

2020年度 卒業論文

Podcastの検索システムの開発

大阪産業大学 デザイン工学部 情報システム学科
情報教育システム研究室

17H055 瀧川 公裕

Podcast の検索システムの開発

17H055 瀧川公裕

1 はじめに

2020 年は新型コロナウイルス感染症の影響で家で過ごす時間が長くなったこともあり在宅勤務が急増し、ラヂオのように“ながら聞き”できる音声メディアの需要が高まっている。特に Podcast は成長が著しく、Google 製 Podcast アプリケーション“Google Podcast”の場合、[1] に示す通り所持ユーザー数が 2019 年 6 月と 2020 年 6 月を比較すると 9.06 倍になっている。しかし、現段階で Podcast のアプリケーションのほとんどすべて“配信者名”“紹介文”“題名”“カテゴリ”でしか検索できない。

2 目的

現状の Podcast アプリケーション又はサービスの検索システムの時点での問題点として、“Podcasting されたエピソードは“配信者名”“紹介文”“題名”“カテゴリ”に乗っていない内容では検索ができない”という点が挙げられる。Podcast がより詳しい内容、エピソードの内容などで検索することが出来るようになれば、“ながら”という忙しい現代において有用なアプローチを掛けることが出来るより魅力的な情報収集できるツールになると考えた。その為、本研究の目的は既存 Podcast サービスの検索機能と違った、任意の Podcast エピソードの情報を素早く取得できるシステムの開発である。

3 castumble

castumble とは、今回のテーマである“Podcast”と今回使用するメディアミックスブログサービス“Tumblr”、そして“転がる”を意味する英単語“tumble”を合わせた造語であり、今回開発したシステムの名称である。castumble は Podcasting された mp3 ファイル、WAV ファイルなどの音声ファイルを Google 製文字起こし API“Speech to Text”を通してテキスト化する。その文字起こしされたデータをメディアミックスプロ

グサービス“tumblr”上にストックしておき、それをもとに文字検索を行うことで内容での検索を可能にし、Reblog 形式で表示することで、Podcast の検索システムにおける問題点の改善を試みて開発したソフトウェアである。

4 検証

100 個の内容が異なる Podcast エピソードを用意し、そのエピソードを castumble に掛け、文字起こしされたテキスト群を castumble アカウント上にてエピソード検索を行う。検索に合致したエピソードに本当にキーワードの単語が出てきたのかを判断した。

5 まとめ

本研究では“Podcast”のエピソードの内容をテキスト化し、収集したエピソードを全文検索を行うことができるシステムの開発を行った。その結果、Podcast にてエピソードの内容での検索ができるようになり、一般的な Podcast アプリケーションと比較した際より広範囲の Podcast を検索対象にすることに成功した。しかし、エピソードの内容とは全く違った文章にて文字起こしをされていることもあり、はじめからすべてが正しく出力される例は非常に少なかった。今後の課題・展望としては、“文節、文脈をくみ取った自動段落とし又は、改行機能の実装”“同音異義語や文節・文脈をくみ取った reblog をより手軽にする文章訂正機能の実装”の以上 2 点が上げられる。

参考文献

- [1] 音声アプリ、ポッドキャストの利用が増加 | google ポッドキャストダウンロードユーザー数、1 年で 9 倍！ <https://lab.appa.pe/2020-07/music-trend-ranking-podcast-apps-20200714.html>.

目次

1	はじめに	1
2	目的	3
3	Podcast とは	4
3.1	Podcasting のメリット	5
3.2	Podcasting に対応している主な podcast 対応サービス	5
4	現在稼働している Podcast の検索システムの比較と現状	6
4.1	現在の検索システムの問題点	14
4.2	問題点に対する現状での対応	14
4.3	過去にあった podcast 検索システム “Podscope” “PodZinger”	14
4.4	類似検索技術 “ワードスポッティング”	15
5	castumble	16
5.1	検索システム “castumble”	16
5.2	castumble の詳細	17
5.3	利用したサービス “Speech to Text” “Tumblr”	19
5.4	Google 製文字起こし API “Speech to Text”	19
5.5	メディアミックスブログサービス “Tumblr”	20
5.6	過去にあった podcast 検索システム “Podscope” “PodZinger” と “castumble” の比較	21
5.7	類似検索技術 “ワードスポッティング” と “castumble” の比較	21
6	castumble の検証	22
7	結果と考察	24
8	結論	28
付録 A	ソースコード	31
A.1	GCS に格納された音声ファイルの文字起プログラム	31

1 はじめに

近年、音声アシスタントや音声アシスタントを組み込みソフトウェアとして専用設計されたスマートスピーカーなどが次々に発売されている。それとともに、音声や動画を使った情報発信や情報収集が注目されている。特に画面を注視しなくてはならない動画よりも作業をしながら情報収集ができるの音声メディアの成長が著しい。特に2020年は新型コロナウイルス感染症の影響で在宅勤務や時短営業が急増し、家で過ごす時間が長くなったこともあり、ラヂオのように“ながら聞き”できる音声メディアの需要が高まっている。特にPodcastは成長が著しく、[1]に示す通り、Google製Podcastアプリケーション“Google Podcast”の場合、所持ユーザー数が2019年6月と2020年6月を比較すると、2019年6月の記録である256ダウンロードと比較し、2020年6月のは2,332ダウンロードと9.06倍になっている。

Google Play 音楽&オーディオカテゴリ内のアプリにおける2020年6月のトレンドランキングにおいてpodcast専用アプリケーションで首位を獲得したGoogle Podcastの利用動向を図1に示す。

しかし、こういった急激な盛り上がりを見せるPodcastではあるが、現段階ではPodcastのアプリケーションの

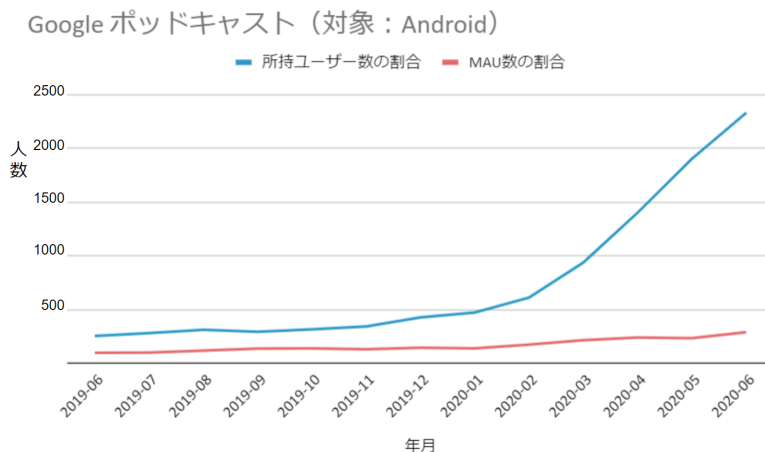


図1 データ元: App Ape (国内約15万台のAndroid端末を分析) / アクティブ数はApp Ape 推定による。
所持ユーザー数: そのアプリの対象期間において、対象のアプリを所持しているユーザーの数
MAU (Monthly Active Users): そのアプリの対象期間における、月間アクティブユーザー数で、月に一度でもそのアプリを起動したユーザーの数
2019年6月の記録である256ダウンロードと比較し、2020年6月のは2,332ダウンロードという9.06倍の記録である。
MAU (月間アクティブユーザー数) と所持ユーザー数の推移を見ても、新型コロナウイルス感染症によって日本国内で外出の自粛が求められていった2020年3月以降、どちらも数字を伸ばしていることが分かる。
また、長らくAndroidでしか使えなかったアプリが、2020年3月にiOS版のアプリも登場したことで、アプリの認知度が高まり、Androidのユーザー増加にも影響したと考えられる。

ほとんどすべてが“配信者名”“紹介文”“題名”“カテゴリ”でしか検索できない。そのため上記以外のフィルタリング項目が存在しないアプリケーションが多いということになり、情報収集ツールとして不便な点が存在する。

第2章では本研究の目的について述べる。第3章では現在のPodcastと昨今の音声メディアの情勢について述べる。第4章では現在のPodcastの検索システムとその精度、そしてその問題点の詳細について述べる。第5章では本研究で開発した検索システム“castumble”について述べる。第6章では本研究で行った検証の概要について述べる。第7章では検証の結果をその考察とともに述べる。第8章には研究の成果とともに今後の課題について

てまとめる。

2 目的

現状の Podcast アプリケーション又はサービスの検索システムの時点での問題点として、“Podcasting されたエピソードへの詳細な検索ができないという点”が挙げられる。Podcast がより詳しい内容、エピソードの内容などで検索することが出来るようになれば、“ながら”という忙しい現代において有用なアプローチを掛けることが出来るより魅力的な情報収集できるツールになると考えた。その為、本研究の目的は Podcast の既存の検索機能と違った、任意の Podcast エピソードの情報を素早く取得できるシステムの開発である。

3 Podcast とは

Podcast とは、米国 Apple 社の iPod と、“放送”を意味する broadcast を組み合わせた造語であり、公開された音声ファイルによるブログ群である。もともとは Internet Radio の1つであったが RSS で配信することが可能なのではないかという発想から RSS にて配信されるようになった。それが iTunes で採用され Podcasting が生まれた。配信に利用されている 音声データファイルのほとんどは標準的な MP3 形式のオーディオファイルであるため、任意の場所に保存して再生できる。現在では動画配信も含めて、Podcast と呼ぶ。一般的には Podcast という言葉で音声や動画のデータファイルをインターネット上に配信する仕組み自体を指し、Podcasting はこの仕組みを利用して実際に配信を行うという行為を指すのに使われる。また、Podcasting は公開するファイルの種類は問わない。iOS には“Podcast”というアプリが標準でインストールされており Android にも“Google Podcast”をはじめとする Podcast アプリが多数存在する。“iTunes”（Win 版／Mac 版）などのアプリがあれば、パソコンでも Podcast を手軽に楽しめる。また、Podcast は場所や時間をえらばず学習用に作成されたコンテンツを簡単にダウンロードしそして持ち歩けるという低コストで学習上有効なテクノロジーである。スマートフォン等の多機能携帯端末を所持する学生数が増加している背景もあり、大学は教育用の Podcast を利用する場として最適な環境であるとも言える。

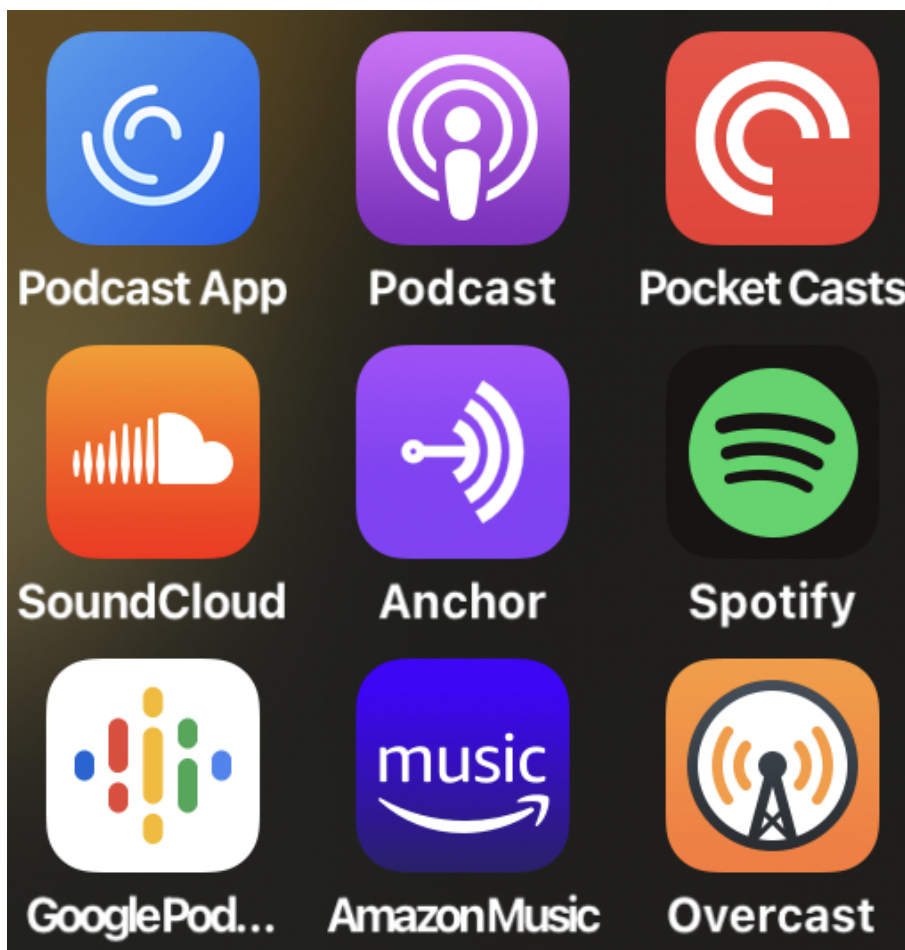


図2 Podcast アプリケーションのアイコン

3.1 Podcasting のメリット

音声や動画を使った情報発信や情報収集の配布手段として Podcasting 以外に以下の方法がある。“Youtube や TikTok 等の動画コンテンツ配信サービスへの投稿” “テキストウェブページへの埋め込み ” “Apple 製音楽ストリーミングサービス iTunes” “Twitter や Instagram などの SNS への投稿” これらにはなく、Podcast だけが持つメリットとして、以下が挙げられる。

- ブラウザや OS に依存せず音声を取得することが可能である。
- 配信されたファイルを利用者が指定した場所に保存することが可能である (USB メモリ等にも保存が可能である)
- 配信された RSS は配信者と視聴者の双方が確認することができる。

3.2 Podcasting に対応している主な podcast 対応サービス

- 1 Apple 製 Podcast アプリケーション “Podcast”
- 2 Google 製 Podcast アプリケーション “Google Podcast”
- 3 Apple 製音楽ストリーミングサービス “iTunes”
- 4 スポティファイ・テクノロジー製音楽ストリーミングサービス “Spotify”
- 5 Amazon 製音楽ストリーミングサービス “Amazon Music”
- 6 Podcast 専用検索エンジン “ListenNotes”

4 現在稼働している Podcast の検索システムの比較と現状

今回のシステムの構築にあたり、主流とされている Podcast アプリケーション及び Podcast をサポートしているサービスの検索条件を列挙し表 1 に示す。

表 1 主流とされている Podcast アプリケーション及び Podcast をサポートしているサービスの検索条件の比較
音楽ストリーミングサービスはまだ Podcast に最適化されておらず、Podcast 専用アプリケーションと音楽ストリーミングサービスとで変化があるのがわかる。

アプリケーションの名称	キーワード検索での表示条件
Apple 製 Podcast アプリケーション “Podcast”	キーワード又はカテゴリ
Google 製 Podcast アプリケーション “Google Podcast”	キーワード又はカテゴリ検索
Apple 製音楽ストリーミングサービス “iTunes”	キーワード又はカテゴリ検索
スポティファイ・テクノロジー製音楽ストリーミングサービス “Spotify”	楽曲も含めたキーワード検索またはカテゴリ検索
Amazon 製音楽ストリーミングサービス “Amazon Music”	キーワード又はカテゴリ検索
Podcast 専用検索エンジン “ListenNotes”	キーワード検索またはカテゴリ内での詳細なキーワード検索

代表例として Apple 製 Podcast アプリケーション “Podcast” にての一エピソードにどのような情報が記載されているのかを確認する。今回は “NHK ラジオニュース” を参考にした。“Apple 製 Podcast アプリケーション Podcast” の検索画面を図 3 に、その検索欄にて “NHK ラジオニュース” と検索を行った後の結果画面を図 4 に、結果として出てきた “NHK ラジオニュース” の番組一覧の画面を図 5 に、“NHK ラジオニュース” の番組一覧にある “エピソードの詳細” に変移した画面を図 6 に示す。

また、アプリケーションではなく Podcast 専用検索エンジン “ListenNotes” でも、同じく “NHK ラジオニュース” にどのような情報が記載されているのかを確認する。Podcast 専用検索エンジン “ListenNotes” の検索画面を図 7 に、その検索欄にて “NHK ラジオニュース” とエピソードで検索を行った後の結果画面を図 8 に、Podcast で検索を行った後の結果画面を図 9 に、結果として出てきた Podcast “NHK ラジオニュース” へ変移した画面を図 10 に、“NHK ラジオニュース” の番組一覧にある “エピソードの詳細” に変移した画面を図 11 にそれぞれ示す。

現在 Podcast を扱っているアプリケーション又はサービスはキーワードが番組の名前、又はエピソードの名前、キャプションまたは概要欄に入っているのかという条件しか検索をすることができない。また Podcast 専用検索エンジンである “Listen notes” は、この条件に加え、図 12 に示すとおり、“カテゴリー” “言語” “地域” “分野” のフィルターをかけることができるのみである。課金をすると、“総配信時間” “総配信期間” “最新エピソードの配信時期” が追加でフィルタリングできるようになるが詳しい内容までは検索できない。

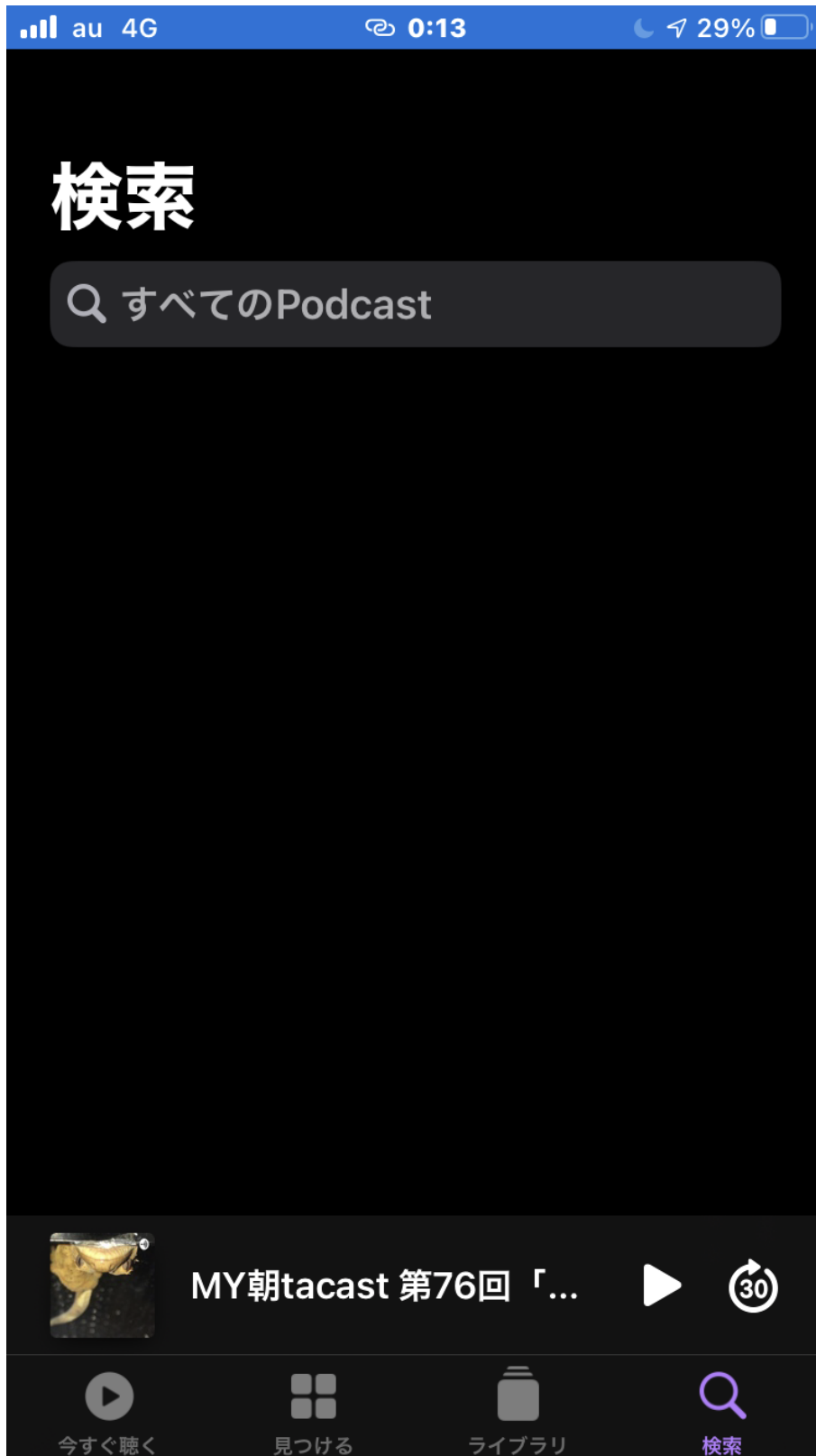


図3 Apple製Podcastアプリケーション“Podcast”での検索画面。フィルタリングなどの項目が無く、検索する項目がキーワードのみでしかできないことがわかる。



図4 Apple製Podcastアプリケーション“Podcast”でのキーワード検索を行った画面。番組とエピソードに分かれており、こちらは番組の一覧が表示されている。それぞれキーワードとマッチングしていないものも混じっている。



図5 Apple製Podcastアプリケーション“Podcast”にて結果として出てきた“NHK ラジオニュース”の番組一覧の画面図。番組紹介とエピソードに分かれており、こちらはエピソードとその紹介文の一覧が表示されている。



図6 Apple製Podcastアプリケーション“Podcast”にて結果として出てきた“NHK ラジオニュース”の番組一覧にある“エピソードの詳細”に移った画面図
番組の紹介文やキャプションがある場合ここに表示される。

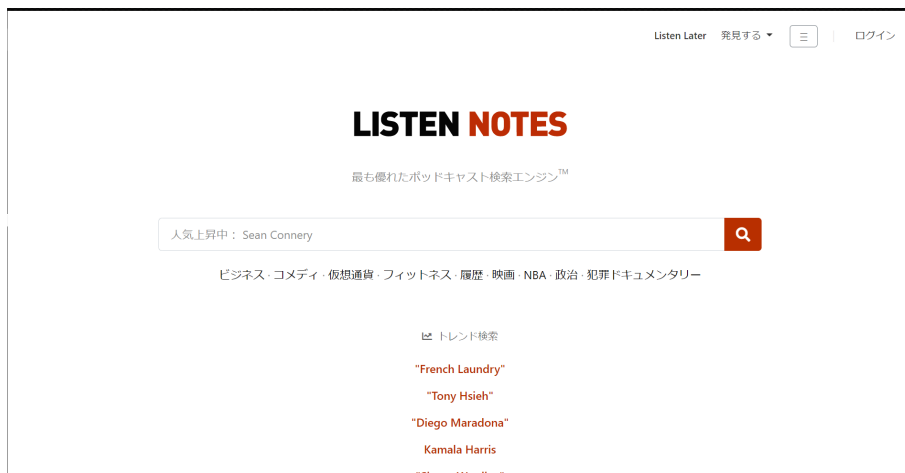


図7 Podcast 専用検索エンジン “ListenNotes”

この時点ではフィルタリング機能をつかうことが出来ず、キーワード又はカテゴリのみでの検索となる。



図8 Podcast 専用検索エンジン “ListenNotes”にて “NHK ラジオニュース”と検索を行った後の結果画面。番組とエピソードに分かれており、こちらはエピソードでの検索となる。

それぞれキーワードとどの部分がマッチングしたのかがハイライトで表示される。検索欄にキーワードを入力が無かった場合、その日のトレンドの単語にリダイレクトされる。その為、“ ”(空欄)などのエピソードは検索することができない。

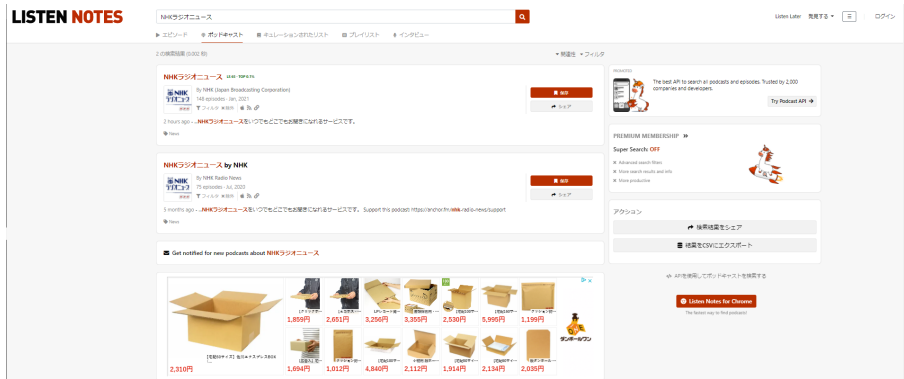


図9 Podcast専用検索エンジン“ListenNotes”にて“NHK ラジオニュース”と検索を行った後の結果画面。Podcastとエピソードに分かれており、こちらはPodcastでの検索となる。それぞれキーワードとどの部分がマッチングしたのかがハイライトで表示される。検索欄にキーワードを入力が無かった場合、その日のトレンドの単語にリダイレクトされる。その為、“”(空欄)などのエピソードは検索することができない。

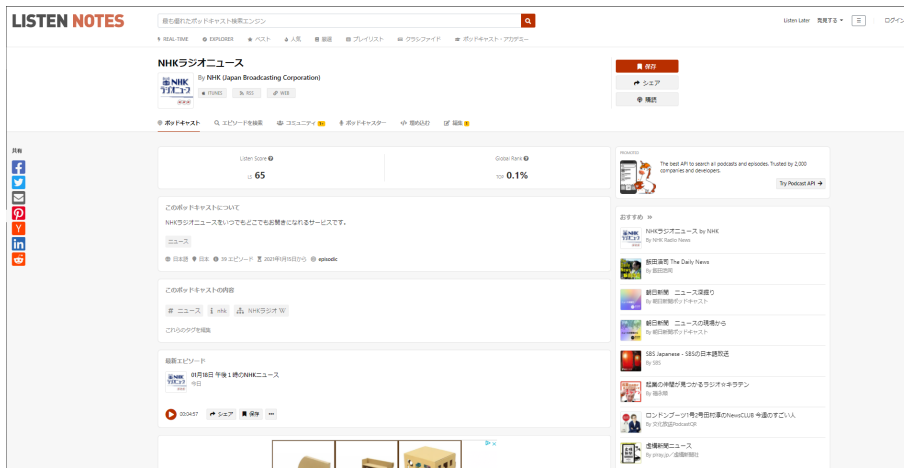


図10 Podcast専用検索エンジン“ListenNotes”にて結果として出てきたPodcast“NHK ラジオニュース”へ変移した画面図



図11 Podcast専用検索エンジン“ListenNotes”にて“NHK ラジオニュース”の番組一覧にある“エピソードの詳細”に変移した画面図

▼ 関連性 ▲ フィルタ

カテゴリー

言語

地域

期間 分
分 最大

分野

特定のフィールドのみを検索

公開日 *

* Premium features リセット 閉
じる

図 12 Podcast 専用検索エンジン “Listen notes”でのフィルタリング項目を移した画面。“カテゴリー” “言語” “地域” “分野” “総配信時間” “総配信期間” “最新エピソードの配信時期”の6つのフィルタが存在し、それぞれのフィルタリングを組み合わせることで詳細なフィルタリングを掛けることができる。その反面、キーワード必須の検索システムであるため、キーワード無しでのフィルタリング検索もできない。

4.1 現在の検索システムの問題点

現在の検索システムの問題点は、検索方法がテキストであるという仕様上、データ形式が大きく異なる“音声”をそのまま検索することができないという点である。音声の検索は難しい理由は検索に必要な索引情報（文やキーワード等）を音声から抽出することが困難だからである。詳細な検索ができる Podcast 専用検索エンジン“ListenNotes”を使ったとしても“配信者名”“紹介文”“題名”“カテゴライズ”といった配信者の定めた文章からでしか検索できず配信者が載せていない、もしくは作者・配信者の意図しない内容の検索が現時点で非常に困難であるというのが現状である。情報収集ツールとして使うには配信者の意識しているしていないに関わらず収集する必要があるため、不特定多数の他者の意図が少なからず混入してしまいかねないこの検索方法では詳しい内容が結局は聞かないとわからない。

4.2 問題点に対する現状での対応

問題点に対する現状での対応の選択肢として調べたいことと類似したことがキャプションに書かれてあるエピソードの視聴などが挙げられるが、様々な理由により、エピソードの必要箇所へたどり着くのが非常に困難である場合がある。エピソードの検索を困難にする要因として“番組の配信が長期にわたっている”“1つのエピソードが長時間である”“配信者が内容とは異なった題名やキャプションを付けている”ということが上げられる。どうしてもその部分がどこにあるのかを知る必要がある場合、SNS にて配信者本人又は他の視聴者へ質問として呼びかけるという方法もあるが、必ずしも返信が返ってくる保証はなく、また、他の視聴者からの回答の場合その回答が正しいという保証もない。

4.3 過去にあった podcast 検索システム “Podscope” “PodZinger”

英語の Podcast に対して音声認識を利用して全文検索を可能にするシステム “Podscope” [2] “EveryZing” [3] が過去に存在した。いずれも、音声認識により音声又は動画コンテンツ内の音声をすべてテキスト化した索引情報を内部に持ち、ユーザが Web ブラウザ上で入力した検索語を含む Podcast の一覧を提示する検索エンジンであった。また、ユーザー検索により、ランク付けされた結果のリストが生成され、最も関連性の高い Podcast と、コンテンツを再生またはダウンロードするためのリンクが提供されるというところは酷似しており、今回目標とする “Podcast のエピソード内容での検索” というものに非常に似た性質を持つ。しかし、これら二つはいくつかの点で問題が発生している。まず、これらのサービスは要点を把握して文章を再構築し、言語の意味を理解することを目標としておらず、検索のキーワードが存在する部分の文章をそのまま抜き出す “抄約” というのが技術体系として近いものである。その上で、Podscope では、Podcast のタイトルだけが列挙され、検索語が出現する直前から再生できるが、音声認識されたテキストは一切表示されない。一方、EveryZing では、これに加え、検索語が出現した周辺のテキストも表示され、ユーザがより効率的に部分的な内容を把握できるようになっている。しかし、せっかく音声認識をしていても、表示されるテキストは一部に限定されており、音声を聞かずに Podcast の詳細な内容を把握することはできなかった。また EveryZing のサービスは英語のみのサービスであり、日本語で同じ事をする場合、日本語は英語と異なり同音異義語があり、IME レベルの漢字変換、及び表記揺れ（コンピューターとコンピュータ）に対応した検索などが必要となる。また、両方とも音声認識では認識誤りを避けることはできないため、Podcast に対して誤った索引付けがなされて検索に悪影響を与えるが、それをユーザが把握したり改善したりすることは不可能であった。

4.4 類似検索技術 “ワードスポッティング”

音声から特定の単語を見つけ出す技術として“ワードスポッティング”が存在する。[4]にあるように、音声入力などに関して、発音される音声全体ではなくキーワードだけを認識し、その単語から文全体の意味を理解する技術の名称である。重要度高そうな単語が使われている文だけを抜き出して並べる“抄約”ではなく、要点を把握して文章を再構築し、言語の意味を理解するために考え出された方式で、意味理解を含む文章自動要約技術として注目されている。

5 castumble

5.1 検索システム “castumble”

Podcast 内での音声そのままでの検索は困難である。なぜなら、検索に必要な索引情報 (文やキーワード等) を音声から抽出することが困難だからである。一方、テキストの検索は既に広く使われており、Google 等の優れた検索エンジンにより Web 上のテキストを含む各種ファイルに対する全文検索が可能となっている。もし Web 上の音声ファイルからその発話内容のテキストを抽出できれば、同様に全文検索が可能であると考えた。

castumble とは、今回のテーマである “Podcast” と今回使用するメディアミックスブログサービス “Tumblr”、“転がる”を意味する英単語 “tumble”を合わせた造語であり、今回開発したシステムの名称である。castumble は Podcasting された mp3 ファイル、WAV ファイルなどの音声ファイルを Google 製文字起こし API “Speech to Text”を通してテキスト化する。その文字起こしされたデータを Tumblr 上にストックしておき、それをもとに文字検索を行うことで内容での検索を可能にし、メディアミックスブログサービス “Tumblr”にて Reblog 形式で表示することで、Podcast の検索システムにおける問題点の改善を試みて開発したソフトウェアである。



図 13 今回の研究にて制作したシステムの castumble の Tumblr 上のウェブページ。左上に検索欄があり、そこにキーワードを打ち込むことで検索をする。

このシステムは、“Tumblr でのテキスト検索エンジンからの検索も可能である” “ユーザが Tumblr 上で検索した Podcast を聞くだけでなく、読むことができる” “誰でも手軽に Tumblr 上で文字起こしされた文章を reblog という形で編集することができる” という 3 つの特徴を持っている。

5.1.1 Tumblr でのテキスト検索エンジンからの検索も可能である

castumble では過去の Podcast エンジンである “Podscope” “PodZinger” などのように索引情報として内部利用するのではなく、Tumblr にてテキストブログの 1 つのエピソードとして公開する。そのため castumble では Tumblr 内の castumble アカウントの web ページからの検索だけではなく、Tumblr 上の検索エンジンからの検索も可能になる。結果として多くの人の検索に該当することとなり、castumble と castumble にて文字起こしをした Podcast 両方の宣伝にもなる。

5.1.2 ユーザが Tumblr 上で検索した Podcast を聞くだけでなく、読むことができる。

音声再生環境がなくても内容を把握することができるほか、Podcast は音声のためエピソードの内容や中身の良し悪しがどうかを聞く前に把握することができるようになる。再生スピードを上げることで聞く時間を短縮す

castumble

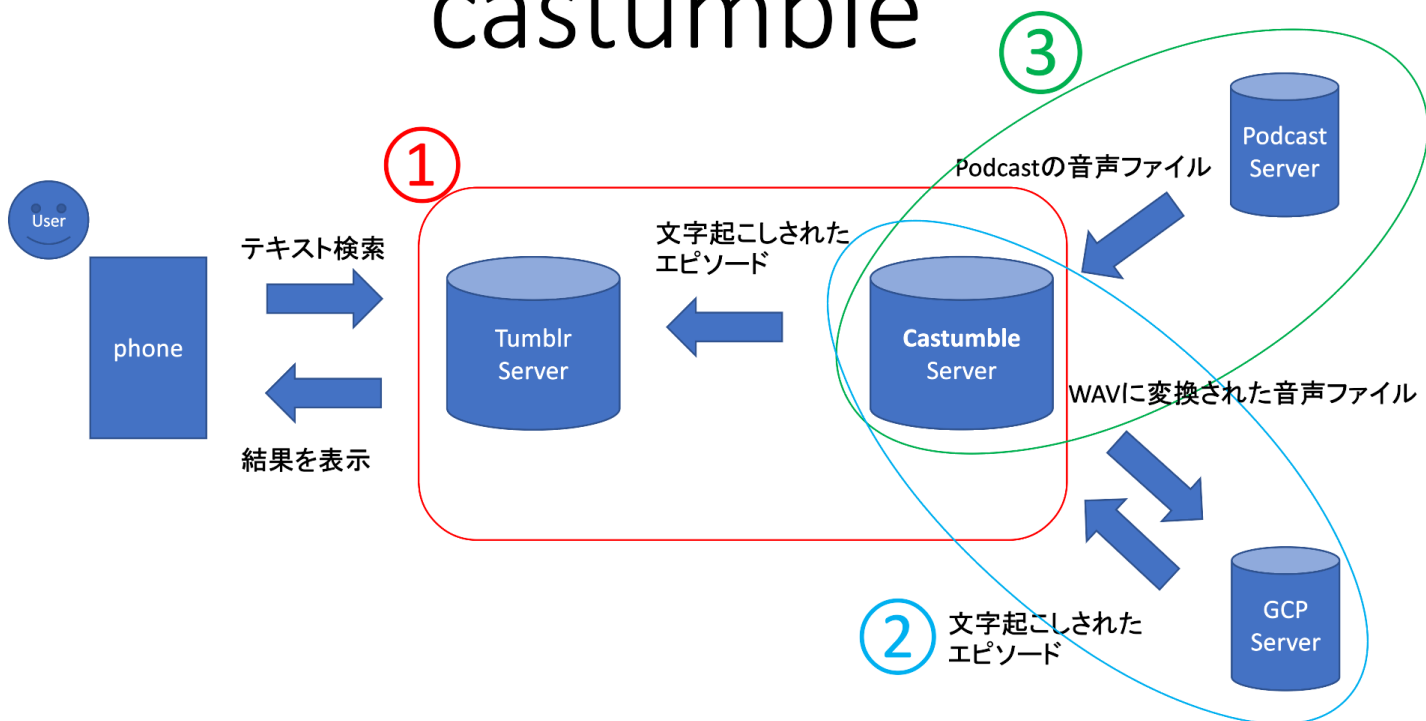


図 15 castumble 概要図

5.3 利用したサービス “Speech to Text” “Tumblr”

本研究ではシステムの構成に Google 製文字起こし API “Speech to Text” とメディアミックスブログサービス “Tumblr” を用いた。この章では、本研究で使用する Google API “Speech to Text” とメディアミックスブログサービス “Tumblr” の概要とシステム内でどのように稼働するかについての説明を行う。

5.4 Google 製文字起こし API “Speech to Text”

Google 製文字起こし API “Speech to Text” とは、機械学習を活用して音声を変換する Google の API サービスである。[5] システム内では、castumble サーバーより音声ファイルを受け取り、その音声ファイルを文字起こしをしてテキストファイルを生成する部分にて使用している。音声を文字起こしするクラウドサービスを選定する際に、声認識の精度が高く、多くの言語にも対応しているということを前提に文字起こし API を扱っているサービスを検討した。その中で、他のクラウドサービスを手掛けている会社と比較して、唯一 Google は Google Podcast や Youtube 等の音声・動画サービスを手掛けているということより、今後の展望として Podcast だけでなく動画のサポートも行う可能性を考慮し選択した。比較対象であった AWS は “castumble” システムの作成段階では Amazon Music は Podcast に非対応であったため見送っている。今回の研究において、GCE の無料インスタンスである f1-micro を利用した。

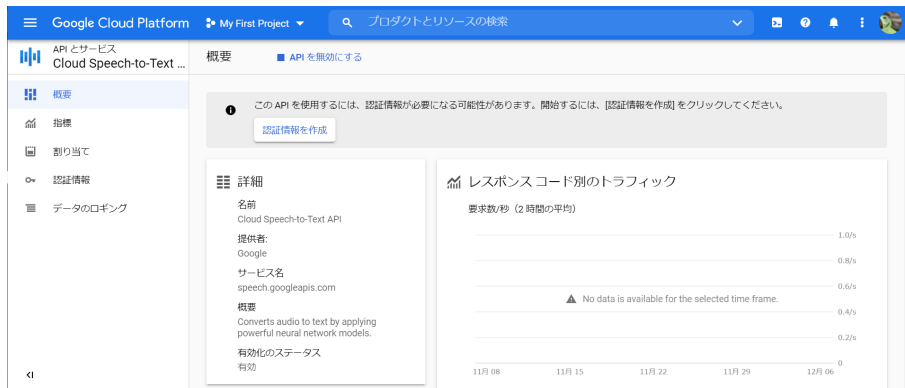


図 16 API “Speech to Text” 画面



図 17 Googlecloud ストレージ画面

5.5 メディアミックスブログサービス “Tumblr”

メディアミックスブログサービス “Tumblr”とは、アメリカ発の“メディアミックスウェブログサービス”といわれるものである。[6] 画像や動画、テキストを引用しシェアできるオンラインのスクラップブックといえる。このサービスを利用した理由は、Podcast の著作権上の問題を解決するためである。今回の研究は Podcast の音声文字起こししてストックしておき、その中にある単語を使って検索を掛けるという作業が、著作権に触れる部分があるという判断をされる可能性がある。これが現時点でのこのサービスの大きな弱点である。しかし、ここで Tumblr の特徴である reblog が大きな意味を持つ。reblog とは、他人の Tumblrlog の内容をそのまま自分の Tumblrlog で “繰り返す”ことである。たとえば誰かがお気に入りの写真を Post したとすると、それをそのまま自分の Tumblrlog にも表示させる機能のことを “reblog”と呼ぶ。他の人が見つけたいい写真・サイト・誰かが書いた文章でさえ、ReBlog をすれば自分のもののように見せることも出来てしまうという機能である。この機能は他人の投稿した Tumblrlog の内容の編集ができてしまう。もちろん文字起こしの元となった Podcast のエピソードのリンクと、その番組の URL・RSS も表示することとすることを前提とすれば、文字起こしてるのも reblog の範囲内である。つまりは Podcast という Voicelog を reblog するという行為に落とし込めるのではないかという狙いがある。



図 18 メディアミックスブログサービス “Tumblr”



図 19 メディアミックスブログサービス “Tumblr” の reblog 画面

5.6 過去にあった podcast 検索システム “Podscope” “PodZinger” と “castumble” の比較

過去にあった podcast 検索システム “Podscope” “PodZinger” と “castumble” の大きな違いは、英語の Podcast に対する全文検索は実現されていたが、音声認識により索引付けされた結果は内部に隠蔽され、各システム上でしか検索ができなかったという点である。仮にそれらのシステムが日本語に対応したとしても、Tumblr 上という制限は着くが、文字起こしされたテキストを誰でも検索、閲覧、編集ができるものではないというのが大きな違いとなる。

5.7 類似検索技術 “ワードスポッティング” と “castumble” の比較

ワードスポッティングと castumble の大きな違いは、ワードスポッティングは連続して発声された音声から特定の単語を見つけ出す音声認識技術であり、Podcast の中から検索のキーワードを認識する能力には長けているが、多数の音声ファイルから単語を抜き出す際、その都度その場にて処理を回さなくてはならない。その上、Podcast は基本ストックされていくため、時間が経過するに連れて処理する量が増えてく。その点、“Speech to Text”を通してのテキスト化によるストック検索では、一度テキスト状にしてしまえばテキスト同士の処理のため高速に行えるという点と、テキストに文字起こしする処理は Podcast の RSS をトリガーにすることによってバックグラウンドでの処理となることが大きな違いになる。

6 castumble の検証

castumble と Apple 製 Podcast アプリケーション “Podcast” を対象に検索結果の差異を求める検証を行った。検証の手順を以下に示す。

6.0.1 検証方法

100 個の内容が異なる Podcast エピソードを用意し、そのエピソードを castumble に掛け、文字起こしされたテキスト群を castumble アカウント上にてエピソード検索を行う。

今回は “NHK ラジオニュース” “01 月 18 日 午後 1 時の NHK ニュース” を参考にした。以下がこのエピソードにて述べられていた内容である。

“一時の Nhk ニュースです韓国のムンジェイン大統領は記者会見で慰安婦問題をめぐって韓国の裁判所が日本政府に賠償を命じたことについて “被害者も同意できる解決策を見出せるよう日本と協議していく” と述べ、外交で解決策を探していきたいという考えを示しました。韓国のムンジェイン大統領は今日午前、年頭の記者会見を開きました。この中で慰安婦問題をめぐり今月 8 日ソウルの地方裁判所が日本政府に対し元慰安婦の女性らへの賠償を命じたことについて “両国間のその他の懸案の解決に向けて努力していたところだったため少し困惑しているのは事実だ。” と述べました。そして “慰安婦問題をめぐる 2015 年の日韓合意は政府間の公式的な合意だと強調し、その土台の上で今回の判決を受けた被害者も同意できる解決策を見出せるよう日本と協議していく” と述べ外交で解決策を探していきたいという考えを示しました。また太平洋戦争中の徴用をめぐる韓国での裁判については、 “強制執行の形で日本企業の資産が現金化されることは両国関係にとって望ましくない。そのような段階になる前に外交的な解決方法を探すことが優先だ。” と述べました。ただ、いずれの問題についても具体的な対応には言及しませんでした。新型コロナウィルスの影響で倒産した企業が去年 2 月から今日正午までで合わせて 900 社に達したことがわかりました。信用調査会社の帝国データバンクのまとめによりますと業種別では飲食店が 141 社で最も多く、次いでホテル旅館と建設工事業がそれぞれ 72 社アパレル小売店が 54 社などとなっています。急速に発達する低気圧の影響で、北日本や新潟県それに北陸では、明日にかけて雪を伴って非常に強い風が吹き荒れた天気となる見込みです。気象庁は暴風雪や高波のほか猛吹雪による交通への影響に警戒するよう呼びかけます。PCR 検査やワクチンの接種を受けられるが予約金が必要だなどとして現金を要求する不審な電話が相次いでいることを受けて警視庁は詐欺被害への注意を呼びかけるポスターを撮影し、都内のスーパーなどに張り出しました。今日は警視庁滝野川警察署の警察官四人が東京北区にある開店前のスーパーを訪れ、新型コロナウイルスに乗じた詐欺に注意などと書かれたポスターを次々に貼っていきました。物入れは去年感染拡大に便乗した不審な電話が 153 件確認された他、今年に入ってから高齢者を対象に “PCR 検査やワクチンの接種を受けられるが予約金が必要だ” などとして現金を要求する電話が相次いでいるということです。今日は人との接触を避ける形で注意を呼びかけることにしたもので警察官はポスターを貼り終えると周辺の住宅を一軒一軒回り、郵便ポストにチラシを投函していました。滝野川警察署の小久江亮生活安全課長は “PCR 検査やワクチン接種には金が必要だなどという電話は全て詐欺です不審な電話がかかってきた直ちに電話を切り警察に通報して欲しい” と話していました。では続いて為替と株の値動きです。週明けの東京外国為替市場の円相場は現在ドルに対しては先週末に比べて 6 銭円安ドル高の 1 ドル 103 円 75 銭から 77 銭で取引されています。またユーロに対しては 59 銭円高ユーロ安の 1 ユーロ 125 円 23 銭から 27 銭で取引されています。東京株式市場の日経平均株価は先週末に比べて 251 円 70 銭安い 28267 円 48 銭になっていますまた東証株価指数 TOPIX は 10.74 下がって 1845.87 となっています。1 時の nhk ニュース大野済也がお伝えしました。時刻 1 時 5 分です。 ”

この内容を要約すると五つのトピックがあることがわかる。

一つ目は、“韓国のムンジェイン大統領が記者会見で慰安婦問題について考えを示した。”という部分。キーワードは“韓国”“ムンジェイン大統領”“記者会見”“慰安婦問題”である。二つ目は“新型コロナウイルスの影響で倒産した企業が去年2月から今日正午までで合わせて900社に達した”という部分。キーワードは“新型コロナウイルス”“倒産”“企業”“900社”である。三つめは“北日本や新潟県それに北陸では、荒れた天気となる”という部分。キーワードは“北日本”“新潟”“北陸”“荒れた天気”である。四つ目は“PCR検査について現金を要求する不審な電話が相次いでいることを受けて警視庁はポスターを張り出した。”という部分。キーワードは“PCR検査”“現金を要求”“電話”“警視庁”“ポスター”である。“為替と株の値動き”という部分。キーワードは“為替”“株”“値動き”である。これらの内容を検索し、“01月18日午後1時のNHKニュース”という題名が検索結果として表示されれば、Podcastがエピソードの内容で検索することができることとなり、castumbleが“Podcastの既存の検索機能と違った、任意のPodcastエピソードの情報を素早く取得できるシステム”であるということが言える。

7 結果と考察

結論として、要約文での検索はできなかったが、キーワードでの検索では検索結果にて広く検索をできるようになり、その中に“01月18日午後1時のNHKニュース”という題名が検索結果として表示された。しかし、為替や株などの非常に多くのエピソード数に含まれるものは検索結果に非常に多くの結果が表示されることになり、目的となるエピソードにたどり着くまでに他の検索要件より時間を要した。

“韓国のムンジェイン大統領が記者会見で慰安婦問題について考えを示した。”の場合の検索結果画面を図20に、“韓国のムンジェイン大統領”や“慰安婦問題”等で検索した検索結果画面を図21に、

“新型コロナウイルスの影響で倒産した企業が去年2月から今日正午までで合わせて900社に達した”の場合の検索結果画面を図22に、“新型コロナウイルス”“倒産”“企業”“900社”等で検索した検索結果画面を図23に、

“北日本や新潟県それに北陸では荒れた天気となる”の場合の検索結果画面を図24に、“北日本”“新潟”“北陸”“荒れた天気”等で検索した検索結果画面を図25に、

“PCR検査について現金を要求する不審な電話が相次いでいることを受けて警視庁はポスターを張り出した。”の場合の検索結果画面を図26に、“PCR検査”“現金を要求”“電話”“警視庁”“ポスター”等で検索した検索結果画面を図27に、

“為替と株の値動き”の場合の検索結果画面を図28に、“為替”“株”“値動き”等で検索した検索結果画面を図29にそれぞれ表示する。

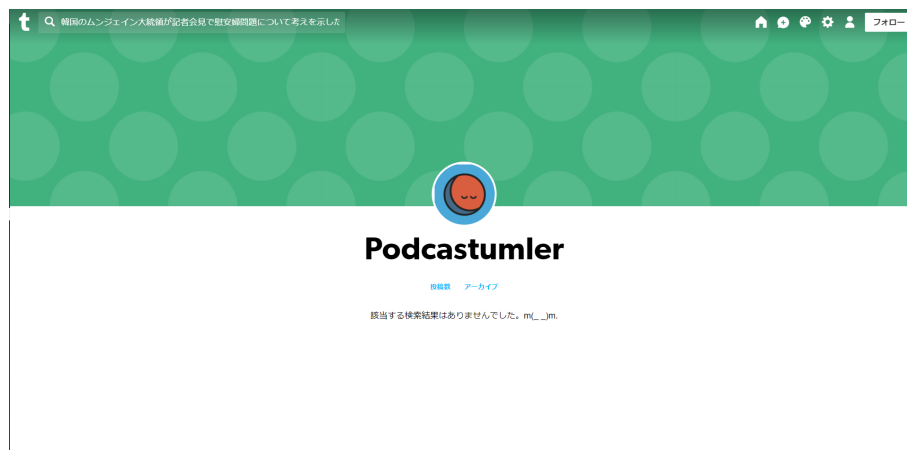


図20 “韓国のムンジェイン大統領が記者会見で慰安婦問題について考えを示した。”の場合の検索結果画面



図 21 “韓国のムンジェイン大統領”や“慰安婦問題”等で検索した検索結果画面

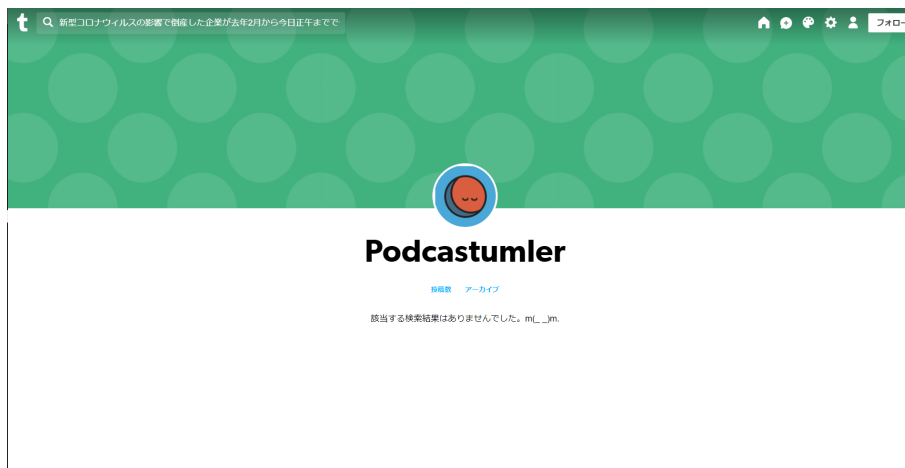


図 22 “新型コロナウィルスの影響で倒産した企業が去年 2 月から今日正午までで合わせて 900 社に達した”の場合の検索結果画面



図 23 “新型コロナウィルス”“倒産”“企業”“900 社”等で検索した検索結果画面



図 24 “北日本や新潟県それに北陸では荒れた天気となる”の場合の検索結果画面



図 25 “北日本”“新潟”“北陸”“荒れた天気”等で検索した検索結果画面



図 26 “PCR 検査について現金を要求する不審な電話が相次いでいることを受けて警視庁はポスターを張り出した。”の場合の検索結果画面



図 27 “PCR 検査” “現金を要求” “電話” “警視庁” “ポスター”等で検索した検索結果画面



図 28 “為替と株の値動き”の場合の検索結果画面



図 29 “為替” “株” “値動き”等で検索した検索結果画面

8 結論

本研究では“Podcast”のエピソードの内容をテキスト化し、収集したエピソードを全文検索を行うことができるシステムの開発を行った。その結果、Podcast にてエピソードの内容での検索ができるようになり、一般的な Podcast アプリケーションと比較した際より広範囲の podcast を検索対象にすることに成功した。しかし、エピソードの内容とは全く違った文章にて文字起こしをされていることが多くあり、具体的には“抜けている箇所がある”“変換がおかしくなっている”“複数人で重なっている部分が抜けている”はじめからすべてが正しく出力される例は非常に少なかった。特に、人間の名前や地名などの固有名詞のものは漢字の変換間違いやカタカナでの表記となる“表記ゆれ”が多数あり、それが検索結果に影響を及ぼしていると思われるが、エピソードを放送している制作者によってその振れ幅が大きいという点があった。また、文節・文脈をくみ取った改行や段落を付ける機能は非実装のため、非常に読みにくい文章として出力される。RSS フィードを出さない音声プラットフォームに依存したコンテンツ、例を挙げると“Spotify 限定配信”などのものは文字起こしをすることができないというのがわかった。BGM やジングルの有無により認識される文字の正誤が変化するという事は無く、それよりも配信主本人の声質や話すテンポ、録音環境による音質の精度が大きく影響しているという点は新しい発見であり、今後の課題へ繋がるものが見つかった。

今後の課題・展望としては、“多言語を織り交ぜたテキスト化と検索機能の実装”“文節・文脈をくみ取った自動段落落とし又は、改行機能の実装”“同音異義語や文節・文脈をくみ取った reblog をより手軽にする文章訂正機能の実装”“エピソードをまたにかけた単語の一発変換機能”の以上4点が上げられる。

謝辞

本論文執筆及び研究室等、研究室での活動の際以外でもご指導・ご協力いただきました大垣斉准教授、情報教育システム研究室の学生、当研究室のOBに深く感謝いたします。

参考文献

- [1] 音声アプリ、ポッドキャストの利用が増加 | google ポッドキャストダウンロードユーザー数、1年で9倍！
<https://lab.appa.pe/2020-07/music-trend-ranking-podcast-apps-20200714.html>.
- [2] Podscope searches podcast content, audio and video. <https://www.searchenginejournal.com/podscope-searches-podcast-content-audio-and-video/1701/>.
- [3] 動画中の音声をテキスト化し全文検索出来る everyzing. <http://www.onflow.jp/cyano/archives/218>.
- [4] 株式会社アニモ 単語集. <https://www.animo.co.jp/glossary/206.html>.
- [5] Speech-to-text の 基本. <https://cloud.google.com/speech-to-text/docs/basics?hl=ja#audio-encodings>.
- [6] Tumblr (タンブラー) とは？機能と活用方法まとめ. <https://saruwakakun.com/it/app/tumblr>.

付録 A ソースコード

A.1 GCS に格納された音声ファイルの文字起プログラム

Listing 1 GCS に格納された音声ファイルの文字起プログラム (transcribe.py)

```
# !/usr/bin/env python
# coding: utf-8
import argparse
import io
import sys
import codecs
import datetime
import locale

def transcribe_gcs(gcs_uri):
    from google.cloud import speech
    from google.cloud.speech import enums
    from google.cloud.speech import types
    client = speech.SpeechClient()

    audio = types.RecognitionAudio(uri=gcs_uri)
    config = types.RecognitionConfig(
        encoding=enums.RecognitionConfig.AudioEncoding.LINEAR16, # 拡張子の設定
wav
        sample_rate_hertz=44100, # 音声ファイルのヘルツ
        language_code='ja-JP') # 日本語の場合

    operation = client.long_running_recognize(config, audio)

    print('Waiting for operation to complete...')
    operationResult = operation.result()

    d = datetime.datetime.today()
    today = d.strftime("%Y%m%d-%H%M%S")
    fout = codecs.open('output{}.txt'.format(today), 'a', 'utf-8')

    for result in operationResult.results:
        for alternative in result.alternatives:
            fout.write(u'{}\n'.format(alternative.transcript))
    fout.close()
```

```
if __name__ == '__main__':
    parser = argparse.ArgumentParser(
        description=__doc__,
        formatter_class=argparse.RawDescriptionHelpFormatter)
    parser.add_argument(
        'path', help='GCS path for audio file to be recognized')
    args = parser.parse_args()
    transcribe_gcs(args.path)
```