

二つの動作を含む重文型命令文を受理するためのサービス指向発話文解析の拡張

但田聖 徳久雅人 木村周平

鳥取大学 工学部

b19t2060u@edu.tottori-u.ac.jp {tokuhisa,kimura}@tottori-u.ac.jp

概要

本稿では、2つの動作を一度に命令する発話文を受理してサービスの提供を行う手法について述べる。スマートフォンや家電に音声認識を使ったサービス要求の命令が可能である。1つの動作を命令する発話文は受理可能である。しかし、2つの動作を命令するためには2回の発話が必要となる。

本稿では、車載器に用いられるサービス指向発話文解析器(従来手法)に対して、文分割、分割評価および命令合成を追加することで、2つの動作を含む命令文であっても適切にサービスの提供が実現できることを示す。

1 はじめに

近年、音声認識を用いた技術が普及している。スマートフォンや家電に、音声認識を使ったサービス要求の命令が可能になっている。

本研究室では音声命令によって操作可能な車載器の開発を進めている[1]。この車載器では、1つの動作のための命令文しか実行することができない。よって、この車載器に2つの動作を命令したい時は、1つずつ動作を命令しなければならず、時間と手間がかかってしまう。

そこで、本稿では車載器で2つの動作を含む重文型命令文が実行できるようにすることを目的とする。

2 先行研究

文分割についての先行研究、発話文解析についての先行研究、および本稿の位置づけについて述べる。

2.1 文分割についての先行研究

先行研究[2]は、複雑な文の可読性の向上や解析処理を目的とした研究である。WEBSPLITという単純な文と複雑な文をペアとするデータセットを提案し、文分割を可能にしたが、実際に複雑な文の解析に適用できるかの確認は行われなかった。

2.2 発話文解析についての先行研究

先行研究[1]では、ユーザからのサービス要求型発話文を受け取ると、発話文の意図するサービスとメソッドを発話文解析によって推定し、小型計算機上でサービス結果を出力する研究が行われた。発話文解析の流れと先行研究の問題点について述べる。

2.2.1 発話文解析の流れ

発話文解析の流れを図1に示す。

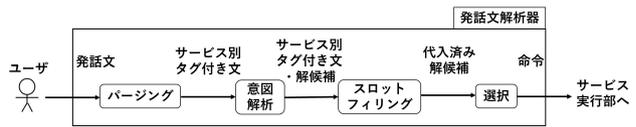


図1 発話文解析の流れ

発話文解析器がユーザから発話文を受け取ると、パーズングが行われる。パーズングではサービス別に用意された辞書を用いて各語句にタグ付けを行い、「サービス別タグ付き文」を出力する。次に、意図解析では、オリジナルコーパスの事例と「サービス別タグ付き文」の比較を行い、意図クラスを求める。オリジナルコーパスは、「発話文、サービス名、メソッド名・引数」という事例で構成され、サービス名とメソッド名・引数の組を意図クラスとする。比較の際、事例*i*と「サービス別タグ付き文」との近さ d_1 を次式で求めて利用する。

$$d_1(U, T_i) = \frac{2|U \cap T_i|}{|U| + |T_i| + \beta} \quad (1)$$

ここで、 U はタグ付き文のタグの集合、 T_i は意図解析用事例*i*のタグの集合、 β は偏りをを持たせるパラメータである。

近さの降順で意図クラスが得られ、解候補として複数が出力される。

次に、スロットフィリングでは、得られた解候補の引数に対して代入を行い、得られたスロット値(引数と代入語句の対)を解候補に加えて出力する。

最後に、選択では、解候補のタグの一致率およびスロットの代入率からスコア d_2 を次式で求めて利用する。

$$d_2(U, T_i, b, s) = \frac{2|U \cap T_i| + b}{|U| + |T_i| + s + \beta} \quad (2)$$

ここで、 b はスロットフィリングにおいて代入されたスロット数であり、 s はスロットフィリングにおいて代入されるべきスロット数である。なお、 U 、 T_i および β は式(1)と同じである。

スコアの降順で意図クラスおよび引数の具体値が得られ、最高スコアの命令(サービス名・メソッド名・スロット値)が選ばれる。命令はサービスの実行部に送られて、ユーザにサービスの提供が行われる。以上の流れで発話文解析を行う。

以下に発話文解析の例を示す。

発話文：鳥取の地図を表示してください
意図クラス：serviceMap.comShow(loc)
候補のスロット値：loc=鳥取
命令：serviceMap.comShow(loc=鳥取)

2.2.2 先行研究の問題点

2つの動作を含む重文型命令文を先行研究の手法に入力すると、1事例につき1意図クラスしかオリジナルコーパスに登録されていないため、1つの命令しか実行されない。2つの動作を含む命令を実行させるには、オリジナルコーパスの発話文を組み合わせる命令文を作り、その1つの命令文につき動作を2つ登録するというアイデアが考えられる。しかし、オリジナルコーパスに登録されている発話文は約3千文であり、その組み合わせは最大で約9百万文となる。これではコストが大きすぎる。

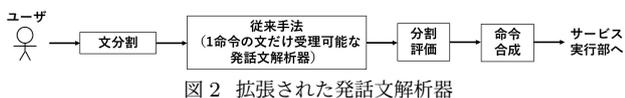
2.3 本稿の位置づけ

本稿では、コストを抑えるために、コーパスへの文の追加ではなく、車載器の発話文解析器の拡張という手段をとる。すなわち、文分割を発話文解析の前に行うことで、2つの動作を含む文を1つの動作を含む2つの文とし、得られる命令を適切に実行することを試みる。ただし、本稿では、2つの動作を含む表現を重文型命令文に限定する。

3 提案手法

提案手法は、従来手法(2.2節)に対し、「文分割」、「分割評価」および「命令合成」を追加した拡張手法である(図2)。拡張のねらいは次のとおりである。

- 文分割：1文1命令となるように文を分割
- 分割評価：分割/非分割のどちらが意味を汲み取ったかを評価
- 命令合成：単機能な複数命令か、高機能な単一命令かの選択



以下に提案手法における処理の要点を例示する。

重文型命令文に対する文分割の結果を示す。

重文型命令文：地図を表示して音楽をかけてください

分割後の文1：地図を表示してください

分割後の文2：音楽をかけてください

分割評価では、重文型命令文を従来手法で解析した結果、および分割後の文1、文2をそれぞれ解析した結果という3つの結果を扱う。地図サービスと仮定すると

「音楽」の意味が汲み取られず、音楽サービスを仮定すると「地図」の意味が汲み取られない。一方で分割後ではそれぞれで意味が汲み取られるため、分割が適切と判定される。

命令合成については、別の例文で説明する。たとえば、「初音ミクをかけてから鏡音リンをかけてください」という場合、命令合成には2つの命令が入力され、代替命令が出力される。

命令1：serviceMusic.comSearch(obj=初音ミク)
命令2：serviceMusic.comSearch(obj=鏡音リン)
代替命令：serviceMusic.compPlayList(obj1=初音ミク, obj2=鏡音リン)

検索再生の命令2つよりも、プレイリストの命令1つの方が容易にサービスの実行が行える。

本節では、文分割、分割評価および命令合成について詳しく述べる。

3.1 文分割

文分割は、発話文に特定のワードが含まれている時、特定のワードの前後で単文に分割する。特定のワードの決定方法、特定のワードの判別方法および文末処理について示す。

3.1.1 特定のワードの決定方法

鳥バンクという日本語と英語の意味類型パターン辞書がある[3]。この中では、節間キーワードという主節と従属節をつなぐワードが定義されている。節間キーワードの出現頻度上位30件の内、重文型命令文に含まれると想定されるものを特定のワードとして選出する。

節間キーワードの出現頻度上位30件は表1の通りとなった。

表1 節間キーワードの頻度表(上位30件)

件数(件)	節間キーワード	件数(件)	節間キーワード	件数(件)	節間キーワード
22,968	て	1,851	もの	728	とは
9,655	と	1,821	ても	717	のを
4,590	連帯形 + 名詞	1,601	0	708	には
3,984	連用中止	1,570	たら	682	疑問後 + か
3,510	~の(が は)	1,363	様に	639	ながら
3,325	事	1,256	連体形 +	534	まで(は)
3,303	ので		形式名詞	527	てから
3,189	が	1,232	人	492	ほど
2,936	ば	912	時	490	こと(が は)
2,247	~事(が は)	891	為に	490	連体形 +
2,043	から	836	のに		形式名詞

表1中の、「て」、「てから」、「たら」、「ながら」の4つの節間キーワードを特定のワードとして選出した。

3.1.2 特定のワードの判別方法

文分割の判断には、特定のワードの品詞、字面およびその前後の単語の品詞で定義される条件を用いる(表2)。

発話文をMeCabで形態素解析し、得られた解析結果と表2を比較して文を分割するか判断する。

表2 分割条件

通番	特定のワード (品詞/字面)	前単語品詞	後単語品詞
1	て (接続助詞/て で)	自立・動詞	-(非自立・動詞) ^ -(助動詞)
2	て (接続助詞/て で) + から (助詞/から)	自立・動詞	
3	たら (助動詞/たら だら)	自立・動詞	
4	ながら (接続助詞/ながら)	自立・動詞	

3.1.3 文末処理

文を分割した際、分割後の1文目の文末が不完全になる。その対応として、分割前の文末表現を、分割後の1文目の文末に追加するという文末処理を行う。以下に、文末表現が「てください」の時の処理の例を示す。

処理前の分割後の文1：音楽をかけ

処理後の分割後の文2：音楽をかけてください

対象となる文末表現は以下のとおりである。

- ~てください | ~てください
- ~ます
- ~です

ここで、「ます/です」による命令文は、ユーザの行動を車載器に受理させる際に使われる。たとえば、「今から鳥取駅に行きます」や「10時に出発です」のように予定を登録する発話に用いられる。

3.2 分割評価

車載器のオリジナルコーパス（先行研究のコーパス）には既に重文型命令文が登録されている。たとえば、「シャッフルして音楽をかけてください」という文がある。これは音楽の再生モードを変更するという、1動作を命令する重文型命令文である。発話文に対して毎回分割を行うと、過剰な分割となる恐れがある。この問題を分割評価によって防ぐ。

非分割文と分割文のスコアを比較する。非分割文のスコアは d_2 を用いて求める。分割文のスコア d_3 は、分割文の文1と文2のスコアを転用した値であり、次式を用いて求める。

$$d_3(U_1, U_2, T_1, T_2, b_1, b_2, s_1, s_2) = \frac{2(|U_1 \cap T_1| - t) + 2|U_2 \cap T_2| + b_1 + b_2}{|U_1| - t + |U_2| + |T_1| - t + |T_2| + s_1 + s_2 + \beta} \quad (3)$$

ここで、 U_1, T_1, b_1, s_1 および U_2, T_2, b_2, s_2 は式(2)と同様であり、それぞれ分割後の文1および文2の値である。 β も同様である。 t は文末処理によって追加された文末表現に対するタグ数である。

もし、 $d_2 \geq d_3$ ならば非分割を採用し、そうでなければ分割を採用する。

3.3 命令合成

命令合成の対象となる2つの命令の例を以下に示す。

命令1：serviceMusic.comSearch(obj=初音ミク)

命令2：serviceMusic.comSearch(obj=鏡音リン)

この2つの命令を連続で実行すると1曲目の再生が中断され、2曲目のみが再生されてしまう。

命令合成では、2つの命令が以下の条件を満たすとき、代替命令を出力する。代替命令は2つの命令の動作を含む高機能な単一命令である。

- 1つ目の命令が並列処理を行う、かつ
- 2つの命令間でリソースの競合がおこる

知識ベースとの照合によって、2つの命令が条件を満たすか判断する。条件を満たす2つの命令と代替命令を知識ベースにまとめた(表3)。代替命令を用意できない場合はエラーとする。エラー時は1つ目の命令のみを出力するという運用にする。

表3 知識ベース

命令1		命令2		代替命令	
サービス	メソッド	サービス	メソッド	サービス	メソッド
serviceMusic	comSearch	serviceMusic	comSearch	serviceMusic	comPlaylist
serviceMusic	comSearch	serviceGuide	comGuide	error	error
serviceGuide	comGuide	serviceBlog	comStart	error	error

4 実験

追加した処理部の動作の妥当性を確認する。評価項目を、命令文の実行誤り率および命令実行の応答時間とする。

4.1 実験準備

実験で用いるテストセットとして、テストセット1~4を用意した。テストセット1~3をそれぞれ表4~6に示す。

テストセット1は、実際に車載器を利用した際に得られた発話ログ1,441件の中から、組み合わせると重文型命令文となる、連続している2つの発話を抜粋したものである(表4)。

テストセット2は、テストセット1の2つの発話を3または4つの特定のワードで組み合わせで作った重文型命令文である。各テスト文中にある()内の|で区切られている単語は、それぞれの単語だけを含む文に展開するという意味で、()内に4つ単語を含む場合は、4つの重文型命令文が得られる(表5)。

テストセット3は、並列処理を行う命令文のペアを組み合わせた重文型命令文である(表6)。

テストセット4は、従来の車載器のオリジナルコーパスである(表なし、総文数3,519文)。

4.2 誤り率についての実験

テストセット2および3を用いることで2つの動作を含む重文型命令文の実行確認を行う。テストセット4を用いることで、元々実行できていたオリジナルコーパスの命令文(1動作の文)が実行できること、すなわち、提

表4 テストセット1

事例数	テスト文
1	“データベースを表示してください”, “エディターを表示してください”
2	“音楽を再生してください”, “天気図を表示してください”
3	“コンソールを消してください”, “降雨レーダーを表示してください”
4	“観光レーダーを表示してください”, “現在地を表示してください”
5	“音楽をかけてください”, “観光レーダーを表示してください”
6	“トレースデータベースを表示してください”, “検索モードにしてください”
7	“音楽をかけてください”, “シャッフルモードにしてください”
8	“音楽をかけてください”, “フォルダーの順にしてください”
9	“ブラウザを起動してください”, “グルメを表示してください”
10	“コンソールを閉じてください”, “音楽をかけてください”

表5 テストセット2

事例数	テスト文
1	データベースを表示し(て てから たら)エディターを表示してください
2	音楽を再生し(て てから たら)ながら天気図を表示してください
3	コンソールを消し(て てから たら)降雨レーダーを表示してください
4	観光レーダーを表示し(て てから たら)現在地を表示してください
5	音楽をかけ(て てから たら)ながら観光レーダーを表示してください
6	トレースデータベースを表示し(て てから たら)検索モードにしてください
7	音楽をかけ(て てから たら)シャッフルモードにしてください
8	音楽をかけ(て てから たら)フォルダーの順にしてください
9	ブラウザを起動し(て てから たら)グルメを表示してください
10	コンソールを閉じ(て てから たら)音楽をかけてください

表6 テストセット3

通番	テスト文
1	「曲名1」をかけて「曲名2」をかけてください
2	「曲名1」をかけてから「曲名2」をかけてください
3	「曲名1」をかけたら「曲名2」をかけてください

案手法に拡張前との互換性があることを確認する。

車載器1~4を用意し、それぞれの車載器で誤り率を求めた。車載器1は従来の車載器、車載器2は従来の車載器に文分割を追加した車載器、車載器3は文分割・分割評価を追加した車載器、車載器4は文分割・分割評価・命令合成を追加した車載器である。誤り率は表7の通りになった。

表7 誤り率の比較

車載器名	テストセット2	テストセット3	テストセット4
車載器1	1.00 (10/10)	1.00 (3/3)	0.00 (0/3519)
車載器2	0.00 (0/10)	1.00 (3/3)	0.39 (14/3519)
車載器3	0.00 (0/10)	1.00 (3/3)	0.00 (0/3519)
車載器4	0.00 (0/10)	0.00 (0/3)	0.00 (0/3519)

※各セルは誤り率(誤り数/入力数)

テストセット2の誤り率は、車載器1では1.00となり、車載器2~4では0.00となった。文分割を追加することで重文型命令文を実行できることが確認できた。

テストセット3の誤り率は、車載器1~3では1.00となり、車載器4では0.00となった。命令合成を追加することによって、並列動作を起こす重文型命令文を適切に実行できることが確認できた。

テストセット4の誤り率は、車載器2では0.39となり、車載器1, 3および4では0.00となった。文分割を追加することで過剰分割がおきていることが確認できたが、分割評価を追加することで過剰分割が抑えられていることが確認できた。

過剰な分割が抑制された命令文を表8に示す。予定登録の命令や、元々登録されていた重文型命令が分割されずに車載器4で処理できた。なお、車載器2で実際に表

8の文を実行すると求めていないサービスが実行された。

表8 過剰分割の抑制が成功した命令文

通番	オリジナルコーパスにおける重文型命令文
1	繰り返して案内してください
2	繰り返して説明してください
3	道の駅に限定してレーダーに表示してください
4	レーダーは道の駅に限定して表示してください
5	保存して閉じてください
6	浜坂はやめて浜村です
7	仁風閣はやめて鳥取港に行きます
8	仁風閣はやめて砂丘海岸に行きます
9	17時に解散して帰ります
10	明後日は8時に京都に向けて出発です
11	12月30日に福岡に向けて出発します
12	シャッフルしてビデオを再生してください
13	ビデオをシャッフルして再生してください
14	サイズをノーマルにして表示してください

4.3 応答時間についての実験

2つの動作を、1つずつ命令した時と、重文型命令によって一度に命令した時の応答時間を比較するために、テストセット1および2の平均応答時間を求めた。結果は表9の通りになった。

表9 平均応答時間

テストセット	1文あたりの平均応答時間(秒)
1	11.67
2	7.57

テストセット2の平均応答時間は、テストセット1の平均応答時間に比べて約4秒の短縮が確認できた。自動車内において操作にかかる時間が4秒短縮されることは有益であると考えられる。

5 おわりに

本稿では、車載器上で2つの動作を含む重文型の命令に対して実行を可能にした。手法としては、従来の車載器の発話文解析器に文分割・分割評価・命令合成の3つの処理を追加した。文分割では、重文型命令文を分割し、解析可能とした。分割評価および命令合成では、文分割に伴う実装上の問題に対応した。以上より文分割は複雑な文の解析に有用であることが分かった。さらに、既存の1動作のための解析器が流用できる点も利点である。

本稿では複雑な文を重文型の言い方に限定したが、複雑な文には重文型以外の言い回しも多くある。今後は重文型の言い方以外の複雑な文にも文分割が有用であるか、研究を進めて確認していきたい。

謝辞

本研究は JSPS 科研費 JP19K12548 の助成を受けたものです。

参考文献

- [1] 徳久 雅人, 木村 周平: 小型計算機におけるサービス指向発話文解析, 自然言語処理, 26(3), pp.545-578, 2019.
- [2] Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorin: Split and Rephrase. Natural Language Processing(EMNLP), pp. 606-616, 2017.
- [3] 池原 悟, 阿部 さつき, 徳久 雅人, 村上 仁一: 非線形な表現構造に着目した重文と複文の日英パターン化, 自然言語処理, 11(3), pp.69-95, 2004.