Vol.2012-SLP-94 No.24
2012/12/21

音声中の検索語検出における 音素トライグラム照合による高速抽出法

鎌田圭祐[†] 斉藤裕之[†] 伊藤慶明[†] 小嶋和徳[†] 石亀昌明[†]
田中和世[†2] 李時旭[†3]

音声中の検索語検出(STD : Spoken Term Detection)において，音素トライグラムを利用した STD の高速化方式を提案する．提案方式では，音声ドキュメントを予め音素認識あるいは音節認識を行っておき、得られた認識結果から 3 音素単位で音素トライグラムとその出現位置を転置インデックスとして保持しておく．クエリが与えられると，クエリの音素列を，1 音素ずつシフトさせながらクエリの音素列を分割することで，トライグラム群を作成する．クエリの各トライグラムを事前に作成した転置インデックスとの照合を行い候補区間を抽出する．抽出された候補区間にのみ連続 DP による精度の高いスコアリングを行うことで検索時間の短縮を図る．MAP による評価を行った実験において，検索性能低下なしで検索時間 86.5%削減し，2.21 秒にすることに成功した．検索結果上位 K 件の正解数による評価を行った実験において上位 1,3,5,10 位いずれについても全ての音声ドキュメントに対して連続 DP による検索を行った時と同性能を 1 秒未満で検索することができた．

Fast spoken term detection by phone trigram matching

KEISUKE KAMATA[†1] HIROYUKI SAITO[†1] YOSHIAKI ITOH[†1]
KAZUNORI KOJIMA[†1] MASAAKI ISHIGAME[†1]
KAZUYO TANAKA[†2] and SHI-WOOK LEE[†3]

We have been conducting a research for Spoken Term Detection (STD), which identifies the target section where query terms are spoken in spoken documents. In STD, Out-Of-Vocabulary (OOV) query terms are one of the most important problems because OOV terms are not correctly recognized by using an automatic speech recognizer and are likely to be query terms. We have proposed a subword based STD system to deal with OOV query terms, where all spoken documents are searched for after query terms are given. It leads to the linear increase of search time according to the amount of spoken documents. The paper proposes a new method for fast STD by using phone trigram.

1. はじめに

近年のマルチメディアデータ特にビデオデータの増加に伴い，ハードディスクレコーダ等の大容量記憶媒体が広く普及している．このような大量のビデオデータを有効に活用するためには，効率的な検索が必要不可欠である．その実現に向け，現在音声中の検索語検出（STD: Spoken Term Detection）の研究が盛んに行われるようになった．米国の NIST による TREC では評価型ワークショップが行われており，2011 年には国立情報学研究所が主催する NTCIR Workshop 9 が日本で開催され，Spoken Doc Task[1]として STD の評価が行われた．2012 年にも NTCIR10 としてさらなる研究の深化が図られている．STD とは，音声ドキュメント中でクエリ（検索語）が発話されている位置を特定することであり，クエリが辞書に登録されている既知語ならば単語認識結果を用いて検索を行えば良いが，クエリが未知語である場合は単語認識では誤認識となり正しい検索は困難であることから，サブワード認識結果を用いて，クエリのサブワード系列と照合する方式が一般的となってきている．STD では辞書に登録されていない未知語の検索が重要であり，本提案方式も未知語を検索するためのサブワード認識に基づく STD システムをベースとしている．

我々のベースとする STD システムでは，音声ドキュメント群を予めサブワードで音声認識しておき，テキストで与えられたクエリをサブワード系列に変換し，サブワード系列の検索対象データと連続動的計画法（連続 DP: Continuous Dynamic Programming）等で照合を行う．このシステムでは，日本語話し言葉コーパス(CSJ: the Corpus of Spontaneous Japanese)177 講演に対する 1 クエリあたりの検索時間は 1.01 秒，全 2702 講演に対する 1 クエリあたりの検索時間は約 16 秒であった．

このように我々が現在行っている連続 DP による音声ドキュメント全体との照合方法では，検索時間は検索対象の音声ドキュメント群のデータ量に比例して増加してしまう．そこで，本稿では STD の高速化のため認識結果から作成した音素トライグラムの転置インデックスを利用した方式を

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

提案する．本提案方式では，事前に音素認識あるいは音節認識を行って得られた認識結果から音素トライグラムとその出現位置を転置インデックスとして抽出し，保持しておく．クエリが与えられると，クエリを 3 音素単位で 1 音素ずつシフトさせながら分割し，（クエリの音素数－2）個の音素トライグラムを抽出する．各音素トライグラムについて転置インデックスとの照合を行い，クエリを含む可能性の高い候補区間を抽出し，抽出された候補区間のみに連続 DP 照合を行うことで全ての音声ドキュメントを連続 DP で検索した場合と比較して検索性能の低下を抑えながら大幅な検索時間の短縮を図る．本提案方式は，高速化の利点以外に構造がシンプルなため，構築が容易であり，そのインデックスもコンパクトな点に特長がある．

本稿では，まず 2 章で我々の STD 手法について概説し．次に提案方式を示す．3 章で，提案方式の有効性を実験により示した後，実システムにて用いる際を考慮し検索結果の上位における正解数で評価を行い，区間が含まれている率を調査し，本提案方式の有効性を示す．

2. 提案方式

2.1 システム概要

本節では，我々が提案している主に未知語のクエリが与えられた場合のサブワードベースの STD システムについて概説する．Fig.1 に我々のこれまでの STD システムの概要を示す．

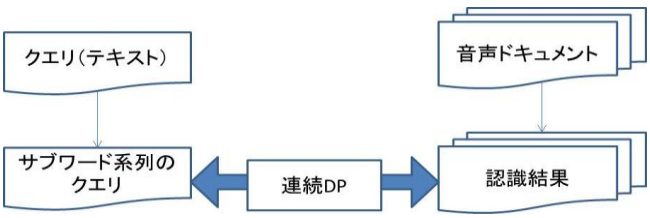


Fig.1. STD System　概要図

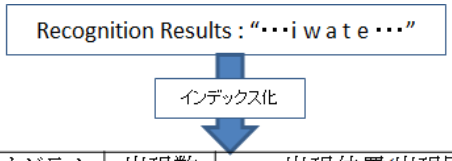
検索対象の音声ドキュメントは，発話毎にポーズによりセグメンテーションし，その発話毎にサブワード認識を行う．認識結果のサブワードモデル系列をサブワード認識結果として予め保持しておく．本稿では，クエリはテキストで与えられるものとし，クエリは変換規則に則り自動でサブワード系列に変換できるものとする．このサブワード系列クエリとサブワード認識結果データベースを連続動的計画法（連続 DP: Continuous Dynamic Programming）で照合を行う．照合時の局所距離には，サブワード間音響距離を用いることによりテキストマッチングによる検索性能の劣化を防いでいる．クエリと各候補発話区間の連続 DP の累積距離が小さい，即ち類似度が高い順にユーザへ候補区間を提示する．

我々の STD システムでは，NTCIR-9 SpokenDoc の STD ALL タスク即ち日本語話し言葉コーパス(CSJ: the Corpus of Spontaneous Japanese)2702 講演に対する 1 クエリあたりの検索時間は約 16 秒であった．我々が現在行っている連続 DP による音声ドキュメント全体との照合方法では，検索時間は検索対象データ群のデータ量の増加と線形に増加する．このため，この規模より大きい音声ドキュメントでは全照合システムは利用可能性が低く，高速化は必要不可欠である．

2.2 提案方式

2.2.1 音素トライグラムを用いたインデックスの構築

本稿で提案する手法では，検索対象の音声ドキュメントに対し，事前に音素認識あるいは音節認識を実行しておく．その認識結果である音素列から 1 音素ずつずらしながら音素トライグラムを抽出し，音素トライグラムの転置インデックスを作成する．この転置インデックスは，Fig.2 のように出現したトライグラムに対しその出現位置を保持させたものである．インデックスを構築する際，音素を 1 つの数字に対応させ，トライグラムはその 3 つの音素の数字で指定できるように 3 次元配列に格納する．



トライグラム	出現数	出現位置(出現区間)
2 37 9 (a t e)	40326	100092, 103884,104036･･･
:	:	:
18 43 2 (i w a)	18466	21935, 116746, 360930･･･
:	:	:
43 2 27 (w a t)	21064	12758 ,131232, 175225･･･

Fig.2　転置インデックス構築の例

2.2.2 クエリのトライグラムの抽出

クエリが与えられると，クエリの 3 音素単位で 1 つのトライグラムを構成し，1 音素ずつシフトさせながらクエリの音素列を分割し，トライグラム群を作成する．1 つのクエリに対して（クエリの音素数－2）個の音素トライグラムを抽出する．

2.2.3 トライグラム照合法

クエリに含まれるトライグラムを 2.2.1 の転置インデックスから参照する際は，2.2.1 同様トライグラム中の各音素の音素番号から，転置インデックス中の配列番号を指定することで直接その音素トライグラムの配列を参照し，その音素トライグラムの位置情報（出現区間）を即座に取得することができる．

2.2.4 トライグラムのヒット数 N 順による候補数の制御

2.2.3 よりクエリ中のトライグラムを 1 つ以上含む発話区間が候補区間となる．ここで，ある発話区間が N 個のトラ

イグラムを含む時その発話区間のヒット数 N と呼ぶ．クエリ中のヒット数がより多い候補区間がクエリを含む可能性が高く，一方，クエリのトライグラムを 1 つだけ含んでいる候補区間は数が多く，正解区間を網羅している可能性は高いが効果的な絞込みが行われていないと考える．また，クエリ毎にその長さ（トライグラム数）が異なるため，一律に N 個以上とすることは難しい．今回は，効果的な候補の絞込みと正解区間の網羅性を実現するために以下のように候補の抽出を行う．

トライグラムのヒット数の最大が Nmax, 候補の下限数 T として，

(0) K の初期値を N とする（K=Nmax）
(1) クエリのトライグラムを K 個含む候補区間を抽出する
(2) 以下のいずれかの条件を満たせば Nmin=K とし終了する
- ●抽出した候補区間数が下限数 T 以上
- ●K＝1

(3) K=K-1 として(1)へ

即ち，ヒット数最大の Nmax から候補を抽出し始め順次ヒット数を少なくしながら候補を抽出し，候補数が候補の下限数 T 以上になるまで(もしくはヒット数が 1 になるまで)候補とするヒット数の条件を緩めていく．

例えばトライグラムの最大ヒット数 Nmax=5，候補の下限数 T=5,000 とした場合，1 つのクエリに対してクエリのトライグラムを 5 つ以上含む候補区間を抽出し，その候補区間数が 5,000 件以上となれば候補抽出を終了する．5,000 件未満であれば，N=4 と条件を緩め候補区間をさらに抽出する．候補数が 5,000 件を超えるまで繰り返しヒット数の制限を緩めるか，ヒット数が 1 になるまで行い，抽出された候補区間に対して連続 DP 照合を行う．

T を大きく設定すれば，正解区間の網羅性を維持した検索を行えるが，連続 DP で詳細な照合を行う必要のある候補区間が多くなりすぎ高速性が失われる．T が小さすぎると，高速な検索が行えるが，正解区間の網羅性が低下するため適切な T の設定が必要であり，次章で実験的な検証を行う．

2.2.5 トライグラムの連続性の考慮

本節ではヒット数による候補の抽出に加えて，クエリのトライグラムの連続性を考慮して候補を抽出する方式を提案する．クエリのトライグラムを複数有する候補区間ではクエリに含まれるトライグラムが順番どおり連続して含まれていることが望ましく，2 つ以上のトライグラムが連続している場合，連続性があると定義する．ヒット数による候補の抽出ではこのような連続性は考慮されておらず，クエリのトライグラムの出現数のみでの評価を行っている．クエリ中のトライグラムの連続性がある候補区間の方が正解区間である可能性が高く，より効果的な候補の抽出を行えると考える．Fig.4 にトライグラムの連続性がある区間，連続性がない区間の例を示す．図のように候補区間 1，2 はどちらもクエリのトライグラムを 3 つ含んでいるが，候補区間 1 では候補区間内のトライグラムの出現位置の連続性は見られない．一方，候補区間 2 ではクエリ中のトライグラム（iwa と wat）が順番どおりに連続して含まれており，この場合では候補区間 2 の方が候補区間 1 に比べてクエリが含まれる可能性が高いとする考え方である．

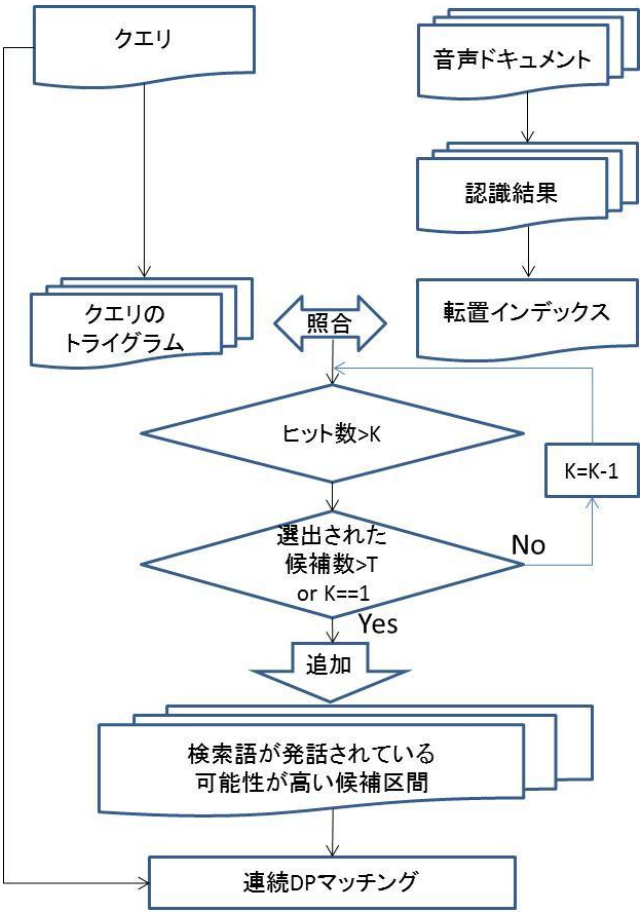


Fig.3 トライグラムのヒット数 概要図

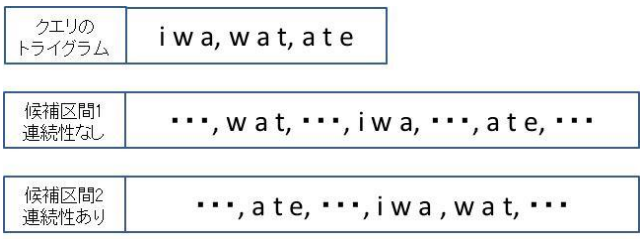


Fig.4 トライグラムの連続性の例

3. 評価実験

3.1 評価用データ

検索対象の音声ドキュメントは CSJ 全 2702 講演・約 604 時間分を用いる．これらの音声ファイルは CSJ 付属の xml

データの IPU 単位でセグメンテーションしてあり，全 2702 講演データでは約 880,391 発話となる．クエリは NTCIR-9 Spoken Doc Task フォーマルランで用いられた全 2702 講演用クエリ各 50 個を用いる．セグメンテーションされた発話内にクエリが含まれていれば正解区間とする．実験条件と現状の検索性能・時間を Table1 に示す．

3.2 性能指標・時間計測

検索性能の評価指標には，MAP(mean average precision)を用いる．クエリ毎に順位が上位の候補から出力した際，正解出現時の適合率を平均すると AP(average precision)が得られ，この AP を全てのクエリに対して平均すると MAP が得られる．MAP の他，検索結果の上位 H で検索して正解率を用いて評価を行う．MAP はシステム全体の検索性能を表し，上位 H は正しく高適合度候補を抽出できているかを表す．後述するが，一般ユーザを対象としたアプリケーションではシステム全体の性能よりもユーザに直接提示する上位 H 件の評価が重要であると考える．

処理時間の計測には，Intel 社の Core i7 2600，メモリ 8G の Linux マシンを使用し，Linux の time コマンドと C 言語関数の gettimeofday を用いて時間の計測を行った．

Table 1　実験条件と現状の検索性能・時間

検索対象音声データ	CSJ・全 2702 講演 (約 604 時間分)
発話件数	880,391 発話
クエリ	NTCIR-9 Spoken Doc Task Formal run 用 50 クエリ
デコーダ	Julius 4.5.1
音響モデル	集約 Triphone(3,500 models)[5]
言語モデル	音節バイグラム ・音節トライグラム
連続 DP による全照合時の性能（MAP(%)）	66.37
1 クエリ検索時間(s)	16.38

3.3 実験結果

3.3.1 インデックスの構築時間とサイズ

約 604 時間の音声ドキュメントのインデックスに文献[3]では約 26Gbyte, 文献[6]では 6.8Gbyte を要している．一方，本提案方式で作成する転置インデックスは構造がシンプルであり，Table2 に示すように 130.3MB と従来の研究におけるインデックスに比べ, 非常にコンパクトになった．また，インデックスの構築時間も 30 秒未満と高速なため, 新たな音声ドキュメントが追加された場合もインデックスを追加するだけなのでインデックス全体を構築し直す必要はない．

Table2　インデックスの構築時間とサイズ

インデックスの構築時間(s)	28.56
インデックスサイズ(MB)	130.3

3.3.2 検索性能と検索時間

(1)MAP による評価

実験の結果，候補の下限数 T による検索性能（MAP）と 1 クエリあたりの検索時間（Time）を Table3 と Fig.5 に示す．図中，検索性能は棒グラフ，1 クエリあたりの検索時間は折れ線グラフで表す．この結果から提案方式による検索性能と検索時間の削減について以下にまとめる．

- 全ての音声ドキュメントを連続 DP で検索した場合（all）と比較し，候補の下限数 T=30,000 でほぼ同じ性能となる．1 クエリ当たりの検索時間は 16.375sec から 1.804sec に削減し，9 倍以上の高速化を実現した．
- T=15,000 の時，検索性能の低下は 0.24 ポイントで検索時間を時間は 16.375sec から 1.172sec と 92.84%削減できた．
- T=10,000 では検索性能の低下を 1 ポイント未満に抑えた上で 1 秒以内の検索を実現できた．

以上のように本手法により，検索性能を維持しつつ検索時間を大幅に削減可能であることが分かる．本手法は事前に作成した音素トライグラムのインデックスを参照し，クエリが存在する可能性が高い区間のみに対して連続 DP によるスコアリングを行うことにより，全ての音声ドキュメントに対してクエリとのマッチングを行わずに検索が可能である．

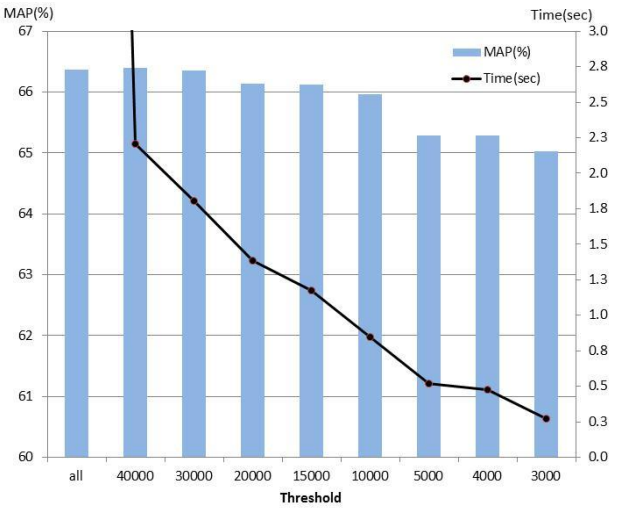


Fig.5　ヒット数　MAP と検索時間の遷移

Table3　提案方式の結果

Threshold	all	40,000	30,000	20,000	15,000	10,000	5,000	4,000	3,000	1,000
MAP(%)	66.37	66.40	66.36	66.14	66.13	66.40	65.29	65.29	65.03	63.52
Time(sec.)	16.375	2.210	1.804	1.382	1.172	0.846	0.518	0.477	0.271	0.164

Table4 上位候補 H 件内の候補区間保持数(50 クエリのヒット総数)

	all	40,000	30,000	20,000	10,000	5,000	4,000	3,000	1,000
H=1	0.94(47)	0.94(47)	0.94(47)	0.94(47)	0.94(47)	0.94(47)	0.94(47)	0.94(47)	0.94(47)
H=3	0.90(135)	0.90(135)	0.90(135)	0.90(135)	0.90(135)	0.90(135)	0.90(135)	0.90(135)	0.90(135)
H=5	0.90(225)	0.90(225)	0.90(225)	0.90(225)	0.90(225)	0.89(222)	0.89(222)	0.89(222)	0.88(221)
H=10	0.78(389)	0.78(389)	0.78(389)	0.78(389)	0.78(389)	0.77(387)	0.77(387)	0.77(387)	0.76(382)
Time(sec)	16.375	2.210	1.804	1.382	0.846	0.518	0.477	0.271	0.164

3.3.3 上位候補による評価

本節では検索結果上位 H 件の正解率による評価を行う．実システムとして運用する際には検索時間と共に上位に正解区間が含まれているかが検索タスクにおいて重要であると考える．音声ドキュメントの検索では，ユーザが目で見て確認ができないため，第一候補を実際に聞いて確認する作業が必要であり，その確認の間に精密な検索を行うことができる．従って，一度に全体の検索を高速に完了する必要性は低く，高順位即ち，高適合度区間に対して高速に検索が完了する手法が有効であると考えられる．

本節では，本提案方式が候補の下限数 T の制御によって高速に上位候補の検索が可能であることを実験により示す．下限候補数 T を小さくすることで整合度の高い上位候補に限定して CDP の詳細な照合を行うことで，高速な高精度区間の検索が実現できる．ユーザがこれらの候補を確認する間により大きい T を用いて検索を行うことで網羅性の高い検索を行う手法となる．

上位 H 件における評価では，上位 H 件以内に正解区間を含む件数を 50 クエリの平均と検索時間で評価する．H は 1,3,5,10 の 4 種について調査する．結果を Table4 に示す．表から，T=3,000 の時でも H=1,3 位に正解区間が all の場合と同等の性能が 0.271 秒で得られた．T=10,000，H=10 で all と同等の検索性能を維持し、1 秒未満で 10 位まで高適合度候補区間を抽出出来ており，高適合度区間に対して高速に検索が完了していることが分かる．このことから本提案方式が実システムの運用においても有効であると考える．

3.3.4 ヒット数，ヒット数＋トライグラムの連続性の比較

ヒット数のみによる候補の抽出とヒット数にトライグラムの連続性を考慮した候補の抽出の 2 つの手法の検索性能の比較を Fig.6 に示す．図中の case1,2,3 はクエリ 1 つ当たりの検索時間が同程度のときを比較したものである．

case3 ではトライグラムの連続性を考慮した場合の検索性能 2 ポイントほどが低下した．トライグラムの連続性を考慮することにより正しく認識された区間に対しては高精度な候補区間を抽出できる．一方，誤認識が含まれている区間ではヒット数のみの抽出に比べて候補区間として抽出されにくくなったためと考える．

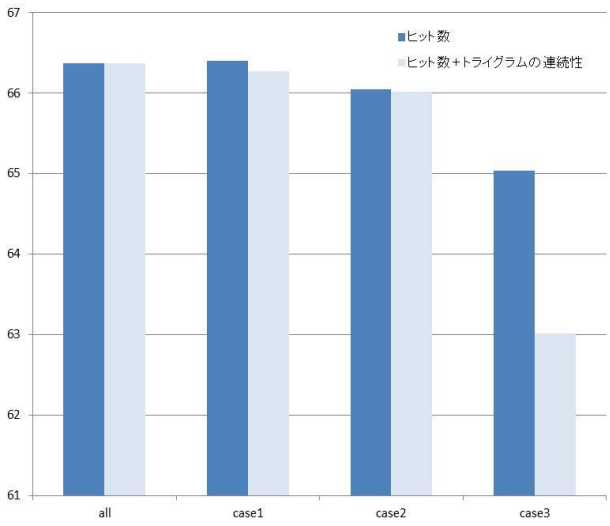


Fig.6 ヒット数，ヒット数＋トライグラムの連続性の検索性能の比較

3.3.5 クエリ長と Nmax，Nmin についての考察

2.2.4 でトライグラムのヒット数 N 順による候補数の制御する一方式を提案したが，N による適切な候補数の制御を検討するため，本節では，クエリ長と Nmax，Nmin について検討を行う．まず Nmin の平均を調べてみると，T=40,000 のとき 1.40，T=10,000 のとき 2.12，T=3,000 のときに 2.72 となり，N=1 に近いところまで候補の抽出が行われていることが分かる．次にこれらの要素間の相関の調査を行ったその結果を Table 5 に示す．

この結果から得られた知見を以下にまとめる．

- クエリ長は Nmin，Nmax，AP のいずれに対しても相関が見られた
- Nmin は Nmax との相関は高いが，AP との相関は弱い

Table 5 各要素間の相関

		Nmin			Nmax	AP
		T=40,000	T=10,000	T=3,000		
クエリ長		0.551	0.666	0.617	0.969	0.508
Nmin	T=40,000				0.518	0.293
	T=10,000				0.653	0.337
	T=3,000				0.617	0.270
Nmax						0.538

- Nmin（T=40,000 と T=10,000，T=3,000 のとき）とクエリ長との相関を比較すると, T=10,000 のときの方が相関が強かった

クエリ長と Nmax、Nmin および AP との強い相関は予想通り確認できた．一方，Nmin（T=40,000 と T=10,000，T=3,000 のいずれのときも）とクエリ長との相関は弱いことから Nmin をクエリ長から与えることは困難と考えられる．このことより, 今回用いた 2.2.4 のトライグラムのヒット数 N 順による候補数の制御は，候補数を適正な数に制御する上ででは有効だったのではないかと考える．

4. おわりに

本稿では音声ドキュメントを予め音素認識した結果から音素トライグラムを抽出し，音素トライグラムの転置インデックスを作成しておき，クエリが与えられると音素トライグラムの転置インデックスとクエリの音素トライグラム群を照合することによって，STD の性能を維持しつつ高速な検索手法を提案した．

クエリのトライグラムを含む可能性の高い候補区間を抽出し，抽出された候補区間のみに連続 DP 照合を行うことで全ての音声ドキュメントを連続 DP で検索した場合と比較して検索性能の低下を抑えながら大幅な検索時間の短縮を図った．

クエリのトライグラムのヒット数による検索候補区間の抽出を行った場合，検索性能の低下無しで検索時間を 16.38sec から 2.21sec(7.41 倍)に, 検索性能の低下 1.0 ポイント未満で検索時間は 16.38sec から 0.85sec(19.36 倍)の高速化を実現した．

高適合度区間の高速抽出として，上位 1，3，5，10 位いずれについても全照合と同性能を 1 秒未満で実施することができ，ユーザには疑似的に待ち時間を感じさせることのない検索が可能であることを示した．今後は音節のバイグラム，トライグラムについても同様の実験，検証を行い，適切な音素, 音節の N グラム数について検証していきたい．

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 Document Task in NTCIR Workshop, NTCIR-9 Meeting, 2011.

[2] 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.48, No.5, pp.1990-2000, 2007.

[3] 神田直之, 住吉貴志, 小窪浩明, 佐川浩彦, 大淵康成, 多段リスコアリングに基づく大規模音声中の任意検索語検出，電子情報通信学会論文誌 D Vol.J95-D No.4 pp.969-981, 2012

[4] 中川聖一, 岩見圭祐, 藤井康寿, 山本一公, "連続音節認識結果の距離つきトライグラムアレイ化による未知語音声の超高速検索", 第 4 回音声ドキュメント処理ワークショップ講演論文集, 2010.

[5] 中野拓也, 伊藤慶明, 小嶋和徳, 石亀昌明, 田中和世, 李時旭他, "triphone 認識を用いた音声中の検索語検出における適切なモデル数の検討", 日本音響学会 2009 年秋期研究発表会論文集 1-Q-28, pp.185-188, 2010.

[6] 斉藤裕之, 伊藤慶明, 小嶋和徳, 石亀昌明, 田中和世, 李時旭他, "複数音節の事前検索結果に基づく音声中の検索語検出の高速化", 日本音響学会 2012 年春期研究発表会論文集 3-7-10，2012.

[7] 岩見圭祐, 山本一公, 中川聖一, "複数音声認識システムを併用した音節 n-gram 索引による検索性能の改善", 第 6 回音声ドキュメント処理ワークショップ, SDPWS2012-10, 2012.