社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－NL－185 (5)
2008－SLP－ 71 (5)
2008／5／23

ポッドキャストを対象とした類似エピソード検索手法

水野 淳太† 緒方 淳‡ 後藤 真孝‡

† 奈良先端科学技術大学院大学
‡ 産業技術総合研究所

†junta-m@is.naist.jp
‡{jun.ogata,m.goto}@aist.go.jp

あらまし 本稿では，エピソードと呼ばれる音声ファイルの集合から成るポッドキャスト（音声ブログ）を対象とした，類似エピソードの検索手法について述べる．動画共有サイト等で，あるコンテンツの再生後に関連・類似したコンテンツを提示する機能を持つものが多いが，それらは書誌情報やタグ，ユーザの視聴履歴に基づいている．本稿では，エピソードを音声認識した結果に基づいて，音声認識結果を confusion network に変換し，そこからエピソードを特徴づけるキーワードセットを抽出して，キーワードセット間の類似度を計算することで，関連エピソードを検索・提示できる手法を提案する．単語正解率や話者数など，傾向の異なるいくつかのエピソードに対して実験を行い，本手法がどのような場合に有効であるかについて評価を行った．本成果は，音声認識に基づくポッドキャスト検索サービス PodCastle で，関連エピソードを提示するためにも利用できる．

A Similar Episode Retrieval Method for Podcast

Junta Mizuno† Jun Ogata‡ Masataka Goto‡

†Nara Institute of Science and Technology(NAIST)
‡National Institute of Advanced Industrial Science and Technology(AIST)

Abstract

Given podcasts (audio blogs) which are sets of speech files called episodes, this paper describes a method for retrieving similar episodes. Although video sharing services usualy have a function of showing a set of relevant/similar content after playing back a piece of content, they are based on bibliographic information, tags, and users' playback behaviours. In this paper, we propose a method that extracts keywords from confusion networks converted from speech recognition results and then retrieves and shows relevant episodes on the basis of similarity between those keywords. We evaluated this method using several episodes including a variety of speech recognition accuracy and the number of speakers. This result can be applied to show relevant episodes on PodCastle, a podcast search service based on speech recognition.

1 はじめに

近年，電子テキストによるブログ (Weblog) が急速に普及してきたが，その音声版に位置付けられるのがポッドキャストであり，ウェブ上の音声データとして多数公開されている．iPod などのデジタルオーディオプレイヤや計算機のメディアプレイヤで視聴することができ，個人の日記からニュースまで，その内容は多岐に渡る．ポッドキャストは，一連のエピソードと呼ばれる音声データ (MP3 ファイルなど) に加え，その流通を促すために，ブログなどで更新情報を通知するために用いられているメタデータ RSS が付与されている (図 1)．エピソードは，作成者 (ポッドキャスタ) が任意のタイミングで追加できる．この仕組みによって個人による音声データの発信，流通，入手が容易にできる点が，ポッドキャストの普及を促してきた．そして，ウェブ上のテキストに対して全文検索サー

メタデータ
　タイトル：「JUNK 爆笑問題 カーボーイ 」
　概要：「レギュラー番組や雑誌の連載を多数抱える超売れっ子… 」
エピソード1
　タイトル：「2006年04月04日 」
　MP3：http://podcast.tbsradio.jp/bakusho/files/20060404.mp3
エピソード2
　タイトル：「2006年04月11日 」
　MP3：http://podcast.tbsradio.jp/bakusho/files/20060411.mp3
エピソード3
　タイトル：「2006年04月18日 」
　MP3：http://podcast.tbsradio.jp/bakusho/files/20060418.mp3
エピソード...
　エピソードは、毎日、毎週など、任意のタイミングで追加されていく

図 1: ポッドキャストの例

ビスが不可欠になっているのと同様に，音声データに対しても全文検索を行う重要性が増している [1].

そうした音声情報検索用 Web サービスである PodCastle[1] では，ポッドキャストを音声認識によってテキスト化することで，ユーザがポッドキャストをテキストで検索，閲覧でき，さらに音声認識結果の誤りを訂正することができる [2][1][3]．現在，PodCastle でエピソードを探すには以下の 3 つの方法がある．

- 任意のテキストを検索語として与え，その検索結果一覧から選択
- ポッドキャスト一覧から選び，そのポッドキャスト内のエピソード一覧から選択
- 各種ランキング（アクセスの多いエピソード，訂正回数の多いエピソード等）から選択

しかし，これらの方法では関連・類似したエピソードを探し出して聴くことは難しい．例えば，同じポッドキャスト内ではエピソード同士の関連性が比較的高く，それらを視聴することはできるが，他のポッドキャストに含まれる関連エピソードを見つけることはできない．

関連するコンテンツを探しやすくするために，Youtube[2]等の動画共有サイトでは，書誌情報やタグ，ユーザの視聴履歴に基づいて，あるコンテンツの再生後に関連・類似したコンテンツを提示する機能を持つものが多い．しかし，ポッドキャストの RSS ではエピソードごとの説明文が不十分なことが多く，説明文を元にした類似エピソードの提示は難しい．また，視聴履歴から，同一人物が一連のコンテンツを視聴したことがわかったとしても，必ずしもそれらが内容的に関連・類似しているとは限らない．

[1] http://podcastle.jp
[2] http://www.youtube.com

そこで本研究では，RSS 中の説明文やユーザの視聴履歴等を一切用いずに，音声認識結果，つまりコンテンツそのもののみを利用して，類似エピソードの検索を行う手法を提案する．具体的には，各エピソードの音声認識結果から，そのエピソードを特徴づけるようなキーワードを抽出し，それをもとに類似度を計算して，類似エピソードの提示を行う．将来的には，この類似度に基づいてクラスタリングをすることで，“ニュース” や “スポーツ” 等のように内容に基づいてエピソードを分類できる可能性もある．

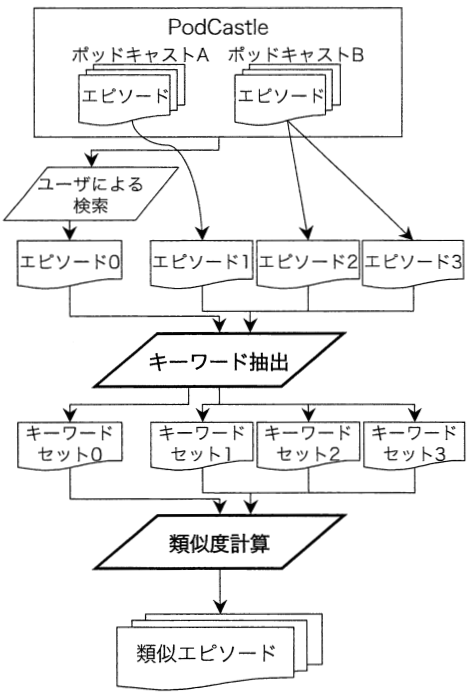

図 2: 類似エピソード検索手法

2　類似エピソード検索手法

類似エピソードを検索することは，自然言語処理研究における類似文検索に相当する．類似文検索には GETA[4][3]や，広い意味では検索エンジンがあげられる．ただし，検索エンジンでは比較する二つの文のうち，片方はクエリであり，短い単語の組み合わせで構成される場合が多い．それに対し GETA では，長短を問わない自然文を入力と

[3] 汎用連想計算エンジン http://geta.ex.nii.ac.jp

して，それに類似した文を抽出することができる．しかし，本研究に直接これらの方法を適用することはできない．ポッドキャストのエピソードには，内容に直接関係しないような挨拶や相づち，フィラーなどが多いため，それらを内容語としては扱えないからである．また，そもそも認識誤りが存在するため，誤認識された単語が悪影響して，検索性能を下げてしまう．

そこで，エピソードを特徴づけるキーワードを抽出し，それをもとにした類似度を計算し，類似エピソードを検索，提示する．本研究で構築した類似エピソード検索手法の概要を図 2 に示す．図 2 の例では，まず，事前にすべてのエピソード (エピソード 0 ～ 3) について，キーワードを抽出しておく (キーワードセット 0 ～ 3)．そして，ユーザが検索などで求めた任意のエピソード (ここではエピソード 0 とする) に対して類似したエピソードを得るには，キーワードセット 0 に対して，キーワードセット 1 ～ 3 のそれぞれについて類似度を計算し，近いものから順に提示する．

3 キーワード抽出

キーワードの抽出には TF-IDF 法を用いる．TF-IDF 法は，ある単語について，それが含まれるドキュメントにおける重要度を付与する手法である．例えば単語 w_i について，その値は以下のようにして計算することができる．

$$\mathrm{TF}_i = \frac{n_i}{\sum_k n_k} \quad (1)$$

$$\mathrm{IDF}_i = \log \frac{\#D}{\#d+1} \quad (2)$$

$$\mathrm{TFIDF}_i = \mathrm{TF}_i * \mathrm{IDF}_i \quad (3)$$

ただし，

n_i	単語 w_iの出現頻度
$\#d$	単語 w_iの出現するエピソード数
$\#D$	全エピソード数

つまり，TF(term frequency) によって単語 w_i のドキュメント内における重要度を頻度によって表し，IDF(inverted document frequency) によってその単語の特定性を表し，それらを積算することで，重要度を計算している．経験に基づいた単純な手法であるが，効果的であることが知られており，広く利用されている [5]．

音声認識文書においても 1-best の結果に対しては，通常の電子テキストと同様にして計算することができる．しかし，音声認識文書は通常の電子テキストと異なり，認識誤りが存在するため，認識結果の 1-best に対する計算では，1-best に含まれない単語はキーワードとして抽出できないという問題がある．

そこで，音声認識器によって出力される confusion network[6] を利用した TF-IDF 法を提案する．

3.1 confusion network

confusion network は，音声認識における中間結果 (複数候補) をシンプルかつコンパクトに表現したものである [6]．confusion network は，単語グラフ (図 3-a) を音響的なクラスタリングによりリニアな形式 (図 3-b) に圧縮することで求めることができる．ここで"sil" (silence) は発話開始，終了時の無音を表し，アルファベット 1 文字はグラフのリンク上の単語名を表している．また，図 3-b の

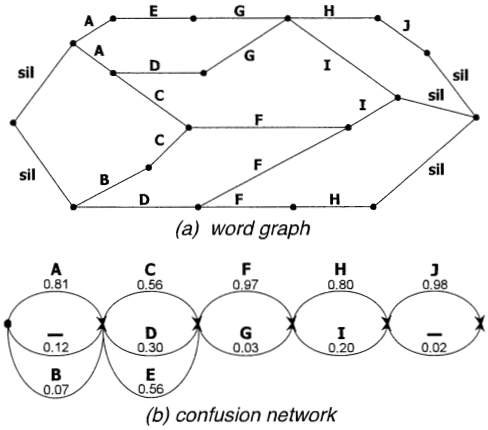

図 3: 単語グラフと confusion network の模式図

ネットワーク上の"-"は削除候補である．音響的クラスタリングは以下の 2 つのステップにより行われる [6]．

1. 単語内クラスタリング: 単語名が同一で，時間的に重なりのあるリンクをクラスタリングする．時間的類似度をコスト関数として用いる．
2. 単語間クラスタリング: 単語名の違うリンクのクラスタリングを行う．コスト関数として単語間の音響的類似度を用いる．

confusion network の各リンクには，クラスタリングした各クラス (単語の区間) ごとに事後確率が算

出され，それらの値は，各クラスでの存在確率，あるいはそのクラス内の他候補との競合確率を表す．各クラスのリンクは，存在確率の大きさでソートされ，認識結果として可能性の高いリンクほど上位に配置される．各クラスから事後確率が最大となるリンクを選択すると，1-best の認識結果 (最尤の単語列) が抽出される．

3.2 confusion network を利用した TF-IDF 法

confusion network において，すべての単語には音声認識結果としての信頼度が事後確率という形で付与されているので，それを TF-IDF の計算に利用することができる．胡らも TF-IDF に confusion network の確率値を用いて音声ドキュメント検索を行い，1-best 結果に比べ，confusion network を利用した場合の方が検索性能が向上することを報告している [7]．ただし，彼らの目的はクエリによる音声ドキュメント検索である．そのため，少ない単語で構成されるクエリと，音声ドキュメントから抽出された多くのキーワードとを比較するために，類似度の定式化について工夫がなされている．

本研究の目的はエピソード同士のキーワードを用いた類似度の算出であるため，類似度の計算は一般的な尺度を用いればよい．ここでは，どのようなキーワードを抽出するかが重要であるため，TF-IDF の定式化が重要である．本研究では特に IDF について 2 種類の定式化を行い，TF-IDF の定式化について実験による評価を行った．

3.2.1 TF の定式化

まず TF について，単語 w_i の TF は，w_i のエピソード内での全確率値を足したものとして，以下のように定式化する．

$$\mathrm{TFp}_i = \sum_k P_k(w_i) \tag{4}$$

ここで，$P_k(w_i)$ はドキュメント内の位置 k における単語 w_i の事後確率である．

これによって，信頼度が低い単語も TF の計算に影響を与えることができ，図 3-b の C の 0.56 のように，信頼度は高いがその数値がそれほど高くない場合，TF に対する影響を低くすることができる．この TF を TFp と呼ぶ．

3.2.2 IDF の定式化

IDF については，2 つの定式化が考えられる．まずは，信頼度の高低に関わらず，すべての認識候補を同率に扱う定式化である．つまり，事後確率の高低に関わらず，一回の出現を一回と数える．この IDF を IDFn と呼ぶ．しかし，IDFn では信頼度の低い単語，すなわち実際には発話されていない可能性の高い単語も一回の出現として数えるため，IDF が低くなりがちである．

そこで，すべての認識候補を同率に一回の出現として数えず，その単語のドキュメント内での confusion network における最大確率値をその出現回数として用いる．単語 w_i について以下のように定式化し，この IDF を IDFp と呼ぶ．

$$\mathrm{IDFp}_i = \log \frac{\#D}{\max(P_k(w_i)) + 1} \tag{5}$$

ただし k はドキュメント内の出現位置

3.2.3 TF-IDF の定式化

ベースラインとして 1-best の場合の TF-IDF の計算も行う．一般的な電子テキストと同様にして TF と IDF を式 1〜式 3 によって計算する．この場合の TF を TFb，IDF を IDFb と呼ぶ．このとき，1-best の認識候補以外は利用されず，音声認識の事後確率も利用しない．

TF-IDF は TF と IDF の積算で計算するので，最終的に以下の 3 種類の TF-IDF を計算する．

TFb-IDFb 音声認識結果の 1-best のみを利用した，ベースラインとなる TF-IDF

TFp-IDFn TF について confusion network の事後確率を利用し，IDF について事後確率に関わらず 1 回の出現を 1 回と数えた場合の TF-IDF

TFp-IDFp TFp-IDFn と同じ TF と，IDF についても事後確率を利用した TF-IDF

TFp-IDFn と TFp-IDFn で異なるのは IDF であり，IDFn の方が低い値となる．

本研究では，ベースラインとなる TFb-IDFb に対して，confusion network を利用した TF-IDF である TFp-IDFn と TFp-IDFp の類似エピソード検索における有効性について実験による評価を行う．

4 類似エピソード検索

3 で抽出されたキーワードを利用してエピソード間の類似度を計算する．類似度はベクトル空間モ

表 1: 評価に用いたエピソード

ID	ポッドキャスト	内容	種類	単語正解率
colm1	森永卓郎 経済コラム	- 住民税と所得税の話題 - 税制改正が裕福な人に優遇されていることへの指摘	コラム	91.32%
colm2	伊藤洋一のビジネストレンド	- ユビキタスについての話題 - 銀行で領収証が出せなくなる		52.40%
news1	読売ニュース ポッドキャスト	- 一日のニュース (複数の話題)	ニュース	85.04%
news2	聴くトク報知	- 風味堂の 1 年にわたる PV 撮影 - 流行語の特許が出願されていた		81.19%
talk1	サイエンス・サイトーク	- 数分程度の CM - ダイエットの話題	雑談 (複数話者)	48.25%
talk2	文化系トークラジオ Life	- プロポーズの話題		32.60%

デルを用いて計算する．このモデルは各キーワードを次元とし，その重みから成るベクトルとして比較する 2 つのエピソードをそれぞれ表現し，2 つのベクトルの成す角度の余弦から類似度を計算する．

5 実験

PodCastle に登録されているエピソードを利用して類似エピソードの検索を行い，その評価を行った．

5.1 データ

実験に利用したエピソードは全部で 12184 個で，その中で TF-IDF の計算を行った．評価は，単語正解率や内容の異なるエピソードを用いた．本実験で扱ったポッドキャストには，大きく分けて “コラム”，“ニュース”，“雑談 (複数話者)” の 3 種類のエピソードがあり，それぞれについて単語正解率度の違いによって 2 個ずつエピソードを選び，計 6 個のエピソードについて類似エピソードの検索を行い，その評価を行った．表 1 にその一覧を示す．

5.2 キーワード抽出

2 つのエピソードについて，抽出されたキーワードの上位 10 個を表 2 ～ 表 4 に示す．

キーワード抽出では，そのエピソードを特徴づけるような単語の TF-IDF が高く，上位となることが期待され，また認識誤りの単語，つまり実際には発話されていない単語や，直接内容に関係しないような挨拶やフィラーなどが下位となることが期待される．キーワードの上位 100 個において，認識誤りの単語は平均して TFb-IDFb で 28%，TFp-IDFn で 23%，TFp-IDFp で 19%含まれていた．TF-IDF に confusion network の事後確率を利用することで認識誤りの単語をキーワードとして下位にすることができている．

キーワードがそのエピソードを特徴づけているかどうかを直接評価することは難しい．エピソードによっては明らかに特徴的な単語があり，それらは上位となるべきである．しかし，特徴的すぎるために，他の類似エピソードには含まれない場合もある．適度に特徴的な単語が，類似エピソード検索においては重要となるが，それらを判別することは人手でも難しい．そのため，抽出されたキーワードについてその性能を直接評価はせず，それらを用いた類似エピソード検索の結果によってキーワード抽出の性能についても評価を行う．

また，3 種類の TF-IDF で同じ順位となっていても，その TF-IDF は異なるため，類似エピソード検索において，その効果は異なる．より特徴的な単語について，より高い値となることが望ましい．

5.3 類似エピソード検索

抽出されたキーワードを利用して類似エピソードの検索を行った．キーワードの上位 100 個を用い，それらの類似度の尺度にはベクトル空間モデルの余弦を用いて実験を行った．なお予備実験では，上位 10 個を用いる場合や全部用いる場合も試したが，ほとんどの場合で上位 100 個の結果の方が優れていた．そして，類似度の高い上位 10 件を，検索結果の類似エピソードとして評価した．

類似しているかどうかの判断は実際に視聴した上で人手で行い，“よく似ている” “似ている” “少し似ている” の 3 段階にスコアとして 3 ～ 1 を割り当て，似ていないものは 0 とした．

エピソードごとにどのような評価尺度を用いるべきかを考える．ユーザの立場にたってみると，類似エピソードとして順位づけをして提示された場合，本当に類似しているかどうか，ユーザが満足するようなエピソードであるかは実際に視聴して

表 2: 抽出されたキーワード：colm2

TFb-IDFb	TFp-IDFn	TFp-IDFp
ユビキタス	ユビキタス	ユビキタス
振り込み	郵便局	郵便局
郵便局	振り込み	振り込み
領収証	領収証	システム
システム	教習所	教習所
教習所	ユビキタスコミュニケータ	領収証
コンピュータ	コンピュータ	コンピュータ
用紙	システム	用紙
振り込む	ドラえもん	社会
支払い	携帯電話	支払い

表 3: 抽出されたキーワード：news2

TFb-IDFb	TFp-IDFn	TFp-IDFp
ＰＶ	ＰＶ	ＰＶ
風味	風味	風味
堂	クリスマス	堂
クリスマス	商標	クリスマス
筑紫	堂	商標
商標	密着	どん
区分	ハラフウミ	筑紫
どん	筑紫	区分
酒造	ユーキャン	密着
出願	ノミネート	出願

表 4: 抽出されたキーワード：talk1

TFb-IDFb	TFp-IDFn	TFp-IDFp
ダイエット	キロカロリー	キロカロリー
キロカロリー	ダイエット	ダイエット
キロ	村田製作所	キロ
カロリー	カロリー	カロリー
村田製作所	主人公	村田製作所
体重	キロ	主人公
主人公	製作所	五十
低い	基礎代謝	低い
ちゃう	翌々日	ちゃう
我慢	見た目	体重

みなければ分からない．そのため，類似度がもっとも高いと判定されたエピソードを聴き，それで満足しなければ次のエピソードを聴く．これを満足するまで繰り返すという行動をとると考えられる．そこで，以下の 3 つの評価尺度を定義し，それぞれについて評価を行った．

- 評価尺度 1 一つめの評価尺度は，類似度スコア 3 のエピソードが何位に提示できたかということである．スコア 3 のエピソードを 1 位として提示できたシステム A は，同じエピソードを 2 位に提示したシステム B よりも優れたシステムであると言える．システム A がスコア 3 のエピソードを 1 位に提示することで，ユーザは 1 つのエピソードを聴くだけで満足し，それ以下のエピソードを視聴しない可能性が高い．

- 評価尺度 2 二つめの尺度は，上位 10 件のうち，いくつの類似エピソードを提示できたかという尺度である．上位 10 件のエピソードをすべて視聴した場合に，ユーザの満足できるエピソードがどれだけあったかという点で，多くの類似エピソードを要求するユーザの満足度として評価できる．スコアに対して MRR(Mean Reciprocal Rank) を計算する．

$$\mathrm{MRR2} = \frac{1}{n}\sum_{k=1}^{n}\frac{1}{4-s_k} \tag{6}$$

ここで，n は上位何件まで評価するかによって変化し，ここでは 10 である．s_k は上位 k 番目の類似エピソード候補の評価値 (0 ～ 3) である．

- 評価尺度 3 三つめの尺度は，評価尺度 2 に順位による重み付けを行ったものである．評価尺度 2 では，スコア 3 のエピソードを 1 位に提示できたか 10 位に提示できたかということを区別しないが，評価尺度 3 ではそれを区別する．評価尺度 2 と同じようにスコアを用い，上位 n 番目のスコアに $1/n$ を積算し，最終的なスコアが 0 ～ 1 になるような正規化を行う．

$$\mathrm{MRR3} = \frac{\sum_{k=1}^{n}\frac{1}{k}\frac{1}{4-s_k}}{\sum_{k=1}^{n}\frac{1}{k}} \tag{7}$$

評価尺度 1 では，類似エピソード候補 10 件に対して，スコア 3 のエピソードが最初に何位に提示できたか，その順位で評価する．評価尺度 1 による評価結果を表 5 に示す．評価尺度 2，3 では，類似エピソード候補の上位 10 件について，最終的に 0 ～ 1 のスコアが得られる．1 に近い方が優れた結果である．評価尺度 2，3 による評価結果を図 4 に示す．

表 5: 実験結果：評価尺度 1

ID	TFb-IDFb	TFp-IDFn	TFp-IDFp
colm1	1	1	1
colm2	3	9	10
news1	1	1	1
news2	2	2	1
talk1	1	1	1
talk2	1	2	2

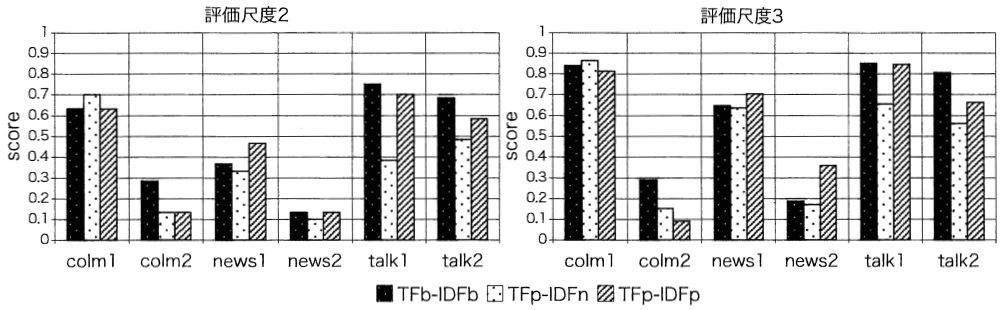

図 4: 実験結果：評価尺度 2,3

6 考察

人手による評価はコストがかかるため，多くのエピソードについて評価を行い，その傾向を観察することは難しい．そこで，エピソードごとにどのTF-IDF を用いた結果が良いかを述べ，confusion network を利用した場合の効果についてそれぞれ考察する．

6.1 エピソードごとの考察

- colm1 単語正解率が高いため，TFb-IDFb でも良い検索結果が得られた．TFp-IDFn，TFp-IDFp では他の認識候補も利用しているが，confusion network の事後確率を利用することで，それらは悪影響を与えていない．

- colm2 特定性の高いキーワードが得られなかったため，全体的に良い検索結果が得られていない．“銀行” のような一般的な単語が，ここでは “ユビキタス” に関連してキーワードとなっている (表 2) が，このような一般的ではない用法の単語はキーワードとして良く機能しないと考えられる．

- news1 一日のニュースであるため，多くの話題を扱っている．ここでは，どれかの話題に類似していれば類似エピソードであると判断した．全体的に良い検索が行えているが，TFp-IDFp を用いることでもっとも良い検索結果が得られた．

- news2 流行語に関する類似エピソードが多く得られたが，キーワードとして “どん[4]”(表 3) が効果的であり，これに対する重みがもっとも高い TFp-IDFp によって良い検索結果が得られた．評価尺度 3 ではスコアの高いエピソードを何位に提示できたかを区別しているので，より顕著な差が出ている．

- talk1 このエピソードは前後に CM があり，他のエピソードにもまったく同じ CM を用いたものがあるため，それらと類似性が高いと判定される結果が目立った．TFp-IDFn のみが低い結果であるが，キーワードが一般的な単語 (“ダイエット”) である場合 (表 4)，IDF を低く評価すると，それらのキーワードへの重みが，IDF の高い他のキーワードに比べて相対的に小さくなるためである．そのため TFp-IDFp では TFb-IDFb とほぼ同じ結果が得られている．

- talk2 すべての評価尺度において TFb-IDFb がもっとも高い．colm2 と同様に，キーワードを特別な意味で利用しているため，TFp-IDFp と TFp-IDFn による検索結果は良いとは言えない．また，単語正解率が低く，訂正候補にも正しい認識単語が含まれていない場合が多いため，confusion network を利用しても改善が行えず，逆に訂正候補が悪影響を与えてしまったと考えられる．

6.2 総評

全体的に，単語正解率がある程度高い場合に confusion network を利用した TFp-IDFp によって良い検索結果が得られた．これは，1-best 以外の他の認識候補中に正しい認識結果が含まれている場合であり，本研究で改善できると考えられるエピソードである．TFb-IDFb に比べて colm2 と talk2 については良い検索結果が得られなかった．この二つのエピソードは単語正解率が低く，そのため confusion network を利用しても正しい認識候補が得られにくい．予備実験において，キーワードの上位 100 個では認識誤り単語数が少なくなったが，上位 10 個では TFb-IDFb よりも TFp-IDFn と TFp-IDFp の方が認識誤り単語を含んでいた．これは TFb は

[4]流行語の “どんだけ ～” が音声認識の過程で “どん” と “だけ” に分かれてしまい，“だけ” は IDF が低い (他ドキュメントでも頻出する) ので，“どん” だけが残っている

TFp や TFn よりも高い値となるためである．同じように単語正解率の低いエピソードには talk1 があるが，このエピソードでは “ダイエット” という明確なキーワード (表 4) があるため，良い検索結果が得られた．

TFp-IDFn と TFp-IDFp との比較だが，TFp-IDFn は IDF に事後確率を利用していないため，IDF を低く計算する傾向にある．そのため，特定性の高いキーワードが得られるが，類似エピソード検索においては TFp-IDFp に比べて良い結果ではない．これは，キーワードが特徴的過ぎるためであると考えられる．特徴的なキーワードはそのエピソードのみにしか表れていない場合もあり，それらは余弦の計算に影響しない．評価エピソードすべてにおいてこの現象が表れていた．

7 まとめと今後の課題

本研究ではポッドキャストを対象として，コンテンツ内容に基づく類似エピソード検索手法を提案し，その評価を行った．confusion network を利用して TF-IDF を定式化することで，単語正解率の高いエピソードについては，高い検索性能が得られることが分かった．しかし，単語正解率が低い場合には，認識候補が悪影響を与えている．これは，式 3 において，TF と IDF の重みを調整することである程度改善できると考えられる．また，confusion network の事後確率を利用することで，エピソードにおける単語正解率が推測できる．これを利用し，単語正解率によって適用する TF-IDF の種類を変更するということも考えられる．

利用するキーワード数について，100 個という一定数ではなく，何らかの尺度で可変にすることが挙げられる．ポッドキャストには数分から数時間の長さのエピソードが混在しているため，それらに対して一定数のキーワードを利用すると，TF-IDF に偏りが生じてしまう．今後は，例えばエピソードの長さに比例した数だけ利用するといったことが考えられる．

検索性能の評価について，現状では人手による視聴を伴うため大規模な評価を行うことが難しい．今回は 1 人による評価であるため，評価に偏りが生じる可能性もある．今後は，複数人による評価を行い，その一致度も評価する必要がある．

さらに，エピソード間の類似度をベクトル空間モデルの余弦として定式化しているので，それをもとに，内容の類似度に基づいたクラスタリングを行うことが考えられる．クラスタの大きさが小さくなるように調整することで，必要十分な数の類似エピソードだけを提示することができ，ある程度大きくすることで，内容によって分類した，ポッドキャスト間を越えたエピソードのまとまりを作成することができると考えられる．これはユーザにとってより多くの他のエピソードを視聴する動機付けとなるだろう．

謝辞: 本研究の一部は，科研費 (19300065) の助成を受けた．

参考文献

[1] 後藤真孝, 緒方淳, 江渡浩一郎. PodCastle の提案: 音声認識研究 2.0 を目指して. 情報処理学会 音声言語情報処理研究会 研究報告 2007-SLP-65-7, pp. 35–40, 2007.

[2] 緒方淳, 後藤真孝, 江渡浩一郎. PodCastle：ポッドキャストをテキストで検索，閲覧，編集できるソーシャルアノテーションシステム. 第 14 回インタラクティブシステムとソフトウェアに関するワークショップ, pp. 53–58, 2006.

[3] 緒方淳, 後藤真孝, 江渡浩一郎. PodCastle の実現: Web2.0 に基づく音声認識性能の向上について. 情報処理学会 音声言語情報処理研究会 研究報告 2007-SLP-65-8, pp. 41–46, 2007.

[4] 高野明彦, 西岡真吾, 今一修, 岩山真, 丹羽芳樹, 久光徹, 藤尾正和, 徳永健伸, 奥村学, 望月源, 野本忠司. 汎用連想計算エンジンの開発と大規模文書分析への応用. 平成 13 年度成果報告集 情報処理振興事業協会, 2001.

[5] 藤田邦彦. 徳永健伸 (著), 情報検索と言語処理, 言語と計算シリーズ 5, 東京大学出版会, 1999 年, ISBN4-1-065405-5. *Journal of Information Processing Society of Japan*, Vol. 41, No. 5, p. 606, 2000.

[6] L. Mangu, E. Brill, and A. Stolcke. Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks. In *Computer, Speech and Language*, pp. 373–400, 2000.

[7] 胡新輝, 呉友政, 柏岡秀紀. Confusion Network を用いた音声ドキュメントの検索及び評価に関する研究. 第 2 回音声ドキュメント処理ワークショップ, 2008.