情報処理学会論文誌 Vol.58 No.10 1735–1744 (Oct. 2017)

音声検索語検出のための検索語拡張法

南條 浩輝[1,a)] 前田 翔[2] 吉見 毅彦[2]

受付日 2016年9月8日, 採録日 2017年7月4日

概要：音声中で検索語がそのまま現れる発話を特定する音声検索語検出（Spoken Term Detection: STD）の研究を行う．STD における大きな問題点の 1 つに検索語ではないものを検出する誤検出問題があげられる．本研究では，この誤検出をできるだけ少なくする方法を研究する．具体的には，検索語拡張を行って拡張語を得たうえで連続 DP マッチングによる拡張語の検索を行い，その検索結果に基づいて検索語が含まれる発話の候補の並べ替え（リスコアリング）を行うことで誤検出を抑制する方法を提案する．本論文では，拡張語の獲得方法として，検索語の前または後に文字列を付加したものを拡張語とする手法を提案する．この手法はどのような検索語に対しても容易に拡張語を自動生成できるため，汎用性が大きいと考えられる．講演音声を対象とした種々の STD 検索タスクで評価したところ，すべてのタスクで検索精度の向上が得られ，提案手法の有効性および汎用性を示した．

キーワード：音声検索語検出，検索語拡張，文字列の付加，リスコアリング

Automatic Query Expansion for Spoken Term Detection

Hiroaki Nanjo[1,a)] Sho Maeda[2] Takehiko Yoshimi[2]

Received: September 8, 2016, Accepted: July 4, 2017

Abstract: This paper addresses Spoken Term Detection (STD), which finds speeches including a specified query term. One of the main STD problems is a false detection problem, which we focus on in the paper. We investigate a method suppressing false detections based on a query expansion (QE) approach, which extracts query-related terms. Specifically, we rescore and rerank speech candidates which may include query term(s) with the results obtained by continuous DP matching between expanded queries and speeches. In this paper, we propose a QE method for STD, that is, making expanded terms by adding words to the original query. The QE approach is widely applicable since it can generate expanded terms automatically for any query terms. On a task of STD from lecture corpus, we confirmed the effectiveness of the proposed method. We achieved STD performance improvements for several STD tasks, which showed a validity and robustness of the proposed method.

Keywords: spoken term detection, query expansion, adding strings, rescoring

1. はじめに

デジタル化された大量の音声や動画（音声ドキュメント）から，ユーザが知りたい特定の区間を取り出す音声ドキュメント検索技術が求められている．音声ドキュメント検索には，大きく 2 つのタスク，すなわち，検索要求（文または語のセット）が示す内容の区間を検索する音声内容検索（Spoken Content Retrieval: SCR）タスク [1], [2] と検索要求の語（以降，「検索語」）そのものが出現する音声区間を見つける音声検索語検出（Spoken Term Detection: STD）タスク [3], [4], [5], [6] がある．いずれのタスクにおいても，音声認識によってテキスト化したものを対象に検索するのが一般的である．その際，音声認識における音声認識誤りは本質的に避けられず，何らかの対処が必要である．

本論文では，STD タスクを対象とする．STD における検索語は 1 つまたは連続する複数の単語（以下，単語列）である．特に STD では，検索語として固有名詞や新語な

[1] 京都大学学術情報メディアセンター
Academic Center for Computing and Media Studies, Kyoto University, Kyoto 606–8501, Japan
[2] 龍谷大学理工学部
Faculty of Science and Technology, Ryukoku University, Otsu, Shiga 520–2194, Japan
[a)] nanjo@media.kyoto-u.ac.jp

どが想定されている．これらは単語単位での音声認識では未知語となる場合も多く，誤認識されやすい．なお誤認識されたときは，発音が近い別の単語列となっていることが多い．この問題に対して単語よりも小さいサブワード（音節や音素など）を単位として音声認識を行うこともある．この場合でも単語を単位としてマッチングを行おうとすれば，音声認識結果を対象としたかな漢字変換が必要となり，固有名詞や新語が正しく変換されない可能性もある．これらのことより，STD ではサブワードを単位としたマッチングが行われるのが一般的である．本研究ではサブワードに音素を採用し，音素単位でマッチングを行う STD を行う．

音声認識誤りにより検索語が音素列が近い別の単語列に認識されているケースを考える．このとき検索語の音素列と検索対象の音素列との間で完全一致する区間を検出しても正しく検出できない．したがって，音素列が最も類似する（たとえば 1 つの音素だけ異なる）音声区間を次に探す．それでも見つからない場合は次に類似する（たとえば 2 つの音素が異なる）区間を探す．このように，音声区間を完全一致するものから徐々に類似するものといったように類似度順にリストにして出力することで，検索語が誤って認識されていてもそれを含む音声を検索結果のリストに含める（検出する）ことができる．しかし，このとき同時に求めたい単語でないものも検索結果のリストに含めてしまう（誤検出する）．たとえば，音声認識の際に「大阪（o: s a k a）」という語に音声認識誤りが起こり，「お酒（o s a k e）」として認識結果に登録された場合について考える．完全一致では検出できないが，音素が 2 つまで異なるものを許容すればこれを検索結果に含めることがきる．しかしこのとき，「お酒」「大崎」「大須賀」などが正しく認識されている音声区間も検索結果に含めてしまう．

また，このような STD では，音声認識誤りに由来する問題以外の問題も発生する．1 つ目は同音異義語すなわち同じ音素列でも意味が異なる単語が存在するという問題である．たとえば，人名の「尾家（o k e）」を検索する場合に一般名詞の「桶」や動詞の命令形の「置け」を検出するという問題である．2 つ目は文字列の部分一致の問題である．これは特に検索語長が短い検索語で問題となると考えられる．たとえば，「タイ（t a i）」という検索語で検索すると，「〜したい（sh i t a i）」「大した（t a i sh i t a）」などを含む音声も検出するという問題である．

本研究では，これらの誤検出への対応を目的とした，検索語拡張に基づく 2 パスの検索語検出アルゴリズムを提案する．これは，検索語拡張を行い，第 1 パスで得られた検索結果（音声区間リスト）に対して，第 2 パスで拡張語の検索結果を参照してスコアを補正することで，検索語の誤検出を抑制する方法である．第 2 パスでのスコアの補正は，検索語が含まれる音声（第 1 パスで見つかった検索結果の候補）に拡張語が出現しない場合に，当該検索結果の候補は不確かという仮定に基づいて行う．このような 2 パスまたはそれ以上のパスを用いて STD を行う研究には文献 [7], [8], [9], [10] などがあげられる．文献 [7] では，高精度な検索結果を高速に得るために，1 パス目で粗く絞って 2 パス目で照合を行っている．文献 [8], [9], [10] では，種々の検出モデルでターゲットの検索語の検出を行って，結果を統合している．これに対し，提案手法は，検索語とは異なる語による検索結果を使って 1 パス目の検出結果の高精度化を目指すものである．本手法はこれらの文献 [7], [8], [9], [10] の手法と組み合わせることも可能であり，先行研究の高精度化も期待できる．

このような検索語とは異なる語の出現情報を 2 パス目の照合に用いる研究としては，小田原ら [11] の研究がある．これは，検索語とよく共起する単語が 1 パス目の検索結果（候補）の周辺に見つかれば，当該候補は確からしいと考える手法であり，本提案手法と考え方は類似している．ただし，共起語を用いる際に外部情報源（WEB など）が必要であることや，共起語が正しく得られないことが問題となる．さらに，共起語情報を扱う際のパラメータが多く，それらの設定に困難がある．

これらに対し，本論文では，拡張語の検索結果で検索語の検索結果を補正するシンプルなアルゴリズムを提案する．提案手法は検索語の前や後に文字列を付加した語を拡張語として用いるものであり，どのような検索語に対しても適切な拡張語が得やすいという利点がある．どのような検索語に対しても拡張語を得やすく，かつ高精度化を実現できる方法はこれまでに提案されておらず，本研究はこの点において新規性を有する．さらに，異なる種々の STD タスクで提案手法の評価を行い，提案手法の有効性を示す．

本論文の構成は次のとおりである．2 章では，一般的な連続 DP マッチングに基づく音声検索語検出について述べる．3 章では，提案手法である音声検索語検出における検索語拡張について述べる．4 章では，検索語拡張の評価を行い，提案手法の有効性を示す．5 章では，種々の STD タスクに対して提案手法を適用し，提案法が広く適用可能であることを示す．6 章で結論を述べる．

2. 音声検索語検出

2.1 概要

音声検索語検出（Spoken Term Detection: STD）とは，音声中で検索語がそのまま現れる音声箇所を特定する処理のことを指す．NTCIR [12] の音声ドキュメント検索タスク [3] においては，この条件を緩和し，学会講演や講義などの長い音声（以降，本論文では「音声ファイル」とする）を 200 ミリ秒以上の無音で区切って複数の音声区間（=IPU: Inter-Pausal Unit，以降，本論文では「発話」とする）に分割しておき，検索語が含まれる発話を特定する処理を STD と定義しており，検出結果は「音声ファイル名+発話番号」

Algorithm 1 連続 DP マッチングを用いた編集距離算出アルゴリズム

for $j = 0$ to u **do**
  $M(0, j) = 0$
end for
for $i = 0$ to q **do**
  $M(i, 0) = i$
end for
$LD = M(q, 0)$
for $j = 1$ to u **do**
  for $i = 1$ to q **do**

$$M(i,j) = min \begin{cases} M(i-1,j)+1 \\ M(i,j-1)+1 \\ M(i-1,j-1)+d(i,j) \end{cases}$$

$$\text{ただし,} \begin{cases} d(i,j) = 0 & \text{if } Q_i == U_j \\ d(i,j) = 1 & \text{otherwise} \end{cases}$$

  end for
  $LD = min(LD, M(q, j))$
end for
return LD

のリストとなる．本論文で扱う STD は後者の定義に基づくものである．

2.2 連続 DP マッチングによる音声検索語検出

1 章で述べたとおり，STD では単語より小さな単位であるサブワード（本研究では音素）を単位とし，サブワード系列どうしを誤りを許容して照合することが一般的である．このような照合方法として，DP マッチング [13] があげられる．この DP マッチングを検出単位となる音声区間（本論文では発話）の始端から順にずらしながら適用すること（連続 DP マッチング [13]）により，その区間のサブワード系列中で検索語と最も適合する部分系列とそのマッチング度合い（距離）を求めることができる．本研究ではサブワードとして音素を採用し，マッチング度合いを示す距離として編集距離（Levenshtein 距離）を採用する．

編集距離は，1 文字の「置換」「挿入」「削除」の操作を繰り返して，ある文字列を別の文字列に置き換えるために必要な最小操作数であり，置換，挿入，削除のコストをすべて 1 とした DP パスを用いた DP マッチングにより求めることができる．具体的には，検索語 Q の音素列（長さ q）と検索対象の発話 U の音素列（長さ u）の編集距離を，Algorithm 1 に基づいて求める．ここで，LD は求める編集距離，M は $(q+1) \times (u+1)$ の行列，$M(i, j)$ は行列 M の (i, j) 要素，Q_j は検索語 Q の j 番目の音素，U_i は発話 U の i 番目の音素である．

本研究で用いる連続 DP マッチングに基づく STD のアルゴリズムを Algorithm 2 に示す．

3. 検索語拡張

情報検索における検索語拡張とは，ユーザが入力した初期検索要求（語のセット）に対していくつかの拡張語を加えることでより検索に適した検索要求を生成する手法である．本研究では，STD において検索語拡張を研究するが，本論文での検索語拡張とは，拡張語を得たうえで連続 DP マッチングを行い，その結果を用いて元の検索語の検索結果を調整すること（リスコアリング）を指すこととする．

本論文では，STD における検索語拡張において検索語の前または後に文字列を付加したものを拡張語とする手法を提案する．本章では，この手法について詳細に述べる．

Algorithm 2 連続 DP マッチングに基づく STD のアルゴリズム

(1) 検索対象となるすべての発話を音声認識し，音素列を付与しておく．
(2) ユーザから検索語 Q を受け取る．検索語が単語列の場合は辞書に基づいて音素列に変換する．音素系列の長さを q とする．
(3) 各発話について，検索語と発話それぞれの音素系列間の編集距離を連続 DP マッチングで計算する．ここで，音声ファイル S の n 番目の発話 S_n と検索語 Q との編集距離を $LD(Q, S_n)$ と表記する．
(4) 各発話 S_n に対し検索語との距離の近さを表すスコア $1 - \frac{LD(Q,S_n)}{q}$ を付与し，その順に出力する．スコアが同じである場合は，発話 ID（本研究では「音声ファイル名_発話番号（たとえば，A02F0038_0026 や 09-17_0189）」の形式）の ASCII 順に出力する．

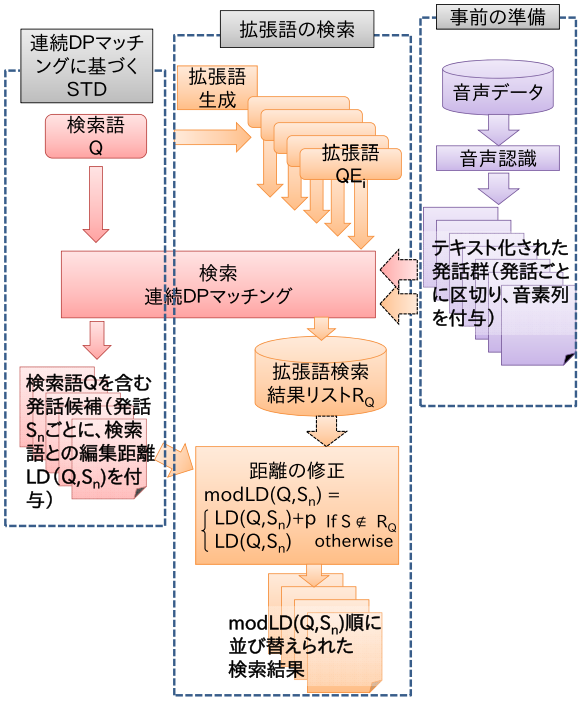

図 1 検索語拡張に基づく STD の概要
Fig. 1 Overview of STD with query expansion.

3.1 音声検索語検出と検索語拡張

検索語拡張に基づく STD の概観を図 1 に示す．また，アルゴリズムを Algorithm 3 に示す．

本手法は，検索語と拡張語が同じ音声ファイル S に出現

Algorithm 3 検索語拡張に基づく STD のアルゴリズム

(1) 検索対象となるすべての発話を音声認識し，音素列を付与しておく．
(2) ユーザから検索語 Q を受け取る．検索語が単語列の場合は辞書に基づいて音素列に変換する．音素系列の長さを q とする．
(3) 各発話について，検索語 Q と発話それぞれの音素系列間の編集距離を連続 DP マッチングで計算する．ここで，音声ファイル S の n 番目の発話 S_n と検索語 Q との編集距離を $LD(Q, S_n)$ とする．
(4) 検索語から複数の拡張語 QE_i（$i = 1 \ldots N$）を生成する．拡張語が単語列の場合は辞書に基づいて音素列に変換する．
(5) 各拡張語 QE_i と検索対象の発話 S_n それぞれの音素系列間で連続 DP マッチングを行い，設定したしきい値以下の距離であれば，音声ファイル名 S を拡張語検索結果リスト R_Q に加える．すでに S が R_Q に登録されていた場合は何もしない．これをすべての i と S_n に対して行う．
(6) すべての発話 S_n について，以下の式に基づいて編集距離を修正し，調整後の編集距離 $modLD(Q, S_n)$ を求める

$$modLD(Q, S_n) = \begin{cases} LD(Q, S_n) + p & \text{if } S \notin R_Q \\ LD(Q, S_n) & \text{otherwise} \end{cases}$$

すなわち，検索語を含む発話の候補 S_n について，同じ音声ファイル S 中に拡張語が見つからなかった場合は，編集距離にペナルティ $p > 0$ を加える
(7) 各発話 S_n と元の検索語（長さ q）の距離の近さを表すスコア $(1 - \frac{modLD(Q,S_n)}{q})$ を計算し，その順に結果を出力する．スコアが同じである場合は，発話 ID（本研究では「音声ファイル名_発話番号（たとえば，A02F0038_0026 や 09-17_0189）」の形式）の ASCII 順に出力する．

しなかった場合に，当該音声ファイル S 中のすべての発話にペナルティを与えてスコアを低くする手法である．たとえば，検索語を「タイ（t a i）」，拡張語を「タイが（t a i g a）」と「タイを（t a i o）」とした場合を考える．ある音声ファイルにおいて「t a i」とマッチした発話があるものの，この音声ファイル中に「t a i g a」と「t a i o」のいずれもが見つからない場合は，この音声ファイル中の「t a i」とマッチした発話は誤検出の可能性が高いと考えてペナルティを与える．本手法はこのようなことを狙うものである．

本手法は，同一ファイル内にいずれかの拡張語が存在したか否かの判断を行い，存在しなかった場合に誤検出と見なして一様に検索スコアを下げるという単純な方法である．拡張語が存在した場合に，その確からしさ（共起の統計量など）を用いて検索スコアの修正を行う方法 [11] も考えられるが，統計量の事前の計算が不要である点で利点がある．さらに，スコアの修正において，本手法と既存手法 [11] を併用することも考えられる．

次に，提案する検索語拡張における拡張語の抽出方法について述べる．

3.2 検索語の前や後に文字列を付加する拡張語

本研究では，拡張語の獲得方法として，検索語の前や後につきやすい語を付加して検索語を長くし，これを拡張語とする方法を提案する．これにより，短い検索語が別の語の一部にマッチして誤検出されることを防ぐ効果が期待できる．また，音声認識誤りにより別の語が検索語として現れていたとしても，前後の文字列まで含めると拡張語にマッチしない場合には，それを検索語でないと見なすことも可能になると期待される．

前後にどのような文字列が現れやすいかを求める方法には様々考えられるが，本研究では，STD のタスクを考えたときに，検索語は基本的に名詞，特に固有名詞となることが多いことに着目し，検索語の前または後に格助詞（10 種類）[*1] を付加して拡張語とする．格助詞を付加する手法には，どのような検索語が与えられても何らかの拡張語を得ることができるという利点がある．ただし，検索語が名詞でない場合は，適切な拡張語とならない可能性がある点には注意が必要である．

この手法を用いることで部分文字列の一致が避けられる可能性がある．たとえば「タイ（t a i）」（= 国の名前）を検索した場合，「〜したい（s h i t a i）」や「大した（t a i s h i t a）」，「スタイル（s u t a i r u）」を含む音声では，これらの一部分と検索語がマッチするので，これらも検索結果として抽出（誤検出）される．ここで，格助詞を前につけた拡張語は「がタイ，のタイ，にタイ」などであるため，「〜したい」や「スタイル」とはマッチせず，このような語の誤検出を防ぐことが期待できる．格助詞を後ろにつけた拡張語は「タイが，タイの，タイに」などであるため，「大した」や「スタイル」とはマッチせず，このような語の誤検出を防ぐことが期待できる．本手法はこのような効果を狙うものである．ただし，名詞どうしの同音異義語（たとえば，タイ（国）と鯛）では，どちらにも格助詞がつくため本手法ではこのような名詞どうしの同音異義語への対応は行えない．

Algorithm 3 では，手順 (3) において「拡張語のそれぞれで連続 DP マッチングを行う」ことを記述している．これはどのような拡張語にも適用できるよう一般化して記述したものである．実際は，提案手法である前や後に語を付与するような検索語拡張では，拡張語での連続 DP マッチング（第 2 パス）ではすべての音声ファイルのすべての発話に対して行う必要はない．これは検索語とマッチする発話のリスト（検索結果候補集合）は第 1 パスの検索語を用いた連続 DP マッチングで分かっているためである．検索結果候補集合中の発話のみに対して行えばよく，さらにいえば，検索語の候補位置の前後を調べるだけでよい．

4. 評価実験

4.1 評価尺度

情報検索システムの検索性能の評価尺度には，正解が検索結果としてどれほど出力されたかを表す再現率（recall）

[*1] 日本語百科大辞典［縮刷版］[14] 225 ページ記載の 10 種類（が・の・に・を・へ・と・で・より・から・や）

と，検索結果中に正解がどの程度含まれているかを表す精度（precision）がある．理想的には，再現率と精度を同時に 1 に近づけることが望ましい．実際には両者はトレードオフの関係にあり，一般的に，検索結果を上位に絞ると再現率は低いものの精度が高く，検索結果を多く出力すると再現率は高くなるものの精度が低くなる，といった傾向がある．このため，ある検索出力数のときの再現率と精度だけで検索性能を評価するのは不十分である．この問題に対して，検索結果出力数を変化させて様々な再現率レベルのときの精度を求め，それらの平均をとった値が評価尺度として広く用いられる．この尺度により平均的に性能が高い検索システムを評価できる．このような評価尺度として，平均精度（Average Precision: AP）がある．

ある検索語 Q に対する平均精度 AP_Q は，式 (1) で与えられる．

$$\mathrm{AP}_Q = \frac{1}{\#cor(Q)} \sum_{t=1}^{N_Q} \mathrm{IsTrue}(Q,t) \cdot P(Q,t) \qquad (1)$$

ここで，$\#cor(Q)$ は検索語 Q に対する正解発話数，N_Q は検索システムが検索語 Q の答え（検索結果）として出力した発話数，$\mathrm{IsTrue}(Q,t)$ は検索語 Q での検索結果の t 番目が正解であれば 1，そうでなければ 0 を返す関数であり，$P(Q,t)$ は Q の検索結果の t 番目までを評価したときの精度（precision）である．

この AP_Q を全検索語（総数 N）で平均したもの（式 (2)）は，MAP（Mean Average Precision）とよばれる．MAP は 0 から 1 をとり，1 に近いほど平均的に精度が高いことを表す．本研究では MAP を評価尺度として用いる．

$$\mathrm{MAP} = \frac{1}{N} \sum_{Q}^{N} AP_Q \qquad (2)$$

4.2 実験

実験データには，NTCIR-9 SpokenDoc [15] のテストコレクションを用いた．これは日本語話し言葉コーパス（CSJ: Corpus of Spontaneous Japanese）[16] の講演音声を対象とした音声ドキュメント検索のためのテストコレクションである．

NTCIR-9 SpokenDoc では，STD タスクとして，検索対象を全講演（2,702 講演）とする ALL タスクと一部（177 講演）を対象とする CORE タスクが設定されており，本研究では ALL タスクを用いた．検索語には dry run 用検索語（100 件）と formal run 用検索語（50 件）があり，ここでは dry run 用検索語（100 件）を用いた．検索対象の講演音声の認識結果には，タスクオーガナイザから配布されているマッチドモデルによる単語音声認識結果（Word Corr. = 74.1%, Word Acc. = 69.2%, Syll. Corr. = 83.0%, Syll. Acc. = 78.1%）[15] を用いた．各発話に対し，音声認識結果の 1-best 候補の音素列を認識結果として登録した．100 件の検索語はすべて音声認識の辞書に含まれる既知語である．

4.3 検索語の前や後に文字列を付加する検索語拡張の効果

4.3.1 評価結果

提案する検索語拡張の評価実験を行った．具体的には，検索語の前や後に文字列を付加する検索語拡張の評価を行った．本研究では，1) 検索語 X の前に 10 種類の格助詞のそれぞれをつけた 10 語（格助詞+X）を拡張語とする場合，2) 後ろにそれぞれの格助詞をつけた 10 語（X+格助詞）を拡張語とする場合，3) 1) と 2) の両者 20 語（格助詞+X，および，X+格助詞）を拡張語とする場合，の 3 通りを試した．なお，1) の拡張語が見つかる音声ファイル集合を R_Q^{Head}，2) の拡張語が見つかる音声ファイル集合を R_Q^{Tail} とすると，3) の 20 語の拡張語が見つかる音声ファイル集合は $R_Q^{Head} \bigcup R_Q^{Tail}$ となる．

また，前と後ろの両方に付加したもの 100 種（格助詞+X+格助詞）を拡張語として用いる方法も考えられるが，この拡張語が見つかる音声ファイル集合は $R_Q^{Head} \bigcap R_Q^{Tail}$ の部分集合となり，検索語と拡張語が同時に見つかるケースが少なくなると考え，本実験では用いていない．

拡張語と検索語が同じ音声ファイルに現れなかった場合のペナルティ（検索語拡張に基づく STD のアルゴリズムの手順 (6) の p）を，$p = 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5$ として実験を行った．検索結果としてスコアの高い順に上位 1,000 件を出力し，MAP を求めた．拡張語を用いた連続 DP マッチングでは，元の検索語による検索結果中の最小編集距離を求め，検索語拡張に基づく STD のアルゴリズムの手順 (5) のしきい値とした．すなわち，元の検索語で連続 DP マッチングを行ったときの最小編集距離が l であったとき，拡張語の連続 DP マッチングではしきい値を l とした．

結果を**表 1** に示す．比較のための従来法には連続 DP マッチングに基づく STD（Algorithm 2）を用いた．3 通りすべてで MAP の向上が見られた．検索語の前または後に格助詞をつけた合計 20 語を拡張語として用いた場合（ペナルティ = 2.5）のときに最も高い精度（MAP = 0.702）が

表 1 前または後に文字列を付加する検索語拡張の効果（MAP）
Table 1 Effect of query expansion based on adding strings to the query (MAP).

従来手法	文字列付加検索語拡張			
	penalty	後のみ	前のみ	前と後の併用
0.628	0.5	**0.661**	**0.656**	**0.663**
	1.0	**0.672**	**0.665**	**0.677**
	1.5	**0.691**	**0.677**	**0.697**
	2.0	**0.638**	0.613	**0.659**
	2.5	**0.691**	**0.672**	**0.702**
	3.0	**0.629**	0.589	**0.657**
	3.5	**0.680**	**0.653**	**0.696**

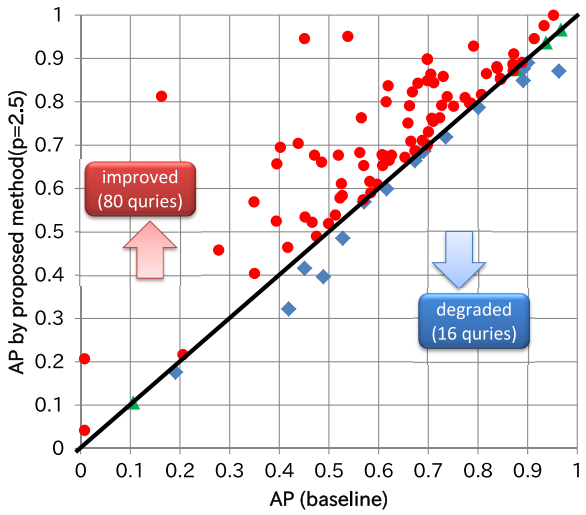

図 2 文字列を付加する検索語拡張の効果（検索語ごとの分析）
前または後に付加した拡張語を併用，penalty = 2.5

Fig. 2 Effect of query expansion based on adding strings for each query (Using head-added and tail-added together, penalty = 2.5).

得られた．ペナルティを 2.0 や 3.0 にした場合は，他のペナルティのときに比べて相対的に精度が低かった．これは，検索語が編集距離 LD で検出されかつ拡張語が見つからなかった場合と，検索語が編集距離 $LD+m$（$m=2.0, 3.0$）で検出されかつ拡張語が見つかった場合の検出結果を，同じ検索順位として扱うことが適切でないことを示している．本提案手法において同順位で見つかった検索候補については，音声 ID の ASCII 順に出力しているためと考えられる．なお，ペナルティが 2.0 のとき，ペナルティを加えられて編集距離が $LD+2$ となった候補集合 T と，ペナルティなしで編集距離が $LD+2$ となった候補集合 S について，T→S の順で（T と S 内では ASCII 順）で出力した場合はペナルティを 1.5 とした結果に一致し，逆に S→T の順で出力した場合はペナルティを 2.5 とした結果に一致する．

4.3.2 検索語の性質と提案手法の効果

精度が最も高かった前または後に付加した拡張語の併用（ペナルティ 2.5）のときの検索精度と検索語拡張なしのときの検索精度を，100 件の検索語それぞれについて比較した．結果を**図 2** に示す．80 件の検索語で精度が向上し，16 件の検索語で精度が低下した（4 件は変化なし）．大きな精度向上が見られた検索語が多数確認できるのに対して，大きく精度が低下する検索語は存在しないことが確認できる．

次に，検索語長に基づいて検索語を 3 グループに分けて評価した結果を**表 2** に示す．検索語長によらず，提案法により MAP が 0.08 ポイント程度向上していることが分かる．これらのことは，前または後に格助詞を付加する検索語拡張の有効性を示している．

表 2 文字列を付加する検索語拡張の効果（検索語長での比較）
前または後に付加した拡張語を併用，penalty = 2.5

Table 2 Effect of query expansion based on adding strings for query length (Using head-added and tail-added expanded terms together, penalty = 2.5).

検索語長	～8 音素	9～12 音素	13 音素～
検索語数	28	34	38
従来法	0.520	0.615	0.720
拡張	0.605	0.673	0.800
改善	+0.085	+0.058	+0.080

表 3 文字列を付加する検索語拡張の効果（検索語のエントロピーに基づく大域的重み値による分類）
前または後に付加した拡張語を併用，penalty = 2.5

Table 3 Effect of query expansion based on adding strings for query entropy global weight (Using head-added and tail-added expanded terms together, penalty = 2.5).

大域的重み	～0.45	～0.6	～0.75	～0.9	0.9～
検索語数	2	18	33	28	19
従来法	0.501	0.653	0.684	0.589	0.579
拡張（$p=2.5$）	0.468	0.675	0.715	0.669	0.780
改善	−0.034	+0.022	+0.032	+0.080	+0.200
拡張（$p=0.5$）	0.507	0.680	0.710	0.631	0.631
改善	+0.006	+0.027	+0.026	+0.042	+0.052

p: penalty

表 3 に，検索語 Q ごとに式 (3) で定義するエントロピーに基づく大域的重み [17] を求め，分類した結果を示す．

$$g(Q) = 1 + \frac{1}{\log Nd}\left(\sum_d \frac{tf(Q,d)}{\sum_d tf(Q,d)} * \log\left(\frac{tf(Q,d)}{\sum_d tf(Q,d)}\right)\right) \quad (3)$$

ここで，$g(Q)$ は，検索語 Q のエントロピーに基づく大域的重み，Nd は全音声ファイル数，$tf(Q,d)$ は音声ファイル d での Q の出現回数である．ある特定の音声ファイルに偏って出現する語ほど $g(Q)$ の値は大きくなり（最大 1），どの音声ファイルにも同様に出現する語ほど $g(Q)$ の値は小さくなる（最小 0）．

表 3 から，エントロピーに基づく大域的重み $g(Q)$ が高い検索語 Q に対してはペナルティを大きくとったときに効果が大きいことが分かる．逆に $g(Q)$ が低い検索語に対しては，ペナルティは小さい方が良いことが分かる．

この理由について考察を行う．$g(Q)$ が高い検索語は，ある特定の音声ファイルに偏って出現する語である．検索語が特定の音声ファイルで何度も出現している場合は，格助詞を付与した拡張語も見つかりやすい一方，検索語が出現しない音声ファイルでは拡張語が見つかりにくいと考えられる．すなわち，拡張語が見つからない音声ファイルでの検索語候補は誤検出である可能性が高いといえる．した

がって，$g(Q)$ が高い検索語に対しては大きいペナルティを用いることで，多くの誤検出の候補の順位を大きく下げつつ，正解の候補について誤って順位を下げることを抑えることができると考えられる．次に，$g(Q)$ が低い検索語について考える．$g(Q)$ が低いということは，多くの音声ファイルで同程度に出現しているということを示している．$g(Q)$ が高い検索語の場合に比べて，拡張語が見つかりにくい「検索語が出現しない音声ファイル」の割合が低い．すなわち，拡張語が見つからない音声ファイルでの検索語候補が誤検出である可能性は $g(Q)$ が高い検索語の場合に比べて低いと考えられる．このときペナルティが大きいと，正しい検索語候補の順位が大幅に下げられることになり，この悪影響が大きいと考えられる．これらの理由より，エントロピーに基づく大域的重みが大きい/小さい検索語については，それぞれペナルティを大きく/小さくとるのが適当と考えられる．

実際には検索対象における検索語のエントロピーに基づく大域的重みの値を調べることはできないが，ユーザが探したい検索語の性質（ある音声ファイルでたくさん出やすいのか，いろいろな音声ファイルでぱらぱらと出てきやすいのか，など）を知っている場合は，ペナルティの値をユーザに選択させるといった応用が考えられる．

5. 他タスクでの評価実験

4 章では，提案する検索語拡張法が効果的であることを示した．本章では，前に格助詞を付与した 10 語と後に付与した 10 語の両方（20 語）を拡張語とする方法（ペナルティ 2.5）を用いて，種々のタスクで検索語拡張の効果を調べる．ここでも比較のための従来法には連続 DP マッチングに基づく STD（Algorithm 2）を用いる．

5.1 NTCIR-9 SpokenDoc formal run

NTCIR-9 SpokenDoc formal run で評価を行った．これは，NTCIR-9 SpokenDoc dry run と検索対象が同じ（CSJ のすべての講演のマッチドモデルによる単語音声認識結果：Word Corr. = 74.1%，Word Acc. = 69.2%，Syll. Corr. = 83.0%，Syll. Acc. = 78.1% [15]）であり，検索語セットが異なるタスクである．検索語の総数は 50 である．この 50 語について，既知語であるか未知語であるかを区別する情報は NTCIR-9 SpokenDoc のデータには含まれていない．

結果を**表 4** に示す．結果の傾向は dry run の結果と同じであり，検索語を変えても提案する検索語拡張手法は効果があることが確認できた．

5.2 NTCIR-10 SpokenDoc2 formal run

NTCIR-10 SpokenDoc2 [18] formal run で評価を行った．CSJ の 2,702 講演を検索する Large-size タスクと音声ドキュメント処理ワークショップ（第 1 回～第 6 回）での講演音声（104 件）を検索する Moderate-size task があるため，この両者で評価を行った．

表 4 NTCIR-9 SpokenDoc formal run ALL task での評価
Table 4 Evaluation on NTCIR-9 SpokenDoc formal run ALL task.

従来手法	文字列付加検索語拡張			
	penalty	後のみ	前のみ	前と後の併用
0.480	0.5	**0.525**	**0.514**	**0.533**
	1.0	**0.531**	**0.522**	**0.541**
	1.5	**0.568**	**0.551**	**0.581**
	2.0	**0.519**	**0.508**	**0.538**
	2.5	**0.573**	**0.548**	**0.587**
	3.0	**0.532**	**0.508**	**0.550**
	3.5	**0.568**	**0.540**	**0.585**

表 5 NTCIR-10 SpokenDoc2 formal run Large-size task での評価（前または後に付加した拡張語を併用，ペナルティ = 2.5）
Table 5 Evaluation on NTCIR-10 SpokenDoc2 formal run Large-size task (Using head-added and tail-added expanded terms together, penalty = 2.5).

全検索語		既知語検索語		未知語検索語	
従来法	拡張	従来法	拡張	従来法	拡張
0.471	0.569	0.562	0.618	0.394	0.528

5.2.1 Large-size task

これは，4 章の実験と検索対象が同じ（CSJ のすべての講演のマッチドモデルによる単語音声認識結果：Word Corr. = 74.1%，Word Acc. = 69.2%，Syll. Corr. = 83.0%，Syll. Acc. = 78.1% [18]）であり，検索語セットが異なるタスクである．検索語の総数は 100 である．検索対象の各発話に対し，当該発話の音声認識結果の 1-best 候補の音素列を登録した．

結果を**表 5** に示す．検索語には，単語音声認識時に単語辞書に含まれる既知語検索語（46 語）と，含まれない未知語検索語（54 語）が存在するため，それぞれでの評価も行っている．既知語，未知語にかかわらず検索語拡張に効果があることが分かる．未知語検索語に対して，精度の大きな向上（0.394 から 0.528）がみられた．

5.2.2 Moderate-size task

これは，4 章の実験とは，検索対象も検索語セットも異なるタスクである．検索語の総数は 200 である．このうち iSTD タスク（存在しないことを見つけるタスク）用の検索語 100 件は用いず，残りの 100 個の検索語（既知語検索語 47 語，未知語検索語 53 語）を用いた．検索対象は音声ドキュメント処理ワークショップの講演音声（104 件）である．音声認識結果として，タスクオーガナイザから提供された下記の 4 種類のもの [18] を用いた．マッチドモデル，アンマッチドモデルはそれぞれ，音声認識に用いたモデルの学習データが検索対象の音声とマッチするものとしない

ものであり，アンマッチドモデルによる音声認識結果は非常に精度の低いものである．なお，マッチドモデルでの音声認識精度も 4 章での対象（CSJ）よりも低い．いずれの音声認識結果を用いた実験でも検索対象の各発話に対し，当該発話の音声認識結果の 1-best 候補の音素列を登録した．

(1) マッチドモデルによる単語音声認識結果 [18]
Word Corr. = 68.4%，Word Acc. = 63.1%
Syll. Corr. = 79.7%，Syll. Acc. = 75.3%
(2) マッチドモデルによる音節音声認識結果 [18]
Syll. Corr. = 72.7%，Syll. Acc. = 67.7%
(3) アンマッチドモデルによる単語音声認識結果 [18]
Word Corr. = 48.4%，Word Acc. = 43.7%
Syll. Corr. = 67.8%，Syll. Acc. = 62.8%
(4) アンマッチドモデルによる音節音声認識結果 [18]
Syll. Corr. = 60.3%，Syll. Acc. = 55.2%

結果を**表 6** に示す．ここでも既知語と未知語のそれぞれの結果も示してある．なお，音節認識では既知語と未知語の区別はないが，単語認識での既知語/未知語の区分をそのまま使用している．これは，既知語の STD では単語認識結果を使って，未知語の STD では音節認識結果を使うといった使い方が想定されるためである．

マッチドモデル，アンマッチドモデルいずれの場合でも，提案法の有効性が確認できる．既知語については，単語音声認識結果を対象に STD を行った場合に検索精度が高く，提案法でさらに改善が得られていることが分かる．未知語については，音節音声認識結果を対象に STD を行った場合に検索精度が高く，提案法でさらに改善が得られていることが分かる．

これらのことより，音声認識の精度にかかわらず本手法は有効であることが分かった．さらに，既知語の STD では単語認識結果を使い，未知語の STD では音節認識結果を使う STD でも本手法は有効であることが分かった．

表 6 NTCIR-10 SpokenDoc2 formal run Moderate-size task での評価（前または後に付加した拡張語を併用, ペナルティ = 2.5）
Table 6 Evaluation on NTCIR-10 SpokenDoc2 formal run Moderate-size task (Using head-added and tail-added expanded terms together, penalty = 2.5).

音声ドキュメント：単語音声認識（マッチドモデル）

全検索語		既知語検索語		未知語検索語	
従来法	拡張	従来法	拡張	従来法	拡張
0.342	0.395	0.499	0.539	0.203	0.267

音声ドキュメント：音節音声認識（マッチドモデル）

全検索語		既知語検索語		未知語検索語	
従来法	拡張	従来法	拡張	従来法	拡張
0.298	0.362	0.368	0.432	0.235	0.299

※音節認識のため既知語と未知語の区別はない．
単語認識での既知語/未知語の区分

音声ドキュメント：単語音声認識（アンマッチドモデル）

全検索語		既知語検索語		未知語検索語	
従来法	拡張	従来法	拡張	従来法	拡張
0.301	0.367	0.367	0.435	0.243	0.306

音声ドキュメント：音節音声認識（アンマッチドモデル）

全検索語		既知語検索語		未知語検索語	
従来法	拡張	従来法	拡張	従来法	拡張
0.314	0.369	0.348	0.400	0.284	0.342

※音節認識のため既知語と未知語の区別はない．
単語認識での既知語/未知語の区分

5.3 NTCIR-11 SpokenQuery&Doc formal run

NTCIR-11 SpokenQuery&Doc [19] formal run においても，STD タスクが設定されている．これは，5.2.2 項の実験と検索対象は基本的に同じであるが（音声ドキュメント処理ワークショップの 98 講演），検索語が大きく異なるタスクである．検索語には音声検索語とテキスト検索語が用意されているが，本実験ではテキスト検索語を用いた．使用した検索語は，iSTD 用と NON タグがつけられた高頻度の検索語を除いた 203 語（既知語 198 語，未知語 5 語）である．検索対象音声ファイルの音声認識結果には，マッチドモデルによる音節認識結果（Syll. Corr. = 79.6%，Syll. Acc. = 71.1%）[19] を用い，各発話に対して 1-best 候補の音素列を登録した．

結果を**表 7** に示す．ペナルティ $p = 0.5, 2.5$ としたときの結果を，検索語のエントロピーに基づく大域的重みの値ごとに集計している．$p = 2.5$ としたときに，203 検索語の MAP が 0.013 ポイント改善された．4 章の実験タスクでは $p = 2.5$ のときの改善が $p = 0.5$ のときの改善よりも大きかったが，本タスクでは $p = 0.5$ のときの改善が大きかった．これは，本タスクでは検索語の 6 割以上が，$p = 0.5$ とする効果が大きい大域的重みの値が小さい（0.6 以下）ものであり，$p = 2.5$ とする効果が大きい大域的重みの値が大きい（0.75 より大きい）検索語の割合が少なかったことが原因と考える．

5.4 評価実験のまとめ

種々のタスクで提案法の有効性が確認できた．提案法について，異なる検索対象，検索語でも効果的であること，既知語，未知語のどちらにも効果的であること，音声ドキュメントの音声認識精度が異なる場合でも効果がみられること，を確認した．

6. おわりに

STD における検索語拡張の研究を行った．ユーザが入力した初期検索語に対していくつかの拡張語を生成し，その拡張語で連続 DP マッチングを行った結果を用いて元の検索語の検索結果を調整する方法を研究した．拡張語の獲

表 7 NTCIR-11 SpokenQuery&Doc formal run タスクでの検索語拡張の効果（前または後に付加した拡張語を併用）

Table 7 Evaluation on NTCIR-11 SpokenQuery&Doc formal run (Using head-added and tail-added expanded terms together).

大域的重み	～0.45	～0.6	～0.75	～0.9	0.9～	すべて
検索語数	86	43	31	28	15	203
従来法	0.498	0.416	0.302	0.377	0.351	0.423
拡張（$p = 2.5$）	0.477	0.442	0.318	0.429	0.444	0.436
改善	−0.021	+0.027	+0.016	+0.052	+0.094	+0.013
拡張（$p = 0.5$）	0.521	0.457	0.330	0.428	0.401	0.457
改善	+0.023	+0.041	+0.028	+0.052	+0.050	+0.034

p: penalty

得方法として，検索語の前または後に文字列を付加したものを拡張語とする手法を提案した．

講演音声を対象としたSTDにより，提案法に効果があることを示した．提案法が，異なる検索対象，検索語でも効果をもつこと，既知語，未知語のどちらにも効果をもつこと，音声ドキュメントの音声認識精度が異なる場合でも効果をもつことを明らかにした．

謝辞 本研究は科研費（15K00254）の助成を受けた．

参考文献

[1] Akiba, T., Aikawa, K., Itoh, Y., Kawahara, T., Nanjo, H., Nishizaki, H., Yasuda, N., Yamashita, Y. and Itou, K.: Construction of a Test Collection for Spoken Document Retrieval from Lecture Audio Data, *IPSJ Journal*, Vol.50, No.2, pp.82–94 (2009).

[2] 西尾友宏，南條浩輝，吉見毅彦：講演音声ドキュメント検索のための擬似適合性フィードバック，情報処理学会論文誌，Vol.55, No.5, pp.1573–1584 (2014).

[3] 伊藤慶明，西崎博光，中川聖一，秋葉友良，河原達也，胡新輝，南條浩輝，松井知子，山下洋一，相川清明：音声中の検索語検出のためのテストコレクションの構築と分析，情報処理学会論文誌，Vol.54, No.2, pp.471–483 (2013).

[4] Noritake, K., Nanjo, H. and Yoshimi, T.: Image Processing Filters for Line Detection-based Spoken Term Detection, *Proc. INTERSPEECH*, pp.2125–2128 (2011).

[5] Natori, S., Furuya, Y., Nishizaki, H. and Sekiguchi, Y.: Spoken Term Detection Using Phoneme Transition Network from Multiple Speech Recognizers' Outputs, *Journal of Information Processing*, Vol.21, No.2, pp.176–185 (2013).

[6] 大野哲平，秋葉友良：音節継続時間を利用した直線検出に基づく音声検索語検出，情報処理学会論文誌，Vol.54, No.2, pp.484–494 (2013).

[7] 三浦成一，桂田浩一，入部百合絵，新田恒雄：Suffix Arrayを用いた高速STDのための検索閾値の調整手法，第8回音声ドキュメント処理ワークショップ，No.6 (2014).

[8] Takahashi, J., Hashimoto, T., Kon'no, R., Sugawara, S., Ouchi, K., Oshima, S., Akyu, T. and Itoh, Y.: An IWAPU STD System for OOV Query Terms and Spoken Queries, *NTCIR-11 Workshop Meeting*, pp.384–389 (2014).

[9] 坂本伊織，工藤祐介，山下洋一：ベクトル量子化に基づいた音声中の検索語検出における検索結果の統合，第8回音声ドキュメント処理ワークショップ，No.8 (2014).

[10] 牧野光晃，山本直樹，甲斐充彦：分布間距離ベクトル表現による音響的類似度を利用したテキスト及び音声クエリからの音声検索語検出の改善，第8回音声ドキュメント処理ワークショップ，No.10 (2014).

[11] 小田原一成，山下洋一：音声中の検索語検出における単語共起情報の利用，情報処理学会研究報告，2016-SLP-110, pp.1–6 (2016).

[12] 東倉洋一（編集長）：本当に必要な情報を，誰もが見つけられる時代をつくるNTCIRが目指す情報検索の姿，国立情報学研究所ニュース（NII Today），No.48, 大学共同利用機関法人情報・システム研究機構国立情報学研究，pp.4–7 (2010).

[13] 古井貞熙：音声情報処理，森北出版 (1998).

[14] 金田一春彦，林 大，柴田 武：日本語百科大辞典［縮刷版］，大修館書店 (1995).

[15] Akiba, T., Nishizaki, H., Aikawa, K., Kawahara, T. and Matsui, T.: Overview of the IR for Spoken Documents Task, *NTCIR-9 Workshop Meeting*, pp.223–235 (2011).

[16] Maekawa, K.: Corpus of Spontaneous Japanese: Its design and evaluation, *Proc. ISCA & IEEE-SSPR*, pp.7–12 (2003).

[17] 北 研二，津田和彦，獅々堀正幹：情報検索アルゴリズム，共立出版 (2002).

[18] Akiba, T., Nishizki, H., Aikawa, K., Hu, X., Itoh, Y., Kawahara, T., Nakagawa, S., Nanjo, H. and Yamashita, Y.: Overview of the NTCIR-10 SpokenDoc-2 Task, *NTCIR-10 Workshop Meeting*, pp.573–587 (2013).

[19] Akiba, T., Nishizaki, H., Nanjo, H. and Jones, G.J.F.: Overview of the NTCIR-11 SpokenQuery&Doc Task, *NTCIR-11 Workshop Meeting*, pp.350–364 (2014).

南條 浩輝 （正会員）

1999年京都大学工学部情報学科卒業．2001年同大学大学院情報学研究科修士課程修了．2004年同博士後期課程修了．同年龍谷大学理工学部助手．2007年同助教．2015年京都大学学術情報メディアセンター准教授，現在に至る．音声認識・理解の研究に従事．電子情報通信学会，日本音響学会，日本バーチャルリアリティ学会，外国語教育メディア学会，IEEE各会員．2008年度日本音響学会粟屋潔学術奨励賞受賞．

前田 翔

2014 年龍谷大学理工学部情報メディア学科卒業.

吉見 毅彦 （正会員）

1987 年電気通信大学大学院計算機科学専攻修士課程修了．1999 年神戸大学大学院自然科学研究科博士課程修了．（財）計量計画研究所（非常勤），シャープ（株）を経て，2003 年より龍谷大学理工学部勤務.