

世代間ギャップの解消を目的とした楽曲再生システムの試作

稲野 友哉^{1,a)} 北原 鉄朗^{1,b)}

概要：

本稿では、年齢の離れた人同士が音楽を通じて世代間ギャップを埋めるシステムを提案する。世代間ギャップを埋めるためには、「コミュニケーションの円滑化」、「価値観の共有」の2点が重要とされているが、これらをITで支援する研究は行われていない。そのため、本研究では、お互いの世代で流行したポピュラー音楽等を聴かせ合うシステムを通じて、世代間ギャップの解消を目指している。2人で構成された同年代の2つのグループ計3組で実験を実施した結果、「相手グループとの会話は取りやすくなったか」、「相手グループの音楽に対して印象が変わったか」というアンケートでは、7段階評価中で平均5以上という有望な結果が得られた。

1. はじめに

近年、世代の離れた人とのコミュニケーションにおいて、世代間ギャップによる世代差が社会問題となっている。世代間ギャップの定義として年齢の離れた世代間で生じる嗜好、価値観、文化などの相違としており、その一例として「話の話題が合わない」、「価値観が合わない」、「言葉が通じない」等がある。そのため、世代間ギャップをどのように埋めたら良いかわからない経験をした人が多いと考えられる。世代間ギャップが生じる原因として、生まれ育った環境による嗜好、考え方の違い、離れた世代に対する先入観等があり、職場等では、世代間ギャップによりコミュニケーションが阻害され、業務を進行する上でマイナスになることが多い。

近年、SNSの発達等から若年層において対面コミュニケーションを苦手とする人が増加している等の影響から、複数人での会話を支援するために、話題の提供や参加者に会話のきっかけを与えるような研究が多くされてきた [1][2][3]。しかし、そのほとんどはユーザの年齢に関して言及されおらず世代間ギャップ、年齢差があるユーザを考慮しているものではない [1][2]。梅田らの研究では高齢者と子供間での、遠隔コミュニケーションにおける会話支援と、世代間ギャップの解消を扱っている。しかし、お互いの価値観の共有、話題の提供を用いて世代間ギャップを解消するような手法は、扱っていない [3]。

また複数人が楽曲を聴取するために楽曲の推薦をするよ

うな研究も過去に行われているが、楽曲を聴くことに限定されている [4][5]。鈴木らの研究では、複数人の聴取履歴から全員の嗜好が考慮されたプレイリストを作成することのみである [5]。

本稿では、音楽を通じて世代間ギャップの解消を目指すシステムを提案する。[6][7]より世代間ギャップを解消するためには「お互いの嗜好を共有し認め合い、理解する」、「コミュニケーションを円滑化する」の2点が有効とされている。ユーザがお互いの世代で流行した歌手の楽曲を聴取することで、価値観を認め合い、提供される歌手に関する話題と質問により、コミュニケーションを円滑化させることで、世代間ギャップの解消につながると考えられる。

2. 提案システム

本研究で提案するのは、世代の離れた人同士が音楽を通じて世代間ギャップを埋めるアプリである。本研究では次のような状況を想定する。

- 数人で構成された同年代のグループが2つあり、グループ間で年齢が10歳以上離れている。
- グループ間で面識のある人が少ない。

また世代間ギャップの解消には [6][7] から、

- (1) 「コミュニケーションの円滑化」
- (2) 「価値観の共有」

が重要であるとわかった。

コミュニケーションに関しては、年齢差があるため、グループ間で共有している体験が異なっており、共通する話題が少ないため、どうコミュニケーション取ればよいのか困難と考えられる [8]。従って、(1)を実現するために、音

¹ 日本大学文理学部

^{a)} inano@kthrlab.jp

^{b)} kitahara@chs.nihon-u.ac.jp

声合成を用いて各グループ間でのコミュニケーションを促進させるような問いかけ、またお互いのグループが10~20代の頃に流行した歌手間(以下グループの世代で流行した歌手)の関連性を収集し、それらの情報から話題の提供を行う。

価値観に関しては、近年、社会的な豊さと情報化が実現したことにより、大きな環境の変化が起こり、多様な価値観が存在しているため、上の世代と若い世代で、ギャップが生じている [7]。一方、音楽は世代を超えて人気のある娯楽である。日本国内で最もよく聴かれている音楽は、いわゆる J-POP と呼ばれているジャンルである [11]。J-POP は世代ごとに人気の曲があり、時代を追うごとに曲の特徴は変化している。しかし、急激に変化するわけではないため、聴ききっかけがあれば、世代を超えて共通の曲を好きになる可能性があると考えられる。

従って、(2)を実現するために、システムを操作するユーザには、相手グループの世代で流行した歌手の曲を聴いてもらう。

以上の議論により、本研究のシステムは図1に示すように各グループが交互に用いる仕様とする。以下に、操作の流れを記す。

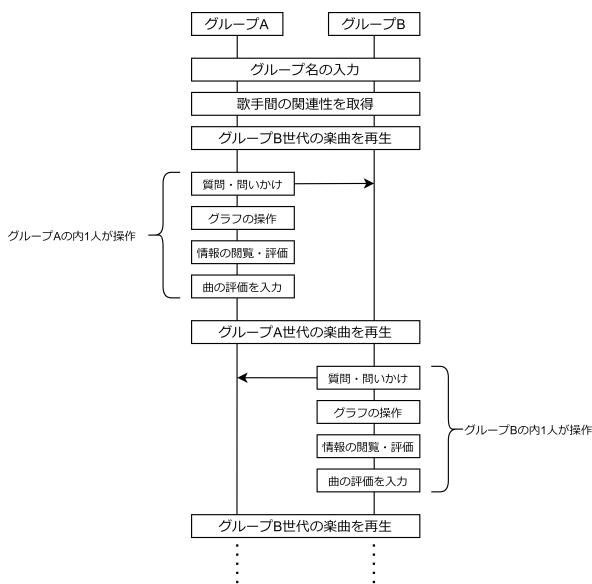


図1 システムの概要図

- (1) 事前にホストとなるユーザが、各ユーザに対して、年齢、名前を聞いておく。またシステムが前処理として、お互いのグループの世代で流行した歌手、及びそれらの歌手の情報と人気曲を取得してあるものとする。
- (2) システムの操作は各グループ1人ずつ交互に行い、システム操作者(以下操作者)の名前を画面上に表示する(図2)。



図2 操作者を画面上に表示(画面右上の赤枠部)

- (3) 操作者が操作を始めると、相手グループの歌手の曲が流れ、画面上に流れている歌手と、その歌手に関連している歌手がエッジでつながれた状態で、グラフ上に表示される(図3)。再生する歌手の順番として、他の歌手とのつながりが多い歌手順で流していく。

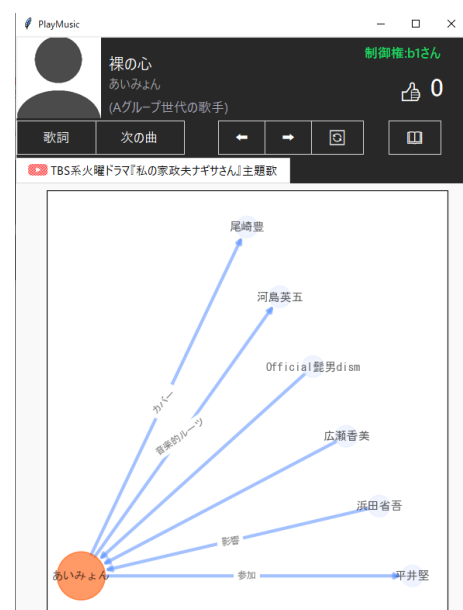


図3 歌手間の関連性を示したグラフ

- (4) 表示されている歌手をクリックすると、その歌手と繋がっている歌手をボタンで表示し(図4)、クリックすると関連性の詳細情報を見ることができる。操作者は表示された情報に対してグッドボタンを押し(最大10回)、情報の評価を行う(図5)。

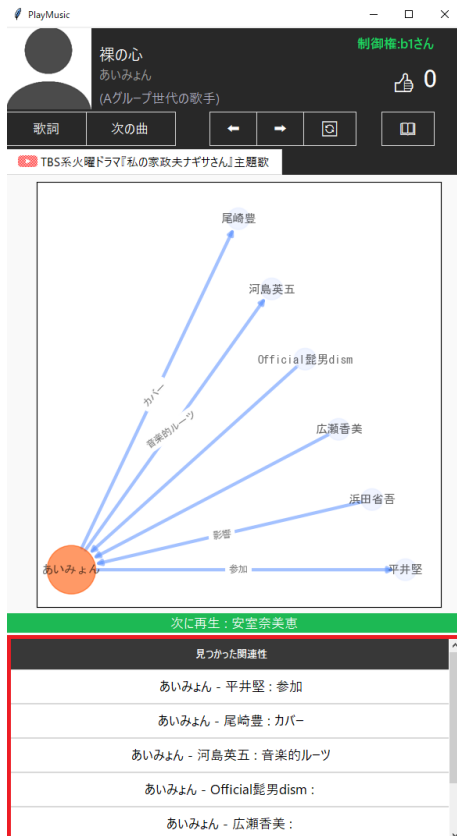


図 4 関連性ボタンの表示画面
(赤枠部)



図 5 詳細情報とグッドボタンの表示画面
(赤枠部)

(5) クリックされた歌手が相手グループの世代で流行した歌手だった場合、「皆さんでクリックされた歌手について話し合ってください」といった文章が画面上に表示され(図 6), また文章が音声合成によって読み上げられる。



図 6 音声合成による問いかけ時の画面

(6) クリックされた歌手が自身のグループで流行した世代の歌手だった場合、次に再生する歌手を、クリックされた歌手に指定することができる(図 7)。



図 7 次に再生する歌手の指定ボタン (赤枠部)

(7) 曲が終わる 30 秒前になると画面上に残り秒数が表示され、カウントダウンが始まる(図 8)。また曲が終わる 20 秒前になった時点で、流れている曲の評価を促すようなメッセージが表示され、画面上にあるグッドボタンで評価を行う(図 9)。曲の評価は次の自分のターンに再生される歌手の曲に利用され、自分が評価した曲の中で、最も評価の高い楽曲の類似楽曲が、次に再

生される歌手の楽曲中から選ばれるようになっている。



図 8 カウントダウンの表示
(赤枠部)

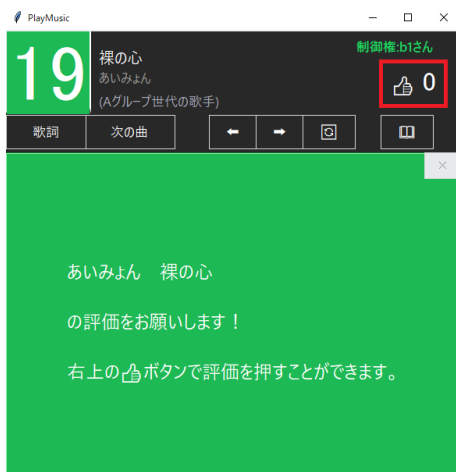


図 9 曲の評価を促す画面表示画面
(赤枠部はグッドボタン)

(8) 曲の再生が終わると、次は相手グループの1人が操作者となる。

2.1 歌手及び楽曲の取得

音楽サービスである Spotify の API メソッドを用いて、お互いの世代で流行した歌手及びその楽曲を取得する。お互いの世代で流行した歌手は Spotify が作成した Karaoke Rewind プレイリストから取得している。Karaoke Rewind プレイリストは、年代ごとにカラオケで最も歌唱された曲の上位 100 曲文が保存されており、Spotify API のメソッドの 1 つである「Get a Playlist」[12] を用いる。上記のメソッドにより、プレイリストに属する歌手名と、Spotify の Artist ID (Spotify 固有の ID) を取得し、お互いの世代で流行した歌手を取得する。(現在の実装では各グループの歌手数は 70 人程) また類似楽曲検索を用いるために、お互いのグループの世代で流行した歌手の曲を取得する。そのために歌手の楽曲中で、最も再生数が多い楽曲の上位 10 曲分の Track ID (Spotify 固有の ID)、及び MP3 形式の 30 秒間の試聴用データを取得しており、Spotify の API メソッドの 1 つである「Get an Artist's Top Tracks」[12] を用いる。

2.2 歌手の関連性の取得

手順として Python の Wikipedia ライブラリを用いて [13]、お互いの世代で流行した歌手の Wikipedia ページ内の文章を句点区切りで取得する。次に取得した文章に対して、新語、固有表現の抽出に有効な mecab-ipadic-NEologd 辞書 [14] をインストールした Mecab [15] を用いて人物名を抽出し、歌手名、抽出された人物名、人物名が出現した 1 文で CSV を作成する。その時に歌手名と抽出された人物名には関連があるとする。また歌手の活動期間、CM や映画等のタイアップ情報、バンドの場合は所属しているバンドのメンバー名といった情報も Wikipedia、及び DBpedia [16] からスクレイピングを用いて取得する

2.3 楽曲の再生

操作者は相手グループの世代で流行した歌手の楽曲を聴く。再生の順番は事前に決定しており、相手グループの世代の歌手と、より関連性が多い歌手の楽曲を優先して再生させている。また PC 版の Spotify アプリを用いて楽曲の再生を行い、Spotify API のメソッド [12] を通してアプリ上で楽曲の再生を行うために、事前に Spotify Premium への登録をしている。

2.4 グラフと関連性の説明表示

歌手間の関連性を図示するために、グラフの描写、作成には、それぞれ Python ライブラリである Matplotlib [17]、NetworkX [18] を用いる。グラフは有向グラフで作成し、歌手がノードとなる。目的の歌手が他の歌手と関連がある場合エッジでつながり、目的の歌手がバンド、もしくはグループだった場合は、所属するバンドメンバーも同様にエッジでつながる。見やすさを確保するためにメンバーと接続しているエッジを除いてエッジの数は 6 本以下に制限しており、エッジラベルには、「尊敬」、「事務所」といった簡易的に 2 人の歌手の関連性が分かるような説明が記載されている。また歌手ノードをクリックすると、その歌手が起点となったグラフに更新され、繋がっているエッジに関する説明文が閲覧できるボタンが表示される、説明文はエッジでつながれた歌手間の関連性を示す 1 文であり、ボタンを押すと表示される。また歌手ノードをクリックした際に、自分のグループの世代の歌手だった場合、次の曲にその歌手の楽曲を再生するように指定ができるボタンが表示される。

2.5 音声合成によるアナウンス

Python の win32com ライブラリを用いて Windows 10 に標準搭載されている Microsoft Speech API より Microsoft Speech API (以下 SAPI) を呼び出し、音声合成を行っている。音声合成では「表示されている歌手の曲で好きな曲はありますか」といった質問の投げかけを通してコミュニケーションの促進と、「曲の評価をお願いします」といった

ユーザの操作を促すことを目的としている。

2.6 評価

表示される歌手間の情報と流れている楽曲には評価ボタンを用いて評価を入力する。どちらもそれぞれ10段階で評価することができ、楽曲の評価は、ユーザごとに記録され、最も高い評価を受けた楽曲に似た楽曲が、次の自分のターンに再生される。

2.7 類似楽曲検索

[19][20]の手法を用いて、類似楽曲検索を実現している。手順として Spotify API から取得した30秒の試聴用MP3ファイルを16kHz、32bitでリサンプリングし、raw形式に変換した後、10msごとに20次元のmfccベクトルを抽出する。次に3000個のベクトルをk-meansを用いて16個のクラスタに分類し、MFCCをシグネチャに変換する。手法として、各クラスタは正規分布と仮定して、その平均ベクトルと分散共分散行列を求め、それを楽曲Pの特徴量とする。式は次のようになっている。

$$P = \{(\mu_{p_1}, \Sigma_{p_1}, w_{p_1}), \dots, (\mu_{p_m}, \Sigma_{p_m}, w_{p_m})\}$$

μ_{pm} : 平均ベクトル, Σ_{pm} : 分散共分散行列,

w_{pm} : 重み, m : クラスタの数

Earth Mover's Distance(以下 EMD)を用いて前述で取得したシグネチャを比較し楽曲間の距離を測定する。EMDは特徴量と重みで構成された集合が2つ与えられた時に、個々の特徴量間の距離を比較し、2つの集合間の距離を求める距離尺度である。出力されたEMD値が最も小さい楽曲を類似楽曲とする。

3. 評価実験

本研究のシステムの有効性を試すために、被験者を募り評価実験を行った。本来であれば、本システムを用いる場合と用いない場合で分けて実験を行うべきである。しかし、どちらが先に実施するか等で結果が変わってしまうことが予想できたため、本システムを用いた場合のみで実験を行った。また被験者は、Zoom越しで本システムを使用し、実施後にアンケートに回答する。

3.1 実験方法

実験は2人で構成された同年代の2つのグループ(20代グループと30~40代グループ)計3組で、3回実施した。またグループ間で年齢が10歳以上離れており、グループ間で被験者同士の面識は無いことを前提としている(1回目の実験のみ、日本滞在2年目になる外国籍の留学生1名に協力を頂いている)。被験者には、グループ間の世代間ギャッ

プを埋めたいという意識を持って実施してもらったために、事前に「世代間ギャップを埋める事が目的なので、被験者同士で積極的に話し合ってもらいたい」というような説明をした。環境は、新型コロナウイルスの影響により、対面での実験が困難だったため、Zoomを用いて行った。また事前に各被験者の端末にZoomアプリをインストールしておく。スケジュールとして、実験の趣旨説明後に、被験者間での自己紹介を5分程度挟み、本研究を用いた音楽聴取を30分程度行ってもらおう。なお、時間の制約があるため、1曲に対する再生時間は90秒としている。本システムの終了後には、Googleフォームを用いたアンケートと各被験者に本実験の感想を聞いた。アンケートは7段階評価で行い、主に本研究のシステムに関する質問と、世代間ギャップの解消に関する質問の2つによって構成される。またログデータを取得する機能を実装しており、下記の内容が記録される。

- 楽曲の評価時：日時、曲名、評価したグループ名、評価値(10段階評価)
- 関連性ボタンの押下時：日時、ボタン名、評価したグループ名、評価値(10段階評価)
- 再生終了時：評価された歌手名、評価値(10段階評価)]

3.2 実験結果と考察

本実験の結果を、アンケートの集計結果、ログを用いて考察する。全3回の実験で得たログの集計結果を、表1~表6に実施回数別で示す。表1~表2では被験者から次に再生する歌手を変更させるような動作が無く、年代も20代、40代のグループで行ったため、3回の実験共に再生された歌手が同一となっている。実験に対してのアンケートの集計結果を表7~9に示す。また、アンケートは7段階で評価される。

表1では各歌手ごとの評価を10段階で入力した値を実施回数別に示している。再生した順番に歌手名を表示しており、結果として時間経過と共に歌手の評価値が上がっていく傾向が見られた。これは、システムの操作に慣れていったためと考える。また話に集中し、評価を押し忘れてしまうと感じる被験者もおり、そのため、評価を入力されていない歌手も多かった。例えば、1回目の実験においてDREAMS COME TRUEは、会話時間が85秒となっているが、評価は0となっており、話に集中してしまったと考えられる(表1,2)。

結果として、歌手ごとの会話時間(表2)は、各歌手の評価(表1)と同じように時間経過と共に会話時間が上がっていく傾向があった。例えば、「鬼滅の刃」の主題歌を歌唱するLiSAでは、「鬼滅の刃の曲」や「前からアニソンを良く歌っている人だよ」といった会話が多くあり、直近のタイアップ曲を歌唱する歌手は、離れた世代の音楽に興味を持つきっかけになりやすいと考えられる。

表 1 歌手ごとの評価 (10 段階評価)

歌手名	1 回目	2 回	3 回目
安室奈美恵	0	1	4
中島美嘉	7	5	4
宇多田ヒカル	3	2	7
LiSA	10	3	6
DREAMS COME TRUE	0	2	3
平井堅	8	2	2
ORANGE RANGE	6	1	0
菅田 将暉	5	3	8
椎名林檎	3	1	5
MONGOL800	10	3	7

表 2 歌手ごとの会話時間 (秒)

歌手名	1 回目	2 回目	3 回目
安室奈美恵	45	60	42
中島美嘉	46	57	50
宇多田ヒカル	60	57	50
LiSA	80	85	77
DREAMS COME TRUE	85	75	69
平井堅	88	83	60
ORANGE RANGE	75	76	69
菅田 将暉	80	72	70
椎名林檎	58	60	57
MONGOL800	84	55	78

表 3 では、Spotify の API[12] から取得した歌手ごとの人気度 (表 4)、歌手ごとの評価 (表 1)、歌手ごとの会話時間 (表 2) との相関係数をそれぞれ実施回数別に示す。歌手ごとの人気度は、Spotify 上での再生回数に基づいて決定しており、最大が 100、最小が 0 になっている。結果として被験者から入力された歌手の評価値が大きくなるにつれて、会話時間も長くなるような、弱い正の相関がみられた (表 3)。また同じように歌手の評価値が大きくなるにつれて、歌手の人気度も高くなるような、弱い正の相関がみられた。

表 3 各項目との相関係数

項目 1	項目 2	1 回目	2 回目	3 回目
歌手の評価	歌手ごとの会話時間	0.41	-0.13	0.13
歌手の評価	人気度	0.06	0.13	0.49
歌手ごとの会話時間	人気度	0.01	0.38	0.1

表 4 歌手ごとの人気度 (最大 100)

歌手名	人気度
安室奈美恵	63
中島美嘉	59
宇多田ヒカル	67
LiSA	78
DREAMS COME TRUE	60
平井堅	59
ORANGE RANGE	56
菅田 将暉	69
椎名林檎	61
MONGOL800	54

表 5 は、関連性の表示回数をエッジラベルの有無で分けた結果である。歌手間の関連性が“事務所”、“尊敬”のように簡易的に表示できる場合のみ、エッジラベルが有り、表示ができない場合はエッジラベルが無しになっている。結果としてエッジラベルの有無で生じる表示回数の差は、わずかであり、エッジラベルによって関連性の表示回数が増えることは認められなかった。原因としては、被験者がエッジラベル付きのグラフを見ることで、歌手間の簡易的な関連性が理解できてしまうために、関連性の情報を表示するまで行わない事例が散見されたためと考えられる。例えば、LiSA の場合、関連する歌手とのエッジに全てエッジラベルが有り、簡易的な関連性が理解できてしまうためか、被験者による評価も 1 度のみしか行われていない。

また表 6 は、計 3 回の実験で表示された 2 人の歌手の関連性の情報に対して入力された 10 段階の評価を降順で表示している。表 5、6 より関連性の情報が表示された回数に比べて、評価された回数が少なくなっている。これは被験者が、評価を忘れていた場合も考えられるが、評価を押すに値するような情報を提示することができなかったためと考えられる。理由として、表示される情報は全て 1 文であり、関連性の詳細までは掴みにくいことが多いためだと考える。また表 6 にある通り、エッジラベルを持つ歌手間の関連性の情報は評価されやすいと考えられる。特に“事務所”、“憧れ”、“影響”といった関連性が高く評価されやすい結果となった。

表 5 関連性の表示回数

エッジラベルの有無	1 回目	2 回目	3 回目
有り	9	4	5
無し	6	8	3
合計	15	12	8

表 6 関連性の評価値

歌手 A	歌手 B	エッジラベル	評価値
福山雅治	桑田佳祐	事務所	4
浜田省吾	尾崎豊	事務所	3
DREAMS COME TRUE	いきものがかり	影響	3
LISA	SPEED	憧れ	2
岡野昭仁	Aimer	参加	2
藤井フミヤ	ISSA	作詞	2
岡野昭仁	Aimer	参加	2
浜田省吾	あいみょん	影響	1
DREAMS COME TRUE	中島美嘉	ファン	1

次に表 7 では、7 段階評価の平均点を示している。問 1、問 2 では、平均点が 4 以上 6 以下となっており、比較的システムの使いやすさ、提供される話題が適切だったと考えられる。また回答に対しての自由記述欄には、「画面をそのままクリックでき、操作もわかりやすい」といった答えや、一方で「再生時間に制限があるので、全機能を理解し、使いこなすのは難しい」といった意見があった。そのため、操作性に関しては、個人差あったと考えられる。

表 8 では、離れた世代の歌手で、「今後聞いてみたい歌手がいた」と答えた被験者が 40%以上となっていた。次に相手グループの世代で流行した音楽に対して印象が変わったかというアンケートに対して(表 9)7 段階評価中 5 近くの評価があるという点から、システムを通して離れた世代の歌手に対して興味を持つきっかけ作りを行えた可能性が高いと考えられる。また表 8 で回答された、今後聞いてみたい歌手としては、「LiSA」、「椎名林檎」といった歌手が挙げられており、傾向として直近のタイアップ曲を歌唱する歌手や活動期間が長い歌手等は、離れた世代に興味を持つきっかけになりやすい歌手と考えられる。最後に相手グループとの会話を取りやすくなったかというアンケート(表 9)では、7 段階評価中 5 段階以上の評価があるという点では、システムを通して、離れた世代の人とのコミュニケーションを円滑化することができたと考えられる。また回答に対しての自由記述欄には、「共通の話題ができたので、会話をしやすかった」、また「もう少し長い時間議論することで、より円滑な会話になる」といった意見が上げられた。

表 7 アンケート結果 [1]

項目	平均	標準偏差
問 1. システムは 使いやすかったか (7 段階)	4.47	1.66
問 2. システムが提供する 音楽の話題は適切だったか (7 段階)	5.25	1.42

表 8 問 3. 離れた世代の歌手で、今後聞いてみたい歌手はいたか (%)

項目	割合 (%)
はい	41.7
いいえ	58.3

表 9 アンケート結果 [2]

項目	平均	標準偏差
問 5. 相手グループの世代の音楽に対して、 印象は変わったか (7 段階)	4.91	1.38
問 6. 相手グループとの会話は 取りやすくなったか (7 段階)	5.50	1.50

4. おわりに

本研究では、世代間ギャップを解消するためのきっかけ作りとして、年齢の離れた人同士が、ポピュラー音楽と歌手間の関連情報を用いて話題の提供と会話の促進を行うデスクトップアプリを提案した被験者実験を行った結果、本研究を用いることで、離れた世代とのコミュニケーションが取りやすくなり、また離れた世代で流行した歌手に興味を持つようなきっかけ作りを行うことができた。実験に対するアンケートでは、世代間ギャップを解消する上で音楽を用いる事は有効という意見も得ることができた。また、直近でタイアップ曲をリリースしている歌手といった、特定の要素を持つ歌手が高い評価を受けやすく、コミュニケーションの促進にも繋がるといった傾向が見られた。そのため、今後再生する歌手を選定していく上で活かすことができると考える。また本稿の実験結果から、曲の評価や、グラフの操作等ユーザによる操作が多いため、覚えるのが困難といった意見があったので、初心者でも、より直観的に使用できるような GUI に改良していきたい。

参考文献

- [1] Rurika Shimizu, Yutaka Takase, Yukiko I Nakano : Audio Based Group Conversation Support System, UBIComp/ISWC'17 ADJUNCT, SEPTEMBER 11-15, 2017
- [2] Gianluca Schiavo: Overt or subtle? Supporting group conversations with automatically targeted directives: Proceedings of the 19th international conference, February 2014
- [3] 梅田 将孝, 田中 文英: 視覚的なイメージを付与することによる世代間遠隔コミュニケーションの円滑化, 2016 年度人工知能学会全国大会 (第 30 回), セッション ID:3J3-OS-18a-2
- [4] 松本 大希, 滝口 恭平, 小高 大典, 北原 鉄朗: 複数人が共有する場のための BGM 選曲手法に関する研究, 日本大学卒業論文 (2012).
- [5] 鈴木 潤一, 末次 尚之, 北原 鉄朗: 複数ユーザーが同一楽曲を聴くための選曲・再生システム, 情報処理学会論文誌, Vol.57, No.12, pp.2526-2530 (2016)
- [6] 斉藤 栄太郎: 特集 みんなかつては"新人類" 世代間ギャップを乗り越えよう, 日経 systems (246), 60-65, 2013-10

- [7] 佐藤 由美子: 世代研究の展開と課題 世代間ギャップと次世代研究, フォーラム現代社会学, 6 巻 (2007), p. 5-14
- [8] 長谷川文雄: 世代間交流の活性化による新たなコミュニティ形成に関する研究, 財団法人ハイライフ研究所, <http://www.hilife.or.jp>
- [9] Jennifer L. Martin Owen Peters: Embracing the Generational Differences to “Bridge the Gap” in the Workplace, IEEE, No. ESW2019-09
- [10] Alo Allik, Florian Thalmann, Mark Sandler: MusicLynx: Exploring Music Through Artist Similarity Graphs, WWW 2018, April 23-27, Lyon, France
- [11] 2019 年度 「音楽メディアユーザー実態調査」報告書公表—日本レコード協会, 入手先 (<https://www.riaj.or.jp/f/report/mediauser/2019.html>)
- [12] Web API — Spotify for Developers, 入手先 (<https://developer.spotify.com/documentation/web-api>)
- [13] wikipedia 1.4.0 — pypl —, 入手先 (<https://pypi.org/project/wikipedia/>)
- [14] mecab-ipadic-neologd, 入手先 (<https://github.com/neologd/mecab-ipadic-neologd>)
- [15] MeCab, 入手先 (<https://taku910.github.io/mecab/>)
- [16] DBpedia Japanese, 入手先 (<http://ja.dbpedia.org/>)
- [17] matplotlib, 入手先 (<https://matplotlib.org/>)
- [18] networkX, 入手先 (<https://networkx.org/>)
- [19] Elias Pampalk: Computational Models of Music Similarity and their Application in Music Information Retrieval,
- [20] Beth Logan, Ariel Salomon: ippA Content-Based Music Similarity Function, Cambridge Research Laboratory, CRL 2001/02