.0 : Query(j) \in PTN(i) \\ 1.0 : Query(j) \notin PTN(i), J \geq 10 \\ 1.5 : Query(j) \notin PTN(i), J < 10 \end{cases} \quad (7)$$

$$Del = \begin{cases} 1.0 : J \geq 10 \\ 1.5 : J < 10 \end{cases} \quad (8)$$

$$Null(i) = \begin{cases} 0.1 : NULL \in PTN(i) \\ 1.0 : NULL \notin PTN(i), J \geq 10 \\ 1.5 : NULL \notin PTN(i), J < 10 \end{cases} \quad (9)$$

$$Null_V(i) = \begin{cases} 0.45 \div Voting(NULL) : NULL \in PTN(i), J \geq 10 \\ 0.675 \div Voting(NULL) : NULL \in PTN(i), J < 10 \\ 1.0 : NULL \notin PTN(i), J \geq 10 \\ 1.5 : NULL \notin PTN(i), J < 10 \end{cases} \quad (10)$$

$$Vot(i, j) = \begin{cases} \alpha \div Voting(p) : \exists p \in PTN(i), p = Query(j) \\ 0.0 : Query(j) \notin PTN(i) \end{cases} \quad (11)$$

$$Acw(i) = \beta \times ArcWidth(i) \quad (12)$$

ここで， Del は脱落誤りコスト， $Null(i)$ は NULL 遷移コスト， $Match(i, j)$ は置換誤りコスト， $Vot(i, j)$ および $Acw(i)$ は誤検出抑制パラメータである． J は検索語の音素数を表し， $Query(j)$ は検索語の j 番目の音素を表す． $Voting(NULL)$ は NULL 遷移を認識していた音声認識システムの数を表す．式 (10) において，音素数に応じて 0.45 と 0.675 を設定しているが，これは経験的に決定した．

脱落・置換誤りに対してコストを 1，正解の場合は 0 と設定したときの編集距離を ED_f と示す．

*1 ただし，これらの遷移コストは開発セットを用いて決定したのではなく評価セットによって決められている．そのためクローズドな実験となっている

表 6 探索パラメータの組み合わせ

検索方法	探索パラメータ
Only EditDist	$ED_f + NULL_f$
Voting1	$ED_f + NULL_f + \text{Voting}$
Voting2	$ED_d + NULL_f + \text{Voting}$
Voting3	$ED_d + NULL_d + \text{Voting}$
Vot+Acw1	$ED_f + NULL_f + \text{Voting} + \text{ArcWidth}$
Vot+Acw2	$ED_d + NULL_f + \text{Voting} + \text{ArcWidth}$
Vot+Acw3	$ED_d + NULL_d + \text{Voting} + \text{ArcWidth}$

本研究では、クエリの音素数によって遷移コストを変化させるため、式 (7)、式 (8) に示すように、音素数が 10 未満の場合の脱落・置換誤りコストを 1.5 に設定し、10 以上の場合は 1 に設定する。この場合の編集距離を ED_d と記す。音素数が小さいクエリのコストを大きくすることで、これらのクエリに対する DP 格子状における直線検出をより厳密にし、誤検出を抑える狙いがある。音素長が小さいクエリは、検出閾値を厳しく設定しても誤検出が多いことから、コストを変更することで厳しい検出閾値^{*1}において、STD の性能改善が期待できる。

挿入誤りは NULL 遷移に基づく。先行研究では、NULL 遷移コストは 0.1 に固定していたが（これを $NULL_f$ と示す）、検索語を構成する音素数によってコストを変化させる（ $NULL_d$ と示す）。また、NULL 遷移にも多数決の概念を導入し、複数の音声認識システムの NULL 判定に応じて NULL 遷移コストを変化させる。それを $Null_v(i)$ とする（式 (10)）。なお、式 (6) において、 $Null(i)$ と $Null_v(i)$ は排他的に利用する。すなわち、どちらかのパラメータを利用するともう一方は 0 になる。

誤検出抑制パラメータについては、式 (11)、式 (12) が該当する。式 (11) の α は、音素数が 10 以上の検索語において、5 つ以上の音声認識システムが同じ音素を認識していた場合に、0.1 よりコストを低くするために 0.5 を設定した。 $Voting(p)$ は $Query(j)$ と一致する p を認識していた音声認識システムの数を表す。式 (12) の β は、音素数が 10 以上の検索語において、全ての音声認識システムが異なる音素を認識していた場合を除き、0.1 よりコストが低くなるように 0.01 を設定した。 $ArcWidth(i)$ は $PTN(i)$ のアークの数を表す。

検索性能の比較を行う検索パラメータの組み合わせを表 6 に示す。

4.2 実験結果

図 6 に、Recall-Precision カーブを、表 7 に各グラフ上において最も高い F-measure と

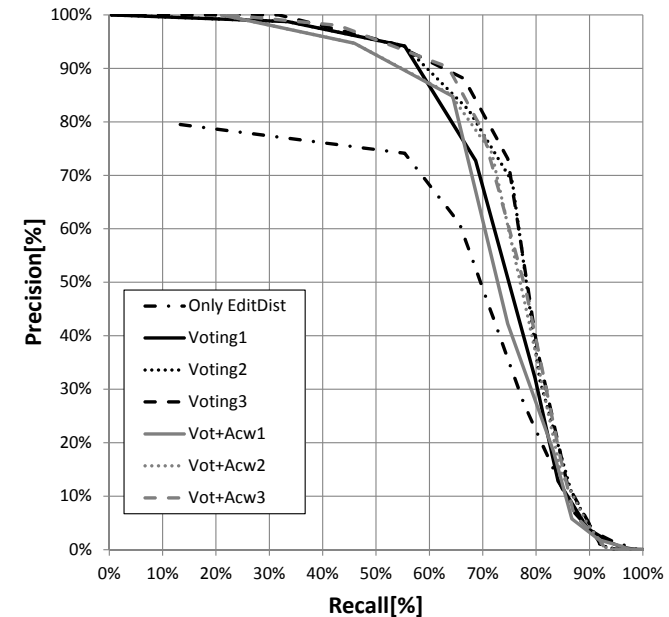


図 6 クエリの音素数に応じたパラメータ調整後の Recall-Precision カーブ

MAP 値を示す。

Only EditDist に比べて Voting1 と Vot+Acw1 の検索性能が高くなっていることから、誤検出抑制パラメータを使用することで、Recall が 80%以下の領域において大幅に検索性能が改善していることが確認できる。また、Voting1 や Vot+Acw1 に比べて Voting2 や Vot+Acw2 の検索性能が高くなっていることから、音素数が 10 未満の検索語に対する編集距離のコストを大きくすることで、湧き出し誤りを抑制し、Recall が 70%前後で検索性能が改善していることが確認できる。さらに、Voting2 や Vot+Acw2 に比べて Voting3 や Vot+Acw3 の検索性能が高くなっていることから、検索性能が改善していることが確認できる。これは、NULL 遷移のコストを Voting によって決定することで、個々の NULL 遷移の信頼度に応じたコストを与えることができたためであると考えられる。また、音素数が 10 未満の検索語に対する NULL 遷移のコストを大きくすることで、F-measure が最大となる閾値を高くすることが可能であり、音素数が 10 以上の検索語における F-measure が

*1 本研究の Recall-Precision カーブでは、Recall が 80%以下の領域にあたる。

表7 図6, 図7, 図8における最大の F-measure と MAP 値

パラメータ	全体		10 音素未満		10 音素以上	
	F-measure	MAP 値	F-measure	MAP 値	F-measure	MAP 値
Only EditDist	0.63	0.80	0.59	0.60	0.77	0.89
Voting1	0.71	0.87	0.70	0.80	0.79	0.90
Voting2	0.74	0.87	0.70	0.80	0.79	0.90
Voting3	0.75	0.87	0.72	0.81	0.78	0.89
Vot+Acw1	0.73	0.86	0.71	0.80	0.79	0.89
Vot+Acw2	0.73	0.86	0.70	0.80	0.79	0.89
Vot+Acw3	0.75	0.87	0.72	0.81	0.77	0.89

最大となる閾値に近づけることができるためであると考えられる。

図7, 図8は, それぞれ音素数が10未満, 10以上のクエリについて求めた Recall-Precision カーブを示す。この結果を見ても分かるように, 音素数が10未満のクエリについて, 誤検出抑制パラメータ (Voting) を導入することによって大幅に改善されている。探索パラメータを調整したり, NULL 遷移のコストを調整したりすることで, さらなる改善が得られている (Voting1 → Voting3 や Vot+Acw3)。

一方で, 音素数が10以上のクエリについても, F-measure 等の性能改善はなされたが, 音素数が少ないクエリに比べると改善の幅は小さい。

以上のことから, 誤検出抑制パラメータを使用することで誤検出が減少し, 検索性能が改善することが示された。また, 音素数が10未満の検索語に対して編集距離のコストを大きくすることで, 検索性能が改善することが示された。さらに, NULL 遷移のコストを Voting によって決定することで, NULL 遷移の信頼度に応じたコストを与えることが可能であり, 検索性能が改善することが示された。このとき, 音素数が10未満の検索語に対する NULL 遷移のコストを上げることで, F-measure が最大となる閾値を高くすることができ, 音素数が10以上の検索語における F-measure が最大となる閾値に近づけることで, 検索性能が改善することが示された。

5. おわりに

本稿では, 我々が以前に提案してきたネットワーク型インデックスに対して, 複数の音声認識システムから得られる情報を利用することで, 誤検出を抑制し STD の性能が向上することを示した。さらに, クエリを構成する音素数に応じて探索のパラメータを調整することで, 特に10未満の音素数を持つクエリに対して検索精度が改善することを示した。

今回は, パラメータの調整に, テストコレクションを用いていたため, オープンな実験

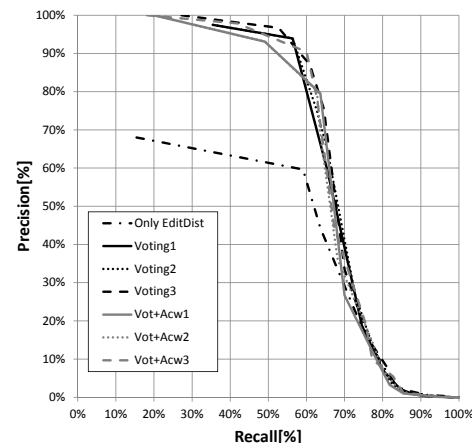


図7 音素数が10未満のクエリに対する Recall-Precision カーブ

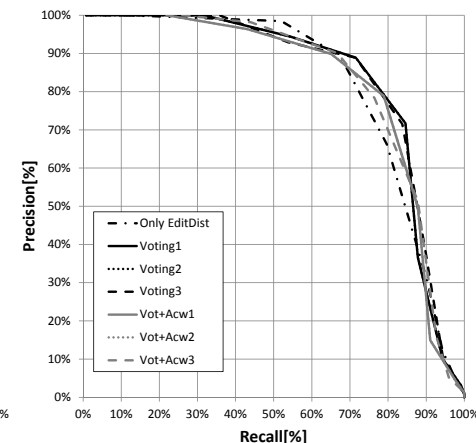


図8 音素数が10以上のクエリに対する Recall-Precision カーブ

とは言えない。今後は NTCIR-9 フォーマルランのテストセットを用いて評価を行いたい。また, 音素数によってクエリを2種類に分けたが, より細かく音素数毎の調整を行ったり, 音素の種類に応じてパラメータを変更することで, さらなる改善を図りたい。

参考文献

- 1) S.Natori, H.Nishizaki, and Y.Sekiguchi, “Japanese Spoken Term Detection Using Syllable Transition Network Derived from Multiple Speech Recognizers’ Outputs,” in *Proc. of the INTERSPEECH 2010*, pp. 681–684, 2010.
- 2) 名取賢, 西崎博光, 関口芳廣, “音声中の検索語検出のための複数の音声認識結果を用いたネットワーク型インデキシング,” 日本音響学会 2010 年秋季講演論文集, pp. 61–64, 2010.
- 3) H.Nishizaki et al., “Spoken Term Detection Using Multiple Speech Recognizers’ Outputs at NTCIR-9 SpokenDoc STD subtask,” in *Proc. of the NTCIR-9 meeting*, pp.236–241, 2011.
- 4) Y.Itoh et al., “Constructing Japanese Test Collections for Spoken Term Detection,” in *Proc. of the INTERSPEECH 2010*, pp. 677–680, 2010.
- 5) T.Akiba et al., “Overview of the IR for Spoken Documents Task in NTCIR-9 Workshop,” in *Proc. of the NTCIR-9 meeting*, pp.223–235, 2011.