

Twitter を用いたラジオ番組圧縮手法の提案

太田 暢[†] 村山 太一[†] 矢田竣太郎[†] 荒牧 英治[†] 若宮 翔子[†]

[†] 奈良先端科学技術大学院大学 〒 630-0192 奈良県生駒市高山町 8916-5

E-mail: †{ota.itaru,og4,murayama.taichi.mk1,s-yada,aramaki,wakamiya}@is.naist.jp

あらまし 音声メディアの代表格であるラジオは、日本においては 2010 年代に各種インターネットサイマル配信サービスが整備されスマートフォンや PC で放送を聴取することが可能となり、地理的・時間的制約を克服したアクセスしやすいメディアとして再注目されるようになった。コロナ禍においてはリモートワークや巣ごもりの息抜きとしてラジオ番組を聴取する事例が増えるなど、近年さらにリスナーを増やしている。放送後に番組を聴取するリスナーのうち、運動中や通勤時などの限りある時間の中で聴取したいと考えるリスナーには、制限時間内に一部始終を一度に聴取出来ず、中断の度に遡って再生することになったり、中断したまま聴取期限が過ぎてしまうといった不便が付き纏う。そこで本稿では、Twitter のデータを用いてラジオ番組の音源を圧縮する手法を提案することでラジオリスナーのユーザ体験向上を試みた。具体的には、放送時間中のハッシュタグ付きツイートの投稿数に基づき番組内で盛り上がった箇所以外を優先的に高速化し、それ以外の箇所は等倍に近い速度に調整して、後から番組を聴くユーザが指定した制限時間内に放送音源を圧縮する手法を二つ提案する。倍速加工を施した音源の比較評価実験の結果、番組における文脈を考慮して再生速度を変更する手法を適用した音源が最も正しく圧縮され、満足度が高いことがわかった。

キーワード ソーシャルメディア, Twitter, 音声メディア, ラジオ, ポッドキャスト, UX

1 はじめに

20 世紀初頭の誕生から音声メディアの代表格であるラジオは、日本では 2010 年代に入ると「radiko」や「NHK ネットラジオ らじる★らじる」といったインターネットプロトコル (IP) サイマルラジオサービスが展開され、スマートフォンや PC でクリアな音質の放送を聴取することが可能になった。2010 年代中頃、「radiko」に全国のラジオ局のリアルタイム放送や過去 1 週間分の放送を聴取する機能が追加されると、ラジオは地理的・時間的制約を克服したアクセスしやすいメディアとしての認知が広まり、2010 年代後半には、スマートスピーカーが普及し始めたことで聴取環境がさらに増加した。コロナ禍においては、リモートワークや巣ごもりの息抜きとしてラジオ番組を聴取する事例が増えている。2020 年 11 月に実施された「radiko」ユーザへのアンケート調査¹によると、コロナ禍において「radiko」の月間アクティブユーザは 150 万人増加し、既存ユーザのラジオや「radiko」の聴取時間も増加したことが報告されている。このように近年のラジオは「ながら」利用が可能な聴覚特化型メディアとして再評価を受け、リスナーを増やしている。

2010 年代はスマートフォンの普及とともにソーシャルメディアの利用者が増加した時代でもあった。その影響は放送メディアにも広がり、テレビやラジオの視聴者が番組の実況や感想をソーシャルメディアに投稿するようになった。主要ソーシャル



図 1 「radiko」におけるソーシャルメディアへのシェア機能

メディアの一つである Twitter においては、番組に関連する単語や文章、ハッシュタグ (#番組名) がトレンドランキングに入ることを企図して、作り手が番組内外で視聴者に投稿を促す事例も見受けられる。なかでもラジオは、視覚を阻害しないという特徴からソーシャルメディアとの相性が高く、番組指定のハッシュタグ付き投稿や番組関連用語を含む投稿は、リスナーの反応を示すデータとして番組制作に用いられている。例として、メールよりも気軽なリスナーとのインタラクション手段としてツイートが放送内で取り上げられたり、関連用語のトレンドランキング入りが番組継続の判断材料となることなどが挙げられる。また、「radiko」の番組シェア機能の画面 (図 1) では、

1: 『コロナ禍でラジオが復権!? 民放ラジオ業界 70 年の歴史を塗り替えた radiko (ラジコ) の「共存共栄型 DX」とは』 Digital Shift Times : https://digital-shift.jp/dx_strategy/210708

Twitter を選択すると番組に関連するハッシュタグが自動でツイートに付帯される仕様となっている。ラジオ局は「radiko」における番組の再生数などのデータを分単位で取得することが可能であり、2022 年現在日本で唯一ラジオ番組の聴取率を調査しているビデオリサーチ社は、「radiko」のユーザデータをモデル化して毎日のラジオ聴取状況を推計した「ラジオ 365 データ」を 2020 年 4 月から導入している。このように従来の聴取率以外で番組の人気を示すデータの価値が高まる中で、ソーシャルメディアは聴き逃し配信プラットフォームへの導線として重視されていることがわかる。

放送後に聴き逃し配信サービスなどを利用して番組を聴取るリスナーが感じる不便の一つに、運動中や通勤時など任意の制限時間内に一部始終の内容を理解しつつ一度に聴取することが難しいことがある。具体的には、通勤の行きと帰りなどに分けて番組を聴取する際、最後に聴き終えた箇所の直前から聴き直す必要があったり、残りの箇所を聴くことを忘れて聴取期限が過ぎてしまうなどの問題が生じる。標準的な解決策として、1.5 倍速、2 倍速などの再生環境に備え付けられている倍速機能を利用してユーザが再生速度を変更する方法が考えられるが、この方法では求める制限時間に収まらない可能性がある。また、一樣に速度を高速化するため、番組内で盛り上がった箇所などのじっくり聴きたい箇所も高速で再生されてしまい、情報を聴き落とすたくないとするユーザにとって望ましくない。別の解決策として、ラジオ番組のホームページなどで事前に公開されているタイムスケジュールに基づいて自動的に任意の箇所を抽出する方法も考えられる。しかし、収録番組であってもタイムスケジュール通りに厳密に進行されることは珍しく、生放送の番組では前後関係を考慮してコーナーをスキップすることが頻繁にあるなど、この方法による解決は難しい。

そこで、本稿ではソーシャルメディアのデータをリスナーのユーザデータとして活用した、ラジオ番組音源の圧縮手法を提案する。ソーシャルメディアには、リアルタイムで不特定のユーザが同一トピックについて複数投稿を行うという性質から、特にラジオ番組聴取との親和性が高いと考えられる Twitter を用いる。「ラジオ番組の面白さは Twitter のツイート数の盛り上がりで反映される」と仮定し、ツイート数に基づき番組の盛り上がり箇所を特定する。番組内で盛り上がった箇所以外を優先的に高速化し、それ以外の箇所を等倍に近い速度に調整して、後から番組を聴くユーザが指定した制限時間内に放送音源を圧縮する手法を二つ提案する。

2 関連研究

2.1 ソーシャルメディアと放送メディアコンテンツの関係性に注目した研究

ソーシャルメディアが普及した 2010 年代には、テレビ番組との関係性に注目した研究が数多く行われた。具体例として、Twitter における投稿を用いて視聴率を評価する研究 [1] や、スポーツ番組のネタバレ検知についての研究 [2] がある。一方で、ソーシャルメディアとラジオ番組の関係については、Bonini ら

[3] の、ラジオ局による Twitter を用いたリスナーエンゲージメント向上施策についての調査があるものの、データを聴取体験の向上に利用した研究は見られない。

2.2 マルチメディアコンテンツの要約タスクの研究

動画の要約タスクについては、Washington ら [4] の、ソーシャルメディアの視覚的・言語的特徴を生かした早送りのパーソナライズを行う研究がある。音声メディアにおいては、Zheng ら [5] や Aneesh ら [6] のように、Podcast 要約についての研究が近年盛んに行われている。なかでも、Reddy ら [7] は Podcast の書き起こし文や説明文における、本編と無関係な箇所を抽出する手法を提案しており、無関係な箇所の特定には Spotify のユーザデータの一つである再生保持率を利用している。しかしながら、これらの研究はコンテンツを書き起こしたテキストの要約であり、リスナーの聴取体験を向上させるための音源圧縮の研究は行われていない。

3 予備実験

3.1 実施内容

ラジオ番組の面白さと Twitter のツイート数にどのような相関があるかを調査するために予備実験を行った。実験では 20 代の男女 4 名の被験者がラジオ番組の音源を聴取し、主観的な面白さを 1 分ごとに評価した。音源には放送時間帯および「radiko」における番組ジャンルを参考にして、首都圏で放送されている 3 番組のそれぞれ一部分を用いた (表 1)。評価には「1 分あたりの音源が面白いと思ったか」という質問に対する「全くそう思わない」「そう思わない」「そう思う」「非常にそう思う」の 4 段階のリッカート尺度 (1~4) を用いて、被験者 4 名の評価スコアを集計した。ツイートデータの取得には Twitter API for Academic Research を用い、番組が公式に周知しているハッシュタグをクエリとして、音源に使用した時間とその後 4 分間の、リツイートを除く全ツイートを取得した。合計件数は音源 1 は 507 ツイート、音源 2 は 455 ツイート、音源 3 は 1981 ツイートであった。

評価スコア平均と番組放送時間中のツイート数の相関の確認には、放送から 0~4 分後のツイート数との相関係数行列を用いた。これは、放送を聴いてからツイートする内容を考えたり、「radiko」のリアルタイム再生が IP サイマルラジオサービスであるためにラジオ放送から数分遅れることなどにより、ツイートがラジオ番組の放送から一定時間遅れて投稿されるという時間ギャップを考慮すべきであると考えたためである。

3.2 結果

評価スコアと 0~4 分遅らせたツイート数の相関係数を表 2 に示す。音源 1 では +2 分との相関係数で 0.6 を上回り、音源 2 では +1 分との相関係数で 0.8 を上回り、音源 3 では +2 分との相関係数で 0.4 を上回った。

各音源におけるツイート数と評価スコアの 1 分ごとの推移を図 2 に示す。評価スコアから 1~2 分遅れて、ツイート数の増加・減少に同様の傾向が見られた。

表 1 番組リスト

音源	番組名	放送日時 (使用した時間)	ハッシュタグ	ジャンル
1	赤江珠緒たまむすび	2022/1/20 13:00~15:30 (27 分間)	#tama954	バラエティー, アナウンサー
2	THE TRAD	2022/1/24 15:00~16:50 (15 分間)	#THETRAD	音楽, ミュージシャン
3	オードリーのオールナイトニッポン	2022/1/16 1:00~3:00 (19 分間)	#annkw	トーク, 芸人

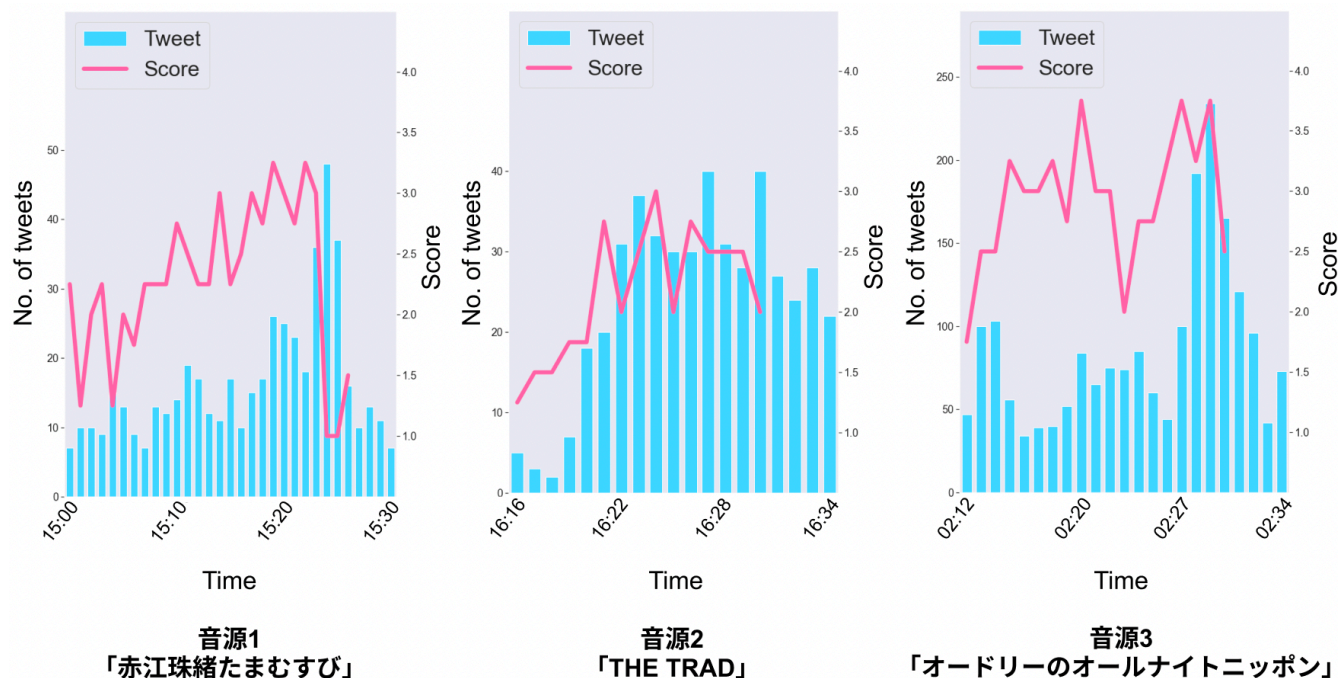


図 2 各音源におけるツイート数と評価スコアの 1 分ごとの推移. 折れ線が被験者の面白さ評価スコアの平均値を, 棒グラフがツイート数を示している. 時間ギャップを確認するため, ツイート数のみ, 使用した音源の時間よりも 4 分間長く図示した.

表 2 時間ギャップを考慮した, 面白さの評価スコアとツイート数の相関係数. 太字は各音源における最大値を示す.

音源	+0 分	+1 分	+2 分	+3 分	+4 分
1	-0.136854	0.370570	0.608066	0.548759	0.471172
2	0.713368	0.815830	0.701268	0.636895	0.428750
3	0.258919	0.258500	0.415752	0.390619	0.092199

以上より, ラジオ番組の面白さは Twitter のツイート数の盛り上がりにより 1~2 分かけて反映されることがわかった.

4 提案手法

予備実験の結果に基づき, 音源の再生速度をツイート数に応じて変化させ, ユーザが指定する制限時間内に収まるように放送音源を圧縮する二つの手法 (分単位ベース手法, 文脈考慮手法) を提案する. ツイート数は, 予備実験で確認した時間ギャップを考慮し, 放送開始から 1 分後以降の番組ハッシュタグ付きツイートを 1 分ごとに集計したデータを用いる.

4.1 1 分間のツイート数の大小を基準とする手法 (分単位ベース手法)

本節では, 1 分ごとのツイート数が多い時点群ほど等倍に近い速度に, ツイート数が少ない時点群ほど高速に変化させる手

法を提案する (Algorithm 1).

STEP 1. 各時点をツイート数の大小により 2 つの時点群に分ける (図 3). 分類の閾値として, 分単位のツイート数の中央値を用いる. 閾値よりもツイート数が多い時点群を P , それ以外を時点群 Q と呼称し, 各時点群の合計時間を T_P, T_Q とする. 番組全体の時間を T_B として式 (1) を定義する.

$$T_B = T_P + T_Q \quad (1)$$

STEP 2. 指定された制限時間 T_L から基準速度 S_S を定める.

$$S_S = \frac{T_B}{T_L} \quad (2)$$

STEP 3. 時点群 P の速度 S_P を式 (3) を満たすよう定義する. 時点群 Q の速度 S_Q は, 定数 C_1 を用いて式 (4) のように定める.

$$\frac{T_P}{S_P} + \frac{T_Q}{S_Q} = T_L \quad (3)$$

$$S_Q = S_S + C_1 \quad (4)$$

4.2 番組における文脈を考慮する手法 (文脈考慮手法)

次に, トークゾーンやコーナー, 天気交通情報など, 番組に

Algorithm 1 分単位ベース手法

Input: 1分ごとのツイート数 $[s_1, s_2, \dots, s_{T_B}]$, 制限時間 T_L

Output: 時点群 P , Q それぞれの1分ごとの再生速度

```
1:  $s_{median} \leftarrow Median([s_1, s_2, \dots, s_{T_B}])$ 
   //STEP 1. 分類
2: for  $s_i$  in  $\{s_1, \dots, s_{T_B}\}$  do
3:   if  $s_i > s_{median}$  then
4:      $P \leftarrow i$ 
5:   else  $\{s_i \leq s_{median}\}$ 
6:      $Q \leftarrow i$ 
7:   end if
8: end for
   //STEP 2. 基準速度
9:  $S_S \leftarrow T_B/T_L$ 
   //STEP 3. 時点群  $P$ ,  $Q$  それぞれの1分ごとの再生速度
10:  $S_Q \leftarrow S_S + C_1$ 
11:  $S_P \leftarrow Condition : T_P/S_P + T_Q/S_Q = T_L$ 
```

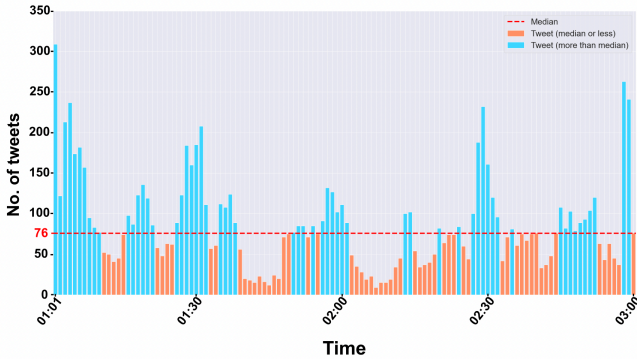


図3 分単位ベース手法における分類の適用例。ツイート数の中央値(点線)を閾値として中央値以上の時点群 P (水色)と中央値未満の時点群 Q (オレンジ色)に分類した。

おける時間的まとまりをセクションと呼称し、番組全体を複数のセクションに分割したのち、1分間のツイート数の大小を基準として倍速加工を施す手法を提案する。手順は二段階に分かれており、第一段階では盛り上がったセクションのグループ (P) の速度とそれ以外 (Q) の速度に極端な差をつける。第二段階ではグループ (P) における盛り上がり箇所の時点群 (PP) とそれ以外の箇所 (PQ) の速度に差をつける。

盛り上がったセクションのグループ (P) とそれ以外 (Q) の速度の差別化:

天気予報や交通情報などの本編と無関係な箇所でツイート数が少なくなる傾向を利用して、盛り上がったセクションのグループ (P) の速度とそれ以外 (Q) の速度に極端な差をつける (Algorithm 2)。

STEP 1. 文脈のある時間的まとまりをセクションと呼称し、番組全体を複数のセクションに分割する。

STEP 2. 各セクションを平均ツイート数の大小により2つのグループに分ける (図4)。盛り上がり箇所以外のセクションを検出するために、グルーピングの閾値として、各セクシ

Algorithm 2 文脈考慮手法における盛り上がったセクションのグループ (P) とそれ以外 (Q) の速度の差別化

Input: 1分ごとのツイート数 $[s_1, s_2, \dots, s_{T_B}]$, 制限時間 T_L

Output: グループ P における基準速度, グループ Q における1分ごとの再生速度

```
//STEP 1. セクションへの分割
1:  $[Sec_1, Sec_2, \dots] \leftarrow Split([s_1, s_2, \dots, s_{T_B}])$ 
   //STEP 2. グルーピング
2: for  $Sec_i$  in  $\{Sec_1, Sec_2, \dots\}$  do
3:    $Sec_{i_{mean}} \leftarrow Mean([s_1, s_2, \dots, s_{Sec_i}])$ 
4: end for
5:  $Sec_{mean_{Q1}} \leftarrow Q1([Sec_{1_{mean}}, Sec_{2_{mean}}, \dots])$ 
6: for  $Sec_i$  in  $\{Sec_1, Sec_2, \dots\}$  do
7:   if  $Sec_i > Sec_{mean_{Q1}}$  then
8:      $P \leftarrow i$ 
9:   else  $\{Sec_i \leq Sec_{mean_{Q1}}\}$ 
10:     $Q \leftarrow i$ 
11:   end if
12: end for
   //STEP 3. グループ  $P$  における基準速度, グループ  $Q$  における1分ごとの再生速度
13:  $S_Q \leftarrow T_B/T_L + C_2$ 
14:  $S_P \leftarrow Condition : T_P/S_P + T_Q/S_Q = T_L$ 
```

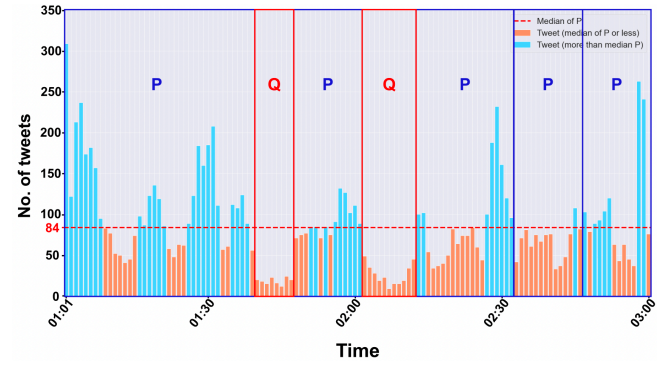


図4 文脈考慮手法におけるセクション分割及び分類の例。 P は盛り上がったセクションのグループ, Q はそれ以外を示す。また、グループ P におけるツイート数の中央値(点線)を閾値として、中央値以上の時点群 PP (水色)と中央値未満の時点群 PQ (オレンジ色)に分類した。

ンにおける1分間の平均ツイート数の第1四分位数を用いる。閾値よりも平均ツイート数が多いセクションのグループを P , それ以外を Q と呼称し、合計時間を T_P , T_Q とする。番組全体の時間を T_B として式 (5) を定義する。

$$T_B = T_P + T_Q \quad (5)$$

STEP 3. 指定された制限時間 T_L を用いて、グループ P の基準速度 S_P を式 (6) を満たすように定義する。グループ Q の速度 S_Q は、定数 C_2 を用いて式 (7) のように定める。 C_2 は、 C_1 , 後に説明する C_3 よりも大きな値を用いる。

$$\frac{T_P}{S_P} + \frac{T_Q}{S_Q} = T_L \quad (6)$$

$$S_Q = \frac{T_B}{T_L} + C_2 \quad (7)$$

グループ P の制限時間 T_{PL} は式 (8) のように定義する.

$$T_{PL} = T_L - \frac{T_Q}{S_Q} \quad (8)$$

グループ P 内における速度の差別化：

次に Algorithm 1 を用いて、グループ P 内における盛り上がり箇所とそれ以外の速度に差をつける.

STEP 4. グループ P の各時点ツイートの数により 2 つの時点群に分類する. 閾値として、グループ P の分単位のツイート数の中央値を用いる. 閾値よりもツイート数が多い時点群を PP , それ以外を PQ と呼称し、合計時間を T_{PP} , T_{PQ} とする. T_P を式 (9) のように定義する.

$$T_P = T_{PP} + T_{PQ} \quad (9)$$

STEP 5. 時点群 PP の速度 S_{PP} を式 (10) を満たすよう定義する. 時点群 PQ の速度 S_{PQ} は、定数 C_3 を用いて式 (11) のように定める.

$$\frac{T_{PP}}{S_{PP}} + \frac{T_{PQ}}{S_{PQ}} = T_{PL} \quad (10)$$

$$S_{PQ} = S_P + C_3 \quad (11)$$

5 実験

提案した 2 種類の手法を適用した音源を評価・比較するために実験を行った². 共著者を含まない 20 代の男女 4 名を被験者とした. 被験者は事前のアンケートにおいて、ラジオを聴く頻度について、1 名が「よく聴く」、3 名が「聴いたことはあるが日常的には聴かない」と答えた. 動画や音声再生の際の倍速機能については、1 名が「全く使わない」、2 名が「ほとんど使わない」、1 名が「たまに使う」と回答した.

5.1 使用音源

音源 3 について、一様な 2 倍速の音源と、制限時間 $T_L = 60$ とした分単位ベース手法、文脈考慮手法による圧縮加工を施した音源の合計 3 種類を用意した. アルゴリズムにおける定数はそれぞれ、 $C_1 = 0.5$, $C_2 = 10$, $C_3 = 0.5$ とした.

聴取対象の箇所として、等倍再生で約 10 分間の、ラジオ番組特有の「トークから天気交通情報, CM, 音楽に移行して再びトークに戻る」音源を抽出した. 実験では、被験者に圧縮手法を示さず、音源を提示した. 以下の設問では一様な 2 倍速を Baseline, 分単位ベース手法による音源を Minute-based, 文脈考慮手法による音源を Context-based と表記する.

5.2 手順

まず、被験者は Baseline, Minute-based, Context-based の順番に音源を全て聴いた後、以下の質問に回答した. 回答には、「全くそう思わない (1)」「そう思わない (2)」「そう思う (3)」「非常にそう思う (4)」の 4 段階のリッカート尺度を用いた.

- Minute-based は、Baseline から改善が見られたか.
- Context-based は、Baseline から改善が見られたか.

次に、Minute-based, Context-based について SD 法に基づき、以下の 2 つの形容詞対を用いて音源を聴いた印象の 7 段階評価を行った.

- 正しい-間違った

非常に間違った (-3), 間違った (-2), やや間違った (-1), どちらでもない (0), やや正しい (+1), 正しい (+2), 非常に正しい (+3)

- 満足な-不満な

非常に不満な (-3), 不満な (-2), やや不満な (-1), どちらでもない (0), やや満足な (+1), 満足な (+2), 非常に満足な (+3)

最後に、Minute-based, Context-based に対する印象を自由回答で記述した.

5.3 結果

一様な 2 倍速と各手法との比較では、被験者平均は Minute-based が 3.0, Context-based が 3.75 であった.

提案手法間の比較では、形容詞「正しい-間違った」では Minute-based の中央値が -1.5, Context-based の中央値が +2.0, 形容詞「満足な-不満な」では Minute-based の中央値が -1.0, Context-based の中央値が +2.0 であった (図 5).

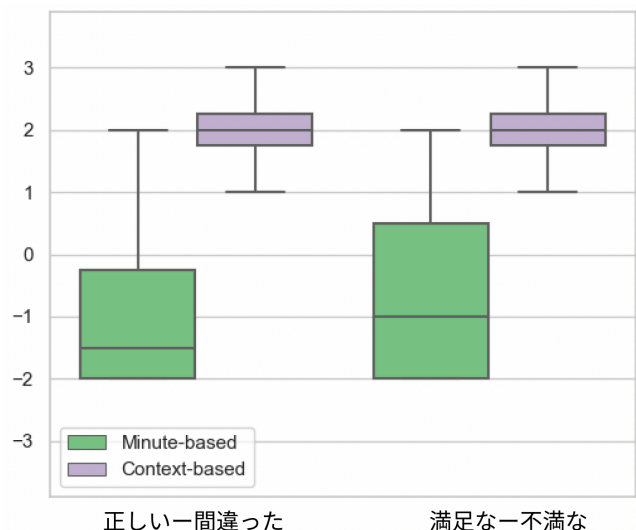


図 5 提案手法間の比較結果. 分単位ベース手法を適用した音源 (Minute-based) は正しさ・満足度ともに評価が低くなる傾向が、文脈考慮手法を適用した音源 (Context-based) では高くなる傾向が見られた.

2: 使用したデータ及び倍速加工処理の詳細:

https://github.com/Ota1022/Compradior_OS

5.4 考察

分単位ベース手法では一様な2倍速からの一定の改善が見られたものの、圧縮の正しさと満足度は低い値であった。印象の自由記述を確認したところ、「本編と無関係な箇所が短い時間で終わる点は良いが、全体的に速く感じて聞き辛く感じた」などの意見が見受けられた。高速化処理により1分毎に倍速の時間が変更されることで、2倍速よりも低速で聞き取りやすくなっている箇所が断片的になってしまい、高速で聞き取りづらい箇所の内容を推測する負担が激しいことが要因であった。

文脈考慮手法では一様な2倍速から大きく改善が見られ、圧縮の正しさと満足度も高い値となった。自由記述を確認したところ、「ポイントを絞って聞き取ることが出来た」「Minute-basedよりも天気交通情報の箇所が飛ばされている感じがおり、全体的に非常に聴きやすい」などの意見が見受けられた。盛り上がり以外の箇所の極端な高速化により、盛り上がり箇所の基準速度よりも高速な箇所が、分単位ベースにおけるそれと比較して低速となり、聞き取りづらさが逡減したことが要因であった。

6 議論

提案手法ではツイート数の統計量を閾値とする分類を行うが、ツイートの意味的内容による分類は行われぬ。つまり閾値や定数次第では、天気や交通情報などの他に、本編のツイート数の少ない箇所も必要以上に高速化されてしまう恐れがある。例えば、文脈考慮手法では盛り上がった箇所以外のセクションを検出する際に第1四分位数を用いたが、番組の形式に応じて変更する必要がある。また、ラジオ番組を構成する要素のうち以下の2点の取り扱いについては特に注意すべきである。

a) 音楽

音楽が流れる箇所はツイート数が少なくなる傾向にあり、高速化されやすい。これは、トークなどの情報が多い箇所に特化して聴取したいと考えるユーザには高速化は好都合であるが、音楽の箇所は等倍で聞きたいと考えるユーザについても想定すべきである。どのようにユーザの嗜好を取り入れるべきか議論が必要である。

b) 広告 (CM)

CMを流すことで民放ラジオの収益が成り立っていることを踏まえると、直前に放送された音源を圧縮対象とする場合には該当箇所は後から聴くユーザに無関係とは言い切れない。本技術を実用化する段階においては放送当時のCMを高速で再生すべきか、等倍の別のCMに置き換えるべきかについて検討すべきである。

7 おわりに

本稿では、ラジオ番組放送時間中に投稿されたツイートデータのみに基づいて音源に倍速加工処理を施し、後から聴くユーザが指定した制限時間内に圧縮する手法を提案した。番組内の面白い箇所の特定に関しては、ツイートデータの他にも音声データや時間帯、放送地域、番組パーソナリティの情報などを素性として機械学習を用いることも考えられるが、教師データ

作成コストの高さや、用いるべき素性の取捨選択の難しさなどの問題が存在する。

今後の展望として、

- 自然言語処理モデルや時系列解析モデル, Nakazawaら [8]が提案した盛り上がり検出アルゴリズムを用いたセクション分割の自動化
 - 滑らかな倍速速度変化を実現するための音響学, 感性工学を用いたアプローチの導入
 - 放送音源の長さユーザが指定する制限時間の比率限界の調査
 - 番組名と放送音源を入力することで自動的に圧縮を行うWebアプリケーションの構築
- などを考えている。

文献

- [1] Shoko Wakamiya, Ryong Lee, and Kazutoshi Sumiya. Towards Better TV Viewing Rates: Exploiting Crowd's Media Life Logs over Twitter for TV Rating. In *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*, page 1–10, 2011.
- [2] Yuji Shiratori, Yoshiki Maki, Satoshi Nakamura, and Takanori Komatsu. Detection of football spoilers on twitter. In *Proceedings of International Conference on Collaboration Technologies*, pages 129–141. Springer, 2018.
- [3] Tiziano Bonini and Toni Sellas. Twitter as a public service medium? A content analysis of the Twitter use made by Radio RAI and RNE. *Communication & Society*, 27(2):125–146, 2014.
- [4] Washington Ramos, Michel Silva, Edson Araujo, Alan Neves, and Erickson Nascimento. Personalizing Fast-Forward Videos Based on Visual and Textual Features from Social Network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3271–3280, 2020.
- [5] Chujie Zheng, Harry Jiannan Wang, Kunpeng Zhang, and Ling Fan. A Baseline Analysis for Podcast Abstractive Summarization. In *Proceedings of the Workshop on Podcast Recommendations (PodRecs)*, 2020.
- [6] Vartakavi Aneesh and Amanmeet Garg. PodSumm – Podcast Audio Summarization. In *Proceedings of the Workshop on Podcast Recommendations (PodRecs)*, 2020.
- [7] Sravana Reddy, Yongze Yu, Aasish Pappu, Aswin

Sivaraman, Rezvaneh Rezapour, and Rosie Jones. Detecting Extraneous Content in Podcasts. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1166–1173. Association for Computational Linguistics, 2021.

- [8] Masami Nakazawa, Maike Erdmann, Keiichiro Hoashi, and Chihiro Ono. Social indexing of TV programs: Detection and labeling of significant TV scenes by Twitter analysis. In *Proceedings of the 26th International Conference on Advanced Information Networking and Applications Workshops*, pages 141–146. IEEE, 2012.