

# 車載器に旅行計画を登録するための発話文解析

徳久雅人 木村周平

鳥取大学 大学院工学研究科 情報エレクトロニクス専攻

tokuhisa@eecs.tottori-u.ac.jp

## 1. はじめに

自動車旅行の最中に巡る内容を話題にすることがある。それらを車載器にメモしておくこと(図1)で、場当たりの旅行であつても見逃し無く旅行することができる。車内ではキーボードやタッチスクリーンは使い難いため、音声入力が見込み。

旅行計画は、日時、場所、および、用件を項目とするレコードの列である。発話文解析ではこれらを抽出する必要がある。発話文において日時は、省略されることがあるため、文脈に応じて補う必要がある。そこで、本稿では、強化学習を用いることで発話文から各項目を抽出する規則を取得すること、および、旅行計画の表管理による日時補完を行なうことを目的とする。



図1 車載器に旅行計画を音声入力に登録する様子。

重複する日時は非表示。発話文から日時を補いながら該当する番目に登録。Androidで音声認識。車載器 (RaspberryPi3) で以降の処理を実行。背景は国土交通省道路技術基準・道路標識より引用。

## 2. 強化学習による発話文解析

### 2.1 解析の概要

トップダウン型で発話文を解析する。まず、発話文に含まれる予約語に応じて、発話文のタイプ(対話行為に相当。車載器のサービス用メソッドに対応)の仮決めを行なう。次に、タイプごとに解析規則集を用いてスロットの充足を行なう(メソッドの引数への代入に相当)。最後に、適合度の高いものから順に、車載器で実行可能性を検査して実行する。

発話文のタイプは、91種類である。各タイプには発話文の例文が対応している。例文はチャンク、予約語(136種 + 6種)、品詞(8種)、意味の各タグが付与されている。予約語からは、Bag of Words (BOW) を生成する。入力発話文と例文とを BOW で比較することで、入力発話文のタイプを仮決めて

きる。候補として複数のタイプが列挙される。例えば「音楽をかけてください」という発話文に対して、次の2つのタイプが候補となる。

- serviceMusic.comPlay(*unit*)
- serviceMusic.comSearch(*name*, *unit*)

「音楽」、「かけ」、「てください」が BOW の基底ベクトルとなっている。*unit*, *name* はスロットである。*unit* はアルバム単位やアーティスト単位という曲をまとめる単位である。*name* は曲を検索する条件である。

スロットの充足は、タイプごとにあらかじめ用意しておいた解析規則集を用いる(次節)。解析規則は、品詞 4-gram, 代入先スロット名 3-gram, などの「状態」を条件部、「認識アクション」を帰結部、および、両者に対する評価値 (Q-Value) の組の集合である。入力発話文の単語を先頭から文末まで順に参照する過程において、解析規則に基づく認識アクションを実行することで、単語とスロットの対応関係を解析する。

例えば「\*/O 松任谷由実/N を/P かけ/R2 てください/AReq EOS/T \*/O \*/O」という発話文は、1単語目に注目している時点では、「O-N-Pを-R2」という品詞 4-gram, 「D-D-D」という代入先 3-gram 等で状態を認識する(O は文頭ダミー文字の品詞, Nは名詞, "Pを" は助詞 "を", R2は予約語#2の意。D は代入不使用の意)。この状態では、解析規則によると「松任谷由実」をスロット *name* に代入するという判断となる。

スロットが代入済みとなった発話文タイプは適合度で順位付け。BOW の個数  $t_1$  と一致数  $t_2$ , 必須スロットの数  $s_1$  と代入数  $s_2$ , 任意スロットの代入数  $s_3$  による適合度  $C$  である。

$$C = (t_2 + \alpha) / (t_1 + \beta) \times (s_2 + s_3 + \alpha) / (s_1 + s_3 + \beta)$$

ここで、 $\alpha = 0.9$ ,  $\beta = 1.0$  とした。全てが一致しても条件数の多い場合の全一致が優位となる。

### 2.2 強化学習による解析規則の獲得

タイプごとの解析規則集は、強化学習を用いて例文から自動生成する。強化学習とは、エージェントが、与えられた環境の中で高い収益を得るためのアクションを学習することである。本稿では、構文解析レベルの解析規則の学習に、強化学習を用いる。入力文の品詞列やスロット代入状況等が環境である。その一部(素性)を参照して状態とする。アクションは、パーキングのための操作であり、「認識アクション」と呼ぶこととする。

#### 2.2.1 認識アクション

認識アクションは次の通りである:

- actForward: 注目位置を 1 つ右に進める。
- actPush: 注目語を一時スロット  $t$  に代入する。

- **actEmpty** :  $t$  を空にする.
- **actApply(s)** :  $t$  の語をスロット  $s$  に代入する.
- **actClear(s)** : スロット  $s$  を空にする.
- **actReconsider** : スロット値等はそのままに, 文頭に戻る.
- **actSubmit** : 解析を終了する.

最初にエージェントを先頭の単語に配置する. **actForward** を選択するとエージェントは隣の単語に移動する. **actPush** を選択するとエージェントの位置の単語を一時スロット  $t$  に代入する. 移動と代入を交互に行なうとチャンクを作ることができる. **actEmpty** はそのチャンクを破棄する操作である.

**actApply(s)** はそのチャンクをスロット  $s$  に代入する. 代入すると一時スロットは空になる. 代入されるスロット  $s$  は追加で単語を代入する. ゆえに, 文の上で離れた単語でも 1 つのスロットに代入できる. **actClear(s)** はスロット  $s$  を空にする. 例えば, 「鳥取じゃなかった島根の地図を表示してください」のように訂正のフレーズを受けてスロット  $loc =$  鳥取 を止めて  $loc =$  島根 という代入状況に変更することができる.

**actReconsider** は文末で有効なアクションであり, 文頭に戻って操作をやり直すこと(再考)ができる. **actSubmit** は代入状況をもって解析を終了することを宣言するためにある.

## 2.2.2 状態と素性

エージェントが把握する状態は次の素性で構成する:

- 品詞 4-gram
- 代入先スロット 3-gram
- **actReconsider** を実行したか, 再考フラグ  $\{T, F\}$
- 各スロット名と代入数  $\{name, unit, loc, \dots\} \times \{0, 1, PL\}$
- 一時スロットの主辞の意味的属性  $\{L, P, R, A\} \cup POS$
- 注目単語の意味的属性
- 名詞述語文の場合での「ハ格」の品詞

品詞 4-gram は, エージェントの 1 つ左の単語から 2 つ右の単語までの 4 単語の品詞の並びである. 2 単語先を見越して現在位置の単語への操作を決めることができる. 代入先スロットは, エージェントの 1 つ左の単語から 1 つ右の単語までの 3 単語の代入先である. 再考フラグにより, 再考をしたことがあるか否かを表す. 2 度目の再考を抑制できる. スロットごとに代入単語数を表す. 2 語以上を  $PL$  とする. 一時スロットの主辞とは代入された最右単語とする. 意味的属性は, 日本語語彙大系の「388 場所」以下の名詞ならば  $L$ , 「2610 場」以下の名詞ならば  $P$ , 場ではない「2422 抽象的關係」以下の名詞ならば  $R$ , 抽象的關係でない「1000 抽象」以下の名詞ならば  $A$ , それ以外の単語は品詞ラベルとする. 例えば, 「博物館に行く」, 「池の辺りに行く」は二格に  $L, P$  がそれぞれ存在し, 移動先であることが明確になる. 「魚釣りに行く」, 「展示会に行く」は二格に  $R$  が存在し, 用件であると区別できる. 「ハ格」の品詞は, 名詞述語文の意味を粗く把握するために導入した.

## 2.2.3 報酬

解析の途中の段階で, 報酬を与える. 部分点を与えることで

学習を早く行なうことを狙う. 報酬には正の報酬と負の報酬があり, 抑制したい行動には負の報酬を与える.

- **actForward** : 文末でさらに右に進もうとすると  $-5$ .
- **actPush** : 文外の単語, または, 既に代入されている単語を一時スロットに代入するならば  $-20$ . いずれかのスロットに代入しなければならない単語ならば  $+5$ . 否定語を含まない文において代入してはいけない単語ならば  $-15$ .
- **actEmpty** : 一時スロットが空ならば  $-20$ . スロットに代入すべき単語を有していたのであれば  $-15$ .
- **actApply(s)** : 一時スロットが空ならば  $-20$ . 一時スロットの単語とスロット名  $s$  が正解と完全一致ならば  $+20$ , 部分一致ならば無報酬, そうでなければ  $-10$ .
- **actClear(s)** : 空スロットならば  $-20$ . 空にすると正解と完全一致にならないならば  $-25$ .
- **actReconsider** : 2回目の再考, 文末以外での再考ならば  $-20$ .
- **actSubmit** : 文末での実行, かつ, 全スロットが正解ならば  $+10$ . 文末以外での実行ならば  $-5$ .

## 2.2.4 試行と更新

強化学習では, 状態とアクションに対する評価値を更新する. ある学習時点  $n$  において, 状態  $x$  におけるアクション  $a$  への評価値を  $Q_n(x, a)$  とする. 一つ前の時点  $n-1$  からの更新を次式で行なう.

$$Q_n(x, a) = (1 - \alpha_n) Q_{n-1}(x, a) + \alpha_n \{r + \gamma \max_b Q_{n-1}(y, b)\}$$

ここで,  $y$  は状態  $x$  の際にアクション  $a$  を実行して生じた状態,  $r$  は報酬,  $\alpha_n$  は学習率,  $\gamma$  は割引率である[1]. なお, 時点  $n=0$  において,  $Q_0(x, a)$  に一定の初期値がある.

本稿では, 1 つの発話のタイプに, 解析規則集が一式対応する. つまり, 規則集一式が一つの  $Q$  関数に対応し, タイプの種類数だけ  $Q$  関数が存在する. なお,  $Q_0(x, a) = 50$ ,  $\alpha_n = 0.8$ ,  $\gamma = 0.9$  とした.

本稿での学習の進め方について述べる. 1 つの発話タイプには複数の例文が対応する. 1 タイプにつき, そのタイプに対応する全ての例文を 1 つずつ入力し強化学習を行なう. 1 文につき 30 回の試行を行なう. 1 回の試行では認識アクションを最大で単語数  $\times 3$  回まで実行する. 規則集を用いて対応する例文の解析を行なう. 誤った文について, 再び強化学習を行なう. 全ての例文が正しく解析できれば, そのタイプに対する学習が終了となる.

なお, 全てを正しく解析できない場合があるので, 再学習を最大 150 回とした. ただし, この事態に陥ったのは, 素性設計が悪い, または, 例文の作成・アノテーションに誤りがあった場合であった.

また, 発話タイプに対応付けられていない例文を受理してしまう場合がある. その場合, その例文をその発話タイプの学習

文として与える。適合すべきスロットがないことを学習させる。その結果、適合度の判定において劣位となるため全体として正しい解析結果を生むことになる。

ただし、適合したスロットは正しいが期待しない発話タイプに例文が受理される場合がある。それは、そもそも発話文が曖昧である可能性がある。より高次元の情報を与える、もしくは、対話的に曖昧性を解消する必要がある。例えば、YMO の「ONGAKU」という楽曲を検索・再生するという意味で解釈 1-2 は正しい。

例 1:「音楽をかけてください」

解釈 1-1: `serviceMusic.comPlay`

解釈 1-2: `serviceMusic.comSearch(name=音楽)`

### 3. 旅行計画の登録

自動車旅行を支援するために旅行計画を音声入力で登録する。発話文は前節の方法で解析する。本節では登録のための発話タイプと登録処理について述べる。

#### 3.1 発話タイプと登録操作

旅行計画登録のための発話タイプは以下の通りである:

- `comOpen`: 旅行計画の登録を開始
- `comClose`: 旅行計画の登録を終了
- `comInformEvent(date, time, loc, info)`: イベントを登録
- `comInformRoute(date, time, loc, info)`: 経路地を登録
- `comSelect(dir)`: 注目するレコードを指定
- `comRemove(query)`: レコードを削除
- `comExchange(query)`: 2つのレコードを交換
- `comRevise(query, correct)`: レコードを改訂

`comOpen/Close` は旅行計画のファイル操作である。「予定を登録してください」という発話で開始となり、「以上です」という発話で終了となる。`comInformEvent` は、日時、場所、用件を登録する。発話では日時が不完全な言い方になることが多い。例えば 2018年1月17日という日にちを正確に言うことは希である。本操作には日時の補完を含む。`comInformRoute` は、経路地を登録する。高速道路名、方面名、分岐点名などである。`comSelect` は、ユーザが注目するレコードを指定するためである。ひとつに、車載器のディスプレイにリスト形式で旅行計画を表示することに用いる。もうひとつに、発話で補う日時の基準に用いる。例えば、「上にずらしてください」というリストをスクロールさせ、「5番を表示してください」というと5番目のレコードを中心とした表示となる。後に続く発話はこの注目したレコードの日時が基準となる。`comRemove` は条件を満たすレコードを削除する。例えば、「今日の予定を削除してください」という発話を受理する。`comExchange` は2つのレコードの順序を入れ替える。例えば、「5番と7番を入れ替えてください」という発話を受理し、5番レコードの場所と7番レコードの場所について項目内容を入れ替え、同様にの用件について入れ替える。日時は入れ替えない。`comRevise` は、日時・場所・用

件の記載内容を変更する。例えば「鳥取砂丘は15時ではなく14時にしてください」というと場所・備考に鳥取砂丘のある最寄りの番目を検出し、その日時を変更する。

#### 3.2 文脈に依存した日時の解析

日時に関する発話として、以下の分類を設ける。

- 絶対的な日時: 例) 2018年1月17日 23時59分

- 相対的な日時:

– 現在からの相対日時: 例) 今日, 明日, 来週, 来月

– 話題からの相対日時: 例) 5分後, その前に, 12時

絶対的な日時は、年月日時を必ず含む。分は「0分」と言わないことが多いため必須ではない。現在からの相対日時は、「今」や「明日」などの幾つかの単語を含む場合に限られる。発話文の多くは、話題からの相対日時である。例えば「1月17日 23時」という発話は、「2018年」と付かないので絶対的な日時ではない。話題が2018年であることを根拠に年を補う。なお、元日や大晦日という単語からでも日にちを具体化する。

ここで、注目している話題を特定しておく必要がある。多くの場合、最後の発話と一致する。レコードの追加のみであれば最後の発話と一致させることで十分である。なお、レコードの削除や旅行計画の参照により、注目先が変わる。そこで、旅行計画を常に表形式でまとめてディスプレイに表示することで、車載器が基準として認識しているレコードをユーザが確認できるようにする(図1でオレンジ色のレコードが注目先)。

#### 3.3 追加

発話文に日時の言及がない場合は、現在注目しているレコードの下に場所・用件のレコードを新規に追加する。このときの日時は注目先のレコードの日時を用いる。日時に言及がある場合は、上記の解析で得た正確な日時に基づき適切な位置を求めて、場所・用件のレコードを新規に追加する。以下のような発話を受理する:

- 今週の土曜日に鳥取砂丘に行きます
- 9時に鳥取砂丘に着きます
- 多鯉ヶ池に行きます
- その前に梨ソフトを食べます

#### 3.4 検索と訂正

旅行計画の確認や訂正のために、条件を指定してレコードを検索することがある。以下のような発話がある:

- 5番を削除してください
- 5番と6番を入れ替えて下さい
- 13時から17時までをキャンセルしてください
- 17時の予定を削除してください
- 2月1日から3日までをキャンセルしてください
- 仁風閣を止めて白兔海岸にします
- 鳥取自動車道は佐用の後です

番号で指定する発話は明確で単純に対処する。時刻を条件とする場合、0分であるのか、0分から59分までの範囲である

のかが不明確である。例えば、「17時の予定を削除して」という場合は「17:00-17:59の予定」であるが、「17時までを削除して」という場合は「17:00」までとする方が良いように思える。しかし、「2月1日から3日まで」という日の場合は「2月3日23:59まで」のとおり広く捉える方が良いように思える。一方、日時の場合、場所を指定した訂正がある。観光先の変更では日時を固定し場所だけを変更する。また、最後の例は、次の文脈での発話である。「鳥取自動車道を通ります。佐用を通ります。鳥取自動車道は佐用の後です。」経路地を表すレコードが2つあり、その順序を訂正している。query="鳥取自動車道", correct="佐用の後" という値で訂正のメソッドが呼ばれる。「の後」を手がかりに「佐用」を検索し、レコードを特定する。

現在のところ、追加・検索・訂正についてのメソッド内で、ルールベースで対処している。

## 4. 実験

### 4.1 発話文解析

コーパスに、発話文、各種タグ、発話文タイプ、および、スロット理想値をまとめた。発話文は 1,308 文、発話文タイプは 91 種類である。発話文タイプに依りスロット数は 0 個から 4 個までである。ただし、発話タイプには、車載器の全サービスを含む。旅行計画のサービス以外については [2] で報告したとおりである。

学習の結果、解析規則集は 91 式となり、合計で約 223 万件となった。ただし、そのうち約 190 万件は発話文タイプ comInformEvent のためのものであった。スロット数が 4 つ、例文数が 302 件であり、言い回しが多様であったためである。規則はハッシュに格納する。状態とアクションをキーとして Q 関数の値をバリューとするハッシュである。学習は 4 GB のメモリを有する計算機で行なった。運用は RaspberryPi3 でありメモリ不足のため GDBM を用いた。

発話文解析のクローズドテストを行なった。タイプ的一致、かつ、スロット値の完全一致となった文数で評価した。正解率は  $99.7\% = (1,304 / 1,308)$  であった。誤り原因は発話文が言葉不足であったことである ([2] で報告済み)。旅行計画の登録においては誤りはなかった。

クローズドテストにて調整を要した点は以下のとおりである:

- 状態の詳細化:
  - 名詞意味属性の追加、予約語の追加
- スロットを一時スロットと確定スロットに分割

状態の詳細化は、強化学習の原理に依存するものである。強化学習はマルコフ決定過程を前提としている。エージェントが観測できる 1 つの状態に対して異なる 2 つのアクションを時々において正解とする場合に学習ができない。2 つのアクションを選択するには異なる状態となるように状態の定義(素性)を拡張する必要がある。旅行計画の登録では、「〇〇に行く」の移動の他に「〇〇を買う」、「〇〇を見る」などの観光の際

の行動の情報も解析対象となる。名詞の意味属性を導入した。また、「〇〇を登る」の場合は場所として抽出するが、「〇〇を見る」の場合は場所として抽出しない。現在は動詞の意味属性を用いておらず、観光時の動詞を予約語とした。

スロットに一時スロットと確定スロットの 2 種類を設けた理由は、格要素に対応させる範囲を認定するためである。品詞 4-gram で注目単語の周辺を観察したが、問題点として、長い語数のものが解析できないこと、および、助詞が見えず代入先のスロットが決定できないことが明らかになった。文末から文頭に向けて解析するならば動詞と助詞が観測された後でチャンクを求めるようにできる。しかし、訂正の発話を扱うことにおいて不利になる。例えば、「鳥根に行きますではなかった鳥取に行きます」において扱い難い。状態の設計で工夫できるかもしれないが、本稿では人間が文頭から理解していくことにならない、文頭から解析する上での状態設計に挑戦した。

### 4.2 車載器での使用感

発話に対する応答速度は問題ない。本稿で目標とする日時、場所、用件の抽出において、「行き、出発、帰り」に関してはある程度正しかった。しかし、日時と場所の語順が異なると抽出できないことがあった。強化学習で状態の定義を厳密なものに近づけるとオープンテストでの少々の違いにおいて別状態として扱われることが原因であろう。状態のマッチングを、現在はハッシュで行なっているが、SQL などを用いると、素性ごとの検索に切り替えることができる。今後、柔軟性を高めつつアンサンブルな手法で解を決定することを試みる。

旅行計画を述べる文と、曲名/観光地名などの検索を命令する文では、自由度が大きく異なる(曲名検索では規則数が 51,365 件、観光ガイド検索では 2,138 件だった)。日時の表現と用件の表現が多様である。今後、発話タイプごとの規則集だけでなく、日時などの品詞・チャンクの解析のための規則集を導入することを検討する。

## 5. おわりに

本稿では、車載器に音声入力で行なう旅行計画を登録するための方法を示した。日時、場所、用件を発話文から抽出する規則は強化学習で作成した。抽出結果からルールベース手法で日時の補完を行いつつながら、旅行計画を適切な順序で追加すること、および、言葉による条件で改訂することを実現した。車載器において、旅行計画の登録の動作を確認した。また、音楽再生や地図表示などの他のサービスのための発話も区別して解析できており、車載器の操作を音声で行なうことができた。

## 参考文献

- [1] Watkins, C., and Dayan, P.: Q-Learning, Machine Learning, Technical Note, Vol.8, pp.279-292, 1992.
- [2] 徳久雅人, 木村周平: 強化学習による車載器への命令発話文解析の試み, 信学技法, NLC2017-38, 2017.