Vol.2014-SLP-102 No.7
2014/7/25

DNN の出力確率を用いた STD のリスコアリング方式

紺野良太[†1]　李時旭[†2]　田中和世[†3]
小嶋和徳[†1]　石亀昌明[†1]　伊藤慶明[†1]

近年，ビデオデータから特定のシーンを検索する機能に対するニーズが高まっており，音声中の検索語検出（STD：Spoken Term Detection）の研究が盛んに行われている．本稿では，従来の STD を行った後，検索結果上位候補を DNN（Deep Neural Network）の出力確率を用いて検索語と照合するリスコアリング方式を提案する．NTCIR-9, 10 の Formal run，Dry run 計 4 種のテストセットを用いた評価実験の結果，リランキング発話数 K を 50 件とした場合，検索精度を表す MAP が 4.11pt～11.61pt 向上し，処理時間はフレーム単位照合で約 0.17 秒，状態単位照合で平均約 0.10 秒と，実用可能な処理時間で検索精度の向上を実現できた．さらに，リランキング対象発話数の増加に伴い，検索精度が向上することも確認できた．K を 2000 件とした場合の MAP は 9.48pt～28.04pt 向上し，検索時間はフレーム単位照合で約 7.24 秒，状態単位照合で約 4.12 秒となった．また，状態単位照合方式は，フレーム単位照合とほぼ同等の検索精度で検索時間を約 1.73 倍高速化できた．以上のように，実用的な処理時間で検索精度向上を実現し本手法の有効性を確認できた．

A rescoring method for STD using output probability of DNN

RYOTA KONNO[†1]　SHI-WOOK LEE[†2]　KAZUYO TANAKA[†3]
KAZUNORI KOJIMA[†1]　MASAAKI ISHIGAME[†1]　and　YOSHIAKI ITOH[†1]

This paper proposes a rescoring method for Spoken Term Detection (STD) using output probability of Deep Neural Network. The experimental results demonstrated the proposed method works well for open test collections that were distributed from National Institute of Informatics (NII) for STD evaluation.

1. はじめに

近年, HDD やブルーレイディスク等の大容量記憶デバイスや WEB 上での動画投稿サイトの利用が一般的となっており，ビデオデータの利用機会が増加している．これに伴い，大量のビデオデータから特定のシーンを検索する機能に対するニーズが高まっており，今後も記憶デバイスの大容量化と共にこのニーズは高まり続けると考えられる．この機能の実現のため，ビデオデータ内の音声情報を用いてクエリを検索する, 音声中の検索語検出（STD：Spoken Term Detection）の研究が盛んに行われている．国立情報学研究所が主催する NTCIR Workshop 9[1]が 2011 年，NTCIR Workshop 10[2]が 2013 年に開催され，STD についての方式が様々な観点から評価された．また，2014 年に NTCIR Workshop 11[3]が開催予定である．

STD とは，音声ドキュメント内で一語以上からなる検索語（クエリ）が話されている位置を特定するタスクである．STD システムの実現方法としては，大語彙連続音声認識システムを用いて音声ドキュメントを予め音声認識しておき，その認識結果を用いて検索を行う方法が一般的である．この場合，単語認識結果を用いるため，クエリが単語辞書に登録されていない未知語であると，未知語の区間は誤認識となるので検索精度が著しく低下してしまう．この未知語のクエリの問題に対応するため，サブワード単位での認識結果を用いて照合を行う方式が一般的である. 一方, 近年, DNN（Deep Neural Network）を用いた音声認識によって認識率が大幅に改善されることが数多く報告されている[4, 5, 6]．DNN とは，多層から成るニューラルネットワークで，従来，多層のニューラルネットワークの学習は困難とされてきたが，事前学習方法の確立等によって学習が出来るようになった[7]．現在では GMM（Gaussian Mixture Model）ではなく DNN を用いて出力確率を推定する音声認識方式が認識率で有利と見られており，本稿では，サブワード認識に DNN を用いるだけでなく，以下に述べるように STD のリスコアリングに DNN を用い, 検索精度の向上を図る．

サブワードベースの STD システムでは, 検索対象の音声ドキュメント群をサブワード認識し，検索対象をサブワード系列として保存しておく．システムにクエリが与えられると，クエリをサブワード系列に変換し，検索対象のサブワード系列と連続 DP（Dynamic Programming）で照合を行う．連続 DP における局所距離は，edit distance が代表的であるが, サブワード間の弁別素性[8]や音響距離[9]を導入することにより, edit distance に比べ高い性能を得られることが報告されており，本稿でも検索精度を向上させるため，サブワード間音響距離を用いる. サブワード間音響距離は，DNN では容易に求めることはできないため，GMM の統計量から求めたものを用いる．

STD において，クエリが与えられてからすべての音声信

†1 岩手県立大学
Iwate Prefectural University.
†2 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology
†3 筑波大学
University of Tsukuba

号と照合することは検索の待ち時間を考えると実用的ではない．このため，音声信号はサブワードに記号化し照合が行われる．サブワード間音響距離は，この記号化された 2 つのサブワード間の距離で音声信号によらず一定の値となる．一方, GMM や DNN から出力される確率は音声信号（フレーム単位の特徴量）によってそれぞれ異なり，精度の高い確率が推定されるため，サブワード間音響距離と比べより高い精度でクエリのサブワード系列と音声信号との比較が行える．

我々の先行研究では，上位の候補に対して発話を再認識することで，クエリのサブワード系列と任意のサブワード系列との尤度比較で受理・棄却を判定し，検索精度を向上させた[10]．本研究は，DNN の出力確率を用いてクエリに対するスコアを求め，上位 K 件を再度順位付けし検索精度の向上を図るものである．

本稿では, 第 2 章で一般的な STD システムについて説明し，次に提案手法について述べる．第 3 章では，提案手法の評価実験を行い，提案手法の有効性を示す．

2. 提案手法

2.1 提案手法の概要

本節では提案手法の概要について述べる．提案手法では, DNN の出力確率を用いて連続 DP を行い精度の向上を図る．DNN の確率計算は，GPU（Graphics Processing Unit）を用いることで高速化が可能であるが，検索対象のドキュメント群すべてに対し，フレーム単位の DNN の出力確率を用いて連続 DP 照合でリスコアリングすると，CSJ（Corpus of Spontaneous Japanese）[11]の 177 講演，約 44 時間の検索対象でも 1 クエリ当たり約 3 分を要し，実用的な検索時間ではない．そこで，従来の STD システムを用いて検索した結果に対し，検索システムでは重要度が高い上位候補に対してのみリスコアリングを行い，検索精度の向上を図ると共に計算時間を削減する．提案方式は以下の処理ステップからなる．①，②については 2.2，2.3 で詳述する．

① 従来の STD システムによってサブワードベースの連続 DP 照合を行い，スコア順にソートする．
② スコアの高い上位 K 件の発話を対象に, DNN の出力確率を用いてフレーム単位あるいは状態単位の詳細な連続 DP 照合を行いリスコアリングする．
③ リスコアリングの結果を上位からユーザに提示する．

2.2 サブワード間音響距離を用いた STD システム

一般的な STD システムでは, 大語彙連続音声認識システムを用いて予め音声認識を行い，その認識結果を用いて検索を行う．クエリとして入力される単語は人名や固有名詞など音声認識システムで利用する単語辞書に含まれていない未知語の場合が多い．そのため，単語認識結果を用いるとクエリが未知語だった場合，検索精度が低下する．そこで，単語よりも小さい monophone や triphone 等のサブワードを単位として音声認識を行い，その結果を用いることで未知語のクエリに対しても検索可能なシステムを実現する．

検索対象の音声ドキュメント群は通常ポーズによりセグメンテーションを行い発話毎に分割しておき，サブワードベースの STD システムではこの発話毎にサブワード認識を行う．認識結果のサブワード系列を検索対象として保存しておく．提案手法では，サブワード認識の際，強制アライメントを行い各サブワードと特徴量フレームとの対応を求め，各サブワードの開始・終了フレーム番号を保持しておく必要がある．

システムにクエリが与えられると，クエリを音節列からサブワード系列に変換し，サブワード系列で保存しておいた検索対象データとクエリのサブワード系列を連続 DP により照合する．連続 DP を行う際の局所距離には，我々はサブワード間音響距離を用いている[9].

サブワード間音響距離は，サブワードを構成する HMM（Hidden Markov Model）の統計量を用いて求める．サブワード HMM は複数の状態から構成されており, まず初めに, i番目の状態間の距離を求める．dを特徴量の次元数，s,tを 2 つのサブワードのi番目の状態，m,nをそれぞれs,tに含まれる分布とし，状態s，分布m，次元dの平均，分散をμ_{smd}，$\sigma^2{}_{smd}$とすると，状態sのm番目の分布と状態tのn番目の分布間距離は，Bhattacharyya 距離（BD）により以下の式で求められる．

$$BD(s_m, t_n) = \frac{1}{4}\sum_{d=1}^{Dim}\left\{\frac{(\mu_{smd}-\mu_{tnd})^2}{\sigma_{smd}^2+\sigma_{tnd}^2}+\log\frac{(\sigma_{smd}^2+\sigma_{tnd}^2)^2}{4\sigma_{smd}^2\sigma_{tnd}^2}\right\} \quad (1)$$

(2)式のように，この分布間距離をi番目の 2 つの状態におけるあらゆる分布間で求め，最小の分布間距離を状態間距離（SD_i）とする．

$$SD_i = \min_{m,n} BD(s_m, t_n) \quad (2)$$

状態間距離をサブワード HMM の同一順番の 2 つの状態間で求め，M個の状態の平均値をサブワード間音響距離（AD）とする．

$$\mathrm{AD} = \frac{1}{M}\sum_{i=1}^{M} SD_i \quad (3)$$

以上のように作成したサブワード間音響距離を用いて連続 DP を行い，検索結果を取得する．このとき，発話内で最もスコアが高くなるサブワード系列を求め，その区間をフレーム番号で保持しておく．フレーム番号は，サブワード認識の際に行った強制アライメントの結果より取得する．次節で述べるように，この情報は，DNN の出力確率を用いたリスコアリングの際に，この区間のみを計算することで計算時間を削減するために用いる．

2.3 DNN の出力確率を用いたリスコアリング

提案手法では，DNN の出力確率を用いて連続 DP 照合する．フレーム単位で全発話を対象として照合すると計算時間を要し実現困難である．そこで，2.2 の STD システムによって得られる上位候補のみを対象に提案手法を適用する．

2.3.1 DNN の概要

DNN とは, 多層から成るニューラルネットワークである．近年, DNN を用いることで音声認識や画像認識の認識率が大幅に改善されることが報告され，大きな注目を集めている．従来，多層のニューラルネットワークは学習が困難とされてきたが，現在では，教師あり学習（fine-tuning）の前に，各層を RBM（Restricted Boltzmann Machine）として教師なし学習（pre-training）を行うことで，深い層を持つニューラルネットワークでも学習が可能となった[7].

音声認識で DNN を用いるためのアプローチの一つに DNN と HMM を組み合わせて用いるハイブリッド型モデルの DNN-HMM がある．DNN-HMM では，入力を MFCC 等の特徴量，出力を HMM の各状態の出力確率とする．入力する特徴量は前後数フレームを追加し，より高次元の特徴量として入力する場合が多い．出力層における各出力ノードは，予め HMM の各状態と対応付けられており，1 フレームの入力が与えられると，各出力ノードから出力確率が得られ，その確率が HMM の各状態の出力確率として利用される．

2.3.2 DNN の出力確率を用いた DP 照合方式

本節では，DNN の出力確率を用いた DP 照合方式について述べる．本手法は以下 2 つの方式により計算時間を削減する．

① 従来のサブワード系列間で照合を行う STD システムによって得られた検索結果の上位候補のみに対して提案手法を適用する．

② 従来のサブワード系列間で照合を行う STD において，連続 DP 照合の際に，発話内で最もスコアが高い区間の開始・終了フレーム番号を保持しておき，その範囲のみに DNN を用いた連続 DP 照合をする．（音声情報の欠落を防ぐため，前後 10 フレームを追加する．）

従来の STD システムでは，クエリ，検索対象ドキュメント共にサブワード単位での照合を行っていた．しかし，DNN に対しては特徴量をフレーム単位で入力するため，DNN からはフレーム単位で出力確率が得られる. クエリのサブワード系列と連続 DP 照合するためには，以下 2 通りの方式でクエリのサブワード系列と DNN の出力系列を一致させる必要がある．

① クエリをフレーム単位の系列に変換する．

② フレーム単位で得られる出力確率を状態単位やサブワード単位に変換する．

方式①では，クエリをサブワード系列からフレーム系列に変換し, フレーム単位での照合を行うことになる. まず，学習データから状態毎の平均継続フレーム数を事前に求めておき，連続 DP の際に平均継続フレーム数分，クエリ方向に確率を積み上げる．図 1 にその例を示す．P_{ij}は，i番目のフレームにおける状態s_jの確率を表す．状態s_a，s_bの平均継続フレーム数はそれぞれ 2, 3 であったとすると，その平均継続フレーム数分，図のように出力確率を縦軸方向に積み上げる．

方式②では，フレーム単位で得られる出力確率を状態単位に変換し，状態単位での照合を行う．各状態における継続フレーム数を学習データより求めると，平均約 2.7 フレームとなった. DNN の出力確率を 3 フレーム毎に計算すれば，フレーム単位で得られる出力確率がおおよそ状態単位に相当する. DNN の出力確率を 3 フレーム毎に計算する方法を採用し，本稿ではこれを状態単位での照合と呼ぶ．図 2 に例を示す. 縦軸には, クエリの状態系列が並んでおり，横軸は，3 フレーム毎に入力フレームが与えられる．この方法では, DNN の出力確率計算時間が 3 分の 1 に削減される．

DP パスは，図 3 の制約のない DP パスと，図 4 の縦横 2 倍の伸縮の制約がある DP パスの 2 種類を検証したが，検索精度に差が見られなかったため，以降の実験では図 3 の制約のない DP パスを用いる．

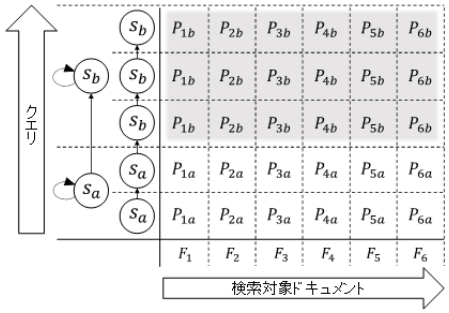


図 1　フレーム単位での照合

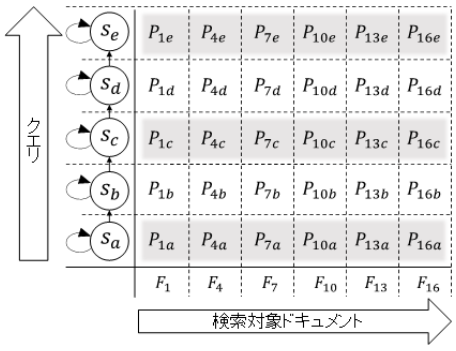
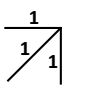


図 2　状態単位での照合



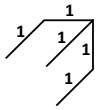


図 3　制約なし DP パス　図 4　制約あり DP パス

3. 評価実験

3.1 実験条件

音響モデル，言語モデルの学習は，CSJ の学会講演と模擬講演を合わせた2702講演から評価に用いる177講演を除いた 2525 講演のうち，偶数講演（1255 講演，約 287 時間）を使用した．

音響モデルは 3 状態の triphone で構成した．DNN のベースとなる GMM の音響モデルは HTK（Hidden Markov Model Toolkit）[12]を使用して作成した．状態数は状態共有を行い 3009 状態とした．GMM の混合数は 32 混合とした．また，入力特徴量の抽出条件は表 1 の通りである．

DNN は，隠れ層 3 層とし，入力特徴量は，39 次元の特徴量に前後 5 フレームを追加し，429 次元（39 次元×11 フレーム）とした. 出力層は, GMM の状態数と同じ 3009 状態とした．DNN の fine-tuning に使用する正解ラベルは，GMM を用いて強制アライメントを行うことで作成した．DNN の学習に使用するプログラムは Python 用ライブラリ Theano[13]を使用して作成した．DNN の学習パラメータは表 2 の通りである．

言語モデルは Palmkit[14]を使用し，音節単位の前向き 2-gram と後ろ向き 3-gram の言語モデルを構築した．認識は，大語彙連続音声認識エンジン Julius[15]を使用した．

CPU は Intel Core i7-4770，GPU は NVIDIA GeForce GTX TITAN，メモリは 16GB のマシンを使用して処理時間を測定した．

表 1　特徴量抽出条件

デジタル化	標本化周波数 16kHz / 量子化 bit 数 16bit
特徴量	MFCC (12dim) + ΔMFCC (12dim) + ΔΔMFCC (12dim) + Power + ΔPower + ΔΔPower (計 39 次元)
窓長	25 msec
フレームシフト	10 msec
窓関数	ハミング窓

表 2　DNN の学習パラメータ

ノード数		入力層 429，隠れ層 2048，出力層 3009
隠れ層数		3
RBM	学習係数	0.004
	モメンタム	0.9
	ミニバッチサイズ	256
	エポック数	10
DNN	学習係数	0.007（前エポックより認識率が下がった場合半減）
	ミニバッチサイズ	256
	エポック数	30

3.2 テストセット

評価には，表 3 で示す NTCIR-9，NTCIR-10 で用いられテストセットを使用した．NTCIR-9 では，CSJ のコア 177 講演（約 44 時間，53,892 発話），NTCIR-10 では，音声ドキュメントワークショップの講演音声（SDPWS：Corpus of Spoken Document Processing Workshop）（約 29 時間，40,796 発話）が検索対象ドキュメントとして用いられた．クエリと正解情報は，NTCIR オーガナイザから提供された NTCIR-9 SpokenDoc の STD タスクにおける Formal run と Dry run 及び NTCIR-10 SpokenDoc の STD タスクにおける Formal run と Dry run の合計 4 つのテストセットを用いた．

表 3　テストセット

	NTCIR-9	NTCIR-10
検索対象データ	CSJ177 講演（約 44 時間，53,892 発話）	SDPWS104 講演（約 29 時間，40,746 発話）
クエリ	Formal run 50 クエリ Dry run 50 クエリ	Formal run 100 クエリ Dry run 32 クエリ

3.3 評価指標

正解の判定は NTCIR で用いられた方法と同様に発話単位で行い，クエリが発話内のどこかで一度以上話されていればその発話を正解とした．検索精度の評価には MAP（Mean Average Precision）を用いた. AP（Average Precision）は検索結果を上位から出力していき，正解が出力された時の適合率を平均したものである．各クエリで AP を求め，それらを全クエリで平均したものが MAP である. AP, MAP はそれぞれ，式(4)，(5)で求められる．

$$AP(q) = \frac{1}{c_q}\sum_{i=1}^{N} \delta_i \times precision(q,i) \tag{4}$$

$$MAP = \frac{1}{Q}\sum_{q=1}^{Q} AP(q) \tag{5}$$

クエリqに対する正解発話数をc_q，Nは検索対象発話数，δ_iはバイナリ関数で，検索結果のi番目の発話が正解なら 1，不正解なら 0 となる．$precision(q,i)$はクエリqのi番目の検索結果出力時点での適合率である．Qはクエリ数である．

処理時間は 3.1 で述べたマシン上で，C 言語の gettimeofday 関数で計測した．

3.4 実験結果と考察

本節では提案手法の評価を行う．NTCIR-9，NTCIR-10 の Formal run と Dry run 計 4 種のテストセットで，2.2 で述べた従来までのサブワード間照合と，2.3 で述べた DNN の出力確率を用いたリスコアリング方式について，検索精度と処理時間を求めた．リスコアリングの対象とする検索結

表 4　NTCIR-9 Formal run の結果

	フレーム単位照合			状態単位照合		
	MAP	up (point)	Time (sec)	MAP	up (point)	Time (sec)
HMM 間照合	84.60	+0.00	0.54	84.60	+0.00	0.54
K=50	91.27	+6.67	0.17	91.03	+6.43	0.10
K=100	92.91	+8.31	0.35	92.68	+8.07	0.20
K=500	93.85	+9.25	1.71	93.60	+9.00	0.99
K=1000	94.29	+9.68	3.51	94.11	+9.50	2.02
K=2000	94.26	+9.66	7.12	94.08	+9.48	4.10
K=ALL	95.22	+10.62	181.07	94.94	+10.34	113.75

表 5　NTCIR-9 Dry run の結果

	フレーム単位照合			状態単位照合		
	MAP	up (point)	Time (sec)	MAP	up (point)	Time (sec)
HMM 間照合	76.20	+0.00	0.53	76.20	+0.00	0.53
K=50	80.66	+4.45	0.18	80.31	+4.11	0.10
K=100	82.88	+6.67	0.35	82.58	+6.38	0.20
K=500	85.80	+9.59	1.71	85.27	+9.07	1.00
K=1000	87.40	+11.20	3.51	86.78	+10.57	2.03
K=2000	87.74	+11.54	7.09	87.08	+10.88	4.16
K=ALL	87.13	+10.92	181.63	85.67	+9.47	114.66

表 6　NTCIR-10 Formal run の結果

	フレーム単位照合			状態単位照合		
	MAP	up (point)	Time (sec)	MAP	up (point)	Time (sec)
HMM 間照合	49.63	+0.00	0.49	49.63	+0.00	0.49
K=50	60.67	+11.04	0.17	61.24	+11.61	0.10
K=100	62.70	+13.06	0.33	63.72	+14.09	0.20
K=500	67.88	+18.25	1.73	68.89	+19.26	0.98
K=1000	69.43	+19.80	3.53	70.46	+20.82	2.01
K=2000	70.68	+21.05	7.12	71.75	+22.12	4.04
K=ALL	73.20	+23.57	141.62	74.28	+24.64	87.35

表 7　NTCIR-10 Dry run の結果

	フレーム単位照合			状態単位照合		
	MAP	up (point)	Time (sec)	MAP	up (point)	Time (sec)
HMM 間照合	56.00	+0.00	0.56	56.00	+0.00	0.56
K=50	66.30	+10.31	0.18	66.28	+10.28	0.10
K=100	67.95	+11.96	0.36	67.89	+11.89	0.20
K=500	77.86	+21.86	1.85	77.76	+21.76	1.01
K=1000	80.34	+24.34	3.71	80.25	+24.25	2.02
K=2000	83.95	+27.95	7.63	84.04	+28.04	4.17
K=ALL	87.29	+31.29	153.20	87.08	+31.09	89.58

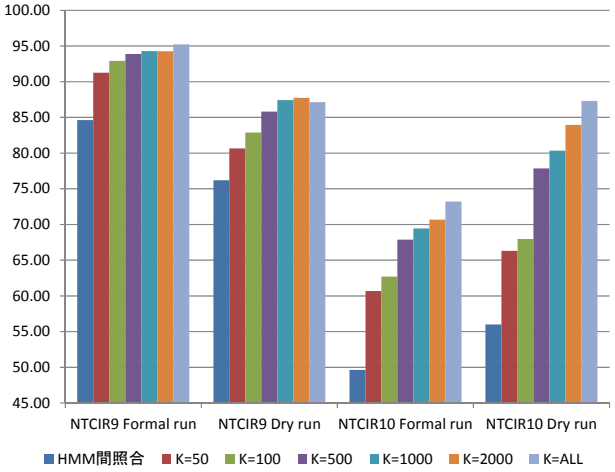


図 5　フレーム単位照合の MAP

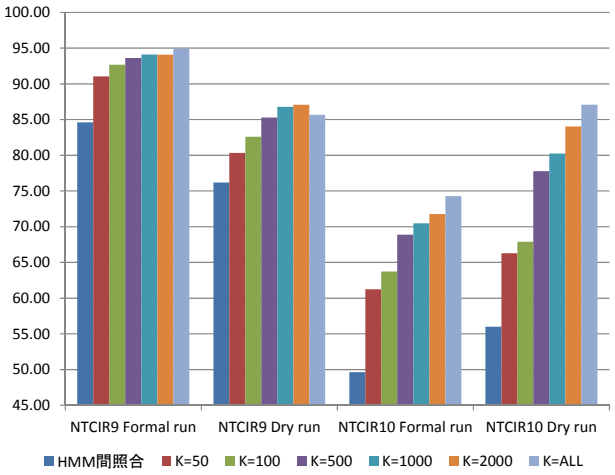


図 6　状態単位照合の MAP

果は上位 K 件とするが，K=50，100，500，1000，2000，ALL（全発話）の場合について実験を行った．4 種のテストセットに対する実験結果を表 4～7 に，表 4～7 のフレーム単位照合の検索精度を図 5 の棒グラフで，状態単位照合の検索精度を図 6 の棒グラフで示す．表中の網掛けは，各評価における最良の検索精度向上幅を示す．

4 つのテストセットのすべての場合で，リスコアリング対象件数 K が 1000 までは，K の増加に伴い検索精度が向上した．NTCIR-9 Formal run で K=2000 とした場合や NTCIR-9 Dry run で K=ALL とした場合に検索精度が若干低下する場合があった．検索精度向上の最大は，表中に示したように NTCIR9 Formal run で 10.62pt, Dry run で 11.54pt, NTCIR10 Formal run で 24.64pt, Dry run で 31.29pt となった．

検索時間はフレーム単位照合，状態単位照合共に K の値によって，想定通り線形に増加した．状態単位照合は，フレーム単位照合に比べフレームレベルで1/3となるが，処理時間には DNN の出力確率計算以外の処理も含まれるため，トータルでは約 1.73 倍の高速化となった．

K=50 のように，リスコアリング対象発話数が少なくても 4.11p～11.61pt 検索精度が向上し，処理時間もフレーム単位照合で約 0.17 秒，状態単位照合で約 0.10 秒と，実用可能な処理時間で検索精度の向上を実現できた．K の増加に伴い，検索精度が向上することも確認できた．K=2000 のとき，9.48pt～28.04pt 検索精度が向上し，検索時間はフレーム単位照合で約 7.24 秒，状態単位照合で約 4.12 秒となった．K=50 と比べ，検索精度は大きく向上したが，検索時間を要するため検索時間の短縮方式が重要な課題と考える．NTCIR-9 Formal run の K=2000 や NTCIR-9 Dry run の K=ALL など, K の増加前と比べ検索精度が低下する原因の解明と対策についても今後の課題としたい．

状態単位照合方式は，フレーム単位照合と検索精度で比べると K=50 の場合, NTCIR-9 Formal run で-0.24pt, NTCIR-9 Dry run で-0.34pt, NTCIR-10 Formal run で+0.58pt, NTCIR-10 Dry run で-0.02pt と，ほぼ同等の検索精度で検索時間を約 1.75 倍高速化できた．本手法では，単語やサブワードを単位とした音声認識結果を用いていない．このため，従来の STD システムのように音声認識システムの誤認識に直接影響を受けない．

本手法では，リスコアリング対象発話をある程度絞り込むことができれば良いため，表 8 に示すように，従来手法である edit distance で候補を絞った後で適用しても精度向上を確認できた．一方，表 4～7 の検索精度の方が明らかに高く，サブワード間音響距離の有効性が顕著に現れた．

以上のように，本手法により検索精度向上を実現し本手法の有効性を確認できた．

表 8　edit distance を用いた検索結果への本手法の適用

	NTCIR-9 Formal run	NTCIR-9 Dry run	NTCIR-10 Formal run	NTCIR-10 Dry run
HMM 間照合	68.55	68.09	35.82	43.67
K=50	78.89	73.42	48.21	53.66
K=100	80.70	76.09	51.88	59.05
K=500	88.05	82.89	58.54	67.33
K=1000	89.53	83.52	61.62	69.94
K=2000	90.04	85.28	63.40	74.40

4. おわりに

本稿では，従来の STD を行った後，上位候補を DNN の出力確率を用いて照合するリスコアリング方式を提案し，フレーム単位，状態単位で照合を行う手法の有効性を検証した．提案手法によって，実用可能な処理時間で検索精度の向上を実現できた．

今後は，K を大きくした場合の検索時間の削減を図る必要がある．また，リスコアリング手法の改善や拡張を行いさらなる精度向上を目指していきたい．

謝辞　本研究の一部は文部科学省科学研究費補助金基盤(C)No.24500124 を受けて実施された．

参考文献

[1] Tomoyosi Akiba, Hiromitsu Nishizaki, Kiyoaki Aikawa, Tatsuya Kawahara, Tomoko Matsui, Overview of the IR for Spoken Documents Task in NTCIR-9 Workshop, Proceedings of NTCIR-9 Workshop Meeting, pp. 223-235 (2011).
[2] Tomoyosi Akiba, Hiromitsu Nishizaki, Kiyoaki Aikawa, Xinhui Hu, Yoshiaki Itoh, Tatsuya Kawahara, Seiichi Nakagawa, Hiroaki Nanjo, Yoichi Yamashita, Overview of the NTCIR-10 SpokenDoc-2 Task, Proceedings of the 10th NTCIR Conference, pp. 573-587 (2013).
[3] National Institute of Informatics, NTCIR-11, http://research.nii.ac.jp/ntcir/ntcir-11/index.html
[4] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, Brian Kingsbury, Deep Neural Networks for Acoustic Modeling in Speech Recognition, IEEE Signal Processing Magazine, Vol. 29, No. 6, pp. 82-97 (2012).
[5] 三村正人，河原達也，CSJ を用いた日本語講演音声認識への DNN-HMM の適用と話者適応の検討, 情報処理学会研究報告, Vol. 2013-SLP-97，No. 9，pp. 1-6 (2013).
[6] 西野大輔，篠田浩一，古井貞熙，ディープラーニングを用いた日本語大語彙話し言葉音声認識，日本音響学会 2012 年秋季研究発表会講演論文集，No. 2-1-7，pp. 71-72 (2012).
[7] Geoffrey E. Hinton, Simon Osindero, Yee-Whye Teh, A Fast Learning Algorithm for Deep Belief Nets, Neural Computation, Vol. 18, pp. 1527-1554 (2006).
[8] 桂田浩一，入部百合絵，新田恒雄，Suffix Array を用いた高速 STD におけるキーワード分割に関する理論的検討，情報処理学会研究報告，Vol.2011-SLP-89，No.16，pp. 1-6 (2011).
[9] 岩田耕平，伊藤慶明，小嶋和徳，石亀昌明，田中和世，李時旭，語彙フリー音声文書検索手法における新しいサブワードモデルとサブワード音響距離の有効性の検証，情報通信学会論文誌，Vol. 8，No. 5， pp. 1990-2000 (2007).
[10] 大竹隆之他，スポッティング区間の再認識に基づく音声検索性能の向上，日本音響学会，日本音響学会講演論文集，pp. 23-24 (2006).
[11] Corpus of Spontaneous Japanese, http://www.ninjal.ac.jp/corpus_center/csj/
[12] Hidden Markov Model Toolkit, http://htk.eng.cam.ac.uk/
[13] Theano, http://deeplearning.net/software/theano/
[14] Palmkit, http://palmkit.sourceforge.net/
[15] 大語彙連続音声認識エンジン Julius, http://julius.sourceforge.jp/