

音声対話コンテンツの利用履歴における頻出パターン解析手法の検討

田中佑太郎[†] 堤 修平[†] 山本 大介[†] 高橋 直久[†]

[†] 名古屋工業大学大学院工学研究科情報工学専攻 〒466-8555 愛知県名古屋市昭和区御器所町

E-mail: tyutaro@moss.elcom.nitech.ac.jp, tsutsumi.shuhei,yamamoto.daisuke,naohisa}@nitech.ac.jp

あらまし 本学では音声対話システムの研究・開発を促進するため MMDAgent と呼ばれる音声対話システム構築ツールキットを開発・公開している。MMDAgent では、状態遷移形式で記述される対話スクリプトを編集することで対話内容を編集することができる。また、そのスクリプトの記述に沿った形式で対話情報を保存する状態遷移履歴の収集を行っている。MMDAgent における音声対話は様々なカテゴリの対話が存在しているが、音声対話コンテンツの作成者は、実際にコンテンツがどのように利用されているのかわからない状態であった。ユーザが音声対話コンテンツを利用する上で、どのような行動パターンを取るのかを知ることで、より利用しやすい対話コンテンツを作成するのに役立つのではないかと考えた。そこで Longest Common Subsequence(LCS) を用いて頻出パターンを抽出するアプローチを行う。LCS を用いることで、頻出単語だけでなく、ユーザがどのような文脈で対話しているのかという情報を保つことができる。本研究の目的は、音声対話における頻出パターンを抽出し、音声対話のユーザがどのような意図をもって利用しているのかを調べることで、対話スクリプトの作成者の支援を行うことである。

キーワード MMDAgent, 音声対話, 状態遷移, ログ, LCS, パターン抽出

1. はじめに

1.1 本研究の概要

Google 社のビッグデータの活用などをはじめとした、ログを活用する取り組みが一般的となっている。さらに、Web アクセスログやライフログを解析し、マーケティングやシステムの改善等に利用する取り組みも行われている。本学では音声インタラクションシステム構築ツールキット MMDAgent [1] [2] [3] [4] [5] [6] の研究・開発を行っている。また、本研究室では MMDAgent を応用し、音声対話と Web サービス等との連携に関する研究を行っている [7] [8] [9]。

MMDAgent における音声対話は様々な種類のコンテンツが存在しているが、音声対話コンテンツの作成者は、MMDAgent のコンテンツを利用する際のユーザの行動パターンがわからない状態であった。ユーザが音声対話コンテンツを利用する上で、どのような行動パターンを取るのかを知ることで、より利用しやすい対話コンテンツを作成するのに役立つのではないかと考えた。

そのアプローチとして、本研究で扱う MMDAgent では、状態遷移履歴と内部メッセージ履歴という形式で、音声対話の利用履歴（以下、音声対話履歴と呼ぶ）を保存している [10]。さらに本研究では、音声対話履歴から Longest Common Subsequence(LCS) [11] を用いて頻出パターンを抽出するアプローチを行う。MMDAgent の音声対話履歴には、状態遷移履歴と内部メッセージ履歴の 2 つが含まれている。状態遷移履歴とは MMDAgent における状態遷移のリストのことであり、内部メッセージ履歴とは MMDAgent のイベントメッセージとコマンドメッセージのリストのことである。特に、本論文では、一連の時間的に連続する状態遷移履歴のまとまりをセッションと

呼ぶ。1 つのセッションには、音声対話システムとユーザの複数の連続する対話が含まれていることを想定しており、状態遷移履歴は複数のセッションから構成される。

LCS を用いることで、頻出単語だけでなく、ユーザがどのような文脈で対話しているのかという情報を保つことができる。また、本研究ではコンテンツ投稿型音声案内デジタルサイネージにおける音声対話履歴について頻出パターンの考察を行う。本研究の目的は、音声対話システムにおける頻出パターンを抽出し、音声対話のユーザがどのような意図をもって利用しているのかを調べることで、対話スクリプトの作成者の支援を行うことである。

1.2 先行研究

音声インタラクションに基づくユーザの嗜好抽出

文献 [12] では、Android 版 MMDAgent [13] [14] におけるユーザの嗜好を抽出する手法を提案している。ここでは、明示的にユーザから情報を収集するアプローチと、音声対話コンテンツにタグ付けによる興味分野との関連付けを行い、対話履歴から暗黙的に情報を収集するアプローチをとっている。本研究では LCS を用いて、前後関係情報を用いた嗜好の抽出を行っている。また、LCS を用いることによって、音声対話コンテンツ自体の情報を必要としない点が文献 [12] と異なっている。

2. 音声インタラクションシステム構築ツールキット MMDAgent

2.1 MMDAgent の概要

MMDAgent はオープンソースツールキットであり、FST スクリプトという対話スクリプトを編集することで音声対話を構築することができる。音声認識に Julius [15]、音声合成に Open JTalk [16] を用い、図 1 のように 3D キャラクターとの対話形

式をとることで、よりインタラクティブ性の高い対話を実現している。

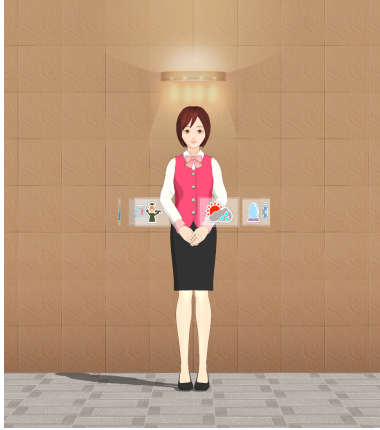


図 1 MMDAgent の画面表示

FST スクリプトは状態遷移をテキスト形式で記述している。現状の状態番号と遷移先の状態番号、状態遷移条件、状態遷移時に実行されるコマンドの 4 つの組で構成されている。図 2 のように、この 4 つの要素を 1 行に空白かタブ文字で区切って記述することで対話の状態遷移を表す。MMDAgent では内部メッセージが発行されており、イベントキューを介した内部メッセージのやり取りによって対話管理が行われている。イベントメッセージは状態遷移条件を表し、コマンドメッセージは実行されるコマンドを表す。ユーザの音声入力から遷移条件となる音声認識キーワードを認識したとき、対話管理部が音声認識によるイベントメッセージを受け取り、FST スクリプトを基に、図 3 のように状態が遷移する。同時にコマンドメッセージが発行され、コマンドを実行する。このように MMDAgent において FST スクリプトと状態遷移、内部メッセージには対応関係がある。MMDAgent の対話スクリプトは一问一答形式で記述されることが多いが、複雑な対話を記述することもできる。本研究における一问一答形式とは、

1. 待機状態でユーザの入力を待つ。
 2. 音声入力によって状態が遷移し、応答発話を返す。
 3. 待機状態に戻り、再びユーザの入力を待つ。
- という対話形式を指している。

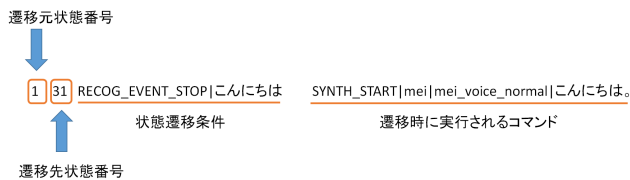


図 2 FST スクリプトの記述例

0031-0040 Hello

```
1 31 RECOG_EVENT_STOP|こんにちは SYNTH_START|mei|mei_voice_normal|こんにちは。
31 32 <eps> MOTION_ADD|mei|action|MotionXmei_greetingXmei_greeting.vmd|PART|ONCE
32 1 SYNTH_EVENT_STOP|mei <eps>
```

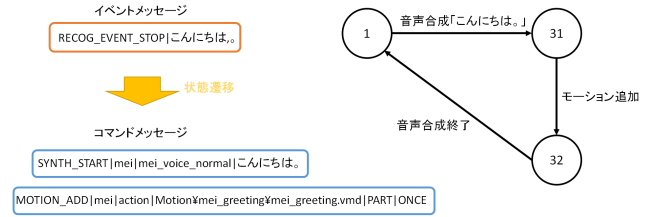


図 3 「こんにちは」という単語を認識した場合

2.2 MMDAgent の音声対話履歴

本研究で扱う MMDAgent では、状態遷移履歴と内部メッセージ履歴を FST スクリプトに対応付けて保存している。状態遷移履歴は、MMDAgent の音声対話における状態遷移が記録される。内部メッセージ履歴は MMDAgent の入出力が記録される。

MMDAgent の対話を構成する状態遷移はイベントメッセージを入力、コマンドメッセージを出力とするため、状態遷移履歴と内部メッセージ履歴には対応する関係がある。しかし、MMDAgent は音声認識や音声合成などが並列に動作するため、同時刻に複数の履歴が発生することがある。そのため、時刻だけで 2 つの履歴を対応させることはできない。そこで、これら 2 つの履歴に対応する実行番号を付加することで、状態遷移と内部メッセージの対応付けを行う。また、状態遷移履歴には、FST スクリプトの記述位置を表す、行番号を付加する。行番号を付加することで FST スクリプトと状態遷移履歴の対応付けを行う。

本研究では、状態遷移履歴を参照することで、MMDAgent のユーザの利用履歴を抽出する。

2.2.1 状態遷移履歴

状態遷移履歴は MMDAgent の対話管理部で実行された遷移を出力したものである。対話管理部では、FST スクリプトを読み込み、記述されている状態遷移を構造体の状態遷移データに格納する。そして格納されたデータを基に、音声対話の状態遷移を処理している。対話管理部で扱うデータを表 1 に示す。履歴生成機能では、これらのデータを取り出すことで状態遷移履歴を構成する。

これらの状態遷移データを、FST スクリプトの記述形式と同様の形で取り出すことで、FST スクリプトと状態遷移履歴の対応付けを行う。状態遷移データを、外部にテキストデータとして出力することで状態遷移履歴を生成する。生成する際に時刻、実行番号、行番号を付加する。

状態遷移履歴の内容は時刻、実行番号、行番号、FST スクリプトに記述されている状態遷移で構成される。状態遷移履歴における状態遷移は FST スクリプトと同様の記述形式で表現する。状態遷移履歴に記述するデータを表 2 に示す。

表 1 FST スクリプトに記述されている状態遷移の構成

データ	説明
m_currentState.number	遷移前状態番号
next_state.number	遷移先状態番号
input_event.type	遷移条件
input_event.args	音声認識結果
output_command.type	コマンド
output_command.args	出力音声内容

表 2 状態遷移履歴の構成

データ	説明
時刻	状態遷移が発生した時刻
実行番号	状態遷移履歴と内部メッセージ履歴に共通する
行番号	FST スクリプトに記述されている位置
状態遷移	FST スクリプトに記述されている状態遷移

3. 提案手法

3.1 提案手法の概要

本研究の手法は、Web アクセス履歴の解析手法[11]を参考に、前述の状態遷移履歴に対して頻出パターンを求めるものである。文献[11]では、Web サイトのアクセス履歴を解析することで、ユーザのアクセスパターンを知り、Web サイトの構成をより好ましくすることを目指している。本研究においては、ユーザのパターンを分析し、より好ましい音声対話とはどういったものなのかを考察するために、LCS を用いた解析手法を行う。

提案手法の特徴は以下の通りである。

特徴 1 音声対話履歴からセッションを抽出する機能を有する。その際、セッションはそのセッション中に現れる音声認識結果の系列によって表現する。

特徴 2 全てのセッションの組み合わせに対して LCS を計算することにより、音声対話履歴における頻出パターンを抽出する。

特徴 3 音声対話履歴における頻出パターンの分析を支援するため、出現回数と構成要素でソートした結果を表形式で提示可能にする。

MMDAgent の音声対話履歴には、状態遷移履歴と内部メッセージ履歴の 2 つが含まれている。状態遷移履歴とは MMDAgent における状態遷移のリストのことであり、内部メッセージ履歴とは MMDAgent のイベントメッセージとコマンドメッセージのリストのことである。特に、本論文では、一連の時間的に連続する状態遷移履歴のまとまりをセッションと呼ぶ。1 つのセッションには、音声対話システムとユーザの複数回の連続する対話が含まれていることを想定しており、状態遷移履歴は複数のセッションから構成される。

3.2 Longest Common Subsequence

文献[11]において、「リスト x の部分系列 x_a とリスト y の部分系列 y_b の中で両方のリストに共通に含まれるものを共通部分系列という。共通部分系列の中でもっとも長いものを最大共通部分系列 (Longest Common Subsequence) と呼び、頭文字をとって LCS と略する。2 つのリストの中に同じ要素が同じ順

序で出現したものが共通部分系列なので、LCS が長いということは、2 つのリストの類似性が高いと解釈できる。」(宇根田, 2002, p.11) と説明されている。

このように、最長の部分系列を求めることで類似性の高い系列を求めることができる。つまり、LCS が十分に長ければそれが頻出のパターンを表していると解釈できる。本研究では、それらを集計することによって音声対話の利用履歴を分析する。

図 4 に、LCS の計算例を示す。

$$\begin{aligned} X &= (A, B, C, D, E) & X &= (A, B, C, D, E) \\ Y &= (A, D, C, E, F) & Y &= (A, D, C, E, F) \\ \text{LCS}(X, Y) &= (A, C, E), (A, D, E) \end{aligned}$$

図 4 LCS の計算例

LCS の特徴として、実際に (A, C, E) や (A, D, E) という系列は X, Y に連続して存在していないという点がある。つまり、間に異なる要素が入っていたとしても類似性の高い共通シーケンスとして抽出することができる。

文献[11]では、WEB アクセスログの解析における、通常のアクセスルートの統計と部分ルートの統計を、LCS による統計と比較している。ここでは、類似性が高いルートを抽出する際に、ルートの類似性が高いにも関わらず、異なるルートとして集計されてしまう問題点を指摘している。また、その解決方法として、LCS を用いた集計を提案している。

この問題点は、音声対話において頻出パターンを抽出する際にも、検討すべきものであると考えられる。そこで、本研究では、この文献に倣い、LCS を用いて分析を行うこととした。これによって、多少の差異を容認しつつ、類似性の高いシーケンスを抽出することが可能であると思われる。

3.3 提案手法の構成

提案手法の構成図は図 5 のようになる。

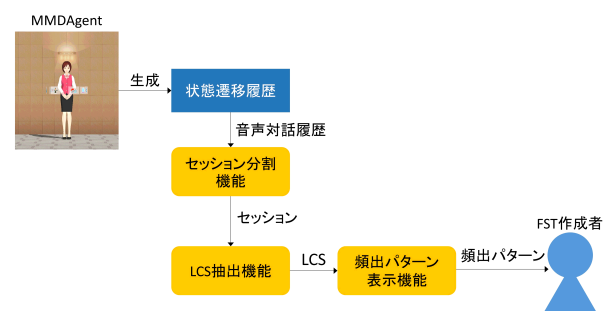


図 5 システム構成図

図 5 より、提案手法の流れは以下のようになる。

1. 状態遷移履歴の前処理 MMDAgent が生成した状態遷移履歴に対して、LCS を計算するための前処理を行う。時間間隔を基に状態遷移履歴をセッションに分割し、セッションにおける音声認識結果の系列を抽出する。

2.LCS の計算 LCS を求めるアルゴリズムを用いて、全てのセッションの組み合わせで LCS を計算する。

3. 頻出パターンの表示 計算した LCS の出現回数を集計し、出現回数順にソートした結果を表形式で表示する。また、前後関係を参照しやすくするため、前後関係順にソートした結果を並べて表示する。

それぞれを、1. セッション分割機能、2.LCS 抽出機能、3. 頻出パターン表示機能とし、説明する。

3.3.1 セッション分割機能

この機能では LCS を計算する前処理として、状態遷移履歴から同一ユーザが行ったと思われる、一連の時間的に連続する状態遷移の系列を抽出する。本研究では、これをセッションと呼称する。それぞれのセッションにおける状態遷移履歴から音声認識結果を抽出したのち、セッションを LCS 抽出機能に渡す。

3.4 セッション

本研究におけるセッションは、状態遷移履歴における一連の時間的に連続する状態遷移の系列を時間間隔に従って分割したものである。セッションは以下の要素で構成される。

表 3 セッションの構成

データ	説明
セッション ID	セッションに付加する固有の ID
状態遷移	状態遷移履歴における状態遷移の行

3.5 セッションの分割

本研究では、同一のユーザは連続して対話を行うものと考え、音声入力の時間間隔が指定した数値を超えない限り、同一のユーザであると解釈する。これは、音声対話を利用するユーザは MMDAgent の応答発話を踏まえたうえで、次の対話を行うと仮定しているからである。

一問一答形式の対話の終了は MMDAgent の発話が終了した時点であると判断することができる。「SYNTH.EVENT.STOP」は、MMDAgent の発話が終了したことを示しており、発話終了時に発行される。つまり、「SYNTH.EVENT.STOP」というイベントメッセージが発行された時点が一問一答形式の対話の終了を示していると解釈することができる。よって、提案手法では、「SYNTH.EVENT.STOP」が発行された時点から、「RECOG.EVENT.STOP」が発行されるまでの時間が N を超えた場合セッション ID を切り替える、という手順をとる。

状態遷移履歴からセッションを抽出する手順は以下のようになる。状態遷移履歴の各状態遷移 L に対して、以下の手順を適用する。状態遷移は発生時刻で昇順になっているものとする。ただし、ここでの時間間隔 N は MMDAgent の応答発話が終了した時間から、次にユーザが音声入力を行うまでの間隔とみなしている。

- 1 状態遷移 L に、セッション ID として S を付与する。
- 2 状態遷移 L の遷移条件が SYNTH.EVENT.STOP のときの、状態遷移 L の時刻を T とする。
- 3 状態遷移 L' の遷移条件が RECOG.EVENT.STOP のとき、状態遷移 L' の時刻を T' とする。

4-1 $T' - T \leq N$ のとき、状態遷移 L' にセッション ID として S を付与する。

4-2 $T' - T > N$ のとき、状態遷移 L' にセッション ID として S' を付与する。

5 手順 2-4 を繰り返す。

以上のようにして、状態遷移履歴の各状態遷移をセッションに分割する。

3.6 音声認識結果の抽出

音声認識結果とは、MMDAgent の状態遷移における音声入力イベントの音声認識結果の文字列のことである。ユーザが MMDAgent に「こんにちは」と話しかけると、「RECOG.EVENT.STOP| こんにちは」というイベントメッセージが発行され、FST スクリプトに記述に従い状態遷移が発生する。つまり、ユーザの行動を表している部分は、音声入力によるイベントメッセージによって状態遷移が発生した部分であると考えられる。

よって、状態遷移履歴の音声認識を遷移条件としている行から、遷移条件となっている音声認識結果の文字列を取り出すことで、MMDAgent のユーザの音声入力結果を抽出する。

例えば、図 6 のように、「こんにちは」「今日の天気は?」「バイバイ」という順番で音声対話を行ったセッションにおける状態遷移履歴があったとき、「こんにちは」、「今日、天気」、「バイバイ」という系列が抽出される。

セッション	
2016/8/25 12:59:18 [88361] line:4252	1 290021 RECOG_EVENT_STOP こんにちは<eps>
2016/8/25 12:59:26 [88456] line:4953	1 600041 RECOG_EVENT_STOP さよなら<eps>
2016/8/25 12:59:31 [88510] line:4974	1 600051 RECOG_EVENT_STOP バイバイ<eps>



(こんにちは), (さよなら), (バイバイ)

図 6 セッションから音声認識結果を抽出する。

上記の手順で抽出した音声認識結果のリストを、セッション ID と紐づけて LCS 抽出機能に渡す。

3.6.1 LCS 抽出機能

この機能では、セッション分割機能で取り出した音声認識結果のリストにおいて、異なるセッション ID を持つリストの全ての組み合わせで LCS を計算する。LCS を計算するアルゴリズムの概要は以下ようになる。ここでの系列 X, Y は異なるセッション ID を持つ音声認識結果のリストを引数として与える。

```
LCS(X,Y){
  array L, S
  for(i=0 to X.size){
    for(j=0 to Y.size){
      if X[i]=Y[j] then L[i+1,j+1] = L[i,j]+1;
```



```

else L[i+1,j+1] = max(L[i+1,j], L[i,j+1]);}}
for(m=X.size, n=Y.size to m!=0 && n!=0){
if L[m,n]=L[m,n-1] then n--;
else if L[m,n]=L[m-1,n] then m--;
else if X[m-1]=Y[n-1] then S.add(X[m-1]);
m--;
n--; }
return S; }

```

L は LCS 長を格納する 2 次元配列であり, S は構成した LCS を格納する配列である. 系列 X と Y を入れ替えと, 同じ長さだが要素が異なる LCS が存在する場合, それを求められる. 計算した LCS を格納したリストを, 頻出パターン表示機能に渡す.

3.6.2 頻出パターン表示機能

この機能では音声対話履歴における頻出パターンの分析を支援するため, LCS の抽出結果を基に頻出パターンを表形式で表示する. それぞれの LCS の出現回数から, LCS 同士で共通部分系列を求め, 出現回数を拡張する. 求めた共通部分系列を出現回数順に表示する. また, LCS を構成する系列の構成要素に関して前後関係順に並べた結果を合わせて表示する.

3.7 出現回数の集計の拡張

LCS 同士で共通部分系列を求め, それを基に出現回数を拡張してソートした結果を表示する.

LCS の抽出結果を集計し, それぞれの LCS について出現回数を得点として与える. 計算した LCS を集計する際, ある LCS_A が他の LCS_B と包含関係になっている場合に, LCS_A の出現回数が LCS_B より小さくなる場合があった. この結果は LCS が最長の共通部分を表しているためであるが, LCS_B が出現しているときは LCS_A も出現しているとみなすことができると考えられる. また, LCS を計算する際に系列を構成する要素の種類が多くなるほど, 共通部分が抽出されにくくなり, LCS の出現回数が少なくなりやすくなる. そこで, LCS_A が LCS_B の部分集合であり, LCS_A の構成要素の順序と LCS_B の構成要素の順序が等しいとき, LCS_B の出現回数を LCS_A の出現回数に加えることで, 頻出パターンの出現回数を補正する. 得点の補正方法のアルゴリズムを以下に示す. LCS とその出現回数の組のリストに対して以下の手順を行う.

- 1 LCS_A の先頭の要素が LCS_B に含まれているとき, LCS_B におけるその要素の位置を R_B とする.
- 2 LCS_A の次の要素が, LCS_B の R_B の後ろにあるとき, LCS_B におけるその要素の位置を R_B とする.
- 3 手順 2 を繰り返し, LCS_A の最後の要素に関して手順 2 が行われたとき, LCS_A は LCS_B に含まれているとする.
- 4 LCS_B の得点を LCS_A の得点に加え, LCS_A とその得点を表示用リストに格納する.

図 7 に例を示す. LCS において, {A,B,C} の系列が {A,B,C,D} に含まれているため, {A,B,C,D} の出現回数を {A,B,C} の出現回数に加える. この手順によって, 頻出するパターンを表示

する際に直観的な結果となる.

LCS	出現数
A->B->C	10
A->B->C->D	20



部分系列	出現数
A->B->C	30
A->B->C->D	20

図 7 出現回数を補正する.

3.8 前後関係順の表示

抽出結果を, LCS の構成要素に関して前後関係順で結果のソートを行う. これは, 音声対話履歴における音声認識結果の前後関係を分析するうえで, 類似した LCS の出現回数を比較することが重要であると考えられるためである.

LCS を構成する要素の系列に対して, ソートの処理を行う. ソートの処理は Java 言語の自然順序付けによって行う. 自然順序付けとは, Java の Comparable クラスの compareTo メソッドによる順序付けのことである.

この並び替えによって, ある要素から特定のパターンが頻出しているとき, それを発見することが可能となる. 一例として, 図 8 を示す. $LCS=\{A\}$ の下に {A,B} と {A,C} が表示されており, A からは C よりも B の方が出現しやすいという結果が読み取れる.

出現数	LCS
10	A
8	A->B
2	A->C
20	B
5	B->A
15	B->C

図 8 前後関係順に結果をソートする.

4. プロトタイプシステム

4.1 プロトタイプシステムの概要

プロトタイプシステムは, Java 言語を用いて実装した. セッション分割機能, LCS 抽出機能, 頻出パターン表示機能を持つプログラムで構成されている.

4.2 セッション分割機能の実装

プロトタイプシステムでは, まず状態遷移履歴のそれぞれの行に, セッション ID を付加する. セッション ID を付加した状態遷移履歴の行を ID ごとに分割し, そこから音声認識結果を抽出する.

本研究では予備調査として, 投稿型音声案内デジタルサイネージ「ゆめルームメイちゃん」の状態遷移履歴において, 一問一答形式の対話間の時間間隔を調べた. (図 9)

この調査によって, 約 80% で 1 分以内に次の音声対話が行われることがわかった. 1 分を超えている場合は別の話題に移つ

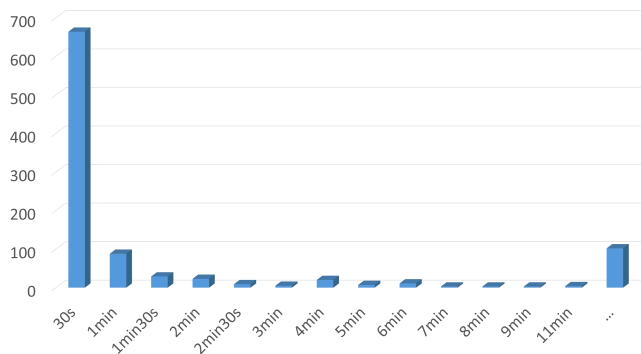


図 9 一問一答形式の対話における時間間隔

たと判断し、セッションを切り替えるのが妥当であると考えた。そのため、プロトタイプシステムではセッションを切り替える時間間隔を 1 分とする。

4.3 LCS 抽出機能と頻出パターン表示機能の実装

LCS 抽出機能では、3 章で示したアルゴリズムを用いて、LCS を計算する。計算した LCS は頻出パターン表示機能に与えられる。

頻出パターン表示機能では、計算された LCS を集計し、出現回数を拡張した結果をソートして表示する。結果は Java の Swing を利用して、表形式で表示される。また、LCS を前後関係順にソートした結果を並べて表示する。

例として名古屋市の観光案内を基にして作成した FST スクリプトを用いて、実際に状態遷移履歴を作成し、LCS の計算結果を表示した。この FST スクリプトは、名古屋市の主な駅と飲食店の音声案内を記述したものである。今回は、駅と飲食店にカテゴリ分けを行った。

表 4 と表 5 に計算結果の一部を示す。

表 4 LCS を出現数順で表示

count	LCS
98	(飲食店),(駅),(飲食店)
74	(駅),(飲食店),(飲食店)
73	(飲食店),(飲食店),(駅)
70	(飲食店),(飲食店),(飲食店)

この結果からわかることの一例として、駅に関する対話が 2 回続けて話されることが少ないことや、飲食店に関する対話が 2 回続いた後に行われる対話は、飲食店よりも駅の方が多いということなどが挙げられる。

5. 実 験

5.1 実験内容

今回は、本学で運用しているコンテンツ投稿型音声案内デジタルサイネージ「ゆめルームメイちゃん」(図 10)を用いて実験を行った。このシステムは実際に本学の学生向けに運用されており、これを用いることで MMDAgent における実際の利用履歴を収集した。本実験では、「ゆめルームメイちゃん」を運用し、

表 5 LCS をシーケンス順で表示

count	LCS
70	(駅)
1	(駅),(駅)
68	(駅),(飲食店)
25	(駅),(飲食店),(駅)
26	(駅),(飲食店),(飲食店)
83	(飲食店)
42	(飲食店),(駅)
31	(飲食店),(駅),(飲食店)
41	(飲食店),(飲食店)
29	(飲食店),(飲食店),(駅)
11	(飲食店),(飲食店),(飲食店)

収集した状態遷移履歴を用いて LCS の抽出を行い、考察する。

5.2 コンテンツ投稿型音声案内デジタルサイネージ

「ゆめルームメイちゃん」

このシステムは MMDAgent を基にした音声案内デジタルサイネージであり、本学のキャラクター「メイちゃん」にユーザが情報を教えるという形態をとる。本システムの特徴は、本学の学生が音声対話コンテンツを自由に WEB から投稿することができるというものである。投稿者は、遷移条件となる任意の音声キーワードと、関連画像やキャラクターの動き、声のトーンを指定することで、FST スクリプトを編集することなく、音声対話コンテンツを投稿できる。投稿されたコンテンツは本学の学生交流スペース「ゆめルーム」に設置された MMDAgent で利用することができる。また、3D キャラクターの背後に、音声対話コンテンツの音声キーワードや関連画像を表示する「バルーン」を出現させ、コンテンツをユーザに提示する。これによってどのようなキーワードで音声対話を行うことができるのかをユーザが把握することができる。



図 10 コンテンツ投稿型音声案内デジタルサイネージ
「ゆめルームメイちゃん」

本システムでは、ジャンルを問わず様々な内容のコンテンツが投稿されているため、LCS を計算する際にコンテンツのカテゴリ分けを行う。カテゴリの分類は以下のように行った。

- 漫画アニメゲーム
- ジョーク
- 時事ネタ

- 挨拶
- 一般知識
- 娯楽
- 学校
- 雑談
- スラング
- 店
- 音楽
- 基本
- 基本 (挨拶)
- 基本 (雑談)
- その他

カテゴリーにおける「ジョーク」とは、ユーザが MMDAgent にジョークを言わせるという目的で投稿したと思われるコンテンツを分類する。

「時事ネタ」というカテゴリーは、時事問題をもとに投稿されたと思われるものを分類する。

「スラング」というカテゴリーは、ネット用語の説明などを行うものを分類する。

「雑談」というカテゴリーは、MMDAgent との雑談を行う目的で投稿したと思われるコンテンツを分類する。

「娯楽」というカテゴリーは、投稿者の趣味に関する言葉を説明するものを分類している。投稿数の多かった、漫画などに関する用語は別カテゴリーの漫画アニメゲームに分類する。

「学校」というカテゴリーは、本学のサークルや行事に関するものを分類する。「店」は飲食店などに関するもの、「音楽」は音楽知識に関するものを分類する。「挨拶」というカテゴリーは、キャラクターとの簡単なやり取りを行うものを分類する。

カテゴリーにおける「基本」とは、MMDAgent にデフォルトで記述されている音声対話であり、バルーンに表示されないという違いがある。これによってユーザがバルーンを利用して対話を行っているかを識別することができる。

5.3 結果と考察

以下に「ゆめルームメイちゃん」の状態遷移履歴における頻出パターンを示す。表 6 は、出現回数順の頻出パターンにおける上位の結果を示している。表 7 は、前後関係順の頻出パターンの結果を示している。紙面の関係で、結果の一部を紹介する。

表 6 LCS の集計結果

count	LCS
87	(漫画アニメゲーム),(一般知識),(一般知識)
71	(漫画アニメゲーム),(ジョーク),(ジョーク)
62	(漫画アニメゲーム),(一般知識),(漫画アニメゲーム)
59	(基本 (雑談)),(基本 (雑談)),(基本 (雑談))
58	(ジョーク),(ジョーク),(ジョーク)
52	(その他),(一般知識),(一般知識)
49	(一般知識),(漫画アニメゲーム),(一般知識)
46	(一般知識),(一般知識),(漫画アニメゲーム)
44	(漫画アニメゲーム),(漫画アニメゲーム),(漫画アニメゲーム)
43	(その他),(漫画アニメゲーム),(一般知識)
43	(一般知識),(一般知識),(一般知識)

表 7 LCS のシーケンス別結果

count	LCS
26	(漫画アニメゲーム),(雑談)
220	(漫画アニメゲーム),(漫画アニメゲーム)
200	(漫画アニメゲーム),(一般知識)
38	(漫画アニメゲーム),(一般知識),(一般知識)
3	(漫画アニメゲーム),(一般知識),(ジョーク)
18	(漫画アニメゲーム),(ジョーク),(漫画アニメゲーム)
32	(漫画アニメゲーム),(ジョーク),(ジョーク)
21	(挨拶)
1	(挨拶),(漫画アニメゲーム)
1	(挨拶),(時事ネタ)
3	(挨拶),(挨拶)
141	(基本 (雑談)),(基本 (雑談))
8	(基本 (雑談)),(基本 (挨拶))
66	(基本 (挨拶)),(基本 (雑談))
112	(基本 (挨拶)),(基本 (挨拶))

以下に考察を述べる。

コンテンツ投稿の傾向 投稿されたコンテンツの総数 264 個において、漫画アニメゲームが 61 個と最も多く、続いて一般知識が 49 個という結果であった。これは、「ゆめルームメイちゃん」がキャラクターに情報を教える、という形態をとっていることから、それぞれの投稿者の嗜好に関連するコンテンツを投稿する傾向が強いと思われる。また、知識を提供するコンテンツが主であり、特定の目的を実現するための音声対話コンテンツは少なかった。例えば、イベント案内情報などを投稿するケースはあまり見られなかった。

カテゴリー間の前後関係 漫画アニメゲームの後に同じく漫画アニメゲームカテゴリーの対話が続いて行われているパターンが 220 回という結果になった。また、漫画アニメゲームカテゴリーから、一般知識カテゴリーに続くパターンが 200 回、雑談に続くパターンが 26 回、という結果となった。挨拶カテゴリーから始まるパターンのうち、最も多いのが同じカテゴリーに続くパターンで、3 回であった。それ以外のカテゴリーに続くパターンはそれぞれ 1 回であった。同じカテゴリーが続くパターンがそれ以外のカテゴリーに続くパターンよりも多くなる傾向があると思われる。しかし、推測の域を出ないため、さらなる検証が必要であると考えられる。

今回の結果から、有用な頻出パターンを明確に述べることはできなかった。今後の課題として、抽出したパターンを解析し、数値化するなどの手法を検討する必要があると考えられる。

基本対話の利用状況 基本対話とは、「ゆめルームメイちゃん」においてバルーンで提示されない音声対話コンテンツのことを指す。「こんにちは」「さようなら」などの基本 (挨拶) カテゴリーや、「かわいい」などの基本 (雑談) カテゴリー、それ以外の基本カテゴリーに分けられている。基本 (挨拶) の後に、基本 (雑談) が発生するパターンが 66 回見られたが、逆の順番のパターンは 8 回となり、少なかった。これは LCS でなければ抽出されなかったパターンであるといえる。基本 (挨拶) は「こんにちは」や「おはよう」などの対話であり、基本の対話を利用

する場合、挨拶から行う場合が多いと考えられる。しかし、基本 (挨拶) の後に、続けて基本 (挨拶) の対話を行うパターンの方が、次に基本 (雑談) の対話を行う場合に比べて多く見られた。これは、ユーザがシステムの応答を確かめるために何度も話しかけるパターンや、「こんにちは」の後に「さようなら」と話しかけるパターンが含まれているためと考えられる。このように、バルーンにより提示された情報以外にも、ユーザは音声対話システムに話しかけることがあると考えられる。

全体を通して この実験により、同じカテゴリーは続けて話されやすいのではないかと推測したが、さらなる検証が必要であると考えられる。また、バルーンの提示に従ってユーザが音声対話コンテンツを利用している場合、コンテンツの投稿順に影響を受けると考えられる。バルーンはコンテンツの投稿順で出現するため、バルーンの出現順が結果に表れている可能性を否定できない。バルーンを用いない MMDAgent における実験や、バルーンの出現順を無作為にするとといったアプローチを取る必要があると考えられる。

6. おわりに

本論文では、MMDAgent の状態遷移履歴をセッションに分割し、Longest Common Subsequence を用いて頻出パターンを抽出する手法を提案した。そして、抽出した LCS をより直観的に閲覧できるように表示する手法を検討した。

また、提案した実現法を基にプロトタイプシステムを実装した。実装したシステムを用いて、コンテンツ投稿型音声案内デジタルサイネージ「ゆめルームメイちゃん」において、実験と考察を行った。実験により、「ゆめルームメイちゃん」における音声対話システムの頻出パターンを抽出した。実験結果から、同じカテゴリーの音声対話コンテンツは続けて話されやすいのではないかと推測ができるが、更なる検証が必要であると思われる。

本研究の今後の課題としては、正確にユーザを識別する方法の導入が挙げられる。本研究では、状態遷移履歴の時間間隔をもとにセッションに分割することによって、ユーザの切り替わりを識別した。しかし、実際には音声入力 of 誤認識や話しかけてもシステムが反応しなかった等のエラーが発生することがあり得る。エラーが発生していたとしても、システムが識別することはできないので、今回の音声対話履歴ではそれらを知ることができない。センサー等を利用して、実際のユーザのシステムの利用を識別することで、より信頼性の高い結果が得られると考えられる。

また、今回はコンテンツ投稿型の MMDAgent において実験を行ったが、別の MMDAgent の音声対話履歴を用いた実験を行い、この手法の有用性を検証することも今後の課題である。

7. 謝 辞

本研究は JSPS 科研費 26330136, 25700009, 科学技術振興機構 CREST, および、総務省 SCOPE の助成を受けたものです。この場を借りて、感謝の意を表します。

- [1] 李晃伸, 大浦圭一郎, 徳田恵一, “魅力ある音声インタラクションシステムを構築するためのオープンソースツールキット MMDAgent”, 『電子情報通信学会技術研究報告. SP, 音声』, 2011.
- [2] Akinobu Lee, Keiichiro Oura, Keiichi Tokuda, MMDAgent - A fully open-source toolkit for voice interaction systems, Proceedings of the ICASSP 2013, pp. 8382-8385, 2013.5.
- [3] 大浦圭一郎, 山本大介, 内匠逸, 李晃伸, 徳田恵一, キャンパスの公共空間におけるユーザ参加型双方向音声案内デジタルサイネージシステム, 人工知能学会誌, Vol.28, No.1, pp.60-67, 2013.
- [4] Ryota Nishimura, Koji Sugioka, Daisuke Yamamoto, Takahiro Uchiya, Ichi Takumi, A VoIP-based Voice Interaction System for a Virtual Telephone Operator Using Video Calls, Proc. of the IEEE GCCE 2014, pp.529-532, 2014.
- [5] Takahiro Uchiya, Daisuke Yamamoto, Ryota Nishimura, Ichi Takumi, Tetsuo Kinoshita ”Proposal of Voice Control-based Interactive Design Method of Agent System,” Proc. of the 13th IEEE International Conference on Cognitive Informatics and Cognitive Computing (ICCI*CC2014), pp.496-500, Aug. 2014.
- [6] Takahiro Uchiya, Masaki Yoshida, Daisuke Yamamoto and Ichi Takumi, Development of Open-Campus System Using Voice Interaction Agent, INWC-2015: The 5-th International Workshop on Information Networking and Wireless Communications, Taipei, Sep. 2-4, 2015.
- [7] Keitaro Wakabayashi, Daisuke Yamamoto, Naohisa Takahashi, A Voice Dialog Editor Based on Finite State Transducer Using Composite State for Tablet Devices, Proc. of IEEE/ACS ICIS 2015, Las Vegas, 2015.6.29.
- [8] 石川博規, 山本大介, 高橋直久, “音声対話コンテンツのパッケージ化とその配信システム”, DICOMO2014 シンポジウム論文集, 2014, 789-795, 2014-07-02
- [9] Tomohiro Yanagi, Daisuke Yamamoto, Naohisa Takahashi, Development of Mobile Voice Navigation System Using User-Based Mobile Maps Annotations, Proc. of IEEE/ACS ICIS 2015, pp. 373 - 378, Las Vegas, 2015.6.30
- [10] 田中佑太郎, 山本大介, 高橋直久, “音声対話コンテンツにおける状態遷移履歴の可視化と簡単化手法”, DICOMO2015 シンポジウム論文集, 2015, 1263-1270, 2015-07-02
- [11] 宇根田純治, 横田治夫, “Web 履歴の共通シーケンス解析”, 電子情報通信学会技術研究報告. DE, データ工学 102(64), 7-12, 2002-05-10.
- [12] 大竹聡史, 打矢隆弘, 内匠逸, “音声インタラクションに基づくユーザの嗜好抽出”, 第 76 回全国大会講演論文集 2014(1), 641-642, 2014-03-11
- [13] Daisuke Yamamoto, Keiichiro Oura, Ryota Nishimura, Takahiro Uchiya, Akinobu Lee, Ichi Takumi and Keiichi Tokuda, Voice Interaction System with 3D-CG Human Agent for Stand-alone Smartphones, Proc. of the 2nd International Conference on Human Agent Interaction, ACM digital library, pp.320-330, 2014
- [14] Takahiro Uchiya, Ryosuke Nakano, Daisuke Yamamoto, Ryota Nishimura, Ichi Takumi, “Extension with Intelligent Agents for the Spoken Dialogue System for Smartphones,” Proceedings of the 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), pp.298-299, 2015. Osaka, Japan, October 27-30, 2015.
- [15] Lee Akinobu, and Tatsuya Kawahara, Recent Development of Open-Source Speech Recognition Engine Julius, Proceedings of the APSIPA ASC, pp. 131-137, 2009.
- [16] 大浦圭一郎, 酒向慎司, 徳田恵一: 日本語テキスト音声合成システム Open JTalk, 日本音響学会春季講演集, Vol. 1, No.2-7-6, pp. 343-344, 2010.