

Mnemonic DJ : 暗記学習のための替え歌自動生成システム

伊藤 悠真^{1,a)} 寺田 努^{1,2,b)} 塚本 昌彦^{1,c)}

受付日 2015年1月7日, 採録日 2015年8月12日

概要: 暗記学習の一般的な手法の1つとして替え歌を用いる手法がある。これまでにさまざまな暗記学習システムが開発されているが、暗記用替え歌の一般的な生成手法や自動で生成するシステムは筆者らの知る限り確立されていない。よって本研究では暗記学習のための替え歌自動生成システムの構築を目的とする。提案システムは、替え歌の元となる楽曲を保持しており、学習者が入力した暗記したい項目が、いくつかの曲の歌詞に割り当てられて替え歌として出力される。本研究では暗記したい項目を歌詞に割り当てる割当てアルゴリズムを提案する。これにより、提案システムは暗記に適した替え歌を生成できる。評価実験の結果、提案システムにより生成された替え歌は丸暗記に比べて効果があることが示唆された。

キーワード: 替え歌生成, 暗記学習, 自動生成システム

Mnemonic DJ: Automatic Mnemonic Song Generation System for Rote Learning

YUMA ITO^{1,a)} TSUTOMU TERADA^{1,2,b)} MASAHIKO TSUKAMOTO^{1,c)}

Received: January 7, 2015, Accepted: August 12, 2015

Abstract: Mnemonic song is known as a memorizing method. Although various systems have been developed for memorization assistance, there is no general method for creating mnemonic songs. Therefore, the goal of our study is to construct an automatic mnemonic song generation system. The system has a database of popular songs as basis of mnemonic song and it outputs mnemonic songs from items to be memorized. Concretely, the system assigns the items to lyrics of the songs. Our proposed algorithm enables the system to generate appropriate mnemonic songs. Evaluation results suggested that the mnemonic songs generated by the proposed system were effective compared with rote memorization.

Keywords: mnemonic song generation, rote learning, automatic generation system

1. はじめに

単語の羅列を暗記する暗記学習には語呂合わせなどさまざまな方法が存在する [1]. その中の1つに替え歌を用いた暗記法というものがあり [2], [3], [4], その有用性も知られている [5], [6]. 情報処理技術の進展により, さまざまな

暗記学習支援システムが開発されている [7], [8] が, 暗記用の替え歌の一般的な生成手法や自動で生成するシステムは筆者らの知る限り確立されていない.

そこで本研究では暗記用の替え歌自動生成システムを構築する. 提案手法では, 替え歌の元となる楽曲データベースを保持し, 学習者が暗記したい単語のリスト (以下単語リストと呼ぶ) を入力すると, 楽曲データベース中のいくつかの曲の歌詞に単語リスト中の全単語が割り当てられて出力される. 加えて本研究では, 暗記学習の種類として単語リストの単語の順序が重要な暗記学習と, そうでない暗記学習の2種類を想定し, それぞれの場合について替え歌生成手法を提案する. これにより, 学習者は2種類の暗記

¹ 神戸大学大学院
Kobe University, Kobe, Hyogo 657-8501, Japan
² 科学技術振興機構さきかけ
PRESTO, JST, Chiyoda, Tokyo 102-0076, Japan
^{a)} yuma-ito@stu.kobe-u.ac.jp
^{b)} tsutomu@eedept.kobe-u.ac.jp
^{c)} tuka@kobe-u.ac.jp

学習を目的とした単語リストに対して、暗記に適した替え歌を生成できる。提案手法では、単語リストを楽曲に割り当てるとき、歌詞を一定の区切りで分割し、その分割した区切りごとで単語リストの割当ておよび割当て時の元の楽曲からの乖離度（以下割当てコストと呼ぶ）の定義を行うことで、単語リストの割当ての前後でなるべく楽曲が変化しないように設計する。本研究では、歌詞の譜割り（音符のリズムの割り振り）に着目した場合と、歌詞と単語リストの韻に着目した場合それぞれの割当ておよび割当てコストを提案している。

以下、2章では関連研究について述べ、3章で提案システムの設計について説明する。4章でシステムの実装、5章で提案システムの評価と考察について述べ、最後に6章で本論文のまとめを行う。

2. 関連研究

2.1 記憶支援に関する研究

本研究のように記憶を支援する手法やシステムに関する研究はいくつかある。文献 [7], [8] では、人間が生活する上で経験するイベントの内容を忘れさせないようにさせるシステムについての研究を行っている。文献 [7] は、facebook や twitter などの履歴から、「あなたは誰とコンサートに行きましたか」などの質問をメールで送ることでユーザに想起を促すもので、文献 [8] は忘れてしまいそうな事柄を時間経過を元に判断して写真を提示するものである。これらの研究では自身の経験を忘れさせないようにすることに着目したものであり、本研究のように暗記そのものの方法を提案したり支援するような場合とは異なる。

暗記そのものの支援として、語呂合わせを自動で生成するものがある [9], [10]。語呂合わせとは、元の単語の羅列をそれらの頭文字をつなげるなどして他の意味を持った文字列に置換することであり、数字列の暗記や専門用語の暗記などでは一般的に用いられている。文献 [9] は任意の単語列の頭文字を抜き出し、それらの文字をすべて含む単語を辞書に登録している単語から探し出すという手法をとっている。入力単語数が長い場合は頭文字を数個ずつに分け、それぞれで単語を決定したのち、文章らしくなるように助詞などを付与することで最終的な結果を出力している。しかし、この研究では長くても8個程度の単語に対してしか自然な語呂合わせが作成できないとしている。

また、文献 [10] では数字列に対する語呂合わせの自動生成が提案されている。これらは数字1つ1つに読み方を割り当て、入力された数字列に対し、辞書から単語を探すというものである。これらのシステムでは比較的長い数字列に対しても語呂合わせが生成できているが、数字列に対してのみ語呂合わせを生成しており、専門用語などの一般的な単語に対しては適用できない。

2.2 替え歌による暗記学習

替え歌による暗記学習は語呂合わせ同様に広く知られた暗記法である [2], [3], [4]。海外では、小学生などの子供がアルファベットやアメリカの50州の名前をアルファベット順に暗記する場合などに替え歌が用いられている [1], [2]。また、日本においては中学や高校での理科や社会の用語の暗記 [3]、税理士試験のような国家試験対策の暗記学習にも用いられている [4]。

暗記学習のために替え歌を用いることの有用性を示した文献はいくつかある [5], [6]。文献 [6] によれば、記憶力を高めるために替え歌を用いることは以下の点で有用であるとしている。

- 新たに暗記する項目とすでに記憶していることとを組み合わせられる。
- 歌やリズムを思い出せば、それに関連する情報も同時に思い浮かぶ。

よって本研究では、替え歌の元となる楽曲として学習者がよく知る楽曲を用いることとする。

文献 [11] では人が音楽を記憶するときのプロセスを認知心理学的見地と情報理論の見地から調査している。人間は記憶を行うとき、数個程度の情報で意味を持つまとまりを作り、そのまとまりを数個集めてさらに大きな意味のあるまとまりとする、など意味のあるまとまりを体系的に組み上げることが長期記憶に結び付くとしている。音楽においてはフレーズや、フレーズをいくつかまとめた楽節などが人間の記憶における意味のあるまとまりに相当し、そのまとまりは人間が楽曲を聴く際には無意識的に行われてると述べられている。つまり、単語リストを楽曲の歌詞に乗せることで、単語リスト内で自動的にまとまりを作ることができ、暗記に有用であると考えられる。

2.3 自動作曲システム

本研究では、単語リストという歌詞の入力に対し、既存の楽曲中から最も適切に割り当てられる楽曲を選ぶことで替え歌を生成するが、入力された歌詞に対しメロディを自動生成する自動作曲システムに関する研究がある [12]。この研究は音楽初心者でも容易に作曲ができるようなシステムを提案しており、歌詞と作曲したい曲のジャンルなどを入力すると、システムが自動的に作曲を行うものである。歌詞に対して曲をつけるという点は本研究と同じであるが、生成されるのは新たに作曲されたものであり、学習者が知っている楽曲ではないため、本研究の目的とは合致しない。

3. 設計

図 1 に提案システムの処理の流れを示す。システムは楽曲が登録された楽曲データベースを保持しており、データベース中の楽曲から替え歌が作られる。2.2 節より、す

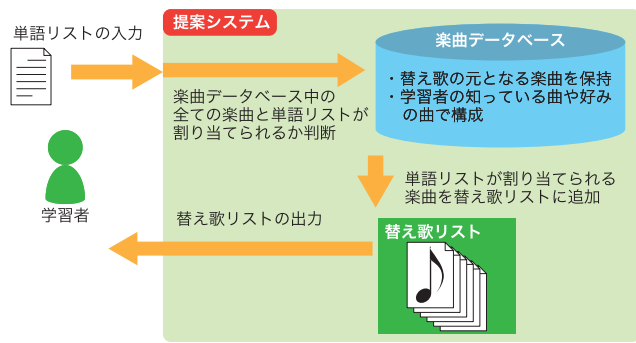


図 1 提案システムの処理の流れ

Fig. 1 Processing flow of the proposed system.

でに記憶している楽曲に対し替え歌を生成することが重要であるため、本研究では楽曲データベースはユーザ個人が記憶している楽曲や好きな楽曲で構成されている。学習者が単語リストをシステムに入力すると、システムはデータベース中の各楽曲と単語リストで音符数や母音などの条件を元に替え歌が生成可能かどうかを判断し、替え歌を出力する。

3.1 割当てアルゴリズムの設計方針

本研究では、入力された単語リストを歌詞として割り当てるアルゴリズムを提案する。

2.2 節より、替え歌を用いて暗記するために楽曲内にある意味のあるまとまりと単語リストの1単語を対応させることで、暗記学習に有効な替え歌が生成できる。替え歌内を作るまとまりは単語リストの1単語の長さと同程度である必要があると考えられ、これは替え歌を生成する言語によって異なる。本研究では、日本語または英語を想定する。楽曲中に設ける意味のあるまとまり（以下区切りと呼ぶ）は日本語では「文節」、英語では「単語」とし、区切りの長さの基準となる単位（以下分割単位と呼ぶ）は日本語では「モーラ」、英語では「音節」を用いる。ただし、実際の楽曲では歌詞にのびし音が含まれて歌われる場合もあるため、本研究ではのびし音も1つの分割単位とし、のびし音を含めた状態で区切りとして扱う。

モーラとは日本語の拍と呼ばれるもので、通常1対の子音と母音からなる。例として「チョコレート」をモーラで分けると「チョコ、レ、ー、ト」となり長さは5となる。また、音節とは英語などの言語で一般的に用いられる連続する言語音を区切る単位の一つであり、1個の母音と複数の子音から構成される。たとえば「information」は音節で分けると「in-for-ma-tion」となり、長さは4となる。

本研究では単語リストを暗記するパターンとして以下の2種類を想定する。

- 単語リストの順番どおりに暗記：歴史上の事件の順番の暗記、元素記号の暗記など
- 単語リストの順番に関係なく暗記：州名や県名などの

あるグループに属する名前の暗記など

提案システムでは、どの単語が楽曲中のどの区切りに割り当てられるかを決定する割当てアルゴリズム（以下、楽曲全体の割当てアルゴリズムと呼ぶ）と、個々の単語内の分割単位と区切り内の分割単位がどのように割り当てられるかを決定するアルゴリズム（以下、区切りごとの割当てアルゴリズムと呼ぶ）の2種類の割当てアルゴリズムによって替え歌を生成する。

次節以降で、楽曲全体の割当てアルゴリズムおよび区切りごとの割当てアルゴリズムについて詳細に説明する。楽曲の区切りを N 個、単語リストの単語数を M 個、 n 番目の区切りを S_n 、 m 番目の単語を W_m とし、 S_n に W_m を割り当てるときの割当てコストを $d(S_n, W_m)$ （以下、区切りごとの割当てコストと呼ぶ）、ある単語リストがある楽曲に割り当てられるときの割当てコストを C （以下、楽曲全体での割当てコストと呼ぶ）とする。また、 S_n に W_m が割り当てられるか否かを割当ての可不可とし、割当て不可能の場合は $d(S_n, W_m) = \infty$ を、割当て可能な場合は $d(S_n, W_m) \neq \infty$ となるように、各割当てアルゴリズムで定義する。ただし、具体的な定義は3.3節で示すこととする。 S_n と W_m はそれぞれ分割単位の系列として表され、それぞれ $S_{n,i}(1 \leq i \leq I_n)$ 、 $W_{m,j}(1 \leq j \leq J_m)$ と表す。

3.2 楽曲全体の割当てアルゴリズム

楽曲全体の割当てアルゴリズムでは、区切りごとの割当てアルゴリズムで計算される区切りごとの割当てコストおよび割当ての可不可をもとに楽曲中のどの区切りに、単語リストのどの単語が割り当てられるかを決定する。楽曲全体での割当てアルゴリズム適用後、単語リストが楽曲に割当て可能であれば、楽曲全体での割当てコストの計算および替え歌の生成を行い、割当て不可能であれば替え歌の生成を行わない。また、いずれのアルゴリズムにおいても、どの単語も割り当てられなかった楽曲中の区切りには日本語では「ラ」を英語では「la」を各分割単位に割り当てることとする。

本研究では入力された単語リストの単語の順番を並べ替えるか否かで楽曲全体の割当てアルゴリズムを分けるが、楽曲全体での割当てコスト C の計算方法は共通で、以下の式で定義する。

$$C = \sum d(S_{n'}, W_{m'})$$

ただし、 $(S_{n'}, W_{m'})$ は最終的に決まった割当ての組合せである。

順番の並べ替えをしない割当てアルゴリズム

単語リスト中の単語を順番どおりに暗記しなければならない場合、順番の並べ替えをしない割当てアルゴリズムを用いる。楽曲全体での区切りおよび単語リストの割当て法を図2を用いて説明する。図2上部の矩形は楽曲の歌詞

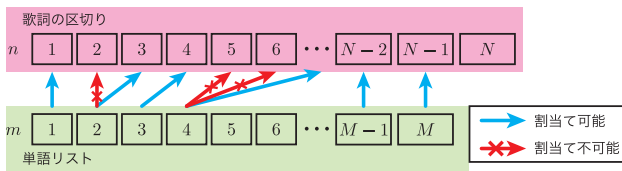


図 2 並べ替えなしの割当てアルゴリズム

Fig. 2 An assignment algorithm that does not change the order.

の区切りを表し、下部の矩形は単語リストの各単語を表す。さらに、青色矢印は暗記項目がその矢印の指す区切りに対し割当て可能であることを表し、赤色矢印は割当て不可能であることを表す。

図 2 に示すように、提案システムは楽曲の歌詞の最初の区切りから順に単語リストの各単語が割当て可能かどうかを判断する。割当て可能な場合は、次の区切りおよび次の単語で割当て可能かを判断し、割当て不可能の場合は、同じ単語が次の区切りで割当て可能かどうかを判断する。図 2 の例では、 $(n, m) = (1, 1), (3, 2), (4, 3), \dots, (N-2, M-1), (N-1, M)$ が割当てとして決定している。このようにして、単語リスト中の単語すべてが歌詞中の区切りに割り当てられた場合、この楽曲に単語リストは「割当て可能」とする。

順番の並べ替えを行う割当てアルゴリズム

単語リスト中の単語を順番に関係なく暗記する場合は、並べ替えを行う割当てアルゴリズムを用いる。この場合、替え歌中のどの場所にも、かつ何回でも単語が出現できる。

提案アルゴリズムでは、まず単語と区切りの全組合せで区切りごとの割当てコストを計算し、ハンガリー法 [13] を用いてどの区切りにどの単語を割り当てるかを決定する。ハンガリー法とは複数人に複数個の仕事を 1 人 1 個ずつ割り当てるとき、最も効率の良い割当て方を決定するアルゴリズムで、ある人にある仕事を割り当てるときのコストを行列状にしたコスト行列と呼ばれるものを用いて計算される。ここで、割当て不可能な単語と区切りのペアに対しては $d(S_n, W_m) = \infty$ となっているので、他に割当て可能な単語と区切りのペアがある限り、割当ては避けられる。

ハンガリー法では 1 つの単語に対して、1 つの割当て先しか決定できない。よってハンガリー法によって割当てが決定した後、 n 番目の区切りに単語が割り当てられていないとすると n 番目の区切りに割り当てられる単語 W_{m_h} を式 (1) によって求める。

$$m_h = \arg \min_{1 \leq m \leq M} d(S_n, W_m) \tag{1}$$

これを単語が割り当てられていないすべての区切りに対して行う。なお、その区切りに対する割当てコストがすべて ∞ の場合はその区切りに割当て可能な単語は存在しないとして単語の割当てを行わない。図 3 に提案アルゴリズム適用の例を示す。図 3 は割当てのコスト行列を表し、行

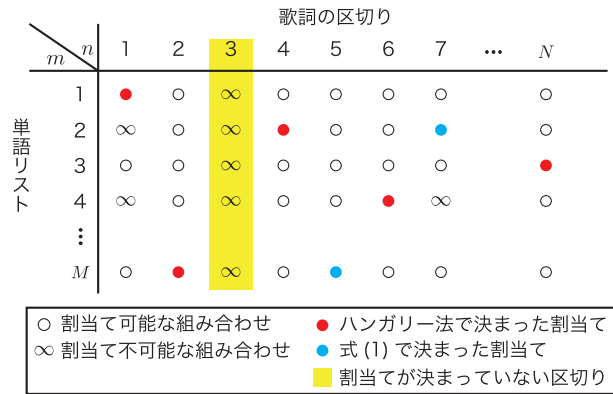


図 3 並べ替えありの割当てアルゴリズム

Fig. 3 An assignment algorithm that changes the order.

は歌詞中の区切り S_n 、列は単語リストの単語 W_m を表している。図 3 では便宜上添え字のみを記してある。また、白丸は $d(S_n, W_m) \neq \infty$ 、 ∞ は $d(S_n, W_m) = \infty$ のときを表し、赤丸はハンガリー法適用後に決定した割当て、黄色の矩形で囲った部分は割当てが決まっていない区切りを表す。さらに、青丸は式 (1) 適用後に新たに決まった割当てである。図 3 の例では、 $(n, m) = (1, 1), (2, M), \dots, (N, 3)$ などがハンガリー法適用後に割当てとして決定しており、その後式 (1) を適用したことで $(5, M), (7, 2)$ が新たに割当てとして決まっている。また、 $n = 3$ の区切りはすべて $d(S_n, W_m) = \infty$ のため式 (1) を適用せず、最終的に割当てが存在しないままとなる。

3.3 区切りごとの割当てアルゴリズム

区切りごとの割当てアルゴリズムでは、ある単語と楽曲中のある区切りにおいて、各分割単位がどのように割り当てられるかを決定する。本研究では以下の 2 種類の指標からアルゴリズムを設計する。

- 譜割りに基づいた割当て：元の楽曲の譜割りと単語割当て後の譜割りが近くなるように割り当てる
- 韻に基づいた割当て：元の楽曲の歌詞の韻と単語の韻をなるべく一致させるように割り当てる

譜割りに基づいた割当てアルゴリズム

譜割りに基づいた割当ては I_n と J_m の大小関係によって異なり、以下で図 4 を用いて説明する。ただし、図 4(a), (b) の上部および図 4(c) の最上部の音符は楽曲のそれぞれの分割単位に割り当てられている音符のリズム要素だけを表したもので、下部の正方形はある単語の各分割単位を表す。

(i) $I_n \geq J_m$ のとき

$I_n = J_m$ のときは図 4(a) に示すように暗記項目の各分割単位を前から順に音符に割り当て、 $I_n > J_m$ のときは図 4(b) に示すように暗記項目の各分割単位を音符列に対し均等に割り当てる。このとき S_n と W_m は「割当て可能」とする。

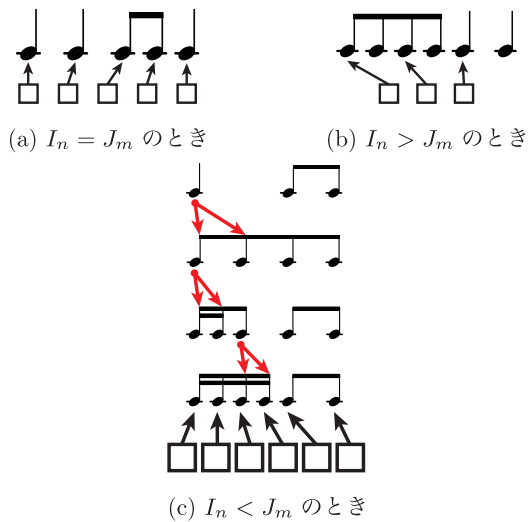


図 4 譜割りに基づいた割当てアルゴリズム
Fig. 4 An assignment algorithm based on rhythm.

(ii) $I_n < J_m$ のとき

この場合、まず図 4(c) の赤矢印で示すように、音符列中で最も長い音符を半分の長さにする事で音符数を増やし、単語の分割単位数と等しくなったところで前から順に各分割単位を割り当てる。ただし、音符列中で最も長い音符が 16 分音符の場合、これ以上音符の長さを短くすると割り当てられた分割単位が聞き取りにくくなってしまふため、分割は行わないこととする。よって、すべての音符が 16 分音符になっても分割単位数の方が多き場合は「割当て不可能」となる。また、割当てコスト $d(S_n, W_m)$ は、

$$d(S_n, W_m) = \begin{cases} |I_n - J_m| & \text{割当て可能な場合} \\ \infty & \text{割当て不可能な場合} \end{cases}$$

と定義する。よって図 4 の例はすべて割当て可能であり、割当てコストは図 4(a) の例では $|5 - 5| = 0$ 、図 4(b) の例では $|6 - 3| = 3$ 、図 4(c) の例では $|3 - 6| = 3$ となる。

韻に基づいた割当てアルゴリズム

韻とは母音と子音の一致度といい換えることができ、本研究ではまず分割単位間の距離を母音と子音に基づいて定義する。

一般的に 1 つの分割単位は母音、母音より前の子音、母音より後の子音の 3 つで構成されているため、それぞれについて距離を計算することとする。ここで、のびし音は直前の分割単位の母音のみで構成されているとする。国際的な音声記号を定義している国際音声記号 (IPA: International Phonetic Alphabet) [14] によると、母音は「舌の高低」、 「舌の前後」、 「円唇母音か否か」の 3 要素、子音は「調音方法」、 「調音部位」、 「有声音か無声音か」の 3 要素で決定される。よって母音、子音を決定している要素ごとに距離を計算し、和をとることで母音間および子音間の距離を計算できる。具体的には「舌の高低」、 「舌の前後」はその程度

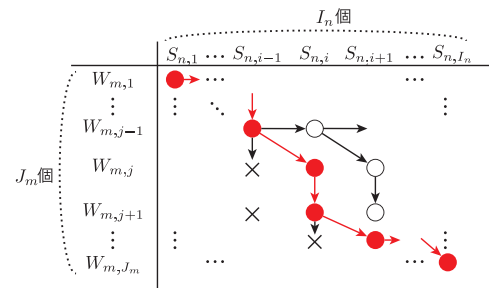


図 5 韻に基づいた割当てアルゴリズム
Fig. 5 An assignment algorithm based on rhyme.

に応じて数値をマッピングし差をとることで、それ以外の要素に関しては排他的論理和のように 2 値が同じであれば 0、異なれば 1 とすることで、距離を計算する。

以上より、2 つの分割単位 $S_{n,i}$ および $W_{m,j}$ 間の距離 $d_u(S_{n,i}, W_{m,j})$ は、それぞれの分割単位から抽出される母音、母音より前の子音、母音より後の子音の 3 つに関して上述の距離を計算し和をとったものと定義する。

次に、分割単位間の距離に基づいて、DP マッチング [15] に基づいたアルゴリズムを用いて韻に基づいた割当ておよび割当てコストを計算する。

図 5 に提案アルゴリズムの処理を示す。図 5 の行は歌詞の分割単位、列は単語の分割単位を示し、最終的に決定された経路を赤丸および赤矢印で示してある。譜割りに基づいた割当てアルゴリズムと同様に、歌詞のある分割単位に単語の分割単位が複数割り当てられた場合、歌詞の分割単位に対応している音符の長さを分割して、音符を割り当てる。ここで、通常の DP マッチングでは、歌詞中のある分割単位に割当てが集中し、最終的に分割された音符が 16 分音符より短くなってしまふ可能性がある。よって本研究では、DP マッチングで用いられる累積コスト $c(i, j)$ を以下のように変えることで、1 つの分割単位に割当てが集中しないようなアルゴリズム設計を行う。

$$c(i, j) = d_u(S_{n,i}, W_{m,j}) + \min \begin{pmatrix} c(i-1, j-1) \cdot p(i-1, j-1) \\ c(i-1, j) \cdot p(i-1, j) + 1 \\ c(i, j-1) \cdot p(i, j-1) + 1 \end{pmatrix}$$

ここで、

$$p(i, j) = \begin{cases} \infty & \text{条件 (A) を満たすとき} \\ 1 & \text{それ以外の場合} \end{cases}$$

とし、条件 (A) は、

$$X > q(S_{n,i}) \text{ または } p(i-1, j-1) = p(i-1, j) = p(i, j-1) = \infty \quad (\text{A})$$

と定義する。ここで、 X は $(S_{n,1}, W_{m,1})$ から $(S_{n,i}, W_{m,j})$ までの経路において、 $S_{n,i}$ に割り当てられている単語の分

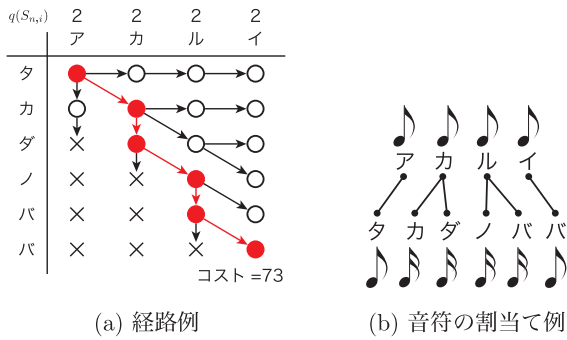


図 6 割当てアルゴリズムの適用例

Fig. 6 An example of the assignment algorithm.

割単位の数で、 $q(S_{n,i})$ はある分割単位 $S_{n,i}$ に対応する音符の長さを 16 分音符の個数で示した値であるとする。図 5 において $p(i, j) = \infty$ の箇所はバツ印で、 $p(i, j) = 1$ の箇所は丸印で示している。最終的に韻に基づいた割当てコストは以下の式で決定する。

$$d(S_n, W_m) = c(I_n, J_m) \cdot p(I_n, J_m)$$

ただし、 $p(I_n, J_m) = \infty$ のときは S_n と W_m は「割当て不可能」とし、それ以外の場合は「割当て可能」とする。

提案アルゴリズムの具体例を図 6 に示す。図 6 の例は「アカルイ」に「タカダノババ」を割り当てる場合であり、図 6(a) の表記は図 5 と同様である。また、図 6(a) 最上部の数字の列は $q(S_{n,i})$ を表す。図 6(b) は実際の音符列および歌詞を上部に、単語と割り当てられた後の音符列を下部に示し、分割単位の割当て関係を黒線で表している。図 6(a) に示すように、1 列目と 2 行目の「ア」と「カ」が交わる場所では $X = 2$ なので $p(1, 2) = 1$ であるが、2 列目と 4 行目の「カ」と「ノ」が交わる場所では $X = 3 > 2$ となってしまうため $p(2, 4) = \infty$ となって経路とはならなくなる。最終的にこの例では割当て可能で、割当てコストは 73、割当ては図 6(b) のようになる。

4. 実装

3 章で述べた替え歌自動生成システムのプロトタイプを実装した。図 7 にプロトタイプシステムのインタフェース概観を示す。プロトタイプシステムは Web アプリケーション*1として実装されており、図 7 に示すように、「替え歌のタイトル」、「並べ替え可か否か」、「割当てアルゴリズムの選択」という替え歌を生成する条件の入力欄、単語リストの入力欄、楽曲データベースとして用いる楽曲の選択欄がある。すべての項目を入力し、画面下部の「入力内容確認ボタン」を押して先へ進むことで、替え歌が生成される。プロトタイプシステムでは、生成された替え歌は musicXML 形式 [16] で出力される。musicXML とは XML 形式の楽譜表記のためのフォーマットで、このファイルを

*1 <http://bit.ly/mnemonicdj>

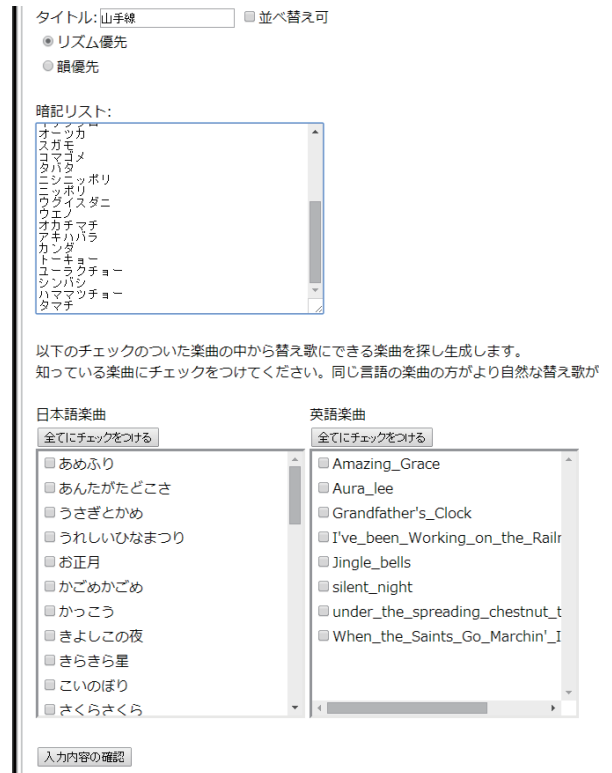


図 7 プロトタイプシステムの画面

Fig. 7 Screen of the prototype system.

楽譜作成ソフトや歌声合成ソフトウェアに入力することで生成結果を楽譜や音声に変換できる。プロトタイプシステムでは楽曲データとしてあらかじめ区切りに分けられた歌詞の csv 形式のファイルおよび、音符データが記された musicXML 形式のファイルの 2 種類のファイルを保持している。なお、日本語歌詞の文節への分割は日本語係り受け解析器である CaboCha [17] を用いており、ソフトウェア開発には Windows7 PC 上の Microsoft Visual C# 2013 を使用し、ASP.NET [18] アプリケーションとして実装した。

5. 評価

本研究では、評価実験として次の 2 種類の実験を行う。

実験 1 「区切りごとの割当てアルゴリズム」と「楽曲全体の割当てアルゴリズム」のどの組合せが好まれやすいかの評価

実験 2 提案手法によって生成された替え歌がどれだけ暗記力を向上させるかの評価

5.1 実験 1

替え歌を用いて暗記するためには、その替え歌が学習者にとって暗記しやすいと感じるか否かが重要になると考えられる。よって本実験では 3.2 節で提案した「楽曲全体の割当てアルゴリズム」2 種類と 3.3 節で提案した「区切りごとの割当てアルゴリズム」2 種類のどの組合せが学習者にとって暗記しやすいと感じる替え歌になるかをアンケート

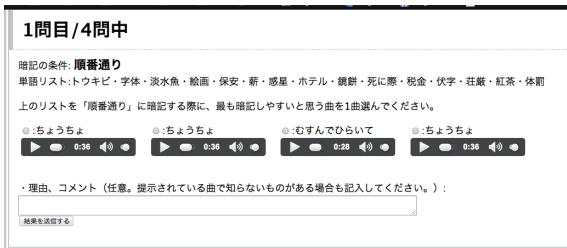


図 8 実験画面例

Fig. 8 An example of a test screen.

トによって評価する。

実験の手順

実験は Web 上のアンケートで行った。図 8 に実験で利用した Web ページの例を示す。図 8 に示すようにページ上部には「暗記の条件」として、「順番どおり」と「順不同」の 2 種類のどちらかを提示している。「順番どおり」は与えられた単語リストの単語を順番どおりにすべて暗記する場合で、「順不同」は順番に関係なくリストの単語をすべて暗記する場合である。単語リストでは 15 個の日本語の一般名詞をランダムに並べて提示している。ページ中部には、生成された替え歌 4 曲が提示されている。4 種類の替え歌は替え歌の元になった楽曲のタイトルのみを示しており、どの楽曲がどのアルゴリズムの組合せから生成されたかは被験者には分からないようになっている。

被験者には提示されている楽曲をすべて聞き、提示されている「暗記の条件」で単語リストを暗記する場合、最も暗記しやすいと感じる替え歌を 1 曲選んでもらった。ここで、4 曲の表示順はページを開くたびにランダムで表示されるようになっており、表示順による回答の偏りがなくなるようになっている。ここまでの手順を 1 つの問題とし、被験者はこの手順を 4 問分繰り返す。ただし、「暗記の条件」は 4 問中 1 問目と 3 問目が「順番どおり」で残り 2 問は「順不同」である。また、各被験者はこの実験を 1 回のみ行う。さらに、ページ下部にはコメント欄として、その替え歌を選んだ理由などを自由に記述できる欄を用意した。

単語リストおよび替え歌

実験に用いた単語リストは形態素解析エンジン MeCab [19] の辞書中の一般名詞 60,477 語からランダムに 15 語選ぶことで作成した。各問題において 4 つの替え歌は同じ単語リストから生成されており、また、これらの替え歌はそれぞれのアルゴリズムで生成した替え歌のうち割当てコストが小さい上位 2 つのうちいずれかをランダムで使用した。さらに、図 8 のように生成された替え歌の元楽曲が同じ場合でも、特に表記は変更せず、被験者には提示されている替え歌の題名は同じでも内容は異なること伝えた。また、替え歌の音声データはプロトタイプシステムが生成した musicXML 形式のファイルから筆者が市販の音声合成ソフトウェア CeVIO [20] を用い作成した。

表 1 実験 1 の結果

Table 1 Result of test 1.

暗記の条件	無-譜割り	無-韻	有-譜割り	有-韻
順番どおり	34	22	11	12
順不同	10	11	21	24

表 2 表 1 の調整された残差

Table 2 Adjusted residual of Table 1.

暗記の条件	無-譜割り	無-韻	有-譜割り	有-韻
順番どおり	3.64**	1.60	-2.59**	-2.94**
順不同	-3.64**	-1.60	2.59**	2.94**

楽曲データベースおよび被験者

登録した楽曲データベースは文献 [21] に記載されている著作権切れの童謡から選んだ 43 曲である。これらの童謡は知名度が高く、学習者がよく知る曲として楽曲データベースに登録する可能性が高いと思われるものを選出した。ただし、被験者が知らない楽曲があればコメント欄に記述するように伝えた。さらに、本実験では各楽曲の 1 番の部分のみをデータベースに登録した。被験者は日本語を母国語とする 20 代から 50 代の男女 40 名である。

結果と考察

表 1 にその組合せが暗記しやすいと感じた人数の集計結果を提示する。表 1 の行は割当てアルゴリズムの組合せを表し、列は暗記の条件を表す。なお、集計において、「知らない楽曲があった」とコメントがあった回答は無効回答として集計からは除外している。ここで、4 つの割当てアルゴリズム「譜割りに基づいた割当てアルゴリズム」、「韻に基づいた割当てアルゴリズム」、「並べ替えをしない割当てアルゴリズム」、「並べ替えを行う割当てアルゴリズム」を便宜上それぞれ「譜割り」、「韻」、「並べ替え無」、「並べ替え有」と呼び、割当てアルゴリズムの組合せを「並べ替え無-譜割り」のように、ハイフンで結んで表記することとする。さらに、表 1 においては「並べ替え無」および「並べ替え有」をそれぞれ「無」および「有」とし、「無-譜割り」のように表記する。 χ^2 検定の結果、集計結果の数字の偏りは有意であった ($\chi^2_{(3)} = 22.90, p < .01$)。よって表 2 に残差分析の結果を示す。表 2 において「**」は統計的に有意かどうかを判断する p 値が $p < .01$ であることを表す。

表 2 の結果より、「順番どおり」での暗記の場合は「並べ替え無-譜割り」が最も好まれ、「並べ替え有-譜割り」および「並べ替え有-韻」は好まれないといえる。まず、並べ替え有のアルゴリズムが好まれないのは、順番どおりに暗記しなければならないのにもかかわらずリストの順番を入れ替えてしまうからである。並べ替え無のアルゴリズムのうち、「譜割り」が選ばれた理由としては、譜割りが自然だったという意見が多く、提案アルゴリズムによって生成される譜割りを自然に感じたため好まれやすいと考えられ

る。さらに、これは被験者の関心が「譜割り」にあったと
いい換えることもできる。理由として今回実験に用いた楽
曲が童謡ばかりであり、リズムやメロディに関してはよく
知っているものの、歌詞までは知らないということが考え
られる。よって今後はポピュラ音楽など被験者が歌詞を口
ずさめるほどよく知っている楽曲を用いて実験を行う必要
がある。また、韻に基づいた割当てアルゴリズムでも譜割
りの考慮はしているが、「譜割り」と「韻」間に差が出て
いる。これは、用いた楽曲で区切り間の譜割りが大きく変
化していないため、「譜割り」では替え歌の譜割りが大きく
変化せず、被験者にとって安定した譜割りであると感じ
させたことが要因として考えられる。一方で、「韻」では歌
詞や単語の韻によって譜割りが大きく左右されるため、
被験者にとって譜割りが不安定であると感じさせたの
ではないかと考えられる。

次に「順不同」での暗記の場合は「並べ替え有」の
場合が最も好まれる結果となっている。これは、並べ
替え有のアルゴリズムが、順番を入れ替えるだけでなく、
可能な限り単語を多く曲中に出すことが暗記のしやす
さを感じさせたと考えられる。また、被験者からのコ
メントとして、「順不同のときは楽曲間にあまり差を感
じない」というものがあり、「譜割り」と「韻」の間に
大きな差異が見られない結果の裏付けとなっている。
並べ替え有のアルゴリズムでは、順番に制約がない分、
アルゴリズムが単語の順番を最適になるように並べ
替えた結果、「譜割り」と「韻」の間には差がほぼな
くなったと考えられる。よって、「順不同」の場合
は「並べ替え有-譜割り」と「並べ替え有-韻」のど
ちらのアルゴリズムでも同等の替え歌が生成されると
いえる。

5.2 実験2

本実験では、提案システムによって生成された替え
歌が暗記に対してどのくらい有効であるかを評価する。

実験方法

被験者には15個の日本語の一般名詞がランダムに
並んだ単語リストを従来手法と提案手法の2種類の
手法で暗記してもらい、一定期間後に記憶できている
単語の数を計測した。各手法の詳細は以下のとおり
である。

従来手法 被験者に単語リストのみを渡し、視
覚情報のみで暗記してもらう手法

提案手法 被験者に単語リストと提案システム
が生成した替え歌の音源を渡し、単語リストと
替え歌両方を用いて暗記してもらう手法

ただし、どちらの手法においても紙に書いて暗
記することは禁止し、それ以外の声を出すこと
などについては制限しなかった。また、従来
手法と提案手法のどちらを先に行うかは被
験者ごとにランダムに行い、いずれの手法
でも被験者にはできる限り順番どおりに
暗記するよう指示した。加えて本実験
では単語リストを2種類用意し、被験者
は従来

手法と提案手法で異なるリストを暗記する。
どちらのリストをどちらの手法で暗記する
かは被験者によって異なるようにして
いる。ここで、実験に用いた替え歌の
生成アルゴリズムは実験1の順番ど
おりに暗記する場合で最も多く票
を集めた「並べ替え無-譜割り」で
あり、被験者に提示した替え歌は
プロトタイプシステムを用いて出
力した替え歌のうち、割当てコス
トが小さい5曲で、提案手法で暗
記する被験者には1曲を選んで暗
記するように伝えた。なお、音源
は実験1と同じ方法で作成した。

実験の手順は以下のとおりである。本
実験では、まず従来手法および提
案手法どちらか一方の手法で以
下の手順を行い、手順(2)終了後、
被験者の準備ができ次第もう一
方の手法での実験を開始した。

- (1) それぞれの手法での暗記方法で
単語リストを暗記できるまで暗
記してもらう。
- (2) 暗記終了から1日後、3日後、
7日後とインターバルを空け、
被験者に単語リストが見えない
状態で、憶えている単語を3分
間で記述してもらう。

手順(1)において、被験者が暗記
できたかどうかは基本的に自己
申告であるが、自己申告を受け
た後に、単語リストを見ない状
態で単語を読み上げてもらうこ
とで暗記できているかをチェッ
クした。加えて被験者が単語リ
ストを見ることができるのは、
この手順(1)のみとした。た
だし、この時点での記憶がその
場限りの記憶である短期記憶
であるか長期的に保持される
長期記憶であるかを判別する
ことはできないので、被験者
に対し、「1日後にテストを行
うこと」および「今後いっ
さい単語リストを見返すこと
はできない」の2点を伝え、
その上で被験者が暗記でき
たと確信できたときに自己申
告するよう指示した。また、
暗記開始からチェック終了ま
での時間も計測した。手順(2)
において、単語を記述する解
答用紙には解答欄を15個用意
し、その解答欄内に解答する
ようにした。インターバルは
暗記用のソフトウェアである
ANKI [22] をもとに設定した。
このソフトウェアでは、人間
の忘却特性に基づいて、暗記
すべき項目の出題や復習の
タイミングを管理しており、
特に難なく暗記できた際の
インターバルである1日後を
本実験でのインターバルとし
た。加えて、1日空けた3日
後、3日空けた後の7日後
でのデータも採取した。な
お、本実験において n 日後
の実験は暗記終了時から $24n$
時間以上 ($24n + 6$) 時間
以内の間に行うとする。

楽曲データベースおよび被験者

替え歌生成の際に使用した楽
曲データベースは実験1の時
に用いたものと同じ童謡43
曲である。被験者は全員日
本語を母国語とし、電気電
子工学を専攻する大学院生
および大学生の男女9名で
ある。被験者の年齢は22歳
から24歳である。

単語リスト

実験に用いた単語リストは
実験1と同様に形態素解析工

表 3 暗記に要した時間 [分]

Table 3 Required time for memorizing [min].

	従来手法	提案手法
被験者 1	5	8
被験者 2	4	14
被験者 3	5	19
被験者 4	6	19
被験者 5	11	16
被験者 6	14	27
被験者 7	8	5
被験者 8	9	15
被験者 9	11	9
平均	8.1	14.6
標準偏差	3.21	6.34

ンジン MeCab [19] の辞書中の一般名詞 60,477 語からランダムに 15 語選ぶことで作成した。被験者に渡すリストには漢字部分にはすべてルビをふってあり、記述する際にはひらがな、漢字など文字の種類は問わないと指示した。なお、用意したリストは以下の 2 種類である。

リスト 1 新鋭, 言葉, 西側, ケーブル, ストレッチ, お茶漬, 教習所, バイク, 空き缶, 小人, 裏山, 朝露, 爪, 悲報, 締切り

リスト 2 持ち物, 故意, 戦車, 初夏, 段違い, バーベキュー, 呪文, 為替, ゴミ捨て, つり銭, 星座, バスケット, 過半数, クッション, 旅路

結果と考察

まず、表 3 に、それぞれの手法において暗記にかかった時間を分単位で示す。下 2 行はそれぞれの手法での時間の平均と標準偏差を示す。また、表 3 中の白色の箇所はリスト 1、灰色の箇所はリスト 2 を暗記したことを示す。暗記に要した時間を手法とリストの 2 要因の分散分析を行ったところ、手法間にのみ有意差が認められた ($F_{(1,14)} = 7.71, p < .05$)。これは、提案手法は 1 曲に一定の長さがあるため、単語リストの反復に時間がかかってしまったことが原因と考えられる。

次に得られた解答を以下の 4 つの値に基づいて数値化する。

正答数 解答と解答欄の位置の両方が正解している数。どれだけ正確に暗記できていたかを表す。

列挙数 解答欄の位置に関係なく正解している数。順番に関係なくどれだけ単語を暗記できていたを表す。

連続正答数 ある正解した解答の直後の解答が正解している数。解答問の相対的な位置を正確に暗記できていたかどうかを表す。

位置誤答数 (列挙数) - (正答数) で求める。解答の位置間違いがどれだけあったかを表す。

4 種類の結果を表 4、表 5、表 6、表 7 に示す。表 4 から表 7 中の各列はそれぞれの手法で何日後に解答された結

表 4 正答数

Table 4 The number of correct answers.

	従来手法			提案手法		
	1 日後	3 日後	7 日後	1 日後	3 日後	7 日後
被験者 1	13	13	13	6	6	6
*被験者 2	3	3	3	15	15	15
被験者 3	15	15	15	10	11	9
*被験者 4	10	9	9	8	8	9
被験者 5	15	13	15	10	15	15
*被験者 6	14	14	14	15	15	15
被験者 7	14	14	10	14	14	14
*被験者 8	2	2	4	11	10	11
被験者 9	14	13	13	12	12	12
平均	11.1	10.7	10.7	11.2	11.8	11.7
標準偏差	4.82	4.64	4.29	2.94	3.12	3.08

表 5 列挙数

Table 5 The number of enumerations.

	従来手法			提案手法		
	1 日後	3 日後	7 日後	1 日後	3 日後	7 日後
被験者 1	13	13	13	10	10	10
*被験者 2	10	10	10	15	15	15
被験者 3	15	15	15	11	11	10
*被験者 4	12	12	12	10	10	10
被験者 5	15	15	15	14	15	15
*被験者 6	14	14	14	15	15	15
被験者 7	14	14	12	14	14	14
*被験者 8	8	8	8	11	10	11
被験者 9	14	14	14	12	12	12
平均	12.8	12.8	12.6	12.4	12.4	12.44
標準偏差	2.39	2.39	2.35	2.07	2.30	2.30

表 6 連続正答数

Table 6 The number of consecutive correct answers.

	従来手法			提案手法		
	1 日後	3 日後	7 日後	1 日後	3 日後	7 日後
被験者 1	11	11	11	6	6	6
*被験者 2	6	6	6	14	14	14
被験者 3	14	14	14	7	8	7
*被験者 4	8	9	9	8	7	7
被験者 5	14	13	14	11	14	14
*被験者 6	12	12	12	14	14	14
被験者 7	13	13	7	12	12	12
*被験者 8	2	1	1	9	7	9
被験者 9	12	11	11	8	8	8
平均	10.2	10.0	9.4	9.9	10.0	10.1
標準偏差	4.09	4.15	4.22	2.98	3.43	3.37

果かを示し、表の下段 2 行はそれぞれの日数での平均、標準偏差を示している。また、被験者列の「*」が付与されている被験者は先に従来手法で実験を行い、後に提案手法で実験を行った被験者で、「*」がない被験者は先に提案手法で実験を行い、後に従来手法で実験を行った被験者であることを表す。

まず、表 4 から表 7 に対して、手法、リスト、日数の 3 要因で分散分析を行ったところ、どの要因にも有意差は認められなかった。よって、単語リストを暗記する場合、どちらの手法を用いても同様の効果が得られることが分かった。以降各被験者ごとの結果について考察する。

表 4 より、提案手法の方に効果があったと認められた被

表 7 位置誤答数

Table 7 The number of wrong answer positions.

	従来手法			提案手法		
	1日後	3日後	7日後	1日後	3日後	7日後
被験者 1	0	0	0	4	4	4
*被験者 2	7	7	7	0	0	0
被験者 3	0	0	0	1	0	1
*被験者 4	2	3	3	2	2	1
被験者 5	0	2	0	4	0	0
*被験者 6	0	0	0	0	0	0
被験者 7	0	0	2	0	0	0
*被験者 8	6	6	4	0	0	0
被験者 9	0	1	1	0	0	0
平均	1.7	2.1	1.9	1.2	0.7	0.7
標準偏差	2.67	2.56	2.28	1.62	1.33	1.25

験者は被験者 2, 被験者 6, 被験者 7, および被験者 8 である。特に被験者 2 と被験者 8 は提案手法により暗記できていた数が 9 から 12 個ほど増えている。この 2 名の被験者は従来手法では単語リストを頭に詰め込むいわゆる「丸暗記」で暗記をしていたと述べている。よって、単語のほとんどが短期記憶にしかとどまらず、従来手法での結果が悪くなり、逆に提案手法では、歌のメロディと単語を結び付けたことで、長期記憶にとどまっている単語の量が増え、結果が改善された。また、これらの被験者は「メロディが想起の助けになった」とコメントしており、提案手法の効果が最もよく表れた被験者であるといえる。

次に従来手法の方に効果があったと認められた被験者は表 4 より、被験者 1, 被験者 3, および被験者 9 である。特に値の差が大きいのは被験者 1 および被験者 3 であるが、これらの被験者は「暗記に使った歌自体を忘れてしまった」「従来手法の方が歌がない分能動的に暗記しようとした」と述べており暗記に対するモチベーションの違いが結果に影響したと考えられる。さらに被験者 1 および被験者 3 は従来手法でリスト 1, 提案手法でリスト 2 を暗記しているが、「リスト 1 の方がストーリーを想像しやすく暗記しやすかった」というコメントもしている。よって単語リストから受ける被験者の主観的なイメージが結果に影響を及ぼすことが考えられる。ただし、前述のようにリスト間に有意差は認められていないので、今回の実験ではリストの違いによって結果が大きく左右されてはいないといえる。

最後に、表 4 よりどちらの手法でも結果が同じだった被験者は被験者 4 および被験者 5 の 2 名である。この 2 名は従来手法での暗記の際に「ストーリーを作って暗記した」と述べており、自身で確立された暗記法を実践した結果、両者に差がでなかったと考えられる。さらに、被験者 2 および被験者 8 を除くすべての被験者が同様の手法を用いて従来手法での暗記を行っていたと述べており、被験者自身で確立された暗記法を実践した被験者はほぼ 10 個以上の単語とその位置を正確に暗記できていたといえる。よって、今後は単語リストからストーリーが想像できそうな単語リストを用いて効果を調べる必要がある。例とし

て、被験者らになじみのない分野の専門用語（医学用語、法律用語、生物学用語など）を使うことがあげられる。

ここまでの考察により、被験者自ら確立された暗記法を実践した場合、従来手法と提案手法での差はなく、逆に従来手法を丸暗記で暗記した場合、提案手法の方が暗記できる数が多い可能性が示唆された。同様の結果が列挙数（表 5）および連続正答数（表 6）についても起こっており、同じ理由によるものと考えられる。

次に位置誤答数に関して考察する。これは（列挙数）－（正答数）で求められる値で、値が小さいほど、暗記していた単語のほとんどについて正確にその位置を暗記していたことになる。被験者ごとの傾向は他の評価値における傾向とはほぼ同じであるが、他の評価値において「従来手法の方に効果が認められた」もしくは「どちらでも同じ」と分類された被験者 4 および被験者 9 が位置誤答数では提案手法に効果があると認められる。よって、これらの被験者は暗記できている単語の絶対数は従来手法の方が多く、そのうち位置まで正確に暗記できている数は提案手法の方が多いといえる。被験者 4 からは、「単語が直接思い出せなくても歌の情報から、何文字空いているかを推測することができたので、正確な場所に解答することができた」というコメントが得られており、提案手法の効果を裏付けるものであるといえる。

最後に、暗記を行った順序が結果に与えた影響について考察する。表 4 から表 7 の結果に関して、暗記した順序、リスト、日数の 3 要因で分散分析を行ったところ、表 4 から表 7 のすべてにおいて「暗記した順序」に関してのみ有意差が認められた（表 4：($F_{(1,14)} = 6.06, p < .05$), 表 5：($F_{(1,14)} = 3.40, p < .1$), 表 6：($F_{(1,14)} = 5.51, p < .05$), 表 7：($F_{(1,14)} = 4.45, p < .1$)). また、1 回目から 2 回目へのそれぞれの場合における平均値は表 4 で 9.4 から 12.7, 表 5 で 11.7 から 13.4, 表 6 で 8.3 から 11.6, 表 7 で 2.3 から 0.41 となっており、1 回目より 2 回目の方に効果が認められる結果となった。原因として、被験者らが本実験の「日本語の単語 15 個をできるだけ順番どおりに暗記する」という課題を行うことに慣れたことが考えられる。

この評価結果より提案手法により生成された替え歌は丸暗記に比べて効果があることが示唆され、また、暗記できた単語の総数に対して単語の位置が正確である割合が高いことも示唆された。

6. まとめ

本研究では暗記学習のための替え歌自動生成システムの設計と実装を行った。提案システムは楽曲中の歌詞の区切りごとに、歌詞の譜割りまたは韻を考慮して単語を割り当てることで、替え歌生成を行っている。さらに、リストの順番を入れ替えるアルゴリズムと入れ替えないアルゴリズムの 2 種類を提案することで、さまざまな暗記対象に対応

できるようになっている。提案システムの評価として2種類の実験を行った。実験1では単語リストを順番どおりに暗記するかどうかによって好みのアルゴリズムが変わることが分かった。実験2では、暗記したランダムな単語の羅列を一定期間後にどれだけ暗記できているかを評価し、提案手法により生成された替え歌は丸暗記に比べて効果があることが示唆された。

今後の課題としては、専門用語のような被験者が完全に新規に暗記する単語リストでの実験を行うことがあげられる。また、現在は1つの区切りに対し1つの単語を割り当てる手法を提案しているが、分割単位数などによっては複数の区切りにまたがって単語を割り当てるなどの手法も検討する必要がある。さらに、英単語とその日本語訳などのように2つで1つのペアになった単語リストなどへの対応方法も検討する必要がある。

謝辞 本研究の一部は、科学技術振興機構戦略的創造研究推進事業（さきがけ）および文部科学省科学研究費補助金基盤研究（A）(23240010)によるものである。ここに記して謝意を表す。

参考文献

- [1] The Learning Center Exchange, available from <http://www.learningassistance.com/2006/january/mnemonics.html>.
- [2] Goldish, M.: Memory-Boosting Mnemonic Songs for Content Area Learning, *Scholastic Teaching Resources* (2006).
- [3] 替え歌で憶える中学受験理科, 入手先 <http://jet-stram.cocolog-nifty.com/blog/>.
- [4] 茂木広貴: 究極の理論暗記法! 税理士試験—替え歌暗記法, Amazon Services International, Inc. (2013).
- [5] Crowther, G.: Using Science Songs to Enhance Learning: An Interdisciplinary Approach, *CBE-Life Sciences Education*, Vol.11, No.1, pp.26–30 (2012).
- [6] Hale-Evans, R.: Mind Performance Hacks: Tips & Tools for Overclocking Your Brain, O'Reilly Media (2006).
- [7] Peesapati, S.T., Schwanda, V., Schultz, J., Lepage, M., Jeong, S. and Cosley, D.: Pensieve: Supporting Everyday Reminiscence, *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI2010)*, pp.2027–2036 (2010).
- [8] Tsai, W., Lee, H., Hsiao, J.C., Liang, R. and Hsu, J.Y.: Framing Design of Reminiscence Aids with Transactive Memory Theory, *Extended Abstracts on Human Factors in Computing Systems (CHI2013)*, pp.331–336 (2013).
- [9] 岡安優弥, 高田雅倫, 渡辺邦浩, 濱川 礼: 品詞による文評価を用いた日本語語呂自動生成手法, 全国大会講演論文集, Vol.72, No.2, pp.511–512 (2010).
- [10] 青木賢太郎: コスト最小法を用いた言葉遊び—数字語呂合わせの自動生成システム, ことば工学研究会, Vol.10, pp.31–35 (2002).
- [11] Snyder, R.: Music and Memory, The MIT Press (2001).
- [12] 深山 覚, 中妻 啓, 酒向慎司, 西本卓也, 小野順貴, 嵯峨山茂樹: 音楽要素の分解再構成に基づく日本語歌詞からの旋律自動作曲, 情報処理学会論文誌, Vol.54, No.5, pp.1709–1720 (2013).
- [13] Bourgeois, F. and Lassalle, J.C.: An Extension of the

Munkres Algorithm for the Assignment Problem to Rectangular Matrices, *Comm. ACM*, Vol.14, No.12, pp.802–804 (1971).

- [14] International Phonetic Association: Handbook of the International Phonetic Association: A Guide to the use of the International Phonetic Alphabet, Cambridge University Press (1999).
- [15] Bellman, R.E.: Dynamic Programming, Princeton University Press (1957).
- [16] MusicXML ホームページ, 入手先 <http://www.musicxml.com/>.
- [17] 工藤 拓, 松本裕治: チャンキングの段階適用による日本語係り受け解析, 情報処理学会論文誌, Vol.43, No.6, pp.1834–1842 (2002).
- [18] The Official Microsoft ASP.NET Site, available from <http://www.asp.net/>.
- [19] MeCab, available from <http://mecab.sourceforge.net/>.
- [20] CEVIO Official Site, available from <http://cevio.jp/>.
- [21] 童謡・唱歌の世界, 入手先 <http://www5b.biglobe.ne.jp/~pst/douyou-syouka/index.htm>.
- [22] Anki ホームページ, 入手先 <http://ankisrs.net/>.



伊藤 悠真 (学生会員)

1989年生。2011年神戸大学工学部電気電子工学卒業。2013年同大学院工学研究科電気電子工学専攻修士課程後期課程修了。現在同大学院工学研究科電気電子工学専攻博士課程後期課程に在籍。ウェアラブル・ユビキタス・コンピューティング、音楽情報科学に興味を持つ。



寺田 努 (正会員)

1974年生。1997年大阪大学工学部情報システム工学科卒業。1999年同大学院工学研究科博士前期課程修了。2000年同大学院工学研究科博士後期課程退学。同年大阪大学サイバーメディアセンター助手。2005年同講師。2007年神戸大学大学院工学研究科准教授。現在に至る。2004年より特定非営利活動法人ウェアラブルコンピュータ研究開発機構理事, 2005年には同機構事務局長を兼務。工学博士。アクティブデータベース, ウェアラブルコンピューティング, ユビキタスコンピューティングの研究に従事。IEEE, 電子情報通信学会, 日本データベース学会, ヒューマンインタフェース学会の各会員。



塚本 昌彦 (正会員)

1964年生。1987年京都大学工学部数理工学科卒業。1989年同大学院工学研究科修士課程修了。同年シャープ(株)入社。1995年大阪大学大学院工学研究科情報システム工学専攻講師。

1996年同専攻助教授。2002年同大学院情報科学研究科マルチメディア工学専攻助教授。2004年神戸大学電気電子工学科教授。現在に至る。2004年より特定非営利活動法人ウェアラブルコンピュータ研究開発機構理事長を兼務。工学博士。ウェアラブルコンピューティングとユビキタスコンピューティングの研究に従事。ACM, IEEE等, 8学会の会員。