**２０２４年度　卒　業　論　文**

卒業論文題目

|  |
| --- |
| 音読教育における英文内容に沿ったコード進行を  ともなった英文歌生成手法 |

　指 導 教 員　　　　　　　鷹野　孝典　教授

神奈川工科大学　情報工学科

　　　学 籍 番 号　　　　　　　2121124

　　　学 生 氏 名　　　　　　　御木　流星

　提 出 日　2024年　12月　23日　　　印

　受 理 日　2024年　12月　23日　情報工学科長　　印

# **論文要旨**

　生成AI技術の発展により，歌詞にメロディを付けた楽曲生成が可能となり，英語教育において，英文にメロディを付与した英文歌を学習に活用することが注目されている．しかし，生成AIが作成した楽曲は，必ずしも歌詞の内容や雰囲気に沿ったメロディやコード進行であるとは限らない．また，英語教師が英語教育に利用するために既存の楽曲の歌詞を英語学習用の文法や英文に変更しても，既存の楽曲のメロディでは音読しやすいとは限らない．本研究では，このような課題を解決するために，Transformerモデルを用いた英文内容に沿ったコード進行を生成する手法を提案する．本研究の特徴は，英文内容に沿ったコード進行を生成することである．また，英語学習で生成されたコード進行を伴奏として，英文にコード進行を付けて歌うことや，メロディを付けて歌うことで，学生が楽しみながら音読できる．そのため，英文歌を英語学習の授業デザインなどに適用することができる．実験では，提案手法の実現可能性と教育的有用性を評価する．

目次

[**1. まえがき** 1](#_Toc185838406)

[1.1 はじめに 1](#_Toc185838407)

[1.2 本論文の構成 2](#_Toc185838408)

[**2. 関連研究** 3](#_Toc185838409)

[2.1 英語教育における歌の活用 3](#_Toc185838410)

[2.2 深層学習を用いた音楽の生成 3](#_Toc185838411)

[**3. 提案方式** 4](#_Toc185838412)

[3.1 提案手法の概要 4](#_Toc185838413)

[3.2 提案手法の手順 4](#_Toc185838414)

[3.3 コード進行に合わせて音読する利点 4](#_Toc185838415)

[3.4 歌詞とコード進行の学習方法 5](#_Toc185838416)

[3.4.1 データの定義 5](#_Toc185838417)

[3.4.2 歌詞のベクトル化 5](#_Toc185838418)

[3.4.3 コード進行のベクトル化 6](#_Toc185838419)

[3.5 コード進行に合わせた音読学習方法の概要 6](#_Toc185838420)

[**4. 実験システム** 7](#_Toc185838421)

[4.1 実験システムの概要 7](#_Toc185838422)

[4.2 実装環境 7](#_Toc185838423)

[4.2.1 環境の概要 7](#_Toc185838424)

[4.2.2 プログラミング言語 8](#_Toc185838425)

[4.3 Webクローリング 8](#_Toc185838426)

[4.4 ニューラルネットワークの構成 10](#_Toc185838427)

[4.5 Transformerモデルの学習 10](#_Toc185838428)

[4.6 コード進行の生成 11](#_Toc185838429)

[**5. 実験** 12](#_Toc185838430)

[5.1 実験目的 12](#_Toc185838431)

[5.2 実験環境 12](#_Toc185838432)

[5.3 実験1 13](#_Toc185838433)

[5.3.1 実験目的 13](#_Toc185838434)

[5.3.2 実験方法 13](#_Toc185838435)

[5.3.3 実験結果 14](#_Toc185838436)

[5.3.4 考察 15](#_Toc185838437)

[5.4 実験2 16](#_Toc185838438)

[5.4.1 実験目的 16](#_Toc185838439)

[5.4.2 実験方法 16](#_Toc185838440)

[5.4.3 実験結果 16](#_Toc185838441)

[5.4.4 考察 19](#_Toc185838442)

[5.5 実験3 19](#_Toc185838443)

[5.5.1 実験目的 19](#_Toc185838444)

[5.5.2 実験方法 19](#_Toc185838445)

[5.5.3 実験結果 20](#_Toc185838446)

[5.5.4 考察 22](#_Toc185838447)

[**6. むすび** 23](#_Toc185838448)

[謝辞 24](#_Toc185838449)

[**参考文献** 25](#_Toc185838450)

# **まえがき**

## はじめに

生成AI（Artificial Intelligence，人工知能）の技術的発展により，歌詞にメロディを付けた楽曲生成が実用的になっている．例えば，OpenAIが公開しているJukebox1)や，BandLab Technologiesが提供しているBandLab2)がある．これらのサービスでは，生成したい楽曲のジャンルや歌詞を入力することで，楽曲を生成できる．また，英語教育において，英文にメロディを付与した英文歌を学習に活用することが注目されている．更に，英語音読学習のための英文歌の生成において，英語教師が意図する発音や文法を学習する内容の英文にメロディを付与することで，学習者を動機づけることが期待できる3)．しかし，課題は二つある．一つ目は，生成AIが作成した楽曲は，必ずしも歌詞の内容や雰囲気に沿ったメロディやコード進行であるとは限らない点である．二つ目は，英語教師が英語教育に利用するために既存の楽曲の歌詞を英語学習用の文法や英文に変更しても，既存の楽曲のメロディでは音読しやすいとは限らない点である．

解決のアプローチとして，英文を入力としてコード進行を出力する深層学習モデルを構築し，既存の楽曲から収集した歌詞とコード進行のデータを学習に用いることで，一つ目の課題を解決する．また，出力されたコード進行を英文に付与することで，二つ目の課題の解決を試みる．

本研究では，Transformerを用いてモデルを構築し，既存の楽曲から収集した歌詞とコード進行のペアデータを入力として学習を行う．更に，英文を入力として，英文内容に沿ったコード進行を生成する手法を提案する．Transformerは，自然言語処理や画像生成など，さまざまな分野で活用されている深層学習モデルであり，Attention Mechanismを用いることで文中の離れた単語間の関連性を効率的に捉えることが可能である．Transformerを用いる理由は，長い文章などのデータが入力されても，記憶して最後まで処理が可能であり，歌詞の内容に沿ったコード進行が生成できると考えたからである．

提案手法の特徴として英文内容に沿ったコード進行を生成することができる．また，英語学習で生成されたコード進行を伴奏として，英文にコード進行を付けて歌うことや，メロディを付けて歌うことで，学生が楽しみながら音読をすることができ，結果的に音読回数や英語学習時間を増やすことにつながる．そのため，英語学習の授業デザインなどに適用することができる．

実験では，提案手法の実現可能性と教育的有用性を評価するために三つの実験を行う．一つ目の実験では, 269曲の歌詞とコード進行のデータをTransformerモデルで学習させる方法と，269曲の歌詞を短く分割し，その歌詞に対応しているコード進行を追加したデータでTransformerモデルを学習させる方法で生成されるコード進行の質の違いを比較する．この比較により，Transformerモデルに学習させるデータはどちらが良いか評価を行う．二つ目の実験では，提案手法で生成されたコード進行が英文の内容や雰囲気に沿っているかを3人で評価を行う．この評価を行うことで，提案手法が生成するコード進行が英文の内容や雰囲気に沿っているかわかる．三つ目の実験では，提案手法により生成されたコード進行を音読に活用した場合の効果を検証する．そのため，五つの英語のクラスをクラスごとにグループ1とグループ2の二つのグループに分けて音読を行う．グループ1はコード進行ありで音読を行い，グループ2はコード進行なしで音読を行う．音読を行った後に，音読した英文の空欄を完成させる形式の30点満点の確認テストとアンケート調査を行い，確認テストの平均点とアンケートの結果をもとに，コード進行がある英文の音読効果を評価する．更に，このモデルを授業で活用することで英語学習の効果を高められると考え，本学の英語教師の協力を得て，英語授業への導入および実践方法についても検討を行う．

## 本論文の構成

本論文は全6章で構成されている．本章では，研究の背景や動機，目的について述べる．第2章では，本研究と関連する研究や技術について述べる．第3章では，提案手法について述べる．第4章では，実験システムの実装について述べる．第5章では，提案手法を評価するために実施した実験についての結論と考察を述べる．第6章では結論と今後の展望について述べる．

# **関連研究**

　本章では，提案手法に関する研究について述べる．なお，英語教育における歌の活用を2.1節，深層学習を用いた音楽の生成を2.2節に示す．

## 英語教育における歌の活用

歌を活用した英語教育では，学習者の英語力を向上させるために，生成AIを用いて英文にメロディを付与した英文歌を取り入れた英語授業デザインが提案された．英文歌を聞いて音読練習を行った場合とメロディのない発話音声を聞いて音読練習を行った場合の音読効果を把握するために比較実験が行われている．比較実験では，100語程度からなる通常英文と40語程度からなる要約英文を用いた音読練習と，通常英文および要約英文の空欄を完成させる形式のテストが実施された．結果として， 通常英文と要約英文の両方で，発話音声を聞いた場合よりも英文歌を聞いて音読練習を行った場合のテストの点数が，それぞれ平均で2.55点，2.20点上昇しており，英文歌を用いた音読練習に一定の学習効果が得られることが確認された3)．また，英語授業に歌の動画を組み込んだ教育実践が行われていた．この実践を取り入れてから授業の出席率や課題提出率が上がった．履修者対象のアンケートでは，68人中56人が知覚動詞や使役動詞の文法理解に動画が役に立ったと回答しており，歌を教材とすることで思考力の育成に貢献しうることが明らかになっていた4)．

## 深層学習を用いた音楽の生成

　深層学習を用いて，メロディ，コード，リズムのそれぞれの関係性を考慮して音楽を生成するモデルが提案された．最初にリズム生成器によってコードとメロディのリズムを予測し，その後，コード生成器がリズムの情報を基にコード進行を生成する．最後に，メロディのリズムと生成されたコードを用いてメロディ生成器がメロディを生成した．このモデルにより，メロディ，コード，リズムを考慮した調和の取れた音楽を生成できた5)．

　コード進行の自動生成を行う研究では，深層学習を用いて，歌詞を考慮したコード進行の生成を試みた．具体的には，Seq2Seqを用いて歌詞を入力して，コード進行を出力するモデルを構築し，コード進行を生成していた．データはWeb上からスクレイピングして収集した歌詞とコード進行のペアを用いていた．また，収集したデータを学習と評価のデータに分け，評価データの予測結果におけるコードの再現率，編集距離から評価を行っていた．結果としては，再現率平均は0.49，編集距離平均は2.92となり，正解のコードに対して，評価データから無造作に抽出したコードを予測結果とするランダム評価よりも再現率平均，編集距離で高い精度となった6)．

# **提案方式**

本章では，本研究で提案する手法についての概要と手順を述べる．

## 提案手法の概要

　提案手法では，既存の楽曲の歌詞とコード進行のペアデータを入力として，Transformerモデルを構築する．このTransformerモデルは歌詞を入力として，その歌詞に適したコード進行を出力するように学習させる．更に，英語教師が作成した英文を入力として，英文内容に沿ったコード進行を生成する．この際，歌詞とコード進行の対応関係を学習し，歌詞の文脈や雰囲気に基づいて最適なコード進行を生成できるようにする． 提案方法の概要図を図 3.1に示す．

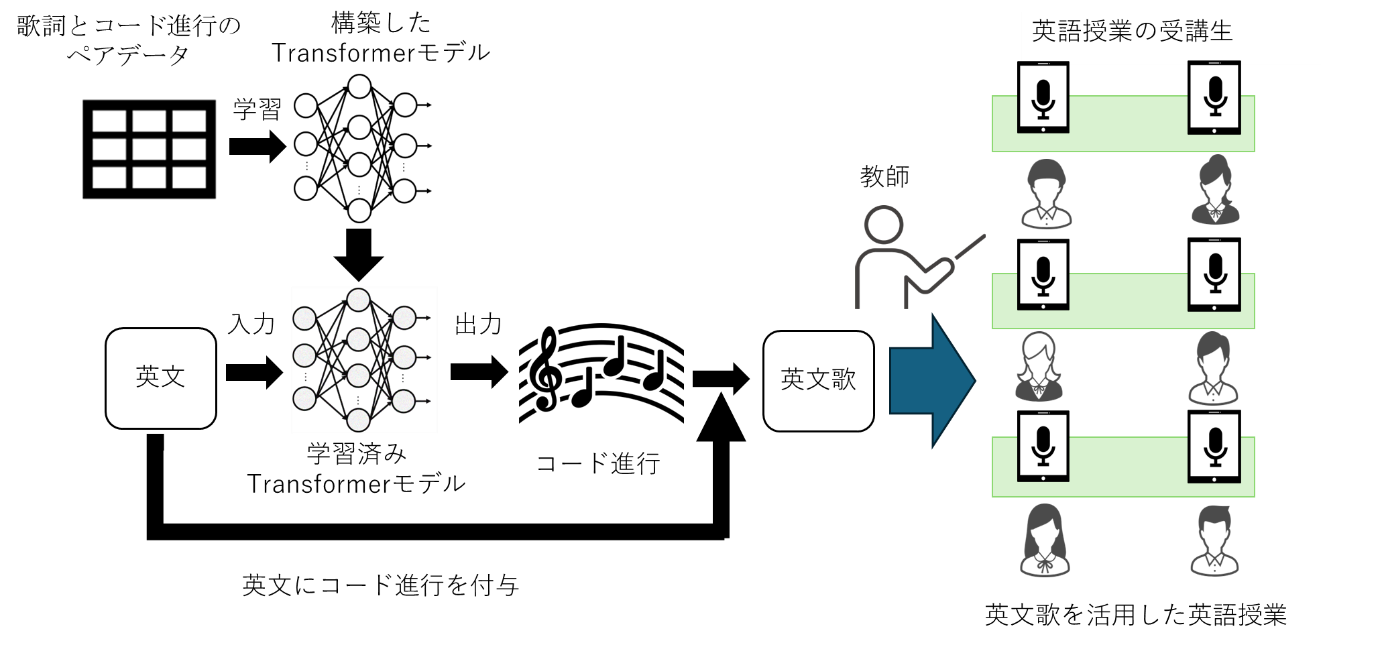


図 3.1 提案手法の概要図

## 提案手法の手順

　手順としては以下のような流れになる．

1. 歌詞データとコード進行データのペアを対象として，それらをモデルの学習に適したデータ構造へと前処理する．歌詞データとコード進行データのペアは，Webサイトで公開されているものを収集して作成する．データ構造を表 3.1に，データ構造の例を表 3.2に示す．
2. 歌詞を入力としてコード進行を出力するTransformerモデルを構築する．
3. STEP-1で前処理したデータを用いて，Transformerモデルを学習させる．
4. 英文をTransformerモデルに入力し，英文の内容に沿ったコード進行を生成する．

## コード進行に合わせて音読する利点

楽曲のメロディに正確に音程を合わせて歌うことは，音楽に馴染みがある人でないと難しい．しかし，コード進行には大まかなメロディが含まれているため，音楽に馴染みがない人でもメロディに乗りながら音読を行える利点があると考えられる．

## 歌詞とコード進行の学習方法

### データの定義

1番目の歌詞をlyrics1，1番目のコード進行をchord1とする．1～n番目までの歌詞とコード進行があった場合，データの構造は表 3.1のように定義する．

表 3.1　データの構造

|  |
| --- |
| [(lyrics1，chord1)，(lyrics2，chord2)，… ，(lyricsn，chordn) ] |

chordは先頭に[start]，終わりに[end]をつける．これはコード進行の開始位置と終了位置をモデルに正確に認識させるためである．また，データの例を表 3.2に示す．

表 3.2　データの構造の例

|  |
| --- |
| ("Never mind, I'll find someone like you", '[start] A E F#m [end]')  ("And you're married now", '[start] F#m D [end]')  ("Come up to meet you, tell you I'm sorry, you don't know how lovely you are", '[start] Bm7 G Gadd9 D Dsus2 [end]')  ('Wait for me to come home', '[start] G D [end]') |

### 歌詞のベクトル化

Transformerのpositional encodingとembeddingにより，歌詞の単語ごとに入力ベクトルに変換する．embedding行列は，単語数×ベクトル次元数の行列である．歌詞のベクトル化の流れを図 3.2に示す．

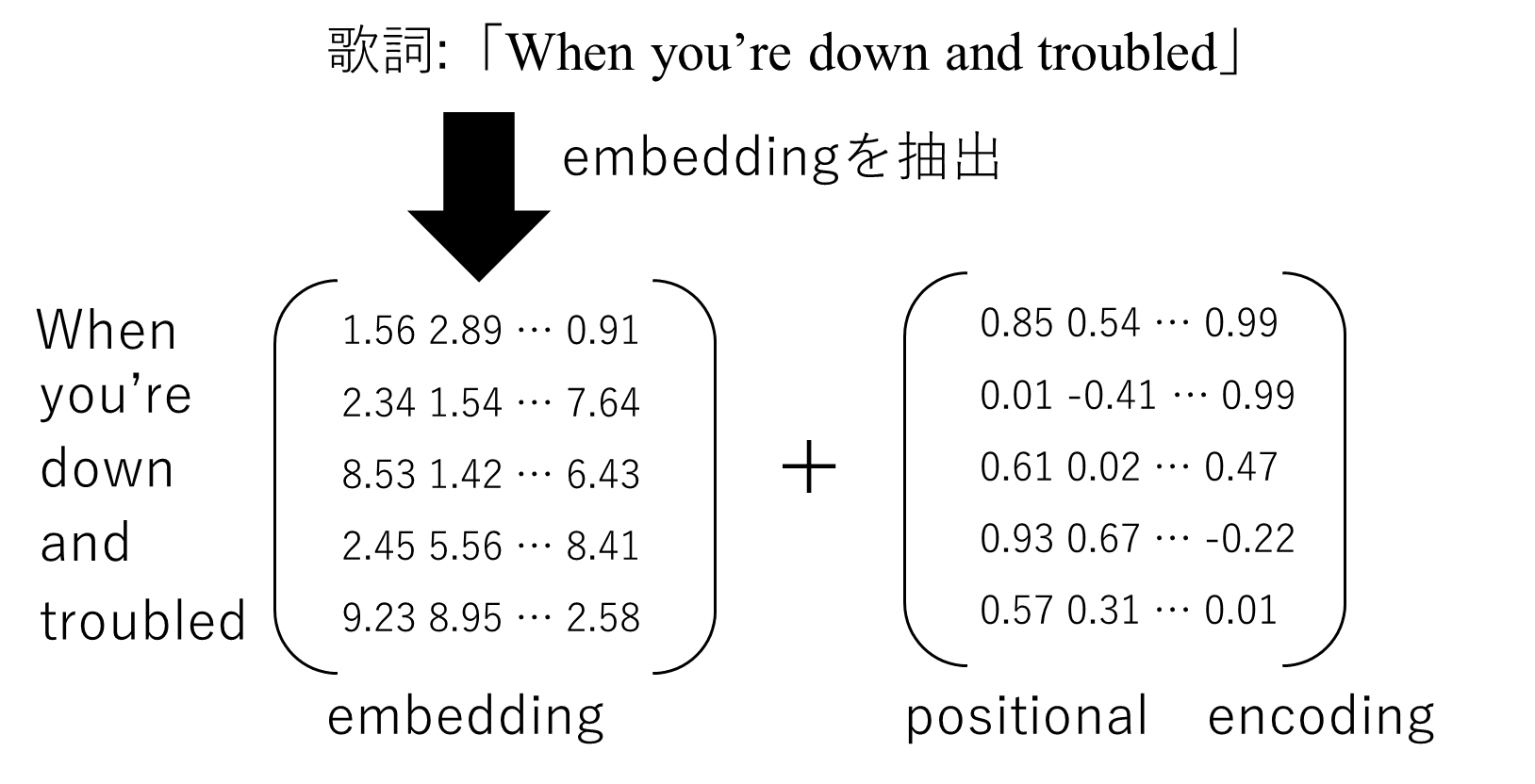


図 3.2　歌詞のベクトル化の流れ

### コード進行のベクトル化

Transformerのpositional encodingとembeddingにより，コード進行のコードごとに入力ベクトルに変換する．embedding行列は，コード数×ベクトル次元数の行列である．コード進行のベクトル化の流れを図 3.3に示す．

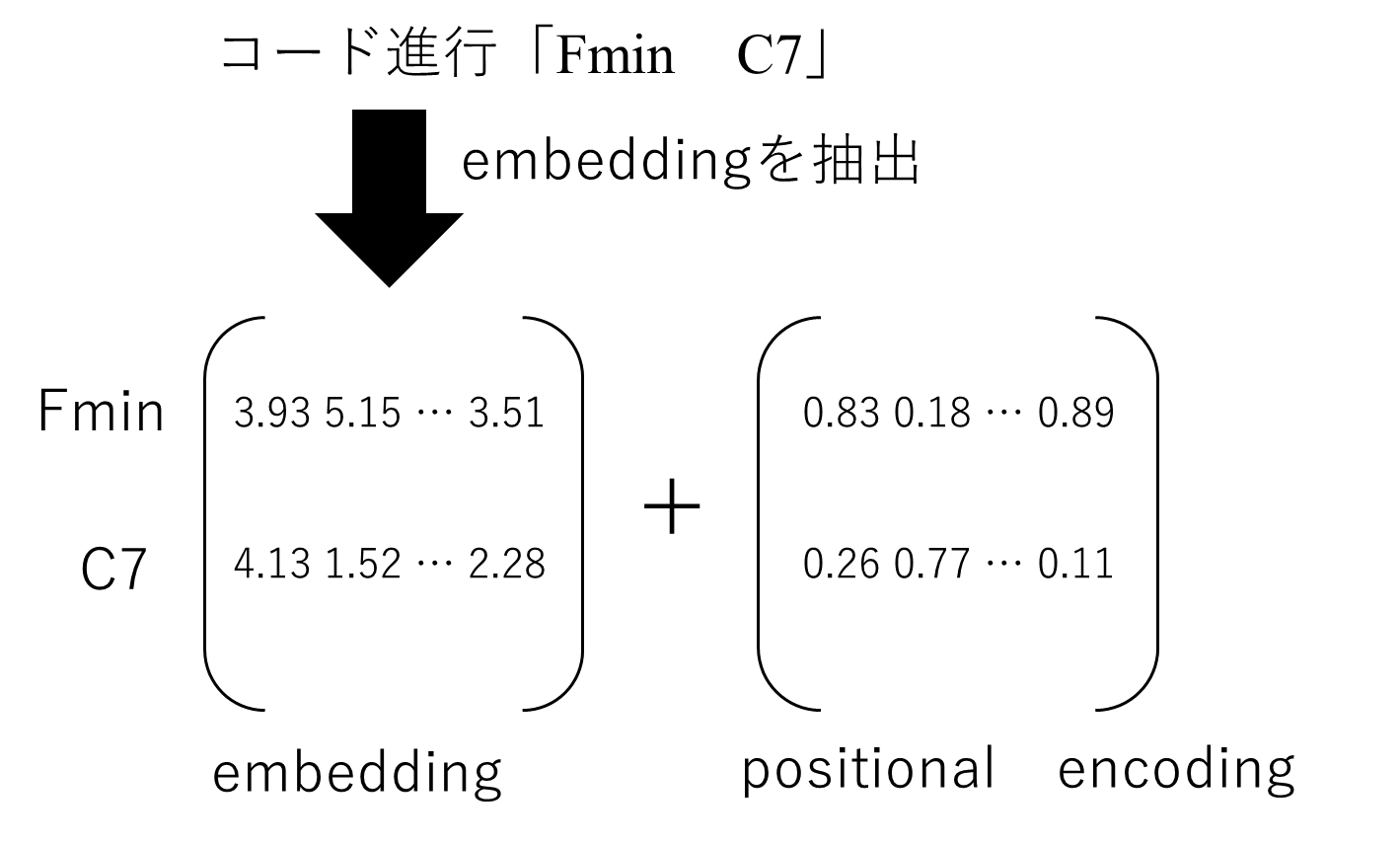


図 3.3　コード進行のベクトル化の流れ

## コード進行に合わせた音読学習方法の概要

本節では，英語授業におけるコード進行に合わせた音読学習方法の概要を記述する．具体的には，まず教師が英文全体の内容を説明することで，学生にこれから読む英文のテーマや目的を理解させる．その後，学生はリズムを取りながら音読を行う．英語教師は英語の発音だけでなく，英文全体のリズムやアクセントを考慮してリズムを作成する．更に，生成されたコード進行の伴奏をかけて，リズムに合わせて音読を行う．このときTransformerモデルで生成したコードを利用する．次に，教師がリズムに合わせて英文を読み上げた後，学生がそれに続き，単語と単語の間で発音が連結する現象であるリエゾンなど英語の自然な発音を意識して音読を行う．最後に，伴奏をかけながら学生がリズムを取りつつ音読を行う．

# **実験システム**

本章では，実験システムの実装について述べる．

## 実験システムの概要

　実験システムでは，Webクローリングによる歌詞とコード進行の収集，歌詞を入力としてコード進行を出力するTransformerモデルの構築，入力した英文に基づくコード進行の生成をPythonで実装した．

## 実装環境

### 環境の概要

PC環境を表 4.1に，プログラミング言語などの開発，実験環境を表 4.2に，実装に使用したPythonライブラリおよびモジュールを表 4.3に示す．

表 4.1　PC環境

|  |  |
| --- | --- |
| 名称 | ハードウェア・OS |
| PC | CPU: 11th Gen Intel(R) Core(TM) i7-1165G7  RAM: 8GB  GPU: Intel(R) Iris(R)  Xe Graphics |
| OS | Windows 11 |

表 4.2　開発・実験環境

|  |  |  |
| --- | --- | --- |
| 名称 | バージョン | 利用用途 |
| Python | 3.10.12 | プログラミング言語 |
| Google Colaboratory | - | 実験環境 |

表 4.3　Pythonライブラリ

|  |  |  |
| --- | --- | --- |
| 名称 | バージョン | 利用用途 |
| numpy | 1.55.1 | 数値計算や配列操作 |
| pandas | 3.8.4 | データの読み込み，加工，および操作 |
| tensorflow | 2.17.1 | Transformerモデルの実装および学習 |
| keras | 3.5.0 | Transformerモデルの構築およびレイヤーの実装 |
| pathlib | 1.0.1 | ファイル操作やパス管理に使用 |
| playwright | 1.49.1 | Webクローリングおよびブラウザ操作 |
| nest\_asyncio | 1.6.0 | Google Colaboratory環境での非同期イベントループの管理 |
| asyncio | - | 非同期処理の実行 |
| random | - | ランダムなデータ生成に使用 |
| string | - | 文字列操作に使用 |
| re | - | 正規表現を用いた文字列のパターンマッチングに使用 |

### プログラミング言語

本研究の実装では，Pythonを利用した．Pythonは他の言語と比較してコーディングがシンプルであるという点や，Transformerモデルの実装に必要なライブラリが充実しているといった利点から使用している．Pythonの実行環境として，Google Colaboratoryを使用している．

## Webクローリング

　Webクローリングを用いて，Webサイトで公開されている既存の楽曲の歌詞とコード進行のデータを収集する．WebクローリングはPythonを用いて行うことが可能である．テキストファイルに保存してあるURLのリストからクローリングを行い，収集したデータはcsv形式で保存する．保存する際に，csvのstatus列に歌詞とコード進行の両方が収集できた場合は1，収集できなかった場合は0を入れる．Webクローリングを行うプログラムのソースコードを表 4.4に示す．また，収集したデータを保存するプログラムのソースコードを表 4.5に示す．

表 4.4　Webクローリングのプログラム

|  |
| --- |
| import nest\_asyncio  import asyncio  from playwright.async\_api import async\_playwright  import re  # URLからWebクローリングする関数  async def fetch\_lyrics\_and\_chords(url):  try:  async with async\_playwright() as p:  # ヘッドレスモードでChromiumブラウザを起動  browser = await p.chromium.launch(headless=True)  page = await browser.new\_page()  # タイムアウトを設定して指定されたURLにアクセスする  await page.goto(url, timeout=90000)  # ユーザーエージェントをセットする  await page.set\_extra\_http\_headers({  'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,  like Gecko) Chrome/58.0.3029.110 Safari/537.36'  })  # preタグ内のすべてのテキストを取得  pre\_tag = await page.query\_selector('pre')  full\_text = await pre\_tag.inner\_text()  await browser.close()  # コード部分と歌詞部分を分離  chords = re.findall(  r'\b[A-G][#b]?(?:maj|min|dim|aug|sus|add|M7|m|m7|7|6|9|11|13)'  　　　　　r'?(?:/[A-G]?(?:[#b]?(?:maj|min|dim|aug|sus|add|M7|m|m7|7|6|9|11|13))?)?\b',  full\_text)  chords = [chord for chord in chords if not re.match(r'Capo|\d+', chord)]  # コード部分を取り除いた歌詞を作成  lyrics = full\_text  lyrics = re.sub(r'CHORDS:.\*?(?=\n\n|\Z)', '', lyrics, flags=re.DOTALL)  for chord in chords:  lyrics = re.sub(r'\b' + re.escape(chord) + r'\b', '', lyrics)  # ギタータブ譜や不要な部分を削除  lyrics = re.sub(r'^[eBGDAE]\|.\*$', '', lyrics, flags=re.MULTILINE)              lyrics = re.sub(r'.\*?(?=Intro:)', '', lyrics, flags=re.DOTALL)              lyrics = re.sub(r'.\*?(?=\[Intro\])', '', lyrics, flags=re.DOTALL)              lyrics = re.sub(r'\[Outro.\*?\]', '', lyrics)              lyrics = re.sub(r'Outro.\*', '', lyrics)              lyrics = re.sub(r'\[.\*?\]', '', lyrics)              lyrics = re.sub(r'(Intro|Verse \d+|Interlude|Pre-chorus|Chorus \d+|Outro):', '', lyrics)              lyrics = re.sub(r'.\*?(?=h)', '', lyrics, flags=re.DOTALL)              lyrics = re.sub(r'.\*?(?=p)', '', lyrics, flags=re.DOTALL)              lyrics = re.sub(r'.\*?(?=s)', '', lyrics, flags=re.DOTALL)              lyrics = re.sub(r'.\*?(?=\)', '', lyrics, flags=re.DOTALL)              lyrics = re.sub(r'\(.\*?\)', '', lyrics)              lyrics = re.sub(r'\d+-\d+(?:-\d+)?', '', lyrics)              lyrics = lyrics.replace('-', '')              lyrics = re.sub(r'x\d+½?', '', lyrics)              lyrics = re.sub(r'.\*?\.', '', lyrics)              lyrics = re.sub(r'\bX\b', '', lyrics)              lyrics = re.sub(r'\bN\.C\.\b', '', lyrics)              lyrics = re.sub(r'[\/|]', '', lyrics)              lyrics = re.sub(r'#', '', lyrics)              lyrics = re.sub(r'%', '', lyrics)              lyrics = re.sub(r'=', '', lyrics)              lyrics = re.sub(r'\\*', '', lyrics)              lyrics = re.sub(r'\d+', '', lyrics)              lyrics = re.sub(r'\s+', ' ', lyrics).strip()  # 歌詞とコードのどちらも取得できるか確認  status = 1 if lyrics and chords else 0  return lyrics, ' '.join(chords), status  except Exception as e:  print(f"Error fetching {url}: {e}")  return None, None, 0 |

表 4.5　収集したデータを保存するプログラム

|  |
| --- |
| for url in urls[start\_index:end\_index]:  lyrics, chords, status = await fetch\_lyrics\_and\_chords(url)  # 取得したデータをCSVに追加  with open(output\_file, mode='a', newline='', encoding='utf-8') as file:  writer = csv.writer(file)  writer.writerow([url, lyrics, chords, status])  # ステータスの確認  if status == 1:  print(f"Data for {url} saved successfully.")  else:  print(f"Failed to fetch data for {url}") |

## ニューラルネットワークの構成

　エンコーダ・デコーダモデルとしてTransformerを適用し，構築する．Transformerモデルの構造概要を表 4.6に示す．また，モデルの構造を表示してコンパイルするプログラムのソースコードを表 4.7に示す．

表 4.6　Transformerモデルの構造概要

|  |  |  |  |
| --- | --- | --- | --- |
| Layer (type) | Output Shape | Param # | Connected to |
| encoder\_inputs  (InputLayer) | (None, None) | 0 | - |
| positional\_embedding  (PositionalEmbedding) | (None, None, 256) | 3,845,120 | encoder\_inputs[0][0] |
| decoder\_inputs  (InputLayer) | (None, None) | 0 | - |
| transformer\_encoder  (TransformerEncoder) | (None, None, 256) | 3,155,456 | positional\_embedding[… |
| functional\_3 (Functional) | (None, None, 15000) | 12,959,640 | decoder\_inputs[0][0],  transformer\_encoder[0… |

表 4.7　モデルの構造を表示してコンパイルするプログラム

|  |
| --- |
| #エポック数  epochs = 40  transformer.summary()  transformer.compile(      "rmsprop", loss="sparse\_categorical\_crossentropy", metrics=["accuracy"]  ) |

## Transformerモデルの学習

　本研究では，Webクローリングにより収集した歌詞とコード進行のペアデータを学習データとして使用し，Transformerモデルに学習させる．ペアデータは表 3.2のように，開始トークンと終了トークンをコード進行に付与することで，Transformerモデルが開始と終了を認識できるようにしておく．モデルを学習するためにfitメソッドを使用し，train\_dsにトレーニングデータセットを入れる．また，epochs でエポックの数を指定し，validation\_data には検証データセットを入れる．Transformerモデルの学習を行うプログラムのソースコードを表 4.8に示す．

表 4.8　Transformerモデルの学習を行うプログラム

|  |
| --- |
| transformer.fit(train\_ds, epochs=epochs, validation\_data=val\_ds) |

## コード進行の生成

　学習したTransformerモデルを用いて，入力した英文に対するコード進行の生成を行う．file\_path に入力する英文のファイルを指定する．output\_path は生成したコード進行を保存するファイルを指定する．decode\_sequence関数でコード進行の生成を行う．また，eng\_vectorizationで英語の入力文をトークン化して，モデルが処理可能なベクトル形式に変換を行う．入力した英文に対するコード進行を生成するプログラムのソースコードを表 4.9に示す．

表 4.9　入力した英文に対するコード進行を生成するプログラム

|  |
| --- |
| max\_decoded\_sentence\_length = 4#出力するコードの最大の数  def decode\_sequence(input\_sentence):  tokenized\_input\_sentence = eng\_vectorization([input\_sentence])  decoded\_sentence = "[start]"  for i in range(max\_decoded\_sentence\_length):  tokenized\_target\_sentence = chords\_vectorization([decoded\_sentence])[:, :-1]  predictions = transformer([tokenized\_input\_sentence, tokenized\_target\_sentence])  # ops.argmax(predictions[0, i, :]) is not a concrete value for jax here  sampled\_token\_index = ops.convert\_to\_numpy(  ops.argmax(predictions[0, i, :])  ).item(0)  sampled\_token = chords\_index\_lookup[sampled\_token\_index]  decoded\_sentence += " " + sampled\_token  if sampled\_token == "[end]":  break  return decoded\_sentence.replace("[start] ", "").replace(" [end]", "")  test\_eng\_texts = [pair[0] for pair in test\_pairs]  file\_path = '/content/drive/My Drive/English.csv'  data = pd.read\_csv(file\_path)  output\_path = '/content/drive/My Drive/English\_with\_Chords.csv'  i=0  for \_ in range(100):  input\_sentence = (data['Lyrics'][i])  translated = decode\_sequence(input\_sentence)  print((input\_sentence))  print((translated))  data.at[i, 'Chords'] = translated  data.to\_csv(output\_path, index=False)  i=i+1 |

# **実験**

本章では，実験目的と実験結果，考察について述べる．

## 実験目的

　提案手法の実現可能性と教育的有用性を評価するために，三つの実験を実施した．Transformerモデルに学習させる歌詞の長さで生成されるコード進行の質を比較することで，どちらの方法が良いかを評価する．また，提案手法によって生成されたコード進行が，入力された英文の内容や雰囲気に沿っているかを評価する．更に，提案手法により生成されたコード進行を音読に活用した場合の効果を検証するため，五つの英語のクラスをクラスごとに二つのグループに分けて音読を行う．音読を行った後に，確認テストとアンケート調査を行い，結果を確認することで，音読効果を評価する．

## 実験環境

　実験で用いた269曲の歌詞とコード進行のデータを**エラー! 参照元が見つかりません。**に示す．**エラー! 参照元が見つかりません。**はlyrics列の一つのセルに1曲の全ての歌詞が入っており，Chords列の一つのセルには1曲の全てのコード進行が入っている．更に，**エラー! 参照元が見つかりません。**のデータに269曲の歌詞を短くわけたものと歌詞に対応しているコード進行を加えたデータを表5.2に示す．なお実験で用いた歌詞とコード進行のデータはUltimate GuitarというWebサイトからクローリングしてきたものである10)．

表 5.1　269曲の歌詞とコード進行のデータ

|  |  |  |
| --- | --- | --- |
| ID | Lyrics | Chords |
| 0 | I found a love for me Darling, just dive right … | G G Em C D G Em C D G Em C G D G Em C D Em C G D … |
| 1 | I heard there was a secret chord That David played ... | C Am C Am F G C G C F G Am F G E7 Am F … |
| 2 | Wise men say, only fools rush in But I can't … | C G Am F C G C C Em Am F C G F G Am F C G C C .. |
| … |  |  |
| 266 | Come up to meet you, tell you I'm sorry You do ... | Bm7 G D Dsus2 A D D6 A E Asus4 A Dmaj7 Dsus2 C... |
| 267 | It's a little bit funny, this feeling inside I ... | A Eb D G A F#m Bm Em7 Bm A Bm G A C D G A G D ... |
| 268 | Karma Police, arrest this man He talks in math ... | Fadd9 Fadd9 F Am D F Em G Am Fadd9 Em G Am D G.. |

表 5.2　269曲の歌詞を短くわけたものとその歌詞に対応しているコード進行を加えたデータ

|  |  |  |
| --- | --- | --- |
| ID | Lyrics | Chords |
| 0 | I found a love for me Darling, just dive right … | G G Em C D G Em C D G Em C G D G Em C D Em C G D … |
| 1 | I heard there was a secret chord That David played ... | C Am C Am F G C G C F G Am F G E7 Am F … |
| 2 | Wise men say, only fools rush in But I can't … | C G Am F C G C C Em Am F C G F G Am F C G C C … |
| … |  |  |
| 268 | Karma Police, arrest this man He talks in math ... | Fadd9 Fadd9 F Am D F Em G Am Fadd9 Em G Am D G.. |
| 269 | I found a love for me | G G Em |
| 270 | Darling, just dive right in, and follow my lead | Em C C D |
| 271 | Well, I found a girl beautiful and sweet | D D G Em |
| … |  |  |
| 892 | Calling out father, prepare as we will | Am7 Em G D C |
| 893 | Desolation comes upon the sky | C Am7 G/B C |
| 894 | Now I see fire, inside the mountain | Am7 G/B C |

## 実験1

### 実験目的

Transformerモデルに学習させる歌詞の長さで生成されるコード進行の質を比較することで，どちらの方法が良いかを評価する．コード進行に変化がある場合はコード進行の質が良い，コード進行に変化がない場合はコード進行の質が悪いとする．

### 実験方法

　実験では方法1と方法2の二つの方法で，コード進行を生成する．方法1は歌詞が長いデータが含まれている**エラー! 参照元が見つかりません。**をTransformerモデルに学習させてコード進行を生成する．また，方法2は歌詞が短いデータが含まれている表5.2をTransformerモデルで学習させてコード進行を生成する．入力として10～20単語からなる100個の英文を用いて，コード進行を生成し，コード進行の質が良かった曲の数をかぞえて二つの生成方法を比較する．入力に用いた英文を表 5.3に示す．

表 5.3　入力に用いた英文

|  |  |
| --- | --- |
| ID | Lyrics |
| 1 | I caught a cold because I kept turning the air conditioning on. |
| 2 | I did not watch the video completely. |
| 3 | I have a sore throat after discussing passionately the topic. |
| 4 | A smartphone does not function when the touch ID is not put correctly. |
| 5 | You have to put in the password every time. |
| … |  |
| 96 | I can't imagine life without a smartphone. |
| 97 | It took a lot of work to prepare presentation assignments in class. |
| 98 | When I showed him my research outline, he gave me much advice. |
| 99 | My classmate's presentation on that theme inspired me. |
| 100 | I have the ambition to become a scientist who works globally. |

### 実験結果

方法1で入力した英文と生成したコード進行を表 5.4に示し，方法2で入力した英文と生成したコード進行を表 5.5に示す．また，コード進行の質が良かった例，コード進行の質が悪かった例を表 5.6，表 5.7に示す．

表 5.4　方法1を用いて入力した英文と生成したコード進行

|  |  |  |
| --- | --- | --- |
| ID | Lyrics | Chords |
| 1 | I caught a cold because I kept turning the air conditioning on. | cm cm cm g g g g g g g g g g g g g g g g cadd9 |
| 2 | I did not watch the video completely. | cm cm cm cm cm |
| 3 | I have a sore throat after discussing passionately the topic. | cm cm cm cm g g g g g g g g g g g g g g g c |
| 4 | A smartphone does not function when the touch ID is not put correctly. | g g g c g g g g g g g g g g g g cadd9 cadd9 cadd9 c |
| 5 | You have to put in the password every time. | cm cm cm cm cm cm |
| … |  |  |
| 97 | It took a lot of work to prepare presentation assignments in class. | cm cm cm cm cm cm cm cm cm |
| 98 | When I showed him my research outline, he gave me much advice. | cm cm cm cm cm cm cm cm |
| 99 | My classmate's presentation on that theme inspired me. | cm cm |
| 100 | I have the ambition to become a scientist who works globally. | cm cm cm cm |

表 5.5　方法2を用いて入力した英文と生成したコード進行

|  |  |  |
| --- | --- | --- |
| ID | Lyrics | Chords |
| 1 | I caught a cold because I kept turning the air conditioning on. | em/b bm ab a7sus4 |
| 2 | I did not watch the video completely. | em/b e7/g e7 d |
| 3 | I have a sore throat after discussing passionately the topic. | em/b bm g7 dm |
| 4 | A smartphone does not function when the touch ID is not put correctly. | em/b a7sus4 a7sus4 d |
| 5 | You have to put in the password every time. | em/b em gadd9 |
| … |  |  |
| 97 | It took a lot of work to prepare presentation assignments in class. | em/b bm ab b |
| 98 | When I showed him my research outline, he gave me much advice. | g cadd9 b7 am |
| 99 | My classmate's presentation on that theme inspired me. | em/b cadd9 cadd9 em7 |
| 100 | I have the ambition to become a scientist who works globally. | em/b d am7 em |

表 5.6　コード進行の質が良い例

|  |  |
| --- | --- |
| Lyrics | Chords |
| I could have done better in the final examination of the subject. | em/b cadd9 b7 e |
| Can you introduce your friend to me? | c f e e9/d |
| Although creating game characters was time-consuming, I found it to be fun. | em/b d am7 asus4 |

表 5.7　コード進行の質が悪い例

|  |  |
| --- | --- |
| Lyrics | Chords |
| Now that I'm a junior, I must consider getting a job. | cm cm cm cm cm cm |
| Can you introduce your friend to me? | am7 am7 am7 am7 am7 am7 am7 am7 am7 am7 am7 am7 am7 am7 am7 am7 am7 am7 am7 am7 |
| His approach to the research is very informative. | cm cm cm cm cm |

### 考察

方法1と方法2の生成されるコード進行を比較してみた結果，方法1は100曲中41曲でコード進行の質が良かった．また，方法2では100曲中98曲でコード進行の質が良かった．方法1よりも方法2の方がコード進行の質が良い曲の数が多いことから，方法2でコード進行を生成する方が良いことがわかる．

そのため，今後の実験および英語の授業では方法2を用いる．

## 実験2

### 実験目的

提案手法により学習させたTransformerモデルによって生成されたコード進行が，英文の内容や雰囲気に沿っているかを確認する．

### 実験方法

　実験では，表5.5のIDが1～30までの30個の英文を使用して，各英文に対して生成されたコード進行が英文の内容や雰囲気に沿っているかを3人で評価する．評価を行う際にコード進行をオーディオで聴くことができるアプリのガレージバンドを使用する．評価する際に，内容や雰囲気に沿っていたら〇，一部雰囲気に沿っていたら△，内容や雰囲気に沿っていなかったら✕をつける．また，〇を1点，△を0.5点，✕を0点として計算を行い，各英文に対するコード進行の雰囲気を3点満点で評価する．なお，それぞれ独立して評価を行い，評価を行う他の人の意見や判断に影響されないように配慮した．具体的には，評価は異なる時間帯で実施し，評価を行う人同士が結果を共有する機会を設けないようにした．

### 実験結果

入力の英文と生成されたコード進行を表 5.8に示す．また，コード進行が英文の内容や雰囲気に沿っているか3人で評価した結果を表 5.9に示す．

表 5.8　入力の英文と生成されたコード進行

|  |  |  |
| --- | --- | --- |
| ID | 英文 | コード進行 |
| 1 | I caught a cold because I kept turning the air conditioning on. | em/b bm ab a7sus4 |
| 2 | I did not watch the video completely. | em/b e7/g e7 d |
| 3 | I have a sore throat after discussing passionately about the topic. | em/b bm g7 dm |
| 4 | A smartphone does not function when the touch ID is not put correctly. | em/b a7sus4 a7sus4 d |
| 5 | You have to put in the password every time. | em/b em gadd9 |
| … |  |  |
| 27 | If you have no idea to solve this math problem, you'd better go to the tutors to ask a question | c f dsus2 cadd9 |
| 28 | He is too busy with club activities to study. | a7 eb f cm7 |
| 29 | Have you finished making the presentation slides for the class? | c f d7 d |
| 30 | Could you check my presentation slides? I want to get your advice. | c a am e9/d |

表 5.9　各英文に対するコード進行の雰囲気の評価

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 一人目 | 二人目 | 三人目 | 〇と評価した数 | 雰囲気の点数 |
| 1 | 〇 | 〇 | 〇 | 3 | 3 |
| 2 | 〇 | 〇 | 〇 | 3 | 3 |
| 3 | 〇 | 〇 | △ | 2 | 2.5 |
| 4 | △ | 〇 | 〇 | 2 | 2.5 |
| 5 | ✕ | 〇 | △ | 1 | 1.5 |
| 6 | 〇 | 〇 | 〇 | 3 | 3 |
| 7 | 〇 | 〇 | △ | 2 | 2.5 |
| 8 | ✕ | ✕ | 〇 | 1 | 1 |
| 9 | 〇 | 〇 | ✕ | 2 | 2 |
| 10 | △ | 〇 | △ | 1 | 2 |
| 11 | 〇 | 〇 | 〇 | 3 | 3 |
| 12 | ✕ | ✕ | △ | 0 | 0.5 |
| 13 | 〇 | 〇 | ✕ | 2 | 2 |
| 14 | △ | ✕ | △ | 0 | 1 |
| 15 | ✕ | ✕ | △ | 0 | 0.5 |
| 16 | 〇 | ✕ | 〇 | 2 | 2 |
| 17 | 〇 | 〇 | 〇 | 3 | 3 |
| 18 | ✕ | ✕ | 〇 | 1 | 1 |
| 19 | 〇 | 〇 | ✕ | 2 | 2 |
| 20 | 〇 | 〇 | 〇 | 3 | 3 |
| 21 | ✕ | 〇 | △ | 1 | 1.5 |
| 22 | 〇 | 〇 | 〇 | 3 | 3 |
| 23 | 〇 | 〇 | 〇 | 3 | 3 |
| 24 | ✕ | 〇 | 〇 | 2 | 2 |
| 25 | △ | 〇 | 〇 | 2 | 2.5 |
| 26 | 〇 | 〇 | △ | 2 | 2.5 |
| 27 | △ | 〇 | △ | 1 | 2 |
| 28 | △ | ✕ | 〇 | 1 | 1.5 |
| 29 | △ | ✕ | △ | 0 | 1 |
| 30 | 〇 | 〇 | 〇 | 3 | 3 |
| 〇と評価した数の平均 | 1.8 | | | | |
| 雰囲気の平均点数 | 2.1 | | | | |

### 考察

　実験結果の表 5.9から，30個中9個のコード進行が英文の雰囲気に沿っていると全員が評価していた．また，英文ごとにコード進行が英文の雰囲気に沿っていると評価した数の平均は1.8となり，雰囲気の平均点数は2.1点となった．更に，30個中21個のコード進行で3人の評価が分かれていた．このことから，提案手法により生成されたコード進行は，英文の内容や雰囲気に沿っていないと考えられる．原因としては，学習データ数が少ないこと，評価する人ごとに雰囲気の解釈が異なった可能性が考えられる．

## 実験3

### 実験目的

英文にコード進行による伴奏を付けることで，英文に楽曲としてのリズムが付与される．コード進行に合わせて音読した際に，メロディがなくても英語の音読学習に効果があるかを確認テストの平均点とアンケートの結果をもとに評価する．

### 実験方法

　実験では，表 5.5のIDが1～9までの9個の英文とそのコード進行を使用して，五つの英語クラスをクラスごとに二つのグループに分けて音読を行った．音読を行った後に，音読を行った英文の空欄を完成させる形式の確認テストを行った．また，コード進行に合わせて音読を行ったグループでアンケートを実施した．確認テストの内容と確認テストの回答の英文を表 5.10と表 5.11に示す．各グループの実験の手順としては以下のような流れになる．

コード進行ありで音読をするグループ(3クラス)

1. リズムを取りながら英文を読み，英文の内容を説明する．
2. 学生ひとりひとりに，教員がリズムをつけて英語を読み，学生にその場でリピートさせる．学生が英文をそのまま読んでいたら，リズムに合わせて音読するよう矯正を行う．
3. その後，クラス全体でリズムつき英文の音読を行う．各英文10回ずつ音読する．
4. 空欄を完成させる形式の確認テストを行う．

コード進行なしで音読をするグループ(2クラス)

1. 通常のように英文を読み，英文の内容を説明する．
2. 学生ひとりひとりに向けて，教員が正しい発音で英語を読み，学生にその場でリピートさせる．学生が間違った読み方をしていたら，その場で正しい読み方を指導する．
3. その後，クラス全体で英文の音読を行う．各英文10回ずつ音読する．
4. 空欄を完成させる形式の確認テストを行う．

表 5.10　確認テストの内容

|  |
| --- |
| 適切な語を選んで空欄を埋め、英文を完成させなさい。（30点） |
| (ア)I \_\_\_\_\_\_\_\_\_\_ a cold \_\_\_\_\_\_\_\_\_\_ I \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_the air conditioning on. |
| (イ)I did not \_\_\_\_\_\_\_\_\_\_ the video \_\_\_\_\_\_\_\_\_\_. |
| (ウ)I have a \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ after \_\_\_\_\_\_\_\_\_\_ passionately \_\_\_\_\_\_\_\_\_\_ the topic. |
| (エ)A smartphone \_\_\_\_\_\_\_\_\_\_ not \_\_\_\_\_\_\_\_\_\_ when the touch ID is not \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_. |
| (オ)You have to \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ the password \_\_\_\_\_\_\_\_\_\_ time. |
| (カ)I was \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ having club \_\_\_\_\_\_\_\_\_\_ after school. |
| (キ)I'd \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ some good \_\_\_\_\_\_\_\_\_\_ tonight. |
| (ク）I \_\_\_\_\_\_\_\_\_\_, so I \_\_\_\_\_\_\_\_\_\_ the first \_\_\_\_\_\_\_\_\_\_. |
| (ケ)\_\_\_\_\_\_\_\_\_\_ you come and \_\_\_\_\_\_\_\_\_\_ our club \_\_\_\_\_\_\_\_\_\_? |

表 5.11　確認テストの回答

|  |
| --- |
| 適切な語を選んで空欄を埋め、英文を完成させなさい。（30点） |
| (ア)I caught a cold because I kept turning the air conditioning on. |
| (イ)I did not watch the video completely. |
| (ウ)I have a sore throat after discussing passionately about the topic. |
| (エ)A smartphone does not function when the touch ID is not put correctly. |
| (オ)You have to put in the password every time. |
| (カ)I was exhausted after having club practice after school. |
| (キ)I'd like to have some good ramen tonight. |
| (ク）I overslept, so I missed the first period. |
| (ケ)Would you come and join our club festival? |

### 実験結果

コード進行ありで音読をするグループで行った確認テストの得点結果を表 5.12に示す．また，実験後にコード進行ありで音読をするグループで実施したアンケート項目を表 5.13に示し，アンケート項目の質問1の結果を図 5.1に示す．

表 5.12　確認テストの得点結果

|  |  |  |
| --- | --- | --- |
|  | 人数 | 平均点 |
| クラス1 | 37 | 16.30 |
| クラス2 | 19 | 10.40 |
| クラス3 | 19 | 18.16 |
| 3クラスの平均点 | 15.28 | |

表 5.13　アンケート項目

|  |
| --- |
| 質問1：生成AIが作成した曲の雰囲気は、英文のイメージとあっていましたか。   * 1. とても合っていた、まったく違和感がなかった   2. まあまあ合っていた、ほとんど違和感がなかった   3. 普通   4. あまり合っていなかった、少し違和感があった   5. 全然合っていなかった、すごく違和感があった   6. わからない   質問2：生成AIの楽曲を使ったリズム音読は楽しかったですか。   * 1. すごく楽しかった   2. 楽しかった   3. 普通   4. つまらなかった   5. すごくつまらなかった   6. わからない   質問3：生成AIの楽曲を使った英文は、音読しやすかったですか。   1. すごく音読しやすかった 2. まあまあ音読しやすかった 3. 普通 4. 音読しにくかった 5. すごく音読しにくかった 6. わからない   質問4：生成AIの楽曲を使った音読は、英語学習に効果があると思いましたか。   * 1. とても効果がある   2. まあまあ効果がある   3. 普通   4. あまり効果がない   5. 全く効果がない   6. わからない |

|  |
| --- |
|  |

図 5.1　アンケート項目の質問1の結果

### 考察

実験結果の図 5.1のアンケートの結果から，クラス1およびクラス3では「まあまあ合っていた、ほとんど違和感がなかった」と回答した割合が最も高かった．しかし，クラス2では「あまり合っていなかった、少し違和感があった」と回答した割合が27%で最も高かったが，「まあまあ合っていた、ほとんど違和感がなかった」と回答した割合は18%で，「普通」および「全然合っていなかった，すごく違和感があった」と並び，2番目に高い割合を示していた．更に，図 5.1の 3クラスの合計割合を見ると，「まあまあ合っていた、ほとんど違和感がなかった」と回答した割合が最も高かった．これらの結果から，コード進行が英文の雰囲気にある程度合っていると考えられる．また，**エラー! 参照元が見つかりません。**と図 5.1の結果から，3クラスの平均点よりも平均点が高かったクラス1とクラス3はアンケートで「まあまあ合っていた、ほとんど違和感がなかった」と回答した割合が一番多かった．そのため，クラス1とクラス3は英文にコード進行が付与された音読に意欲的に取り組めた可能性がある．

# **むすび**

生成AIが作成した楽曲は，必ずしも歌詞の内容や雰囲気に沿ったメロディやコード進行であるとは限らない．また，英語教師が英語教育に利用するために既存の楽曲の歌詞を英語学習用の文法や英文に変更しても，既存の楽曲のメロディでは音読しやすいとは限らない．本研究では，このような課題を解決するために，Transformerモデルを用いた英文内容に沿ったコード進行を生成する手法を提案した．実験では，1曲の歌詞のデータだけでなく，歌詞が短いデータもTransformerモデルに学習させた方法が，生成されるコード進行の質が良くなることを確認した．しかし，全員が英文の雰囲気に沿っていると評価したのは30個中9個のコード進行であった．そのため，提案手法により生成されたコード進行は，英文の内容や雰囲気に沿っていないと考えられる．原因としては，学習データ数が少ないこと，評価する人ごとに雰囲気の解釈が異なった可能性が考えられる．

今後の展望としては，本実験では，Transformerモデルの学習データとして歌詞とそのコード進行を用いたが，コード進行の個々のコードの音の長さやリエゾンを意識して区切った英文とそのコード進行も学習データに加えることが挙げられる．これにより，コード進行だけでなく個々のコードの音の長さや音読の区切りが付与され，音読学習においてより効果的な英文歌を生成できると考えられる．更に，生成したコード進行と音調情報を音声合成技術と統合することで，メロディの生成や音声出力をともなった教材も生成できると考えられる．これらの研究により，英語教育で英文歌を用いた音読効果の向上が期待される．

# 謝辞

本研究を行うにあたり，多くのご指導ご鞭撻をくださった鷹野孝典教授に心から感謝いたします．

英語科河野先生には，本研究を行うにあたり，適切なご助言を賜りました．ここに深謝の意を表します．また，研究を共に進めながら様々な意見を出し合い，本研究の方向性の確立にご協力くださった鷹野研究室の４年生と大学院生の先輩に深く感謝いたします．

# **参考文献**

1. OpenAI，<https://openai.com/index/jukebox/>，(2024/12/17)．
2. BandLab，<https://www.bandlab.com/songstarter/>，(2024/12/17)．
3. 鷹野孝典，河野智子：楽曲生成AIの活用による英文歌を取り入れた英語授業デザイン，教育システム情報学会2023年度第5回研究会予稿集，p.24-30 (2024.01)．
4. 河野智子：サイバースペースでの外国語教育における歌の動画の活用，ITを活用した教育シンポジウム講演論文集，Vol. 15，p.2(2020)．
5. 松村昂輝，大谷紀子，木村司，福井健一，沼尾正行：リズム・コード・メロディの三段階深層学習による音楽生成，研究報告音楽情報科学(MUS)，2019-MUS-125，No.5，(2019.11)
6. 世良拓也，大井雄介，浜野祐介，谷口浩平：深層学習による歌詞を考慮したコード進行の自動生成，第84回全国大会講演論文集，p.99-100 (2022.02)．
7. Ultimate Guitar，<https://www.ultimate-guitar.com/>，(2024/12/20)．