

■競技部門「力合わせる六万人」

登録順	タイトル	学校名	指導教員	参加学生
1	驚くべきチーム名を思いついたが、この余白はそれを書くには狭す	佐世保	嶋田 英樹	豊田 虎 (2年)、鳥居 蓮心 (2年)、柴田光太郎 (2年)
2	10倍高速なプログラムを開発します	大阪公大	窪田 哲也	田村 唯 (3年)、池原 大貴 (3年)、年澄 莊多 (2年)
3	聖徳太子の手も借りたいX2	香川 (高松)	柿元 健	竹内 歩夢 (専1年)、大住 陽雲 (3年)、大石 悠馬 (2年)
4	お茶汲み同好会	福井	斉藤 徹	高橋 知也 (3年)、小川 賢仁 (3年)、彦坂 天塁 (3年)
5	せふとくたいしのつくりかた	松江	橋本 剛	青木 蓮樹 (専2年)、三島 知樹 (5年)、安部 滉人 (4年)
6	令和の聖徳太子	舞鶴	森 健太郎	松田 陸 (4年)、植原 一希 (4年)、杉本 辞 (3年)
7	ちはやくれ	呉	藤井 敏則	宮下 翔 (4年)、城 ジュニアスプラタマ (3年)、 宮川 聖章 (3年)
8	過学習	有明	森山 英明	齋藤 健吾 (2年)、猿渡 優衣 (3年)、古川蒼太郎 (3年)
9	とり天Heads!!!!	大分	徳尾 健司	菊池 空 (4年)、北村 玲英 (4年)、工藤 優花 (4年)
10	チームカニ	群馬	川本 真一	水野 哲郎 (4年)、島田慎太郎 (4年)、中野 友晴 (4年)
11	44Cn	鳥羽商船	中古賀 理	酒徳 和輝 (4年)、正住 将太 (4年)、 モハマド シャリザル (4年)
12	Lynx lynx	鹿児島	原 崇	山田 裕貴 (3年)、酒匂 悠宇 (3年)、文榮 来智 (3年)
13	業火の締切	香川 (詫間)	宮崎 貴大	長野 匡吾 (4年)、小原 崇靖 (3年)、則包 創太 (3年)
14	プレーメンの学術隊	津山	宮下 卓也	川上 功介 (3年)、森中 智己 (3年)、瀬良 瑞葉 (3年)
15	Ascension	奈良	岩田 大志	藤本 光 (4年)、松田 蒼太 (4年)、和田 颯太 (2年)
16	聞き分けのいいゴボウ	和歌山	森 徹	江阪 治人 (3年)、武本 龍門 (3年)、本間 陽季 (3年)
17	チーム末代	東京	松崎 頼人	古川 紘基 (3年)、平田健太郎 (3年)、三浦 理稀 (3年)
18	ソフトウェア研究部会	仙台 (名取)	北島 宏之	佐藤 至 (3年)、鈴木 佑 (2年)
19	NIT-ishikawa	石川	越野 亮	堀 彰悟 (専1年)、坂井 俊介 (専1年)、 佐久間 葉 (専1年)
20	競技部門の話は耳に入りましたよー。詳しい方に確認しますね。	八戸	細川 靖	二川目裕太 (4年)、中村 亮太 (3年)、小笠原涼太 (2年)
21	418. I'm a teapot.	小山	干川 尚人	横井 蛭汰 (3年)、飯野 雅翔 (3年)、加藤 裕槻 (2年)
22	聖徳太子. tar.gz	苫小牧	山本 椋太	推名 晃 (3年)、梶谷 征矢 (3年)、寺井 勇真 (3年)
23	while文ってなんだっけ？	熊本 (八代)	小島 俊輔	嶋中 海人 (3年)、小山 智寛 (3年)、辻中 煌希 (1年)
24	サボリ推奨委員会 Lv.7	新居浜	占部 弘治	神田 凌佑 (5年)、青野 真弥 (1年)、前田 隼 (1年)
25	八尺瓊蟹蒲	神戸市立	朝倉 義裕	入本 聖也 (4年)、西ヶ峰克隆 (4年)、山本 大地 (3年)
26	チョコクロワッサンはオープンで1分30秒くらいがちょうどいい	木更津	大枝 真一	越智 優真 (2年)、大原 陽治 (2年)
27	アルミホイールで身を守ろう	仙台 (広瀬)	力武 克彰	門馬 琢磨 (4年)、吉田 悟 (3年)、渡邊 天 (2年)

競技部門のルール

今年の競技部門では、コンピュータによる「かるた」を行います。

かるたには群馬県の郷土かるたである上毛かるたを使用します。上毛かるたは44枚の取り札と読み札で構成されています。日本語版と英語版の2種類があり、本競技では両方を使用します。

通常のかるたは1枚ずつ読み上げますが、本競技では複数枚を同時に読み上げるので、読み上げを聞き分けなければなりません。

問題データと分割データ

事前に用意された読み札を読み上げた音声データいくつか重ね合わせて問題データを作成します。同じ絵札の日本語版と英語版の両方が含まれることはありません。重ね合わせる際に読みデータの位置をずらしたり、冒頭と末尾の一部を削除したりすることがあります。重ね合わせは最小で3、最大で20です。

問題データをいくつかの分割データの分割します。分割データの長さは一様ではなく、分割数は最小で2、最大で5です。分割数は試合開始前にお知らせします。競技者は取得するデータ数を指定することができ、使用した分割データ数に応じて、ボーナス係数が与えられます。ボーナス係数は使用した分割データ数が少ないほど高い係数です。

取りと変更

問題データに含まれる読みデータと関連した絵札を選択することを取りと言います。取りは未選択の札のほか、前の問題データの取り札や変更札も取ることができます。未選択の札を取った場合はそれを取り札と言います。問題データの読みデータ数まで札を取ることができます。複数の問題データに対して同一の絵札を取ることはできません。

前の問題の取り札や変更札を取ることを変更と言

い、変更した札を変更札と言います。変更すると前の問題で取った札が減りますが、その分を追加することはできません。変更札は枚数に応じて減点されます。変更札を別の問題で再度変更しても2重には減点されません。

正解とお手付き

すべての問題が終了すると各問題の取り札と変更札が確定します。すべての問題が終了したときにすべての44枚の札が使われているとは限りません。すべての問題が終了後、各々の問題データに含まれている札を示します。

問題ごとにデータに含まれる取り札と変更札の両方を正解とし、正解した札のことを正解札と言います。正解札にはポイントが与えられ、1枚当たりのポイントにボーナス係数を乗じたものが正解ポイントとなります。

問題データに含まれていない札を取り札もしくは変更札に選択することをお手付きと言います。お手付きした札のことをお手付き札と言います。お手付き札は枚数に応じて減点されます。

制限時間と問題数

問題ごとに回答の制限時間を定めます。制限時間は2分～10分の予定です。

1試合を複数の問題で順次行います。問題ごとに、重ね合わせ数や分割数、ボーナス係数、制限時間などが異なることがあります。

順位決定方法

勝敗判定は以下の優先順位で決定します。

1. 合計ポイント
2. 正解ポイント
3. 変更減点
4. サイコロの目で勝負

競技部門の組合せ

■ 1回戦

ブース	第1試合	第2試合	第3試合	第4試合	第5試合
1	八戸	石川	富山(射水)	小山	鹿児島
2	大阪公大	神戸市立	明石	群馬	長野
3	津山	和歌山	松江	阿南	大分
4	佐世保	サレジオ	苫小牧	長岡	大島商船
5	東京	香川(高松)	有明	熊本(熊本)	豊田
6	沼津	米子	熊本(八代)	鶴岡	久留米
7	仙台(広瀬)	鳥羽商船	奈良	新居浜	木更津
8	鈴鹿	都立(品川)	弓削商船	富山(本郷)	舞鶴
9	呉	旭川	香川(詫間)	福井	

※ 各試合上位4チームが準決勝へ進出する。5位以下は敗者復活戦へ。

■ 敗者復活戦

ブース	第1試合	第2試合	第3試合
1	1-1-5	1-3-5	1-5-5
2	1-2-5	1-4-5	1-1-6
3	1-3-6	1-5-6	1-2-6
4	1-4-6	1-1-7	1-3-7
5	1-5-7	1-2-7	1-4-7
6	1-1-8	1-3-8	1-5-8
7	1-2-8	1-4-8	1-1-9
8	1-4-9	1-3-9	1-2-9

※ 各試合上位2チームが準決勝へ進出する。

■ 準決勝

ブース	第1試合	第2試合	第3試合
1	1-1-1	1-3-1	1-5-1
2	1-2-1	1-4-1	1-1-2
3	1-3-2	1-5-2	1-2-2
4	1-4-2	1-1-3	1-3-3
5	1-5-3	1-2-3	1-4-3
6	1-1-4	1-3-4	1-5-4
7	C-1-1	1-4-4	1-2-4
8	C-2-2	C-2-1	C-3-1
9		C-3-2	C-1-2

※ 各試合上位3チームが決勝へ進出する。

■ 決勝

ブース	第1試合
1	S-1-1
2	S-2-1
3	S-3-1
4	S-1-2
5	S-2-2
6	S-3-2
7	S-1-3
8	S-2-3
9	S-3-3

※ X-Y-ZはX回戦-第Y試合-第Z位を表します。

ただし、X部分のCは敗者復活戦、Sは準決勝を表します。

・提出された原稿をそのまま印刷しています。

1	驚くべきチーム名を思いついたが、この余白はそれを書くには狭す	佐世保	豊田 虎（2年）鳥居 蓮心（2年） 柴田 光太朗（2年）嶋田 英樹（教員）
---	--------------------------------	-----	--

1. 概要

開発するシステムは、問題データに対して音源分離を行い、個々の音声データを生成し、これらの音声データをスペクトログラムに変換した後、事前に機械学習を行って得られた学習データを用いてカルタの判別を行う。

2. 音源分離

音源分離は Deep Clustering という手法を用いて行い、入力データが問題データ、出力データが問題データから分離された音声データとなる。Deep Clustering を用いた音源分離は、他の手法と違い、話者に依存せず、今大会の「日本語版と英語版の読み札で話者が異なる」という点に適していると考えられたため、採用した。

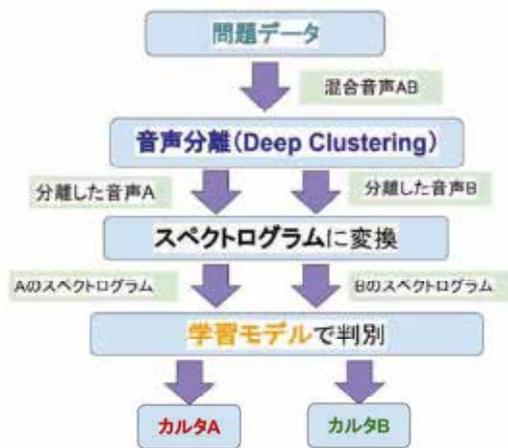
3. スペクトログラムに変換

分離した音声データをスペクトログラムに変換し、音声の特徴を画像から判断できるようにする。スペクトログラムを使用した理由は、音の強弱や時間の観点から音声データの特徴を画像として表現でき、機械学習での画像判別が容易に可能であると考えたためである。

4. 機械学習によるカルタの判別

事前に機械学習によって全ての各カルタのスペクトログラムを学習させ、学習データを作成する。この学習データを用いて、分離した音声から生成されたスペクトログラムに対して画像認識を行い、カルタの読み札を判別する。

の耳でカルタの判別が正しいかどうかを確認できるようにする。



図：問題データから読み札検出の過程

5. 開発環境

[言語] python
[IDE] Visual_Studio_Code・Anaconda

2	10倍高速なプログラムを開発します	大阪公大	田村 唯（3年）池原 大貴（3年） 年澄 莊多（2年）窪田 哲也（教員）
---	-------------------	------	---

1. この問題は何ですか

事前に与えられたベクトルがいくらか「ずらされて」足し算されているので、どのベクトルが含まれているか当ててください。

2. 適当に計算したら解けませんか

厳密に計算しようとするとう部分和问题が出てきます。ところで部分和问题は NP 完全に含まれる非常に難しい問題であり、そのうえ今回は入力サイズの時点で非常に大きいです。このアプローチは絶望的です。

3. じゃあどうしますか

今回はノイズが少ない読み上げの音声なので、わりと整った波形です。これをうまく利用した評価関数を作ることによって、各ベクトルが含まれている可能性を計算することができます。これを利用すると少なくとも貪欲法で処理できる形になります。

4. 計算量が大きすぎませんか

大きいので、PC 間通信や並列処理を最大限用いて時間短縮します。

5. 機械に全部任せて大丈夫ですか

試合進行の処理まで任せるのは開発作業が大変なので、代わりに GUI を作ってその場で入力できるようにします。GUI の実装はオープンソースのライブラリを活用します。

6. 10 倍速いプログラムは作れますか

10 倍なんて差はめったに出るものではありません。諦めましょう。何よりも今回の問題は答えるまでの時間が得点に反映されないで、時間を考える前に精度を上げるべきです。

3

聖徳太子の手も借りたい X 2

香川
(高松)

竹内 歩夢 (専1年) 大住 陽雲 (3年)
大石 悠馬 (2年) 柿元 健 (教員)

1. はじめに

今回の競技部門は、コンピュータによる「かるた」である。しかし、一般的な「かるた」とは異なり、複数枚同時に読み上げられ、それらの音声を聞き分け、対応する絵札を選択しなければならぬ。アルゴリズムの方針と GUI について以下に示す。

2. アルゴリズム

事前に公開された音声データベースをすべて連結し、周波数を行、音声データベースをすべて連結した時の時間を列とした行列を A、音声データベースをすべて連結した時の時間を行、問題データの時間を列とし、各要素が 0 か 1 の行列を B とし、行列 A と行列 B の積を行列 C とする。この時、行列 C が問題データと限りなく近くなるように行列 B を求める。また、問題データは音声データベースの一部

を切り取ったものであるという制約を活用する。さらに、行列 C から音声を作成し、ヘッドホンを着いて元音声と比較し確認を行う。

3. GUI

今回はどの絵札をすでに選択したか、分割データを何個取得したかの状態や、正解ポイント、ボーナス係数などのパラメータが多くあるので、それらをわかりやすく表示する。

4. 開発環境

OS : Windows10/Arch Linux

使用言語 : Python/C++

IDE : Visual Studio Code/Visual Studio

Framework : OpenSiv3D

4

お茶汲み同好会

福井

高橋 知也 (3年) 小川 賢仁 (3年)
彦坂 天壘 (3年) 齊藤 徹 (教員)

1. はじめに

今回の競技は、(a) 合成前の音声で学習させた音声認識モデルを作成する。(b) 音声の序盤から 1 のモデルを使用し一つの解を確定させる。(c) (b) の逆位相波を元の波に合成し音声を分離させる。という三行程を繰り返すことで解を得られると考えた。

2. 音声認識モデル

音声を一度文字列に変換し、文字列から解の確定を行うことで、人の手による修正が容易になる。そこで DNN-HMM による音声認識を採用した。

また、今回のモデル作成にあたって高島遼一著 python で学ぶ音声認識のプログラムを参考にし、改良を施した。

3. 読み札との一致度を調べるアルゴリズム

1. 行程 (b) では以下のようなアルゴリズムを行っている。
与えられた文字列 (検索文字列) をそれぞれ、 $N=1$ として、

N 文字目と $N+1$ 文字目のペアが含まれる札を予め用意しておいた json ファイルを元に絞り込む。そして、 N を 1 増やし再度絞り込み、 N が一つ小さい状態の時の絞り込んだものと比較して被っていない札を消去する。これを、札が一つになるか、 $N+1$ が各検索文字列の最後の文字になるまで繰り返す。終了後、残りの札の種類と 1 つの札に含まれている量に応じてそれぞれの札に『信用値』を加算する。これを全ての検索文字列において、実行後『信用値』順にソートして、検索文字列の個数だけ読まれた札の個数として、返す

4. 開発環境

OS: Windows10, 11

言語: Python

IDE: Visual Studio Code

1. システム概要

日本語と英語の読み札が合わせて 88 個あるため、88 個の畳み込みニューラルネットワーク (CNN) モデルを学習させ、それぞれのモデルが対応する読みデータが含まれているかを学習する。88 個の学習済みモデルに問題データを入力し、それぞれの出力値から問題データに含まれる読みデータを推測する。

2. アルゴリズム

2.1 機械学習モデル

ResNet は、画像分類において高い精度を実現することで知られる CNN モデルである。ResNet を参考に、一次元データに対する CNN モデルを作成する。このモデルは、入力に与えられたデータに目当ての読みデータが含まれる確率を出力する。このモデルを 88 個作成し、すべての読みデータについて学習する。

2.2 取りアルゴリズム

88 個のモデルの出力値を見て、大きいものから取り札にする。ここで閾値を定め、出力値が閾値を超えなかった場合は取り札にせず、お手付きや変更を防ぐ。

3. 学習方法

問題データは読みデータを位置ずらしや切り抜きした後に重ね合わせることで作成され、読みデータは公開されているため、問題データ生成プログラムを作成し、これによって生成された問題データを用いて学習を行う。今回の競技では未知のデータは与えられないため、過学習のリスクがないと考えられる。そのため、延々と学習を繰り返しても精度が上がら続けると考えられる。

4. 開発環境

OS : Windows10, Windows11

言語 : Python3

1. はじめに

同一の問題に対して、アルゴリズムによる分析と機会学習による分析の 2 種類のアプローチで解析を行うことでより高い精度での解析を実現する。

2. 分析方法

2.1 アルゴリズムによる分析

アルゴリズムによる分析は与えられる問題に予めサンプルで与えられた音声の読み札を重ね合わせる。もとの音声データの内、新たに重ね合わせた音声データの波形とのズレが小さいものが問題に含まれている読み札とする。

2.2 機械学習による分析

問題の音声データから MFCC を求め、デルタケプストラムも含めて特徴量とし、機械学習を用いて各音素が含まれるかを二値分類で求める。モデルの学習は、自分たちで合成した疑似的な問題データを用いて行う。

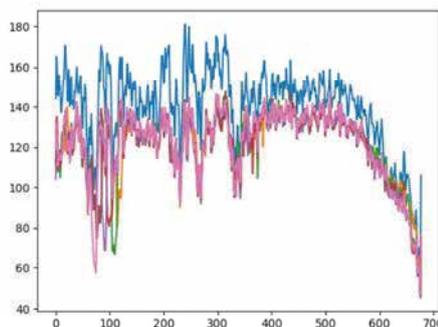


図 1 波形を比較している様子

3 UI

GUI で UI を作成してコマンドラインで操作できるように実装する。機能として問題の取得、アルゴリズムや機械学習から推察された読み札の候補などの表示を行う。

4 開発環境

OS: ArchLinux, Windows10 言語: Python3

7

ちはやくれ

呉

宮下 翔 (4年)
城 ジュニアスプラタマ (3年)
宮川 聖章 (3年) 藤井 敏則 (教員)

1. はじめに

今回の競技内容は音声の解析であり、音声解析においてよく使われる手法である短時間フーリエ変換を用いたシステムを開発した。

2. 音声の解析

今回は音などの波形データから特徴を抽出するときによく用いられる手法である短時間フーリエ変換を用いて音声の解析を行う。

今回は複数の音源の重ね合わせがあるが、音源を重ね合わせる際に特定周波数の信号の強さは大きくなることはあるが小さくなることはない。そのためそれぞれの読みデータの信号が強い部分を記録しておき、それが含まれているかどうかで取る札を決定していくようにした。

また、今回は短時間フーリエ変換を行う際にパラメータ

を変更したものを2つ用意し、2台のパソコンを用いて解析を行い、結果を比較することでより正確な判断を行えるようにした。

3. GUI について

今回、GUI は音声の解析のため手動で何かを操作したり、修正したりといった必要がほぼないことから工数削減のために短時間フーリエ変換を行ったあとのスペクトログラムを確認できる程度の簡単なものにした。

4. 開発環境

[使用言語]C++, Python

[実行環境]Windows subsystem for Linux(Ubuntu)

競技部門

8

過学習

有 明

齋藤 健吾 (2年) 猿渡 優衣 (3年)
古川 蒼太郎 (3年) 森山 英明 (教員)

1. システムの概要

今回の競技部門ではかたるの読みの一部を重ね合わせた音声を扱う。そこで、使用された音声の特徴などを捉えるため、畳み込みニューラルネットワーク (CNN) という機械学習の手法を用いる。このとき、音声データのままだでは機械学習を行うことが困難であるため、問題データに近い音声を作成するプログラムで音声を用意し、画像に変換することで機械学習を容易にすることを旨とする。

下の図1はシステム概要を簡易的に表したものである。

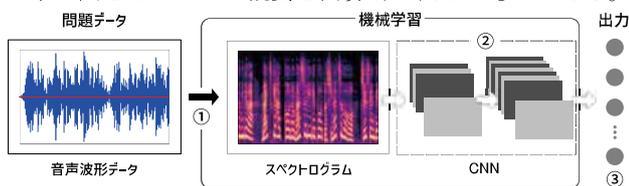


図1. システム概要

2. データセットと学習用画像の作成

機械学習を行うには、入力する画像と正解となるラベルが対応しているデータセットが必要になる。このため、問題データに近い音声を生成するプログラムを作成する。

正解となるラベルは音声とともに生成し、読みデータの総数が日本語と英語がそれぞれ44個の計88個あるため、1か0の88個の整数値の配列とする。

生成した音声からスペクトログラム分析により、周波数と時間を軸にとるグラフを画像として出力する (図①)。ここで出力した画像は機械学習に用いるものである。

3. 機械学習

畳み込みニューラルネットワーク (CNN) を使用する。これは、入力する画像の特徴を捉えて判別を行うものである。

画像の隣り合ったピクセル同士には波形のように連続性がある。その特徴を生かし、データセットとして作成した各画像に対し、様々なフィルタをかけて画像の枚数を増やした後、ピクセル数を小さくしていくことでどの音声データの特徴を有しているか、識別しやすくする (図1②)。

最終的に機械の予測値は使用された音声は1、使用されていない数字を0に近い値にし、88個出力する (図1③)。

これを正解となるラベルとともに損失関数に渡すことで予測値と正解がどのくらい離れているのかの誤差をとる。誤差が小さくなるほど、正しく予測ができていないと判断する。また、1つのデータセットを何度も繰り返し学習させることで、機械が正解となるラベルに近い予測値を出力できるように適応させる。

なお、大会本番で用いる出力は、予測値が指定した値以上であれば1、指定した値未満であれば0の整数値に変換することで、どの音声を使用したのかの可能性が高いか判断する。最終的に別のプログラムに結果を渡すことでの絵札を選択する。

9

とり天Heads!!!!!!

大分

菊池 空（4年）北村 玲英（4年）
工藤 優花（4年）徳尾 健司（教員）

1. 概要

声を読み込み画像化する段階と、画像化した音声にどの読み札が含まれるかを推測する段階の2つに分けて考える。

2. 音声の画像化

音声の画像化を行うために、短時間フーリエ変換を用いてスペクトログラムの作成を行う。この作業は78通りの各読み札の中で他の読み札に含まれない文字列(固有文字列)が読まれている部分の音声と、サーバから受け取った読み札が複合されている音声全てに行う。

3. 複合された音声の推定

受け取った音声にどの読み札の音声が含まれているかを調べるため、スペクトログラムを入力とし、各読み札がスペクトログラム化された音声に含まれている確率を出力とするモデルをCNNによって作成する。

4. 読み札の選択

競技が始まると音声をサーバから受け取ってスペクトログラム化を行い、モデルを使って確率を求める。その後、受け取ったデータに含まれる確率がしきい値以上の読み札に対応する取り札を選択していき、複合されている読み札の数に達したら探索を終了する。このときに取り札の選択数が複合されている読み札の数より少ない場合は、しきい値を下げて再度取り札を選択していく。それでも取り札の選択数が複合されている読み札の数より少ない場合は、再度サーバから音声を受け取って同じ処理を繰り返す。サーバから音声を受け取ることができなくなった場合にも探索は終了する。

5. 開発環境

開発環境：PyCharm, VSCode

言語：Python

10

チームカニ

群馬

水野 哲郎（4年）島田 慎太郎（4年）
中野 友晴（4年）川本 真一（教員）

1. 解法の概要

問題データをスペクトログラムに変換し、読みデータのスペクトログラムと比較を行うことで、読みデータを推測する。

2. アルゴリズムの詳細

2.1 読み札の推察

問題データのスペクトログラムは、読みデータのスペクトログラムの一部を線形に足し合わせたものになると考えられる。問題データのスペクトログラムから、読みデータと類似するパターンを発見することで、問題データに含まれる読みデータを推測する。

類似するパターンの発見は、問題データと読みデータのスペクトログラムのユークリッド距離やコサイン類似度、相関係数などの複数の要素を加味して行う。

2.2 分割データ取得の戦略

分割データを取得するタイミングは、前述のアルゴリズムが出力した結果を評価して数値化し、これを人間が確認することによって決定する。結果の評価は、正解として出力された読みデータと問題データの類似度をもとに算出する。

2.3 GUI

多様な戦略に柔軟に対応できるように、問題データの取得からアルゴリズムの実行、評価、分割データの取得、問題の提出までをGUIによって実現する。

3. 開発環境

OS：Windows11, macOS

言語：Rust, Python

GUI ライブラリ：TkInter

1. 実現方法

1.1 音声分離

(1) 日本語と英語の音声分離

男性と女性の声の周波数の違いを利用し、日本語と英語の音声を分離する。この分離された音声は、複数の音声が混ざっている状態である。

(2) 各言語の分離

日本語と英語に分離した音声は複数の問題が混ざった状態であるため、それぞれ独立成分分析を行うことで音声を問題ごとに分離する。

1.2 分離音声から札を推定

(1) 分離した音声のノイズ除去

独立成分分析をおこなった音声にノイズ入っていると、

近似度を求める際に影響を及ぼすため、ノイズの除去を行う。日本語・英語の各言語によって設定した一定の高さの周波数をノイズとみなし、それを除去する。

(2) 音声の近似度から札を推定

分離された音声を事前に用意しておいた音声データの波形と比較し類似度の高い音声データが正解の札とする。独立成分分析に時間がかかるため、ここでは類似度を使うことで時間を短縮して正解の札を推定することができる。

2. 開発環境

言語：python

OS：windows10

1. はじめに

今回の課題は、与えられた分割データから、44枚の取り札のどれを取るべきかを推測することだと言える。本システムでは、このマルチラベル分類問題を解く。

2. 音声解析モデル

分割データがすべてそろっていない場合、与えられている分割データのサンプル数の平均をとり、その平均が欠けている分割データのサンプル数であると考えて0でパディングする。

欠けている分割データを0でパディングした全体のデータに対して短時間フーリエ変換を行い、スペクトログラムを作成する。この際、短時間フーリエ変換の窓幅と移動幅を調整することにより、波形データの時間的な長さによらずスペクトログラムのサイズが一定となるようにする。

44枚の絵札がそれぞれ含まれているかどうかを出力と

する畳み込みニューラルネットワークにスペクトログラムを入力し、どの絵札を取るかを決定する。

畳み込みニューラルネットワークの構成については、様々な構成のネットワークを構成して実験的に最良のネットワーク構成を求める。

3. GUI

音声解析モデルによって出力された取るべき絵札は、間違いが含まれることがある。音声解析モデルの精度が高いとしても数枚は誤判定されると考えられる。このような場合に対応するために、音声解析モデルによって出力された回答を人間が吟味するためのGUIを開発する。

4. 開発環境

OS：Windows10

言語：Python

ライブラリ：TensorFlow、librosa

13 業火の締切

香川
(読問)

長野 匡吾 (4年) 小原 崇靖 (3年)
則包 創太 (3年) 宮崎 貴大 (教員)

1. システム概要

出題された音声ファイルを配列に変換し、その配列からメルスペクトログラムを生成する。生成したメルスペクトログラムを元に、問題の音声ファイルを構成する読みデータを特定する。

2. 読みデータ特定のアプローチについて

2.1 ディープラーニングを用いた解法

事前に配布されている読みデータを使って実際の問題と同じような音声ファイルを予め生成し、「システム概要」で記述したメルスペクトログラム画像を学習データとして保存する。これらの画像から、それぞれの読みデータが含まれているとき、どのような特徴が表れているかを、CNNを用いて学習させ、読みデータの特定を行う。

2.2 遺伝的アルゴリズムを用いた解法

言語と読み始めの時間を遺伝子とする。この遺伝子の情報から音声データを生成し、メルスペクトログラムを生成する。このメルスペクトログラムと、問題の音声ファイルから求めたメルスペクトログラムとの差を二乗し、総和を求める関数を評価関数とする。この評価関数の値を元に問題の音声ファイルを再現することで、問題の音声ファイルを構成する読みデータを特定する。

3. 開発環境

環境 : Visual Studio 2019, Visual Studio Code,

Google Collaboratory

使用言語 : Python3

主要ライブラリ : librosa, wave, NumPy,

Matplotlib, TensorFlow

14 ブレーメンの学術隊

津山

川上 功介 (3年) 森中 智己 (3年)
瀬良 瑞葉 (3年) 宮下 卓也 (教員)

1. 音声の解析アルゴリズム

今回の問題は事前に配布される読みデータを使った、テンプレートマッチングの実装により、問題データにどの音声データが含まれているかの解析を行う。読みデータをフーリエ変換して値を配列に保存する。送られてきた問題データもフーリエ変換して配列に保存する。そして各サンプルどうしの差の総和を求める。この値が小さければ読みデータと問題データの違いが少ないことがわかるので、どの札が読まれているか推察できる。また、問題データは読みデータを伸縮せずに重ねているため、読みデータとの差が0の区間がある可能性がある。もしあればどの札が読まれているか確定できる。今回の問題は重ね合わせる際に多少ずれが生じる。その対策として、テンプレートマッチングに端点フリーを導入する。例えば、音声の重なりが少ない区間に目をつけてそれ以外の区間を無視してテンプレ

トマッチングをし、音声の重なりが増える瞬間までを無視してテンプレートマッチングをするなどをする。こうすることで推定の精度を高めることができる。

2. その他

精度を上げる方法として、問題データを短い区間で区切りその区間が英語に近い日本語に近いかを識別する方法をとる。まず読みデータをフーリエ変換し、日本語読みデータと英語読みデータの違いを検出する。次にその違いを使い問題データをフーリエ変換したときに日本語と英語のどちらが多く含まれているかを調べる。最後に英語のテンプレートを使うか日本語のテンプレートを使うか両方を使うかを選択する。それぞれを短時間ごとのスペクトルで区切り問題データと読みデータを比較することで時間を短縮させる。

15 Ascension

奈良

藤本 光 (4年) 松田 蒼太 (4年)
和田 颯太 (2年) 岩田 大志 (教員)

1. はじめに

私たちは、問題データの解析アルゴリズム部分と、札の選択アルゴリズム部分に問題を分割できると考えた。ここではそれぞれのアルゴリズムについて記述する。

2. 問題データの解析アルゴリズム

問題データの解析アルゴリズムでは、問題データと全ての読みデータの両方を複数方法で解析し、各札に対する類似度を求める。解析では、周波数や振幅、音声の連続するデータの差などを問題データと読みデータに対して求める。また、読みデータに関しては時間短縮のため予め解析しておきデータをローカル上に保存しておく。類似度の計算では、問題データと読みデータそれぞれの解析結果の距離を利用して各読みデータの問題データとの類似度を算出する。

3. 札の選択アルゴリズム部分

問題データの解析アルゴリズムの精度が最重要とされるため、札の選択アルゴリズムは基本的に問題データの解析結果による各札の類似度と過去の選択した札の類似度、をもとに過去を含めた取り札の類似度の合計が最大となるように今回の問題データに対応する取り札を選択する。選択方法としてはビームサーチや焼きなまし法などを用いて最適な解を求めていく。また最適な解を求める上で、札ごとにお手つきなどの重みなども考慮する。

最終的には人間の目で各札の類似度の値から取り札を判断して、解答が怪しい場合は人間の手で修正する。

16 聞き分けのいいゴボウ

和歌山

江阪 治人 (3年) 武本 龍門 (3年)
本間 陽季 (3年) 森 徹 (教員)

1. 概要

音源を数値化し、数値を比較することで、どの音声が含まれているのかを判別するプログラムです。

2. プログラムについて

2.1 二次元配列化

時間ごとの周波数と音の大きさを数値化し、全音源データを二次元配列化する。

2.2 一致率の測定

句データ配列に問題データ配列を重ね合わせた際の数値の差異を合計し、差異が最も少ないものを一致データとする。

重ね合わせは問題データの先頭を配列 1 サンプルずつずらし、句データの中で一致率が一番高くなる場所を探す。

2.3 音声の消去

問題データの該当する部分に一致率が最も高い句データの逆位相をかけて該当する句の音声を問題データから消去する。また、消去した句の種類を記録しておく。

問題データから完全に音声が消えるまで一致率の測定に戻る。

3. 開発環境

言語: Python3.8

開発環境: VisualStudio2022 VisualStudioCode

Windows10

17 チーム末代

東京

古川 紘基 (3年) 平田 健太郎 (3年)
三浦 理稀 (3年) 松崎 頼人 (教員)

1. はじめに

今回の競技は取得した問題データ以外に、その試合内で既にとった札やまだ取得していない問題データを考慮する必要がある。そのため問題データから読み札を抽出することと、抽出結果や取札をもとに最適な行動をさせることの2つを考える。

2. 読み札の抽出

メルスペクトログラムに変換した問題データをConv-TasNetをベースにした深層学習モデルにかけることで88の各読み札の音声データに抽出する。

3. 行動の選択

抽出したデータと読み札のサンプルデータからCCFを用いて類似度を出す。閾値を超えた類似度を持つデータの数が、指定された読み札の数を超える場合は問題データを追加で取得する。また、複数の問題データが閾値を超えた類似度を持つ場合も同様に取得する。

3. 開発環境

OS : Windows Linux macOS
言語: C++ Python

18 NIT-ishikawa

石川

堀 彰悟 (専1年) 坂井 俊介 (専1年)
佐久間 葉 (専1年) 越野 亮 (教員)

1. はじめに

深層学習による音源分離モデルと音声認識モデルを用いて、①問題音声の分離、②分離された音声の認識、2段階構成のシステムとして開発している。

2. 音源分離モデル

2.1 SepFormer (Separation Transformer)

Transformer を音源分離に応用した深層学習モデルで、2話者と3話者を分離できるモデルがある。

2.2 4話者以上の問題音声の場合

4話者以上の場合は、SepFormer にかけて分離した音声を繰り返し分離させて音声認識にかける。繰り返した結果、音声の品質が劣化する場合は、分離する前の音声を認識にかける。

3. 音声認識モデル

3.1 YAMNet

YAMNet は、MobileNet という軽量かつ高性能な畳み込みニューラルネットワークで、AudioSet という Youtube 動画の約 5800 時間の音声データで学習された事前学習モデルを用いる。

3.2 YAMNet による読み札の認識

事前学習モデルに、上毛かるた読み上げ音声データ (日本語と英語の計 88 ラベル) を用いて、ファインチューニングさせたところ、重なっていない音声であれば切り出した音声でも認識できることを確認した。

1. システム概要

事前に用意した学習済みモデルによる予測を PC 上で実行する。試合では 3 台の PC でそれぞれ実行することによって複数の回答を作成する。

2. データの前処理

問題データは、それぞれの札の読み札の音声ごとに分類を行う。その後、分離したそれぞれの音声データの周波数特性をとり、モデルに学習させやすい形式、サイズに変更を行う。

3. 音声分類

周波数特性をとったデータをディープラーニングによって学習させ、音声の分類が可能ないように学習をしていく。

なお、本競技では日本語の読み札音声と英語の読み札音声が入り混じることがないため、日本語用のモデルと英語用のモデルの 2 種類を用意し、それぞれを学習させる。

また、音声分離をせずに特徴量をとって学習させるモデルも用意し、2 種類の解答を作ることによって解答の信頼性の向上を図る。

4. 開発環境

言語: Python

エディタ、実行環境: Anaconda Jupyter Notebook

主要ライブラリ TensorFlow, Keras, Pydub, tkinter

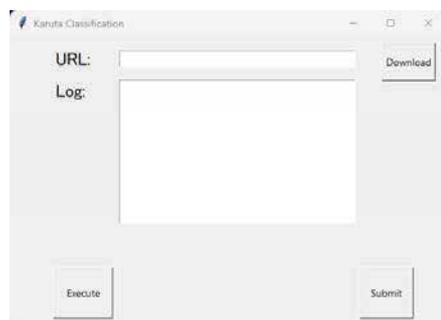


図 GUI の例

1. はじめに

今回、問題データから読みデータを特定するために、音源分離と探索の二つの部分に分けた。つまり、音源分離を用いて問題データを分離し、探索で読み札を特定し取り札を取る。

2. システムについて

2.1 音源分離について

音源分離は半教師あり NMF を用いて行う。教師データは事前に公開される読みデータとし、問題データから一つずつ読みデータを取り除いていく。分離の際の距離関数はユークリッド距離を用いる。正しい教師データであったかどうかは分離したデータと読みデータの一致度から判断する。また、処理の高速化を図るために GPU を用いた並列処理や既に取り除いた読み札を教師データから除くなどする。

2.2 探索について

分離した音源を高速フーリエ変換処理し、音源中に音圧がしきい値を超える点の時間、周波数をフィンガープリントとする。事前に J01.wav ~ J44.wav 及び E01.wav ~ E44.wav を同様の処理しておき、ハッシュテーブルを作成する。

分離された音源のピーク値をハッシュ化し、ハッシュテーブルより探索する。

2.6 開発環境

OS: Windows11

言語: Python, C++, Cuda C/C++

仕様ライブラリ: librosa, Siv3D

エディター: VisualStudio, VisualStudioCode

プロジェクト管理: Git

1. はじめに

今回の競技では問題である合成音声データ(以下、問題音声データ)に含まれるかるたの読み音声データ(以下、読み音声データ)が事前に渡されるため、読み音声データの逆位相による音声の相殺が可能であることに着目して解法の方針を立てた。

2. 逆位相による正解データの推測

基本的な方針としては問題音声データに対して任意の開始位置、任意の読み音声データの逆位相を合成し、全体の振幅の減少度合いから含まれる正解データを推測する。

逆位相による相殺が発生した際には、合成した区間での振幅が著しく減少する。その区間での平均振幅がしきい値以下になった場合に逆位相として合成している読み音声データが正解データとして判定する。しきい値としては、問題音声データの合計振幅から読み音声データの合計振

幅を引き、問題音声データの秒数で割った値をベースに使用している。

3. 正解データの開始位置予測

問題音声データに対する任意の開始位置と任意の問題音声データを走査し処理を行うと計算時間がかかるため、開始位置を絞り込むことで高速化する。開始位置予測にはかるた特有の抑揚を利用する。日本語の読みデータにおいてはどの読みデータも抑揚が近いことから音声のピークの時刻がある程度絞られる。それを利用し、平均的な日本語読みデータモデルを作成することで問題音声データの1つのピークから複数の予測開始位置を提示することができる。

4. 開発環境

言語: Python

開発環境: Visual Studio Code

1. システム概要

事前に配布された「問題データ」を模した学習用データを「読みデータ」から大量に作成し、コンピュータに学習させる。人間の音声知覚の特徴を考慮している点を加味し、今回は MFCC(メル周波数ケプストラム係数)を学習データとする。音声データを画像データとして扱うことで、CNNでの学習・認識が可能となる。

2. 解析アルゴリズム

学習用データの音声ファイルから MFCC を求め、グラフ画像化する(以下、中間画像)。アーキテクチャは VGG 系を使用する。入力を中間画像、出力に含まれる読み札を 1 とする正解ベクトルとして、CNN にてモデルを作成する。これらの中間画像と組み合わせた音声データの数を CNN に与え、認識・学習し解を求める。

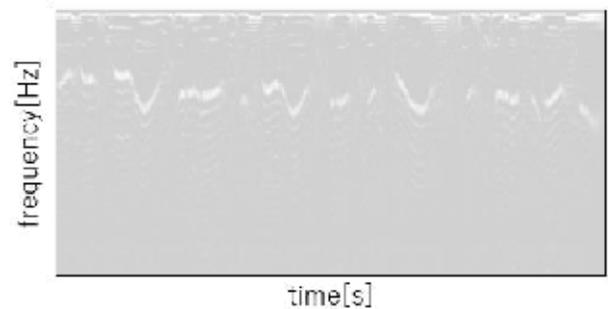


図 1. MFCC 画像

3. 回答

回答は問題データを学習用データ同様に中間画像化し、各読み札が含まれる確率を求め、上位 N 個の取り札を回答として出力する。

4. 開発環境

OS:Ubuntu 20.04, エディタ:Vim, 言語:Python3

使用ライブラリ:PyTorch, LibROSA

23 サボリ推奨委員会 Lv.7

新居浜

神田 凌佑 (5年) 青野 眞弥 (1年)
前田 隼 (1年) 占部 弘治 (教員)

1. 概要

このかるたシステムは、提示された問題の判定結果と試合全体を通じた回答状況の両方を利用して最尤な取り札の取り方を探索することを目標に構築しました。

開発プログラミング言語は Python、機械学習フレームワークは Tensorflow を使用しています。

2. システム構成

2.1. ニューラルネットによる判定

畳み込みニューラルネットワークを構築し、かるたの札の読み上げ音声を与えた機械学習したモデルを用いています。音声を図1のようなメルスペクトログラムの画像へと変換したものが入力で、それぞれの読み札の音声がどの程度の確度で含まれているかを出力としています。

2.2. 判定結果と問題の回答状況による取り札決定

44 種類の取り札の確度とこれまでの回答状況から、確

率論的にどの取り札を取ることが尤もらしいのかを決定して蓄積・回答します。また、ニューラルネットや通信などのエラーで回答状況が喪失しないように、回答状況をファイルに随時バックアップしています。

3. その他の施策

取り札決定とその処理に必要なデータ構造について単体テストを実施し、些末なバグを抑止しています。

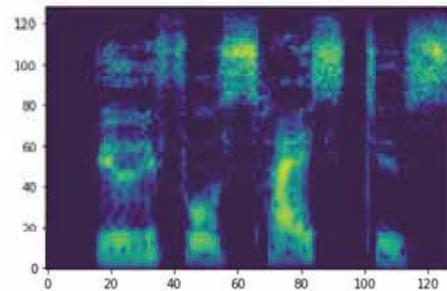


図1. 読みデータのメルスペクトログラム

24 八尺瓊蟹蒲

神戸市立

入本 聖也 (4年) 西ヶ峰 克隆 (4年)
山本 大地 (3年) 朝倉 義裕 (教員)

1. プログラムについて

我々、神戸市立の解答用プログラムはニューラルネットワークの一種である長・短期記憶 (Long short-term memory) を基軸としてサンプルデータを入力し、問題に解答するものである。実装は主に Python のライブラリの一種である Pytorch を使い、他のモジュールも必要に応じて活用することによって学習データの作成を行った。以下に学習時に入力したデータの一例を示す。

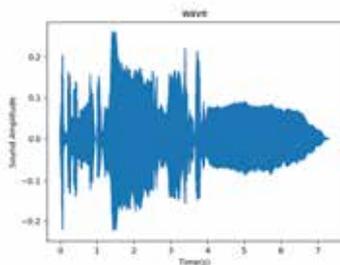


図1 入力データ例1

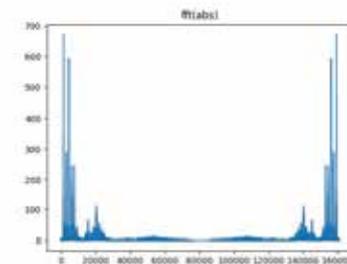


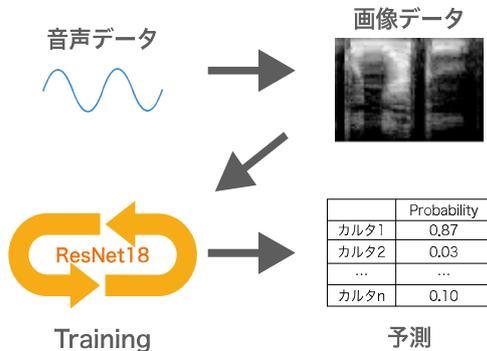
図2 入力データ例2

2. データの送受信について

プログラムの全体において trio を利用した非同期・イベント駆動の設計を採用しています。

これにより、例えば時間のかかるタスクを呼び出した時でも、アプリケーション本体はタスクの実行にかかる時間に関わらず動き続けることができ、ユーザーが操作しやすくなります。

1. はじめに



我々は、今回扱う重畳した同一話者による音声の識別タスクをDeep Learningを用いたソリューションで対応する。学習データの生成から予測までの流れを抽象的に上図に表した。

2. アルゴリズム

2.1 学習データの生成

我々は、音声データを画像データとして扱う。具体的には、メルスペクトログラムと呼ばれる、スペクトログラムからより人間の聴覚に近くなるように調整された手法を用いて、音声データから画像データに変換する。

2.2 モデルアーキテクチャ

音声を1つ重ねたものから20個重ねたものまで同時に学習を行うことで、幅広い特徴抽出を可能にする。具体的には、重畳された音声をメルスペクトログラムで2次元画像に変換し、マルチラベル画像分類タスクに変換する。アーキテクチャはResNet18を採用し、ImageNetで事前学習された重みを用いて今回のデータセットでファインチューニングする。

訓練する際には、マルチラベル画像分類の損失関数が重要になるので、損失を独自に開発し対処した。かるたの選択には、モデルの出力の上位のラベルを採用する。

1. はじめに

今回の競技は、通常とは異なる上毛かるたである。異なる点は、主に「音声の重複している点」、「札が途中から読み上げられる点」の2つである。

2. 戦略概要

分割データに含まれる読みデータの札と開始時間が分かっている場合、その時間だけずらして合成した音声は分割データと完全に一致することを前提とする。

私たちはこの前提のもと、2段階に戦略を分けた。

まず、すべての読み札に対してその札が分割データに含まれているとした場合の開始時間を推定する。

次に、開始時間を考慮した読みデータを合成させた音声と分割データが類似する札の組み合わせを探索する処理を経て、正解の札を割り出す。

私たちは、この戦略によって分割データ1つのみでの正

解を目指す。

3. 開始時間の推定について

このステップでは分割データと読みデータの相互相関を計測することで、読みデータの開始時間を複数個推定する。候補を複数にすることで、処理時間が増えるが、正確性を高めることができる。

4. 札の組み合わせ探索について

このステップでは「PBIL」を用いる。PBILとは各世代を確率モデルとして表現し、評価、更新、突然変異を繰り返して最適解を探索するアルゴリズムである。

実行時間を減らすために、並列化、評価の緩和を行った。

5. 開発環境

OS: Windows

エディタ: Anaconda Navigator, Visual Studio Code

使用言語: Python

27 WHITE SPACE

豊田

宇井 颯汰 (3年) 片野 蓮香 (3年)
鈴木 健太 (3年) 平野 学 (教員)

1. システム概要

問題の音声データに含まれている札を読みデータから推測するプログラムと、より解析に時間をかけられるようにするために問題データの受信や解答の送信をスムーズに行えるプログラムを作成した。

2. 解析アルゴリズム

音声の解析では分割データをすべて取得し、問題データに復元したものを使用して行う。

読みデータから問題データに重なっている分だけランダムに音声を選び、合成した推測データと問題データとの不一致度を計算する。問題データとの不一致度がより小さくなるように推測データを決めていき、そのデータに含まれている札を答えとして出力する。

3 台の PC を使用してプログラムを実行し、推測データと問題データとの不一致度が最も小さかったものを答え

とする。

3. 通信プログラムについて

Python3 の requests モジュールを用いて、競技用サーバーに GET リクエストと POST リクエストを送信し、必要な情報を受け取る。

また意図せぬ送受信を防ぐため、エラーが発生したらすぐにプログラムを終了するようにしている。

4. 開発環境

使用言語: C++, Python3

開発環境: Visual Studio Code

OS: Windows, MacOS, Ubuntu

28 チリツモ高専

都立
(品川)

藤川 純 (3年) 笹川 駿 (2年)
石井 一肇 (3年) 佐藤 喬 (教員)

1. はじめに

今回の問題では、かるたの読みがいくつか重ね合わせられた音声データを扱う。しかし、音声データは波形であり、波形のまま問題データに特定の読みデータが含まれているか判別することは難しい。そこで、フーリエ変換や統計学的方法を音声に対して適用することにした。

2. 周波数スペクトルを用いて音声を比較する方法

問題の分割データと読みデータから一部を切り出してフーリエ変換をし、周波数スペクトルを得る。2 つの周波数スペクトルの差分の総和と読みデータの周波数スペクトルの成分の総和の比を求める。その比を読みデータの切り出した区間での評価値とし、最も評価値が高くなる区間での評価値をその読みデータの評価値とした。

3. マルチチャネル時変ガウスモデルを用いて合成前の音声を推測する方法

マルチチャネル時変ガウスモデルを用いた音源分離は音源分離アルゴリズムの中でも、音声を取得するマイクロホンの本数より音源数が多くても使用できるという特徴がある。この方法により、問題の音声データを合成前の音声に分離する。このとき、評価値を分離後のデータと読みデータの類似度を評価値とした。

4. 選択札の決定方法

2 つの方法の内、より評価値が大きい方を解答とする。これに加え、GUI を用いて 2 つの方法によって求めた解答を、人間が確認できるようにした。

5. 開発環境

使用言語: Python, C++, JavaScript

環境: Visual Studio Code

主要な使用ライブラリ: NumPy, SciPy, Vue.js

29 アスノカルタ解析班

熊本
(熊本)

中村 春喜 (3年) 西村 淳志 (3年)
中村 颯太 (3年) 藤井 慶 (教員)

1. はじめに

本システムでは、「音声を重ね合わせてもスペクトログラム上では重なりにくい」という性質を利用し解析を行う。具体的には、与えられた音源をスペクトログラムに変換→事前に用意したサンプルデータのスペクトログラムと比較→判明した音声を減算というプロセスを繰り返していく。

2. スペクトログラムへの変換

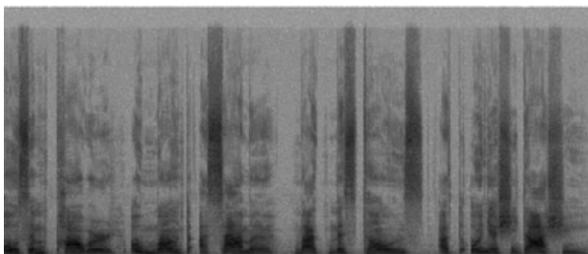


図1 スペクトログラムへ変換したデータ

離散フーリエ変換の原理を利用し、数値計算を行う。

3. 照合について

スペクトログラムに変換したデータを比較する際にデータの数値が取り得る値の幅が極めて大きいため、比較した際に値の誤差が認識されない場合が存在する。それらを考慮して、対数を用いて比較することで照合率の正確性を高めた。

4. 開発環境

開発言語: Python C/C++

ライブラリ: Numpy Opencv Scipy Matplotlib

OS: Windows

エディター: Visual Studio Code

30 みなかた

弓削商船

黒河 天晴 (3年) 甲賀 悠一郎 (3年)
CHALEUNKIT THANASINE (3年)
長尾 和彦 (教員)

1. 概要

システムは、問題データを読みデータと比べる部分と問題データから読みデータを除去する部分の2つからなる。それぞれのシステムは以下に示す通りである。

2. システムについて

2.1 読みデータとの比較

問題データをメルスペクトログラムに変換し、画像問題として処理する(図1)。SIFT アルゴリズムを改良したものをを用いて、最も問題データに含まれている確率の高い読みデータを選ぶ。

2.2 読みデータの除去

選んだ読みデータの逆位相を問題データと重ね合わせるにより問題データの除去を行う。またこの際、問題データ上での読みデータの正確な位置を探索する。

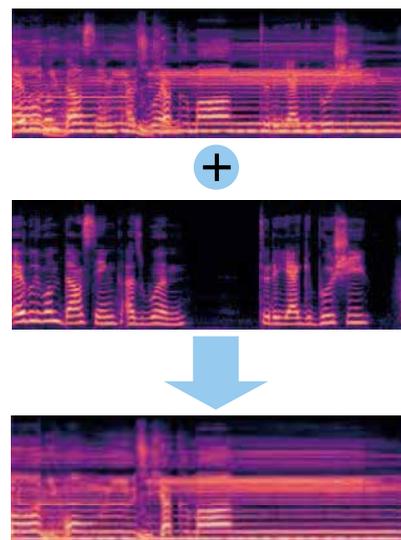


図1:メルスペクトログラム

3. 開発環境

言語: Python3

IDE: IntelliJ IDEA

1. はじめに

本大会の競技部門における最大の問題はその制限時間にある。力技で解こうとした場合、現在のCPUでは計算量が多く、制限時間内に解答を出力することは困難である。そこで我々は並列演算に特化して、なおかつ入手が容易なGPUを用いて本競技に挑むことにした。また、我々は本大会の競技部門において全チーム中最小の電力消費で回答を行うことを目指した。

2. 解法

まず問題の分割データと事前に配布された読みデータを用意し、分割データから読みデータを順に減算する。そして残った波形を積分することで得られた面積が小さいものを、重ねられているものとして判断する。

面積の最小順に並べられたあとは外れ値検出を用い、事前に公開されている重ね合わせ数に満たない場合は再度

分割データを取り寄せ、最初から処理を行う。

3. GUI

競技会場のネットワーク不調時用のファイル選択ダイアログを開いたり、現在の処理状態を表示したりするためにGUIを製作した(図1)。

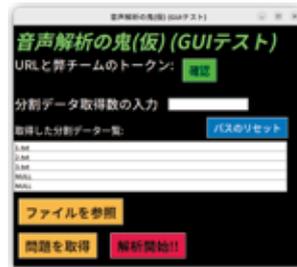


図1 GUI画面(開発時のもの)

4. 開発環境

言語: C++, Python, CUDA

使用デバイス: Jetson Nano

1. はじめに

今回作成したシステムについて、「特徴点作成」「特徴点利用」「特徴点照合」に分けて以下に示す。

このシステムを使うことによって特徴点作成のプロセスの手法を多数行い、それを組み合わせて予測結果の精度を高めることが出来る。

2. 特徴点作成

音声を扱いやすい形式に変形させる。音声をfftなどの変換手法を用いて変換する。その後、変換させた状態のピークを取って特徴点を作成する。

3. 特徴点利用

作成した特徴点を一定時間内の点を繋ぎ、星座のように結ぶことで次項における照合時の堅牢性を高める。また、

この処理を行った問題データをローカルに保存することによってここまでの処理を省略することができる。

4. 特徴点照合

結ばれた特徴点を問題の音声データ、分割データともに作成し、それらを照合する。この処理によって問題データごとの正解率を求める。

33 でじけんっ！

富山
(射水)

土橋 晴人 (2年) 南 理久 (4年)
佐々木 智大 (4年) 篠川 敏行 (教員)

1. はじめに

今回使用する音声解析のアルゴリズムは 3 つの要素で構成される。作業の順番に「元データの音声処理」「AI による読まれた文字の推測」「読み札の選択」の順で下記に説明する。

2. 元データの音声処理

データを扱いやすくするために音声データをサンプリングし、標本化する。また、標本化したデータは 20ms ごとにまとめて扱う。データをフーリエ変換し、それぞれの 50Hz が 20ms のオーディオ片にどれだけエネルギーを持つかを数値で表す。つまり、低域から高域までそれぞれの周波数帯がどれだけ重要か数値化する。

3. AI による読まれた文字の推測

ニューラルネットワークに 20ms ごとのデータを入力し、その区間にどの文字 (日本語なら、「あ」、「い」、英語なら、「A」、「I」など) が読まれたかを予測し、候補とその候補の一致率をだす。

4. 読み札の選択

読み札の全ての連続部分文字列に対して、その部分文字列と一致する、読まれたと推測される文字の連続した区間が存在するかを探索し、一致した読み札を選択する。

5. 開発環境

言語 : C++20, Python3.8.5

IDE : Visual Studio 2019, Visual Studio Code

OS : Windows10

34 三人寄れば聖徳太子

大島商船

嶋 諒大 (4年) 岩政 綾馬 (4年)
清水 蒼馬 (4年) 重本 昌也 (教員)

1. はじめに

重ね合わされた音声データにどの札が含まれているのかを推測するために、以下の複数のアルゴリズムを用いて問題音声を解析する。

2. 音声の解析アルゴリズム

2.1 波形画像を学習する方法

あらかじめ各読み札のスペクトログラム画像を作成し、ニューラルネットワークにより各読み札の分類器を作成する。問題音声が表示された際は同様にスペクトログラム画像を作成し、分類器と照合することで含まれている読み札を推測する。

2.2 音声を重ね合わせて推測する方法

問題音声の波形から頂点の位置や振幅などの情報を取得し、それぞれの札の音声と比較し、類似している読み札を A*アルゴリズムや機械学習によって推測する。

2.3 視覚的な表現

GUI により音声の波形を表示し、人の目視で特徴的な部分を見つけ、各札の特徴的な音声部分と照合する。

3. その他

複数のアルゴリズムにより複数の解を導き、より適切と思われる解を選択する。アルゴリズム間で解に矛盾が生じた場合は、あらかじめ各手法の精度に優先順位を設け、最適解を選択する。プログラムが導き出した解答が正確ではない可能性もあるので、最終的に人が聞いて確認も行う。複数の PC により並列処理を行い、高速化を目指す。

4. 開発環境

OS : Windows10

エディタ : Visual Studio 2019, Visual Studio Code

使用言語 : C++, Python など

1. はじめに

今回の競技では与えられた混合音源の分離と、分離された音源から取り札を特定するという 2 つのステップが必要となる。

以下にそれぞれのステップについて説明を示す。

2. 音源の分離

今回の制約では空間の情報やマイク配置などが与えられず、混合音声のみからそれぞれの音源信号に分離しなければ行けないため、音源分離の中でもブラインド音源分離(BSS)に分類される。

BSS にはいくつかの手法が存在するが、今回は互いに独立な音源の分離であるため、独立成分分析(ICA)を用いて音源の分離を目指す。

また、今回は短時間フーリエ変換(STFT)を施した音源に

対して時間周波数領域で ICA を行う、周波数領域 ICA (FDICA)を用いる。

3. 取り札の特定

取り札の特定については、分離された音源に対して文字起こし(STT)を行い、文字が少なくとも 2 文字以上ある場合には読み札の文章と文字の出現パターンの比較を行い分離音源から対応している取り札の特定を目指す。2 文字に満たない場合においては追加の分割データを取得し、音源の分離を再度行う。

4. 開発環境

言語 : Python3

ライブラリ : NumPy, SciPy

1. はじめに

今回の問題では事前にサンプルデータが渡され、どのサンプルデータの節が読まれたかがわかればよい。かるたの読み上げの音声認識、音声の分離は行わず、どの読みデータが合成された問題データなのかに注目して、札を特定する。

2. システム概要

音声データを 1 次元配列として扱う。合成された音声データはこの配列の加算によって作られたデータになっている。そこで各読みデータ数ごとに配列の要素の和が符号付き 16 ビット以内の大きさになるデータの組み合わせを事前に記録しておく。これはクリッピングを無視して考えるためである。そして分割データの配列と一致する札の組み合わせを、この事前に記録しておいた組み合わせと照合し探す。記録しておいた配列の中から検索するイメージ

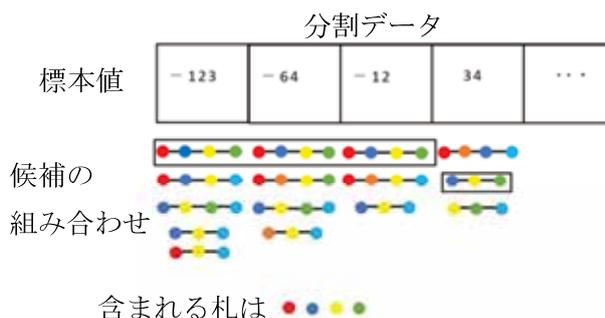
である。また分割データ内に読みデータ数全てのデータが含まれているとは限らないため、与えられた読みデータ数以下の組み合わせも照合の範囲に入れ、含まれる札を判定する。

3. 開発環境

言語 : C, Python

エディタ : Visual Studio Code

OS : Windows 10 Pro



39 とっぽ

鶴岡

庄司 昊平（2年）長谷川 智也（2年）
五十嵐 秀人（2年）金 帝演（教員）

1. はじめに

今回の競技では、事前に公開されている参照データと、本番に渡される分割データのメル周波数ケプストラム係数（Mel Frequency Cepstral Coefficient；MFCC）の解析を行い、MFCC を用いて分割データに含まれる参照データを導き出すプログラムを作成した。

2. MFCC (Mel Frequency Cepstral Coefficient)

声道特性を表す特徴量であるケプストラムの分析において、人間の音声知覚の特徴を考慮したものが MFCC と呼ばれる。ケプストラム分析よりも MFCC で解析した方が高精度かつ高速であることから本手法を用いる。

3. 分割データと参照データとの比較による音声認識

MFCC を用いて分割データと参照データのスペクトログ

ラムを算出し、それぞれのスペクトログラムを比較する。分割データに参照データが入っている場合、スペクトログラムを時間軸の伸縮を考慮しながら比較すると類似度が高い。しかし、分割データに参照データが入っていない場合は類似度が低い。このような手法を用いて音声認識を行う。

4. 開発環境

OS：Windows10

IDE：Visual Studio

音声解析に用いるライブラリ：

Librosa, Numpy

競技部門

40 チームサトウ

富山
(本郷)

田中 煌雅（3年）道上 裕貴（3年）
堀田 海智（3年）佐藤 圭祐（教員）

1. はじめに

私たちのチームでは、2つの方向から正解を求めることで、より正確性の高い回答を目指す。

2. アルゴリズム

2つの方法とは、1つは人間が見て考える方法。もう1つは機械が合成された音声をもととの音声に戻し、正解の札の候補を上げる方法。この2つを参照して回答をする。

2-1. 文字起こし

音源ファイルを何も加工しないまま文字に起こしてそれを人間が読み取り、大方の検討をつけ正解を予想する方法。

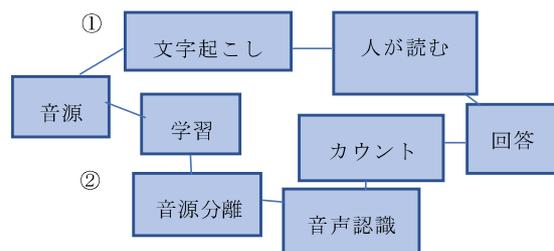
2-2. 音源分離

2つ目は、事前に合成されていない日本語、英語それぞれの読み札のファイルを学習させる。その時、

ファイルは開始位置をずらして1秒間ずつ分割して、1つのファイルを複数学習させる。そして、分離した音声の開始位置をずらしながら2秒間ずつに分けて、それぞれの時間ごとに合成されている音声の数だけ候補を出し、時間ごとの候補をカウントして、多いものから表示させる。

3. 開発環境

言語：python



図：アルゴリズム

41 長岡高専が通りま〜す

長岡

吉田 俊輝 (2年) 星 貞樹 (2年)
星野 咲空 (2年) 竹部 啓輔 (教員)

1. はじめに

今回の競技では、音声の解析として焼なまし法を用い、その精度を高めるために cnn (畳み込みニューラルネットワーク) を用いる。

2. cnn

自作した問題音声をメルスペクトログラム化し長さを統一したものを cnn の入力に使用し、各読み札の尤度を出力するようにする。また、学習の序盤では問題音声の長さが長く、個数が少ないものが作成されやすいようにした。

3. 読み札を選択するアルゴリズム

cnn によって出力された読み札を、尤度の高い順に選択し問題音声との類似度が高くなるように切り取る。未選択の読み札と選択している読み札を入れ替え、最終的に類似度が高くなるように読み札を選択する。少し結果が悪くなる場合でも読み札の変更を確率的に行うことで局所的最

適解を防ぐ。

4. 開発環境

OS : Windows10/macOS Big Sur

言語 : C++ / Python3

エディタ : VSCode/Atom

42 λ

久留米

田中 伶佳 (4年) 叶 弥悟 (3年)
二又 康輔 (3年) 田中 諒 (教員)

1. 概要

今回のシステムは主に音声を分離するフェーズと、分離した音声に読みの音声があるかないかを判定するフェーズの2つのパートから構成されている。以下にそれぞれの詳細を記す。

2. 音声の分離

音声の分離には非負値行列因子分解 (NMF) を用いる。まず、事前に公開されたすべての読みデータに通常の NMF を適用して、個別の教師基底を作成しておく。競技時に問題データを分離するときは、事前に分離した教師基底を固定の基底行列として用い、全教師あり NMF を適用してアクティベーション行列のみの最適化を行う。これにより、それぞれの教師基底に対応するアクティベーション行列を得ることができる。

3. 読みの音声のあるなし判定

分離後の音声にはノイズやエコーが乗っている。ノイズやエコーは読みの音声と同じような周波数帯であるので削除することが困難である。そこで事前に、畳み込みニューラルネットワーク (CNN) を用いて分離後のスペクトログラムを受け取り、読み音声のあるなし判定を行うモデルの学習を行う。競技時は、問題データを分離したあと、得られたすべてのスペクトログラムを判定モデルに渡し、読み音声のあるなし判定を行う。

4. 実験・開発環境

OS: Ubuntu, Windows11, MacOS

言語: Python3, C++

計算ノートブック: Jupyter lab

エディター: PyCharm, CLion, etc...

43 留学生パワー

阿南

アルカンドル イマワン（4年）
アンゼンチェン（4年）
エカテイコムキット ナッタキット（3年）
平山 基（教員）

1. はじめに

今回の競技では最初に読みデータを解析し、それらの解析結果を基盤として問題データを聞き分ける。そして、正しいと思われる絵札を選択する。

2. 読みデータの解析

与えられた読みデータを高速フーリエ変換(FFT)で解析し、各読みデータを周波数領域に変える。そして、短時間フーリエ変換(STFT)を利用して、時間に対しての周波数の分析結果を得る。

3. 問題データを聞き分ける

与えられた問題データをフーリエ変換して、周波数のデータを調べる。調べた周波数は行列で表すことができるため、線形計画法という手法でどのような読みデータが含まれているのかを調べることができる。

4. 絵札の選択

解析して聞き分けた問題データと読みデータを比べて、最も似ている読みデータの絵札を選択する。

5. 開発環境

OS: Windows 11, Windows 10

使用言語: C++

IDE: VisualStudio2019, VisualStudio2022

44 十字架天使

サレジオ

根津 嘉一（4年）可児 謙一（4年）
竹長 嵩（3年）須志田 隆道（教員）

1. はじめに

今回開発するシステムは、「上毛かるたの読み手音声のデータベースの構築」、「2通りの音声解析・分類」の2つの要素により構成される。以下にそれぞれの説明を記す。

2. データベースの構築

解析した音声データと解答となる絵札の照合を行うため、データベースを構築する。振幅・周波数などを用いた指標を作成し、音声データを特徴毎に分類する。

3. 機械学習を用いた音声解析

メルスペクトログラムや振幅などを特徴量として機械学習を行う。始まりの部分の特徴を解析することで、回答までの時間を短縮させる。始まりの部分だけでの分類が困難な場合、続きのデータを解析し、精度を向上させる。

4. 数値微分を用いた音声解析

個々の音声データと合成された音声データの数値微分を算出し、最大値と最小値を特徴量として合成データと比較し、特徴値周辺の情報を用いて合成されている個々の音声データを1つずつ抽出する。

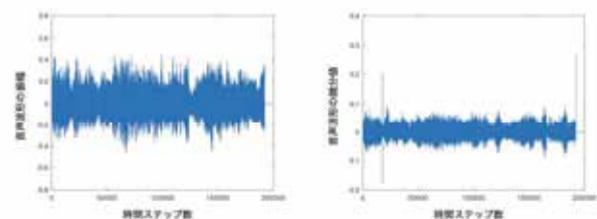


図1. 合成データの波形とその数値微分の波形

5. 開発環境

IDE: Xcode / 言語: Python 3, GNU Octave

45 MacOS

ハノイ
国家大学

Nguyễn Tất Đạt, Nguyễn Thái Bình,
Trần Mạnh Hiếu, Ma Thị Châu (教員)

Abstract– This guidebook includes needed information on our Karuta Card Game project and some basic instructions for use.

I. Introduction.

Karuta are [Japanese playing cards](#). In a regular Karuta game, the cards are read out loud one at a time. But in this competition, multiple cards will be read at the same time.

In this project, we create a predictive model for this problem. When experiencing the products we create, users can identify the Karuta cards using superposed audio data in each match.

II. System Requirement.

- Laptop: Dell Inspiron 7000
- Adapter
- USB Hub
- Lan Cable
- OS: Window 10

III. Operating Guide.

In each match, after receiving the problem audio, we conduct principal component analysis of each divide data that has been announced. The important features are used as inputs to the predictive model. Finally, we submit the results using interface.

46 Kosen元気ですか

タイ高専

Krittin Jeecomwan,
Saptawan Sombanguay,
Rawiphon Kiatdechawit,
土居 茂雄 (教員)

1. Introduction

Our program consists of two parts. First part is MFCC part, and the second part is Convolutional Neural Network part which is a part that prepare the deep learning module.

2. Parts of Programs

2.1 MFCC

From wav files input, the program will extract the data in Amplitude-Frequency domain and Amplitude-Time domain with Mel-frequency cepstral coefficients (MFCC).

3. User's part

The programming challenges us by using many speakers in a question and superimposed sounds. So, the code is developed to identify those speakers, then separate each speaker's sentence. The users can use this program by input the voice as a wav file. Since different sources in general dominate different time-

frequency bins, the speaker identifier that made by deep learning to identify and separate every speaker's content. After separating, the program shall go through the process of MFCC mentioned above. And then the program will compare the data obtained with the data in possession to consider the right cards to pick.

4. Roadmap and Improvement

This program can extract and organize sounds, however, there are still a lot of pitfalls as the factors of AI training and the knowledge that can be used to develop this program are not enough. If we practice the AI more, we can increase the accuracy of the program as well. The Methodology of the program in some stages is difficult to adjust to make it easier for users to be considered. Code organization is also another development guideline for ease of editing and improvement.