

2019 年度 卒業論文

対話状態追跡における
対話行為タグを用いた重要対話履歴抽出

指導教員

李 晃伸 教授
酒向 慎司 准教授

名古屋工業大学 工学部
情報工学科

2016 年度入学 学籍番号 28114006

石島 侑弥

No.37

BACHELOR DISSERTATION

EFFECT OF EXTRACTING SIGNIFICANT
DIALOG HISTORY BASED ON DIALOG ACT
IN DIALOG STATE TRACKING

BACHELOR OF ENGINEERING

FEBRUARY 2020

YUYA ISHIJIMA

Supervisors: Professor Akinobu Lee
Associate Professor Shinji Sako

Department of Computer Science
Nagoya Institute of Technology

目 次

1	まえがき	1
2	対話システムにおける対話状態追跡	3
2.1	タスク指向型対話システム	3
2.2	対話状態追跡	4
2.3	ルールベースの対話状態追跡	5
2.4	統計的機械学習による対話状態追跡	6
2.5	深層学習による対話状態追跡	7
2.6	Dialogue System Technology Challenge 8	9
3	BERT に基づく対話状態追跡	12
3.1	BERT とは	12
3.2	BERT に基づく対話状態追跡	13
3.2.1	スキーマ埋め込みモジュール	13
3.2.2	状態更新モジュール	14
4	対話行為タグを用いた重要対話履歴抽出	17
4.1	対話行為	17
4.2	対話行為タグを用いた重要対話履歴抽出	18
5	対話状態追跡における各対話行為の重要性調査	21
5.1	データセット	21
5.2	実験手法	21
5.3	調査結果	23
6	評価実験	26
6.1	実験条件	26
6.2	評価指標	27
6.3	実験結果	27
7	むすび	31
	謝辞	32
	付録	36
A.1	各サービスのスキーマ	36

図 目 次

2.1	タスク指向型対話システムの基本構成	4
2.2	対話管理部の処理の流れ	4
2.3	if-then ルールによる対話状態追跡の例	5
2.4	対話状態の確率的表現である信念 b のイメージ図	6
2.5	部分観測マルコフ決定過程 (POMDP) の流れ	7
2.6	言語理解と対話状態追跡を共同で学習するモデル [12]	8
2.7	スキーマの構成	10
2.8	DSTC8-Track4 のイメージ図 [3]	10
3.1	Fine-Tuning の例 [16]	13
3.2	ベースラインモデルの構成	14
4.1	重要対話履歴抽出モジュールを加えた提案モデル	18
5.1	1 対話あたりの各対話行為の発話中でスロット値候補が出現した回数	24
5.2	1 対話あたりの各対話行為の発話中で出現したスロット値候補が実際に スロット値に反映された回数	24
6.1	ベースラインと OFFER を用いた提案手法の出力結果	28

表 目 次

4.1	SGD データセットにある対話例の一部	19
5.1	SGD データセットのドメインごとの対話数（括弧内はサービスの個数）	22
5.2	SGD データセットの学習用，検証用が扱うサービスのリスト	22
5.3	SGD データセットの学習用と検証用データにおける各対話行為の出現回数	23
6.1	各対話行為で提案手法を行った際の結果	27
6.2	従来手法との比較	29
A.1	各スキーマのスロットとインテント	36

第1章

まえがき

対話システムは、ユーザが持つ要求を対話によって達成することを目的とするタスク指向型対話システムと、対話そのものを目的とする雑談対話システムに大別される。タスク指向型対話システムの例として Apple の「Siri」、NTT の「しゃべってコンシェル」などがあるが、その多くが行う対話は一問一答型である。一問一答型は簡単に実用化できるため多くのシステムで用いられているが、ユーザと応答を繰り返してユーザの細かな要求に対応するという対話にはできない。これはシステムが、ユーザが過去の発話で発した要求などといったこれまでの対話の流れを理解していないからである。ゆえに、システムがユーザとの対話の流れを理解するための技術である対話状態追跡 (Dialogue State Tracking) が必要とされている。

一問一答型ではないタスク指向型対話システムでは、対話状態と呼ばれる対話中のユーザの目的や要求を保持するための枠組みを用いて対話の流れを理解する。対話システムにおける対話状態追跡は、ユーザが発話する度に対話履歴を考慮してその対話状態を維持あるいは更新する役割を担う。そしてシステムは、更新された対話状態を見て次にユーザに対して行う行動や発話を決定することで、ユーザの目的達成に向かった対話を行う。対話状態追跡を行わない場合、システムが過去の対話で得た情報を保持できないのに加え、次の行動を選択できない。そのため、対話状態追跡はタスク指向型対話システムにとって重要な要素である。

現在の対話状態追跡はニューラルネットワークなどの機械学習に基づく手法が盛んに行われている。中でも、深層学習を用いた End-to-End 型のモデル [1, 2] が優れた結果を残しており、発話文から直接対話状態を出力することが可能となった。このようなモデルでは、対話の流れを捉えるために対話履歴を入力に用いる。

対話履歴を全て入力することで対話の流れを捉えられるが、現在の発話との依存関係がない発話が含まれる上に計算量が多くかかる。そのため、従来研究では対話履歴として直近の数発話を入力とする場合が多い。しかし、直近の数発話では対話状態の推定に貢献しない不要な発話が含まれ、必要な発話が含まれない可能性がある。

そこで本研究では、発話者の意図を示す対話行為タグを用いて、対話履歴の中から

対話状態の推定に貢献すると思われる発話を抽出する手法を提案する．対話行為タグによって各発話を意図ごとに分類し，特に状態に影響を与える発話を選択的に抜き出すことで，計算量を抑えつつ長い対話履歴を考慮することが可能であると期待できる．

対話システムの国際コンペティションである Dialog System Technology Challenges 8 Track4 Dialogue State Tracking (DSTC8-Track4)[3] において，対話状態追跡のデータセットとベースラインモデルが公開されている．本研究では DSTC8-Track4 のデータセットとモデルを用いて実験を行い従来法との比較を行う．

以下，第 2 章では近年の対話システムにおける対話状態追跡とその問題点について，第 3 章では本研究におけるベースラインモデルについて述べる．また，第 4 章では提案手法について述べる．第 5 章ではどの対話行為タグを持つ発話が重要なのかを調査した予備実験の結果を示し，第 6 章では提案手法の結果と分析，さらに従来手法との比較を示す．第 7 章では，むすびと今後の課題について述べる．

第2章

対話システムにおける対話状態追跡

本章ではタスク指向型対話システムと対話状態追跡の説明をした後，これまでの対話システムがどのような手法によって対話状態追跡を行ってきたか，また何が問題であったかを順に説明する．そして，本研究が取り扱ったコンペティション DSTC8-Track4 について説明する．

2.1 タスク指向型対話システム

ユーザとシステムが行う対話は大きく分けると，タスク指向型対話と雑談対話の 2 種類が存在する．タスク指向型対話とは，ユーザが特定の目的を持ち，ユーザとシステムが応答を繰り返すことでその目的を達成する対話のことである．雑談対話とは，対話そのものを目的として行うもので，明確な目的が存在しない対話のことである．また，前章で述べた一問一答型の対話はタスク指向型対話の一部で，対話状態を用いずに一回の応答でユーザの要求に応える対話のことである．一問一答型の対話は一回の応答で対話が完結するため，ユーザの細かい要求に応えられないという問題がある．

タスク指向型対話システムとは，タスク指向型対話を行う対話システムのことであり，ホテル案内，レストラン案内，交通案内など様々なドメインを扱う．タスク指向型対話システムの実用例として，バス案内システムとして実運用されている「Let's Go システム」[4] や大学のキャンパス案内を行う「正門メイちゃん」[5] が挙げられる．

タスク指向型対話システムの基本構成を図 2.1 に示す．対話システムは大きく分けて，音声認識部，言語理解部，対話管理部，応答生成部の 4 つのモジュールで構成されている．音声認識部はユーザの発話音声を認識し発話文に変換し，言語理解部はその発話文をシステムが理解できる形に変換する部位である．言語理解部が行うことは，発話文からユーザが行おうとしているタスクのドメインが何か推定するドメイン推定，そのドメイン中でのユーザの目的は何か推定するユーザの目的推定，発話中に含まれるスロットの値候補を見つけるスロット抽出の 3 つである．スロットとはドメインごとに与えられる属性のことで，システムはスロットと値の組によってユーザの要求を

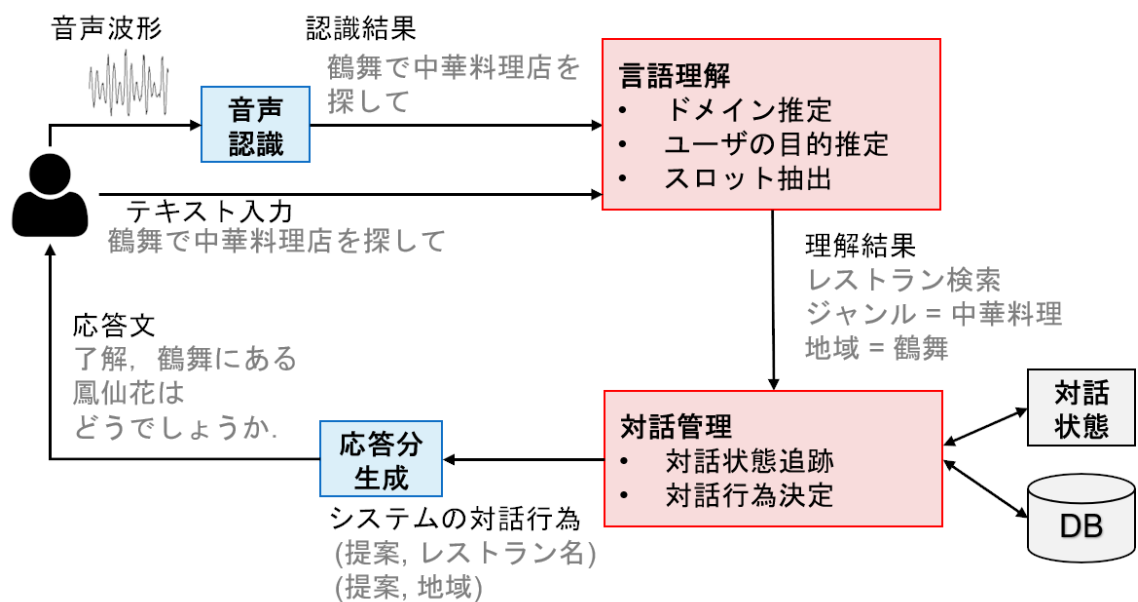


図 2.1 タスク指向型対話システムの基本構成

表現する。そして対話管理部は、過去の発話も考慮してユーザの目的と要求を推定する対話状態追跡と、対話状態から不足している情報を検出し次のシステムの行動を決定する対話行為決定を行う。最後に、応答文生成部でシステムの対話行為に沿った応答文を生成する。このようにタスク指向型対話システムは、4つのモジュールでユーザとのタスク指向型対話を可能にする。

2.2 対話状態追跡

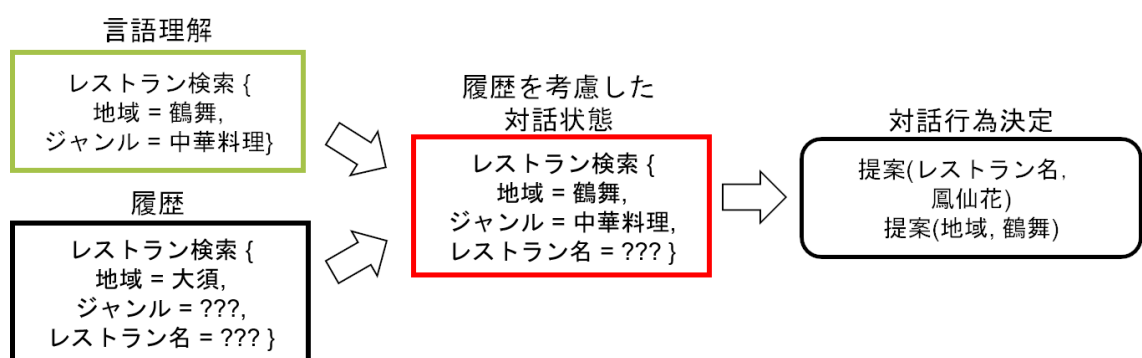


図 2.2 対話管理部の処理の流れ

対話システムにおける対話状態追跡は、図 2.1 に示した通り対話管理部の機能の1つである。対話管理部の処理は図 2.2 のような流れになっている。対話状態追跡は言語理

解の結果と履歴を考慮して、対話状態を出力する。この図において、履歴は一例として前のターンの対話状態を用いている。対話状態はレストラン検索などといった目的と、スロットと値の組を用いた辞書形式で表現される。システムは対話状態にある目的を達成するために必要なスロットを埋めていく作業を行う。つまり、対話状態追跡は対話状態のスロット値を言語理解の結果から得たスロット値候補に変更するか否かを判断する役割を担う。また、対話行為決定では対話状態の不足している情報を検出して、目的達成に向けて必要なシステムの対話行為を選択する。

システムは対話状態を見て次の行動を決めるため、対話状態追跡が対話状態の推定を誤ると、ユーザの要求にそぐわない対話を行うことになる。ゆえに対話状態追跡はタスク指向型対話システムにとって重要な要素である。

2.3 ルールベースの対話状態追跡

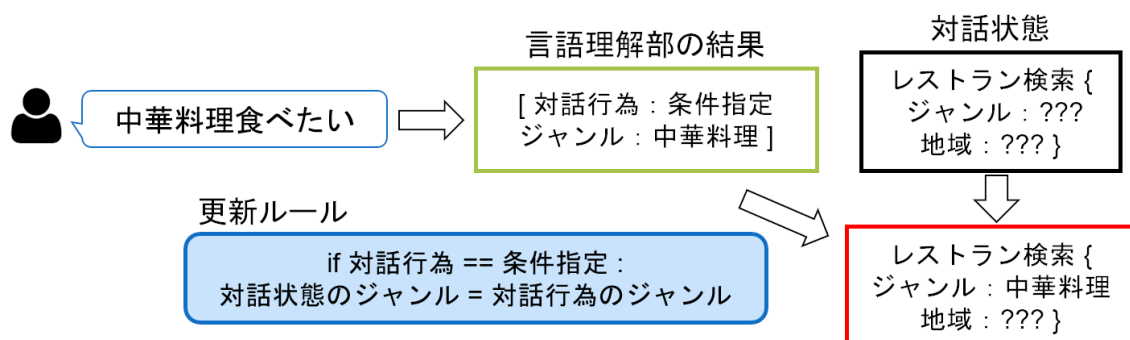


図 2.3 if-then ルールによる対話状態追跡の例

ルールベースの対話状態追跡とは、予め人間が決めた規則に従って対話状態を更新する対話状態追跡のことである。その規則は更新ルールと呼ばれ、プログラムやグラフによって定められる。

一例として、if-then 文で記述されたプログラムを用いた対話状態追跡を図 2.3 に示す。ルールベースの対話システムは図 2.3 のように、言語理解部でユーザの対話行為も推定する。そして対話状態追跡は、言語理解部から得たユーザの対話行為によって条件分岐を行い、対話状態を更新していく。

ルールベースの対話状態追跡は、後述する統計的機械学習や深層学習による対話状態追跡に比べて容易に行えるため、ほとんどの商用対話システムで採用されている。ただし、複雑な対話を扱うには多くの規則を定義する必要がある。また、言語理解部の結果に大きく依存するため、言語理解部で誤りが発生すると、自ずと間違った情報が対話状態に追加されるという問題がある [6]。

2.4 統計的機械学習による対話状態追跡

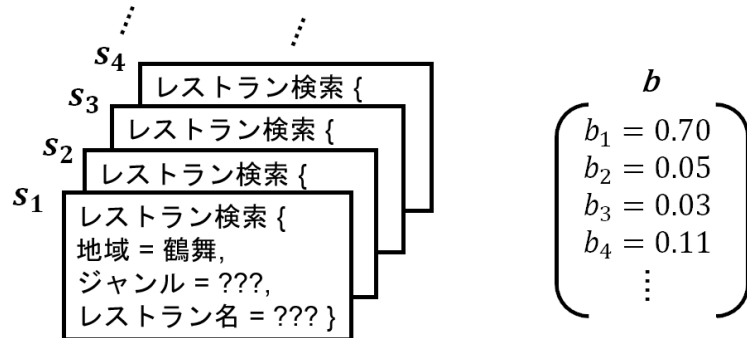


図 2.4 対話状態の確率的表現である信念 b のイメージ図

統計的機械学習による対話状態追跡では、言語理解部からの誤りに対する頑健性を得るために、対話状態を確率的に扱う。これは、ユーザ発話の解釈に不確実性が存在するため、対話状態を直接観測できないとしているからである。対話状態の確率的表現は、全ての対話状態にわたる確率分布である信念 b を用いる。信念に関する図を図 2.4 に示す。また、言語理解部の出力はノイズを含むユーザ入力 of 観測 o とし、対話状態の集合を I_s 、システムの行動の集合を K とする。対話状態の推定は、ターン t の対話状態を $s^t \in I_s$ 、システムの対話行為を $a^t \in K$ 、観測状態を $o^t \in I_s$ 、ユーザの対話状態が s である信念（確率変数）を b_s^t とし、式 2.1 を推定する問題となる。

$$b_s = P(s|o^{1:t}) \quad (2.1)$$

$P(s|o^{1:t})$ は、これまでの観測状態から推定した全てのユーザの対話状態にわたる確率分布である。つまり、履歴としてこれまでの観測状態を用いている。

統計的機械学習による対話管理の一例として、部分観測マルコフ決定過程（Partially Observable Markov Decision Process ; POMDP）[7, 8] を用いた手法を説明する。POMDP ベースのモデルは、信念状態の追跡と強化学習という 2 つのキーアイデアを組み合わせている [8]。強化学習とは、システムの行動に対して報酬を与えることで、システムに報酬を最大化する行動を学習させるという機械学習の一種である。

POMDP を用いた対話管理は図 2.5 のように、言語理解部の出力を全ての対話状態にわたる確率分布 $p(o'|s')$ で表す。そして、信念状態の追跡には対話状態を最新の対話状態に更新する状態更新関数を用いる。状態更新関数は現在の対話状態 s^{t+1} をこれまでの観測状態系列 $o^{1:t+1}$ から求める式である。状態更新関数は式 2.2 に示す。

$$b' = P(s^{t+1}|o^{1:t+1}) \propto P(o'|s'_j) \sum_{s_i} P(s'_j|s_i, \hat{a}_k) b^t \quad (2.2)$$

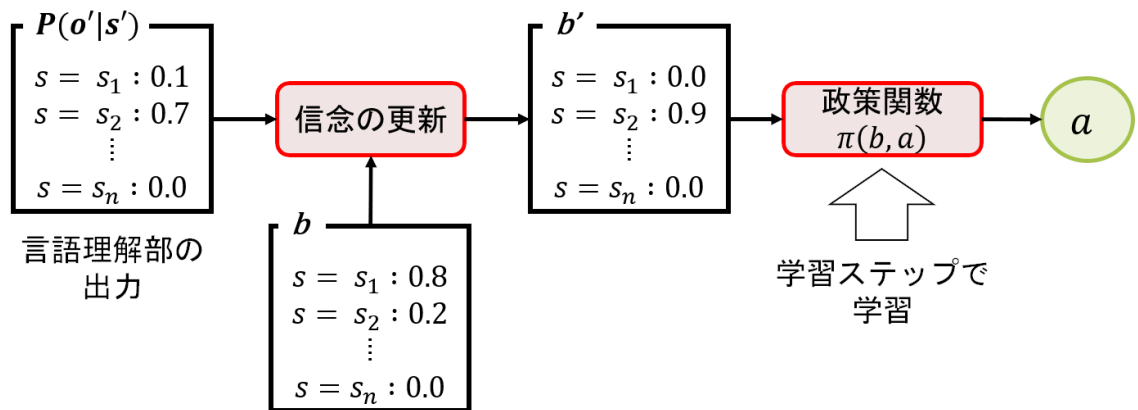


図 2.5 部分観測マルコフ決定過程 (POMDP) の流れ

式 2.2 では、 $P(o'|s'_j)$ が観測確率、 $P(s'_j|s_i, \hat{a}_k)$ が状態遷移確率、 b_t が現在の信念を表す。信念の更新をした後、状態 s で対話行為 a を選ぶ確率である政策関数 $\pi(b, a)$ を用いて次のシステムの対話行為を決める。POMDP での報酬は状態行動価値 (Q 値) などを用いる。Q 値とは、ある状態 s のとき対話行為 a を行い得られる報酬の期待値である。POMDP を用いた対話管理は、その報酬を用いて政策関数を強化学習により最適化していき、最適なシステムの対話行為を選ぶことを目指す。

POMDP を用いた対話管理は、全ての状態にわたる信念の分布を維持することにより、システムは全ての可能な対話経路を効果的に並行して追求し、最も可能性の高い状態だけではなく、全ての状態にわたる確率分布に基づいて次の対話行為を選択することが可能になる [8]。問題としては、複雑な対話を取り扱うと対話状態のパターンが増加するため、信念の追跡が困難になることがある。また、ルールベースほどではないが言語理解部の誤りにより性能が悪化する。

2.5 深層学習による対話状態追跡

深層学習は、人間の神経細胞の仕組みを模したシステムであるニューラルネットワークがベースになっている [9]。そして、ニューラルネットワークを多層にして用いることで、データに含まれる特徴を段階的により深く学習することが可能になる。深層学習を行う場合、ラベル付けされたデータが必要になるが、現在は DSTC2[10] や Multi-Woz[11] など、ユーザの対話状態やシステムの対話行為などをラベル付けした対話データセットが公開されている。このようなデータが公開されたことで、深層学習による対話状態追跡の研究が盛んに行われている。

深層学習による対話状態追跡では、通常発話文を入力として直接対話状態を出力す

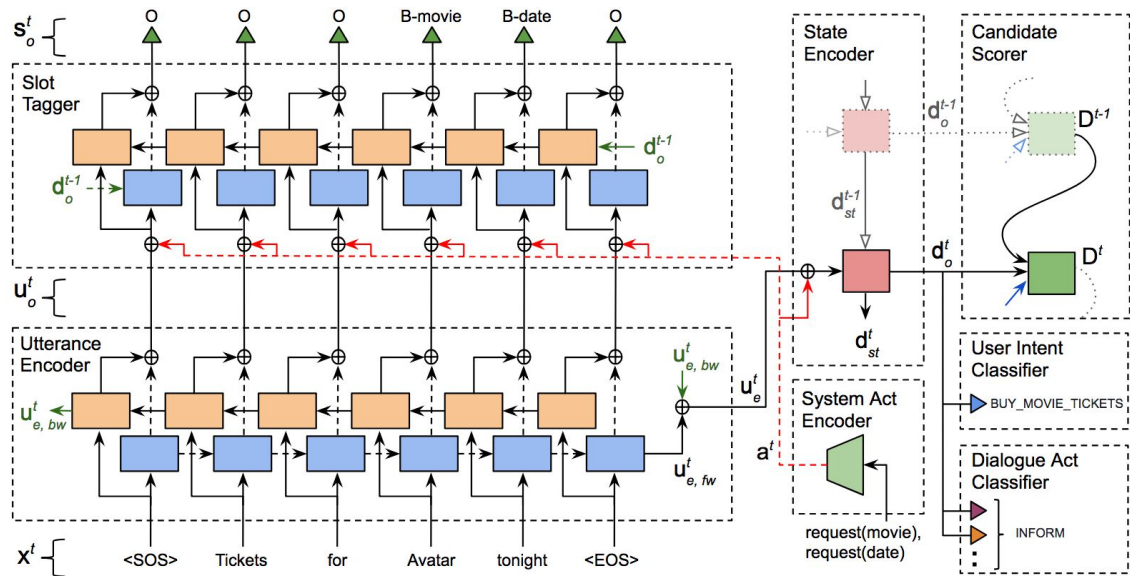


図 2.6 言語理解と対話状態追跡を共同で学習するモデル [12]

る [1, 2]. 対話状態であるインテントや各スロットの値に関してはデータセットによって事前に候補が与えられている. そのため, 発話文から得た情報を基にそれぞれの全候補にわたる確率分布を出力し, 確率最大のものをインテントあるいはスロット値として選択する. しかし, そのようなモデルは候補として与えられていない値を推測できないため, 言語理解などを用いて候補外の値も推定可能にするモデル [12, 13, 14] の研究が行われている. そのようなモデルの例として, Rastogi ら [12] のモデルを図 2.6 に示して説明する. このモデルは, Utterance Encoder, System Act Encoder, State Encoder によってユーザの発話, システムの対話行為, 対話状態の特徴量を得る. これら特徴量は言語理解部と対話状態追跡部で共有される. 言語理解部である Slot Tagger は発話中の単語ごとにラベルを付けてスロット値の候補となる単語を見つける. ラベルは, スロット値の開始地点を示す B ラベル, スロット値の継続を示す I ラベル, スロット値でないことを示す O ラベルで構成される IOB ラベルが用いられる. B ラベルと I ラベルはスロットの数だけ作成される. Slot Tagger で見つけられたスロット値はスロット値候補リストに追加される. 対話状態追跡部は, User Intent Classifier でユーザの目的を推定し, Dialogue Act Classifier でユーザの対話行為を推定し, Candidate Scorer でスロット値候補リスト中の候補のランク付けを行いスロットに値を割り当てる.

言語理解と対話状態追跡を共同で学習することで, 言語理解部で発生した誤りが対話状態追跡部に伝搬するのを防ぐことが可能になる. また, 言語理解部と対話状態追跡部でエンコーダのネットワーク層を共有できるため, 性能の向上とネットワーク層のパラメータの数の削減が同時に行える [12]. しかし, このようなモデルは新しいドメ

インに適応させたい場合、言語理解部の IOB ラベルを増やして再学習する必要がある。したがって、現在は新しいドメインに再学習なしで適応可能な拡張性のある対話状態追跡の研究が注目されている。

2.6 Dialogue System Technology Challenge 8

Dialog System Technology Challenges 8 (DSTC8) [3] とは対話に関するコンペティションである。2013 年に最初の Dialog State Tracking Challenge（対話状態追跡タスク）が開催された。第 6 回から対話状態追跡だけでなく、対話破綻検出や応答生成といった対象となるタスクの幅が広がり、略称は DSTC のままで現在の Dialog System Technology Challenges へと名称が変更された。

ここでは本研究で取り組んだ対話状態追跡タスクの DSTC8-Track4 について説明する。昨今、ホテル案内や交通案内などのドメインを行えるサービスが多く存在している。サービスごとに様々な バックエンド API が定義されるため、同じドメインでもシステムが理解できず、対応させるには再学習が必要となる。DSTC8-Track4 では、現在の対話システムの課題の 1 つである新サービスへの拡張を再学習なしで可能にすることを目的としており、そのための対話状態追跡モデルを作成する。

本タスクでは、ユーザが対話で達成したい目的をインテントと呼ぶ。インテントはサービスごとにいくつか存在する。例えばレストランサービスであれば、レストラン検索（FindRestaurant）とレストラン予約（ReserveRestaurant）が存在する。システムはサービスに対応するインテントの変化を追跡することで、ユーザの目的に合った質問や返答を返すことが可能になる。

また、本タスクではサービスを定義する枠組みであるスキーマが用意されている。スキーマは、サービスごとにスロットのリストとインテントのリストを持つ。スキーマの構成は図 2.7 に示す。スキーマには、サービス、スロット、インテント全てに説明文が付与されている。この説明文によって、システムが未知のサービス、スロット、インテントを理解することを可能にする。スロットに関してはスロット値候補を持つ候補ありスロットと、候補を持たない候補なしスロットに分けられる。候補ありスロットはスロット値を事前に与えられている候補から選択するが、候補なしスロットはスロット値を発話文中から抜き出す必要がある。図 2.7 では、乗り換えの回数を表す “number_stops” スロット、航空会社を表す “airlines” が候補ありスロット、出発地を表す “origin” スロット、到着地を表す “destination” スロットなどそれ以外が候補なしスロットにあたる。また、各インテントの達成に必要なスロットという情報を持つため、システムはスキーマによってサービスの バックエンド API が何を必要としているかを理解できる。

サービス名 : Flight_A 説明文 : Find your next flight		サービス
名称 : origin 説明文 : Starting city for the trip スロット値候補 : []	名称 : destination 説明文 : Ending city for the trip スロット値候補 : []	スロット
名称 : depart 説明文 : Start date for the trip スロット値候補 : []	名称 : return 説明文 : Date of the return flight スロット値候補 : []	
名称 : number_stops 説明文 : Number of layovers in the flight スロット値候補 : ["0", "1"]	名称 : airlines 説明文 : Name of airline スロット値候補 : ["United Airlines", "Delta Airlines", . . .]	
名称 : SearchFlight 説明文 : Search for round-trip flights 達成に必要なスロット : [origin, destination, depart, return]	名称 : ReserveFlight 説明文 : Reserve a round-trip flight 達成に必要なスロット : [origin, destination, airlines, depart, return]	インテント

図 2.7 スキーマの構成

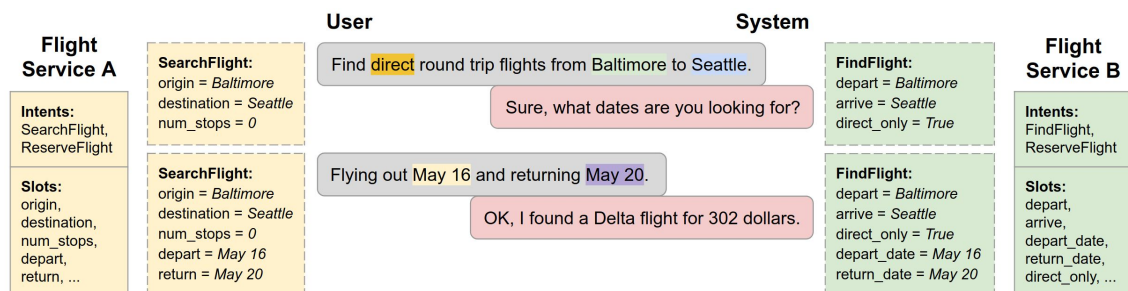


図 2.8 DSTC8-Track4 のイメージ図 [3]

本タスクは、学習時、検証時、テスト時それぞれで扱うサービスが異なる。図 2.8 に示した通り、同じ “Flight Service” でもインテントの候補とスロットが異なる名前で定義されている。そして、どちらのサービスを扱うとしても、ユーザの目的と要求を同様に推定できる対話状態追跡モデルを作成する必要がある。

参加者はベースラインモデルと整形された対話データセットが配布され、そのデータセットなどを用いて作成したモデルの学習と評価を行い、モデルと評価結果を提出する。

配布されたデータセットは SGD データセットと呼ばれ、人とシステムの間のラベル付きタスク指向型対話を持つ。各対話はユーザとシステムが交互に 1 発話ずつ行い、1 発話につき 1 ターンとしている。1 ターンごとに、発話中に存在するスロット値候補の位置を範囲で示すスロット値範囲リストが与えられる。スロット値範囲リストは以

下の要素を持つ.

- スロット - スロットの名前
- 開始位置 - 発話中にあるスロット値候補の開始位置
- 終了位置 - 発話中にあるスロット値候補の終了位置

対話状態はユーザのターンのみ与えられる. 対話状態は以下の要素を持つ.

- インテント (None を含む)
- 要求スロット - ユーザが値を要求しているスロットのリスト
- スロット値 - 各スロットに当てはまる値

対話状態の推定は, この3つの要素を全て推定する必要がある. また, 対話行為はシステムのターンのみ与えられる. 本タスクの対話行為は, システムがあるスロットに対してどのような行動をしたのかを示すため, 以下の要素を持つ.

- 対話行為タグ - 対話行為の種類を示す
- スロット - 対話行為が関与しているスロット
- スロット値 - 対話行為で用いたスロット値

例えば “OFFER (origin, Baltimore)” ならば, 出発地として “Baltimore” を提案するという行動を示す.

本タスクでは, 発話文, スキーマ, 対話行為を入力として, 対話状態であるインテント, 要求スロット, スロット値を推定する. スロット値に関して, 候補ありスロットではスロット値をスキーマから与えられるスロット値候補から推定し, 候補なしスロットではスロット値となる文字列の範囲を発話中から推定する. 候補なしスロットの推定では, 既にスロットに割り当てられている値と同じ意味の値 (例: “at afternoon 1:30”, “1:30 pm”) を推定することがあるため, 候補なしスロットには複数の値が割り当てられることもある.

第3章

BERT に基づく対話状態追跡

本研究で提案する手法を試すために，DSTC8 Track4 で公開されたベースラインモデル [15] を用いた．本章では，ベースラインモデルとして用いた BERT に基づく対話状態追跡について述べる．

3.1 BERT とは

BERT (Bidirectional Encoder Representations from Transformers) とは，Google が 2018 年に発表した自然言語処理モデルである [16]．Recurrent Neural Network などを使わずに Attention 機構のみを用いた Transformer [17] と呼ばれるモデルがある．Transformer が 文脈を単一方向しか考慮しないのに対して，双方向考慮するように改良したのが BERT である．BERT の特徴として，事前学習モデルであることが挙げられる．従来のニューラルネットワークを用いた自然言語処理モデルは，言語理解や感情推定など特定のタスクのみに対応しており，タスクに応じた出力を行う．しかし，事前学習モデルである BERT は既存のタスク実行モデルに付け加えるだけで，そのモデルの性能を向上させることが可能である．

BERT を付け加える際には，入出力をタスクに合わせたものに置き換えて追加学習する Fine-Tuning という手法が用いられる．そのようなモデルの中での BERT は，入力文の埋め込み表現を得るために用いられる．自然言語処理における埋め込みとは，文や単語，文字など自然言語の構成要素に対して，何らかの空間におけるベクトルを与えることであり，ベクトルには文や単語の特徴量が格納される．図 3.1 のように，BERT は 2 つの文章を入力として，単語単位に分割する．分割したものをトークンと呼ぶ．また，入力した文の初めに “[CLS]”，2 つの文章の末尾に “[SEP]” という特殊なトークンを挿入する．そして，BERT は文章ペア全体の埋め込み表現とトークンレベルの埋め込み表現を抽出する．BERT に付け加えたタスク実行モデルは，これら埋め込み表現を用いて各タスクに合わせた出力を学習していく．また，BERT 自体もタスクに合うように埋め込み表現を更新していく．

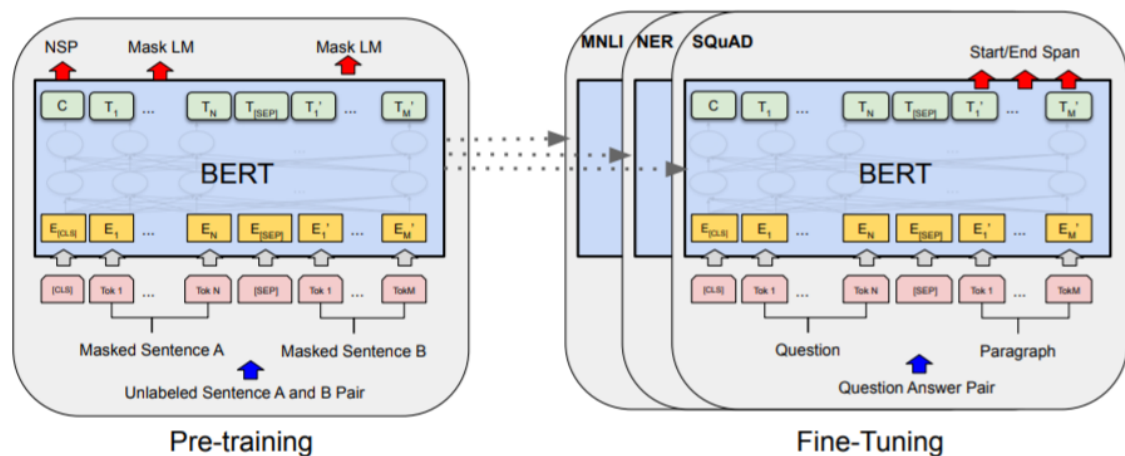


図 3.1 Fine-Tuning の例 [16]

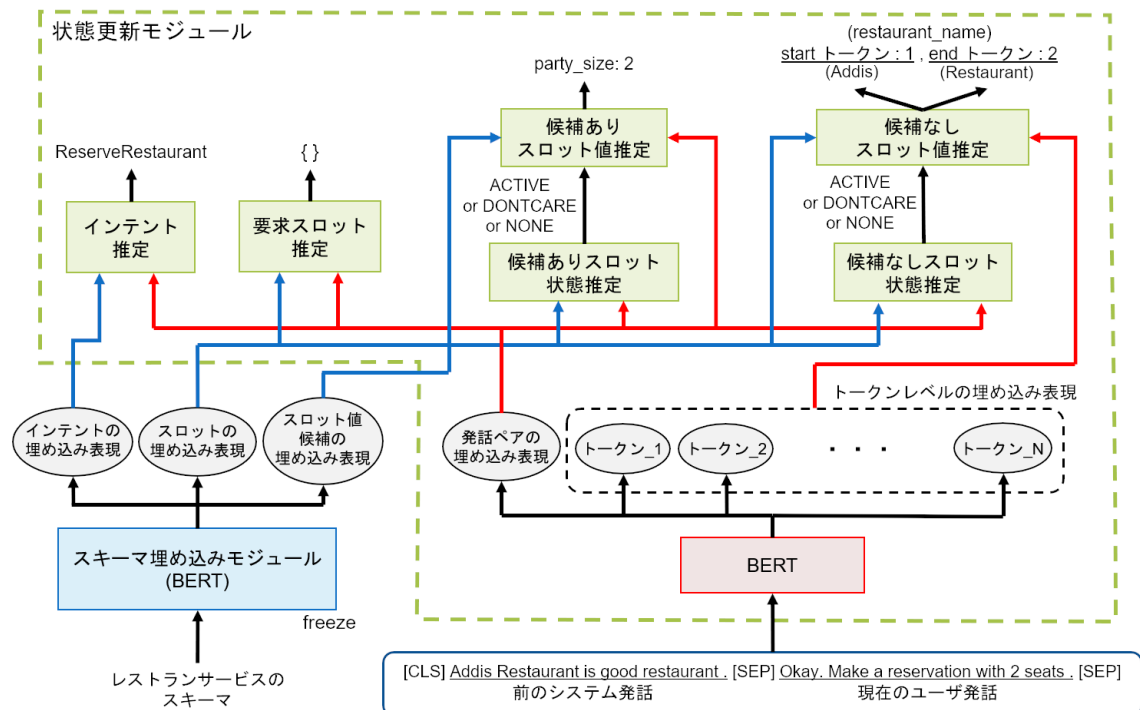
BERT は汎用性の高さが評価され注目を浴びている．加えて，少ないデータで追加学習を行うだけで動作するため，自然言語処理分野の長年の課題であったデータ不足を解決する．自然言語処理分野でデータが不足していた原因は，データ収集やラベルの付与に多大なコストと時間がかかることが挙げられる．しかし，BERT の事前学習では Wikipedia や BooksCorpus などから得た大量のラベル未付与のデータを用いて，前後の文脈を加味したトークンの埋め込み表現や2つの文章の関係性を事前に学習する．そのため，少量のラベル付きデータで追加学習を行うだけでタスクの実行が可能になる．ゆえに，本研究でも BERT による対話状態追跡モデルをベースラインとして用いる．

3.2 BERT に基づく対話状態追跡

本トラックで提供されたベースラインモデルは，BERT を用いた Fine-Tuning で対話状態追跡を行うモデルである．ベースラインモデルは図 3.2 のように，スキーマ埋め込みモジュールと状態更新モジュールの2種類のモジュールで構成されている．

3.2.1 スキーマ埋め込みモジュール

スキーマ埋め込みモジュールは，図 3.2 のように，BERT を用いてスキーマに含まれているインテント，スロット，スロット値候補の埋め込み表現を抽出する．インテントは，{[CLS] [サービスの説明] [SEP] [インテントの名称] [インテントの説明] [SEP]} を BERT への入力として，埋め込み表現を抽出する．スロットは，{[CLS] [サービスの説明] [SEP] [スロットの名称] [スロットの説明] [SEP]} で入力し，埋め込み表現を得る．



BERT に送られ、BERT は発話ペアの埋め込み表現とトークンレベルの埋め込み表現を出力する。その後、インテント推定、要求スロット推定、スロット値推定を共同で学習する。モデルは直近の発話 2 つを入力として使用するため、前のユーザターンからの対話状態の差異を推定するように学習する。そのため、スロット値の推定はスロットが更新されるか否かを出力するスロット状態推定と更新されると判断されたスロットの値を推定するスロット値推定の 2 段階で行われる。また、スロット値の推定は候補ありスロットと候補なしスロットに分けて行われるため、状態更新モジュールは合計で 7 つの推定を共同で学習する。

- インテント推定は、現在のターンにおけるインテントを推定する。インテント推定では、まず初めに、発話ペアの埋め込み表現とスキーマの埋め込みに含まれる各インテントの埋め込み表現を連結する。各インテントの埋め込みには、インテント無しを表す NONE インテントの埋め込み表現も含まれる。連結された埋め込み表現は訓練可能な射影を使用してロジット値に射影される。ロジット値 (logit) とは、確率 p で起こる事象 A について、 A が起こる確率と起こらない確率の比の対数 $\log(p/(1-p))$ である。得られた全てのインテントのロジット値は Softmax 関数で正規化され、サービス中の全てのインテントにわたる確率分布が生成される。そして、確率が最大のインテントがそのターンにおけるインテントであると推定される。
- 要求スロット推定は、ユーザがスロット値を要求しているスロットを推定する。要求スロット推定では、発話ペアの埋め込み表現と全てのスロット埋め込みを用いて、インテント推定と同様の方法でロジット値を取得する。そして、各ロジット値は Sigmoid 関数を使用して正規化され、その値を各スロットのスコアとする。スロットのスコアが 0.5 を超える場合、そのスロットは要求スロットであると推定される。
- スロットの状態として、NONE, DONTCARE, ACTIVE を定義する。NONE はスロット値が変更されないことを示し、DONTCARE はユーザがそのスロットに対して関心がないことを示し、ACTIVE はスロット値が更新されることを示す。スロット状態推定では各スロットに対してどの状態であるかを推定する。入力、候補ありスロット状態推定ならば候補ありスロットの埋め込み表現、候補なしスロット状態推定ならば候補なしスロットの埋め込み表現を発話ペアの埋め込み表現と連結する。連結した埋め込み表現から、訓練可能な射影を使用して、NONE, DONTCARE, ACTIVE のロジット値を取る分布を取得する。スロットの状態が

NONE と推定された場合、スロット値が変更されない。DONTCARE と推測された場合、スロット値として dontcare が割り当てられ、ACTIVE と推測された場合、次に行われるスロット値推定でスロット値が推定される。

- 候補ありスロット値推定は候補ありスロットに割り当てられる値を推定する。候補ありスロット推定では、発話ペアの埋め込み表現とスロット値候補の埋め込み表現を連結して、インテント推測と同様の方法で各スロット値候補のロジット値を得る。得られた全てのロジット値は、Softmax 関数を使用して正規化され、スロット値候補の確率分布が生成される。そして、確率が最大である値がスロットに割り当てられる。
- 候補なしスロット値推定は、入力文中の単語からスロット値となる文字列を抜き出す必要があるため、スロット値となる文字列の範囲を推定するように学習する。範囲の推定では、BERT モデルから取得したトークンレベルの埋め込み表現と候補なしスロットの埋め込み表現を連結する。そして、連結した埋め込み表現は訓練可能な射影を使用してロジット値に変換される。得られた全てのトークンのロジット値は、Softmax 関数を使用して正規化され、全てのトークンにわたる確率分布にされる。この分布は、範囲の開始位置である start を推定するように学習する。また、異なる重みセットを使用した同様の手順で、範囲の終了位置である end の分布も推定する。推論中、start と end の確率の和が最大になるトークンの組み合わせが推定され、その範囲にある文字列がスロットに割り当てられる。

これら7つの推定では、それぞれで交差エントロピー誤差を計算する。計算した7つの誤差の和をこのモデルの誤差として学習に用いる。

第4章

対話行為タグを用いた重要対話履歴抽出

本章では，SGD データセットに含まれる対話行為タグの種類を説明したのち，本研究で提案する対話行為タグを用いた重要対話履歴について説明する．

4.1 対話行為

SGD データセットに存在する対話行為タグは 10 種類である．SGD データセットでは，システムの発話にのみ対話行為タグが与えられる．対話行為タグの種類と意味に関しては以下に示す．

- INFORM - スロットの値をユーザに通知
- REQUEST - ユーザにスロットの値を要求
- CONFIRM - ユーザから得たスロットの値を確認
- OFFER - ユーザにスロットの値を提案
- NOTIFY_SUCCESS - 目的達成に成功したことをユーザに通知
- NOTIFY_FAILURE - 目的達成に失敗したことをユーザに通知
- INFORM_COUNT - ユーザの要求に合致する対象の個数を通知
- OFFER_INTENT - ユーザに新しいインテントを提案
- REQ_MORE - 他に何かするかをユーザに質問
- GOODBYE - 対話を終了

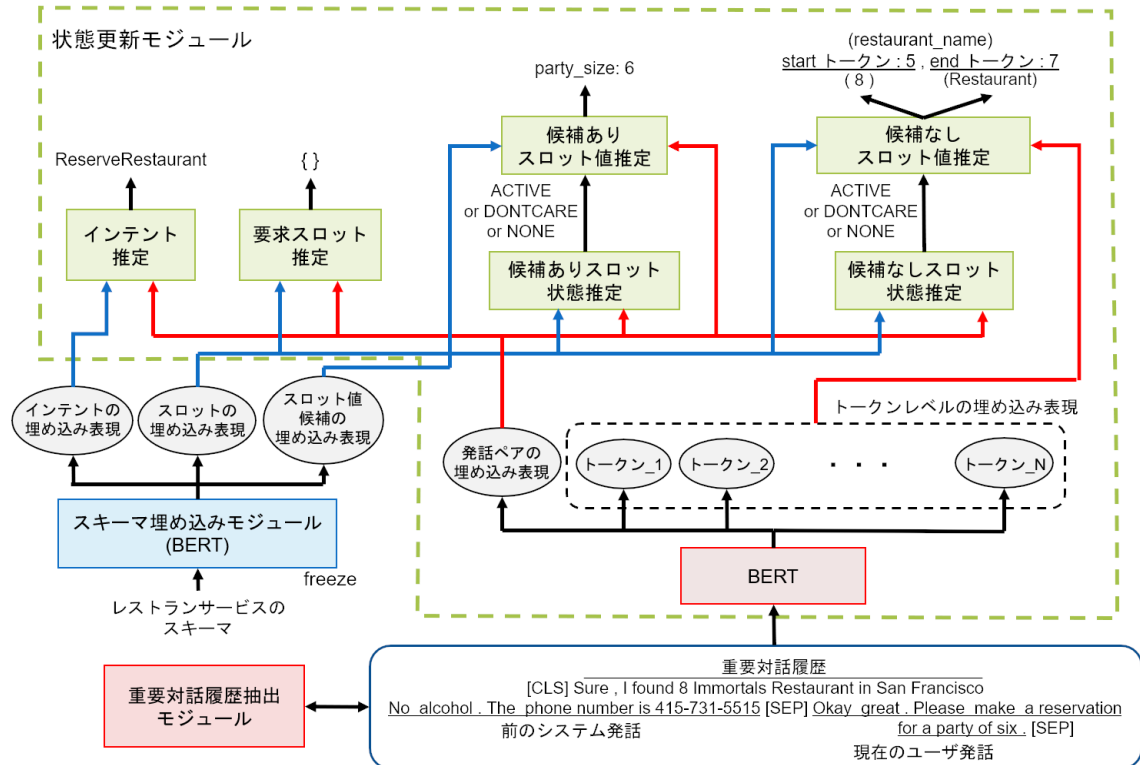


図 4.1 重要対話履歴抽出モジュールを加えた提案モデル

4.2 対話行為タグを用いた重要対話履歴抽出

ベースラインモデルが過去の発話中に存在するスロット値を推定するために、対話履歴を用いる。深層学習による対話状態追跡で対話履歴を用いる場合、直近の数発話を履歴として入力に用いる [13, 14]。しかし、直近の数発話では計算量の観点から多くの発話を入力するのは困難である。そこで、本研究では対話状態追跡に特に重要な発話をモデルへ入力するために、対話行為タグを用いた履歴抽出を提案する。この手法により、履歴として用いる発話を対話状態の推定に必要な情報を持つ発話のみにする。そのような発話を重要対話履歴とする。

図 4.1 に示す提案モデルでは、ベースラインモデルの入力文を作成する箇所に重要対話履歴を抽出する重要対話履歴抽出モジュールを加えている。このモジュールは直前の最も重要度の高い履歴を 1 つ入力に加える。方法としては、入力文中にある前のターンのシステム発話が特定の対話行為タグを持つ場合にその発話を保持し、既に発話を保持している場合は新しい発話に更新する。そして、元々保持されていた発話を重要対話履歴として入力に加える。入力、重要対話履歴、前のターンのシステム発話、現在のターンのユーザ発話の 3 発話となる。

SGD データセットの対話例を表 4.1 に示す。表 4.1 では、 n 回目のユーザ発話を Un ,

表 4.1 SGD データセットにある対話例の一部

発話文 (U : ユーザ, S : システム)	対話状態 (オレンジ色) or システムの対話行為 (水色)
U1: Can you pull up a list of places to eat?	{ インテント : FindRestaurant, 要求スロット : [], スロット値 : {} }
S1: Sure, what city should I search? And what kind of food would you like?	{ REQUEST(city, []), REQUEST(cuisine, []) }
U2: Search San Francisco for Asian Fusion food	{ インテント : FindRestaurant, 要求スロット : [], スロット値 : { city: San Francisco, cuisine: Asian Fusion } }
S2: Sure, I found 8 Immortals Restaurant in San Francisco	{ OFFER(restaurant_name, 8 Immortals Restaurant), OFFER(city, San Francisco) }
U3: Is there live music?	{ インテント : FindRestaurant, 要求スロット : [has_live_music], スロット値 : { city: San Francisco, cuisine: Asian Fusion } }
S3: Unfortunately no.	{ INFORM(has_live_music, False) }
U4: Do they serve alcohol? And what's their phone number?	{ インテント : FindRestaurant, 要求スロット : [phone_number, serves_alcohol], スロット値 : { city: San Francisco, cuisine: Asian Fusion } }
S4: No alcohol. The phone number is 415-731-5515	{ INFORM(serves_alcohol, False), INFORM(phone_number, 415-731-5515) }
U5: Okay great. Please make a reservation for a party of six.	{ インテント : ReserveRestaurant, 要求スロット : [], スロット値 : { city: San Francisco, cuisine: Asian Fusion, party_size: 6, restaurant_name: 8 Immortals Restaurant } }
⋮	⋮

システム発話を S_n と表す。対話状態はユーザ発話が入力された後の状態を示していて、システムの対話行為はシステム発話に付与されているものを示している。この対話例では、S2 でシステムが提案したレストラン名に対して、U5 で了承してレストラン名が対話状態に反映される。この時、ユーザはレストラン名を過去の発話から暗黙的に参照しているため、発話中にレストラン名は含まれていない。そのため、S2 での発話を履歴として入力に加えて、レストラン名を示す “restaurant_name” のスロット値となる “8 Immortals Restaurant” を推定可能にする必要がある。

図 4.1 では、例として表 4.1 のシステムの対話行為 “OFFER” を持つ発話を抽出している。入力に加えられる重要対話履歴は、“8 Immortals Restaurant” を含む S2 での発話になる。入力文にスロット値となる文字列が含まれるため、スロット値として

抜き出すことが可能になる．直近の数発話を対話履歴として用いる従来手法の場合は，直近の 6 発話を入力に加える必要があるが，提案手法では 3 発話で推定が行える．また，図 4.1 では 5 ターン前の発話を抽出しているが，提案手法はさらに前の発話を抽出することも可能である．ゆえに，提案手法は計算量の削減と同時に性能の向上が期待できる．

第5章

対話状態追跡における各対話行為の重要性調査

予備実験として、どの対話行為が対話状態追跡にとって重要なのかを調査する。本実験では、対話状態追跡においてスロット値を変更する候補を持つ発話が重要であると仮定する。ゆえに、対話行為ごとにスロット値候補の出現回数と出現したスロット値候補が実際にスロット値に反映された回数を調べて比較する。本章では、この2つの数値の調査方法の説明と調査結果を示す。

5.1 データセット

データセットはDSTC8-Track4で公開されたSGDデータセット[18]を用いる。対話データはユーザ役の対話シミュレータとシステム役の対話シミュレータによる対話を収集したものであり、各種ラベルもシミュレータによって付与されたものである。現在は学習用と検証用データのみ公開されている。SGDデータセットは、対話中で1つのドメインしか扱わないSingle-domainの対話と複数ドメインを扱うMulti-domainの対話という2種類を持つ。対話の長さは、Single-domainで平均15.15ターン、Multi-Domainで平均22.94ターンである。SGDデータセットは17個のドメインを扱う。そして、ドメインごとにサービスがいくつか存在し、スキーマが用意されている。ドメインごとのサービスの個数と、学習用、検証用データセットの対話数を表5.1に示す。また、SGDデータセットでは学習時、検証時、テスト時で扱うサービスが異なる。学習時と検証時で扱うサービスは表5.2に示す。スキーマに関しては付録の表A.1に載せている。

5.2 実験手法

予備実験ではSGDデータセットの学習用と検証用のデータを全て使用した。対話数は18,624対話、そのうちシステム発話は189,345発話ある。調査する発話は、対話行為タグが与えられるシステム発話に限定し、各発話の対話行為は発話が持つ対話行為

表 5.1 SGD データセットのドメインごとの対話数（括弧内はサービスの個数）

	学習用の対話数			検証用の対話数		
	Single domain	Multi domain	All	Single domain	Multi domain	All
Alarm (1)	NA	NA	NA	37	NA	37
Banks (2)	207	520	727	42	252	294
Buses (2)	310	1,970	2,280	44	285	329
Calendar (1)	169	1,433	1,602	NA	NA	NA
Events (2)	788	2,721	3,509	73	345	418
Flights (3)	985	1,762	2,747	94	297	391
Homes (1)	268	579	847	81	99	180
Hotels (4)	457	2,896	3,353	56	521	577
Media (2)	281	832	1,113	46	133	179
Movies (2)	292	1,325	1,617	47	94	141
Music (2)	394	896	1,290	35	161	196
RentalCars (2)	215	1,370	1,585	39	342	381
Restaurants (2)	367	2052	2,419	73	263	336
RideSharing (2)	119	1,584	1,703	45	225	270
Services (2)	551	1,338	1,889	44	157	201
Travel (4)	NA	1,871	1,871	45	238	283
Weather (1)	NA	951	951	35	322	357
TOTAL	5,403	10,739	16,142	836	1,646	2,482

表 5.2 SGD データセットの学習用，検証用が扱うサービスのリスト

	扱うサービスのリスト
学習用	[Banks_1, Buses_1, Buses_2, Calendar_1, Events_1, Events_2, Flights_1, Flights_2, Homes_1, Hotels_1, Hotels_2, Hotels_3, Media_1, Movies_1, Music_1, Music_2, RentalCars_1, RentalCars_2, Restaurants_1, RideSharing_1, RideSharing_2, Services_1, Services_2, Services_3, Travel_1, Weather_1]
検証用	[Alarm_1, Banks_2, Buses_1, Events_1, Flights_3, Homes_1, Hotels_1, Hotels_4, Media_2, Movies_2, Music_1, RentalCars_1, Restaurants_2, RideSharing_1, Services_4, Travel_1, Weather_1]

表 5.3 SGD データセットの学習用と検証用データにおける各対話行為の出現回数

	Single-domain	Multi-domain	Combined
INFORM	5,735	19,798	25,533
REQUEST	9,772	29,841	39,613
CONFIRM	6,417	18,213	24,630
OFFER	10,276	33,887	44,163
NOTIFY_SUCCESS	4,215	12,254	16,469
NOTIFY_FAILURE	629	2,131	2,760
INFORM_COUNT	4,020	12,519	16,539
OFFER_INTENT	2,527	8,555	11,082
REQ_MORE	3,655	6,890	10,545
GOODBYE	6,239	12,385	18,624
全システム発話	47,258	142,087	189,345

タグで決定する．複数種類の対話行為タグを持つ場合は，対話行為も複数にしている．そのようにした場合の各対話行為の出現回数を表 5.3 に示す．このように各対話行為の出現回数には偏りがあるため，予備実験で示す結果は全て 1 対話あたりとする．なお，予備実験では，対話行為の“INFORM_COUNT”と“OFFER_INTENT”が与えるスロット値はスロット値の更新に関係ないため除外している．

まず，各対話行為のスロット値候補の出現回数を調査した．スロット値候補は，各システム発話に与えられているスロット値範囲リストと対話行為リストを用いて重複を許さずに抽出される．各対話行為の発話からスロット値候補を抽出して，スロット値候補の出現回数をカウントしていく．全てをカウントしたのち，対話数で割ることで，1 対話あたりのスロット値候補の出現回数を各対話行為ごとに求める．

続いて，各対話行為で出現したスロット値候補が実際にスロット値に反映された回数を調査した．まず，現在のターンの対話状態と前のターンの対話状態を比較して，現在のターンで追加されたスロット値を抽出する．そして現在のターンまでに各対話行為で出現したスロット値候補が何個含まれているかをカウントしていく．こちらも，スロット値候補の出現回数と同様に 1 対話あたりの数値を求める．

5.3 調査結果

結果は図 5.1 と図 5.2 に示す．この図では，Single-domain, Multi-domain, 加えてその 2 つを合わせた All で結果に差異がないか調べるためにデータを分けて 3 つの結果を示している．図 5.1 を見ると，“INFORM”と“CONFIRM”，“OFFER”，“

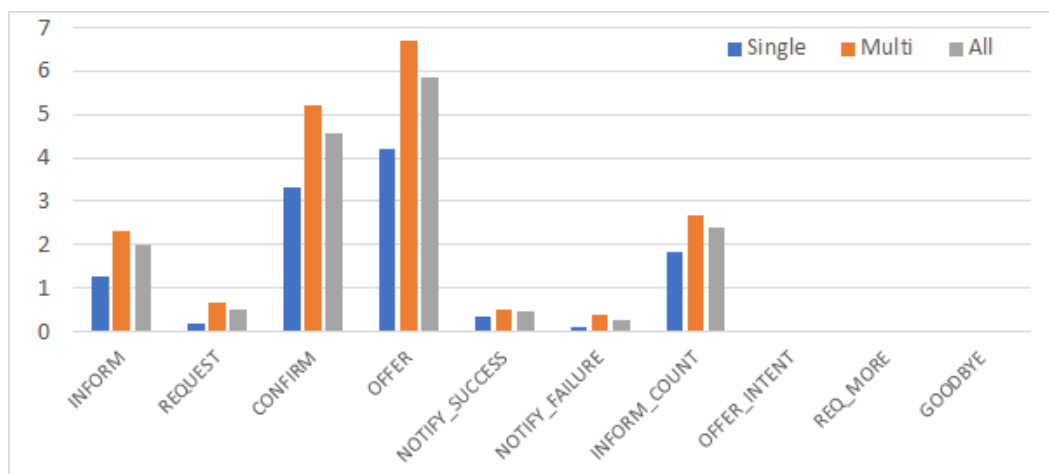


図 5.1 1 対話あたりの各対話行為の発話中でスロット値候補が出現した回数

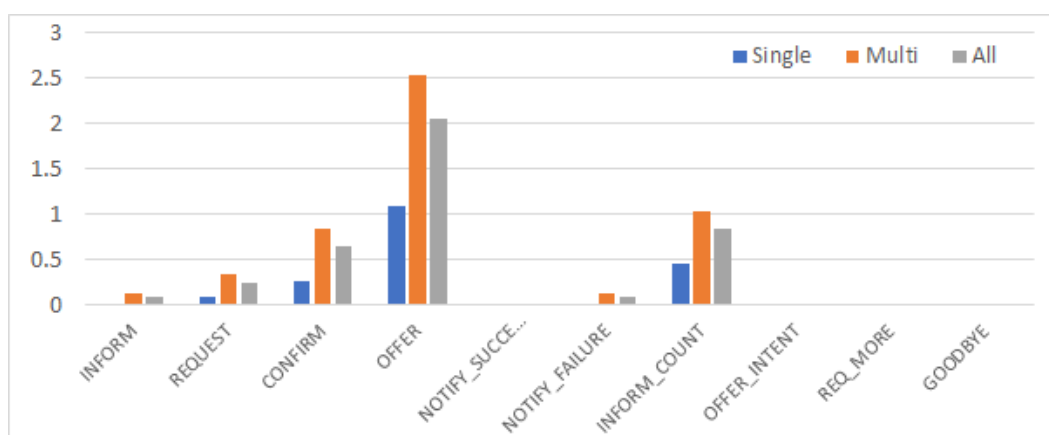


図 5.2 1 対話あたりの各対話行為の発話中で出現したスロット値候補が実際にスロット値に反映された回数

“INFORM_COUNT”では他の対話行為に比べて多くのスロット値候補が出現していることが分かる。“INFORM”、“CONFIRM”、“OFFER”は、スロット値に関する発話を行うため、スロット値候補が多く出現している。ただし、“INFORM_COUNT”はユーザの要求に合致する対象の個数を通知する対話行為であるため、それ自体がスロット値候補を与えることはないが、“OFFER”と同じ発話で共存していることが多い。そのため、“INFORM_COUNT”の発話でもスロット値候補が多く出現している。残りの対話行為は、スロット値に関する発話を行わず、他の対話行為と共存することも少ないため、発話中にスロット値候補が出現することが少ない。

図5.2を見ると、“INFORM”と“CONFIRM”は出現したスロット値候補に対して約10分の1しか対話状態に反映されていないことが分かる。これは両者とも対話状態の

更新に必要でないスロット値候補を多く与えているからである．例えば，“INFORM”が与えるスロット値候補はユーザが指定したものや場所の補足情報であり，対話状態を更新する情報ではないことが多い．また，“CONFIRM”の発話は既に対話状態に含まれているスロット値の確認を行うため，与えるスロット値候補の多くは既に対話状態に反映されており，対話状態が更新されない．ただし，“CONFIRM”が与えるスロット値候補は，ユーザが発した単語の言い換え（例：“at afternoon 1:30” → “1:30 pm”）を含むため対話状態にスロット値として追加されることもある．反対に，“OFFER”と“INFORM_COUNT”は与えたスロット値候補の約4分の1が対話状態に反映されていることが分かる．したがって，両者どちらかの対話行為タグを持つ発話で出現するスロット値候補は対話状態に反映される可能性が高く，その発話は他の対話行為の発話よりスロット値の推定に重要であると考えられる．ただし，“CONFIRM”の説明で述べたように，対話行為ごとに与える情報に特徴があるため，一概に“OFFER”と“INFORM_COUNT”を持つ発話だけが重要とは言えない．また，以上に示した結果はSingle-domain, Multi-domain, 加えてその2つを合わせたAllでも同様の結果となっている．

第6章

評価実験

本章では、実験条件を述べた上で提案手法を用いたモデルの性能について自動評価した結果を示す。その際、各対話行為タグで抽出した場合の性能比較、従来の対話履歴の使用法との性能比較を行う。自動評価の指標は DSTC8-Track4 に従っている。

6.1 実験条件

モデルの実装には Tensorflow を用いている。実行環境は、産業技術総合研究所が構築し運用する、人工知能処理向け計算インフラストラクチャである AI 橋渡しクラウド (AI Bridging Cloud Infrastructure; ABCI) [19] を利用している。ABCI の計算資源はタイプ別に分かれているが、本研究では、CPU (Intel Xeon Gold 6148 プロセッサ 2.4 GHz) が 5 コア、GPU (NVIDIA Tesla V100 for NVLink 16GiB HBM2) が 1 個、メモリが 240 GiB、ローカルストレージが 180 GB の G.small を用いている。

モデルの設定は、与えられたベースラインモデルとミニバッチサイズ以外全て同じにしている。BERT のハイパーパラメータは BERT の論文 [16] に記載された BERT_{BASE} の構成をそのまま使用する。つまり、BERT は隠れ層が 12 層、隠れ層のサイズが 768 次元である。学習はミニバッチ学習を行い、ミニバッチサイズを学習時は 32、検証時は 8、テスト時は 8 とする。ただし、従来の対話履歴の使用法との比較の際は、学習時のミニバッチサイズを 24 にして実験を行う。入力発話の最大長は、2 発話入力の場合で 80 とし、1 発話増えるたびに 40 増加させる。学習率は $1e-4$ とする。誤差関数には、交差エントロピーを使用し、最適化アルゴリズムは重み減衰によって過学習を抑制する Weight Decay という手法を用いた Adam [20] を使用する。過学習を防ぎモデルの汎化性能を上げるために、dropout (dropout 率は 0.1) を用いる。モデルは 80 epoch の学習を行い、学習を終えた段階のモデルを評価に使用する。

データは Single-domain のみを用いて、学習用として 5,403 対話、検証用として 836 対話を学習に用いる。テスト用のデータはまだ未公開のため、今回は検証用データで評価している。

6.2 評価指標

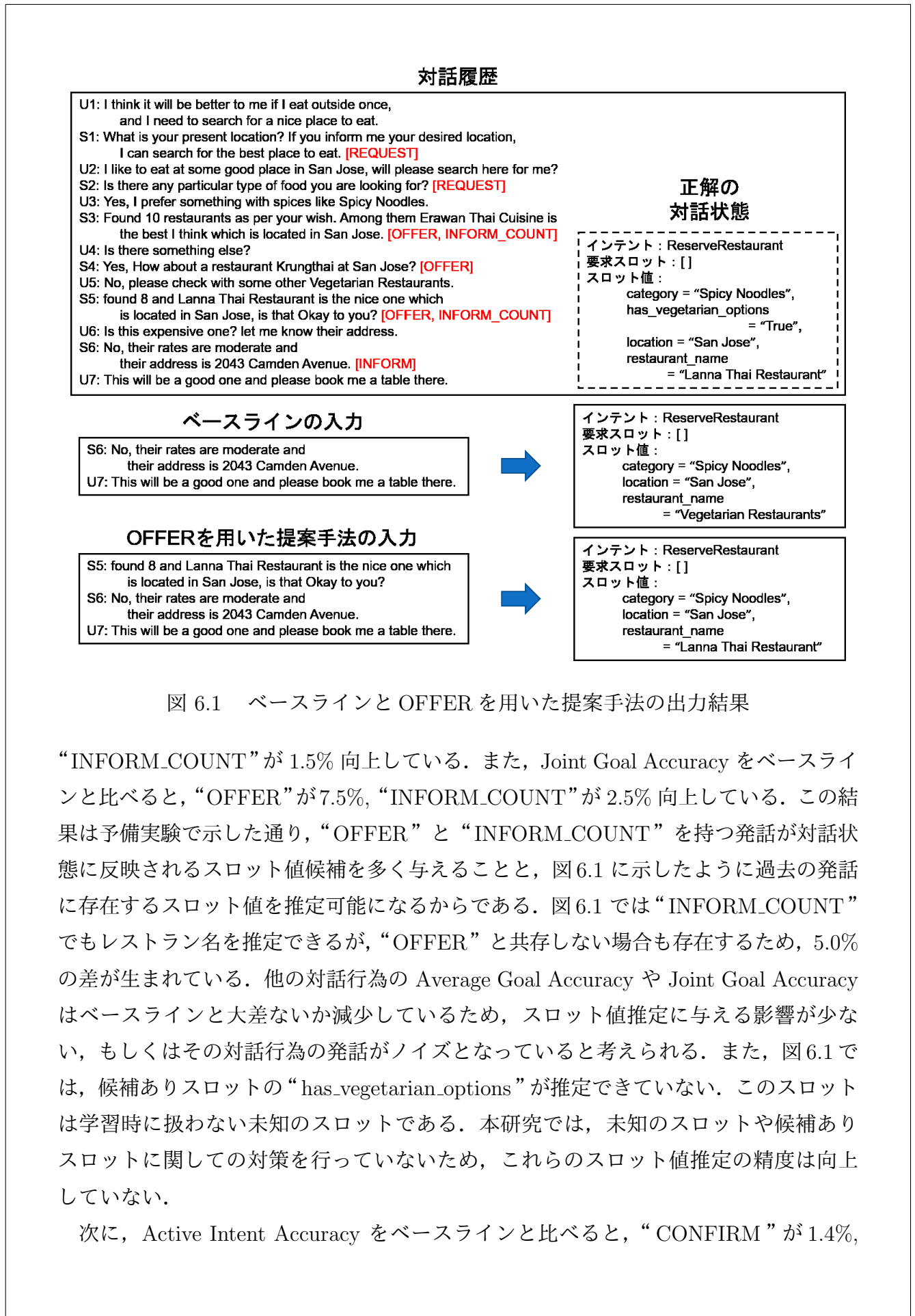
モデルの自動評価は DSTC8-Track4 にて提供される自動評価スクリプトを用いた。そのスクリプトで計算される評価指標は 4 種類ある。1 つ目はインテント推定の精度を示す Active Intent Accuracy, 2 つ目は要求スロット推定のマクロ平均 F1 スコアを示す Requested Slot F1, 3 つ目は各ターンでの対話状態の各スロット値の平均精度を示す Average Goal Accuracy, 4 つ目は各ターンでの対話状態の全スロット値の平均精度を示す Joint Goal Accuracy である。Requested Slot F1 は, あるターンで予測結果と正解データどちらも要求スロットが存在しない場合そのターンをスキップするため, 報告する数値はスキップされていない全てのユーザターンでの平均 F1 スコアとなる。Average Goal Accuracy は, 正解データの対話状態で空でないスロットのみを対象としている。DSTC8-Track4 では Joint Goal Accuracy によってモデルの優劣をつけるため, 本研究でも Joint Goal Accuracy の数値を重要視する。

6.3 実験結果

表 6.1 各対話行為で提案手法を行った際の結果

	Active Intent Accuracy	Requested Slot F1	Average Goal Accuracy	Joint Goal Accuracy
ベースライン (直近 2 発話入力)	0.946	0.946	0.809	0.497
INFORM	0.949	0.947	0.818	0.499
REQUEST	0.946	0.946	0.807	0.495
CONFIRM	0.960	0.946	0.807	0.497
OFFER	0.957	0.945	0.851	0.572
NOTIFY_SUCCESS	0.956	0.948	0.813	0.500
NOTIFY_FAILURE	0.948	0.946	0.815	0.504
INFORM_COUNT	0.943	0.945	0.824	0.522
OFFER_INTENT	0.951	0.947	0.789	0.502
REQ_MORE	0.945	0.947	0.803	0.498
GOODBYE	0.932	0.947	0.803	0.492

初めに, 各対話行為タグを持つ発話がどの推定の性能を向上させるのかについて調べた。ベースラインモデルの結果と各対話行為タグで提案手法を行った際の結果を表 6.1 に示す。Average Goal Accuracy をベースラインと比べると, “OFFER” が 4.2%,



“OFFER”が1.1%，“NOTIFY_SUCCESS”が1.0% 向上しているなど多くの対話行為で精度の向上が見られる．インテント推定はユーザの目的を推定しているので，対話の流れをより良く理解できる発話を履歴として入力できれば精度の向上が見込まれる．ゆえに，過去の発話を用いて情報量を増やした提案手法がベースラインの結果を超えている．ただし，対話の終わりに用いられる対話行為である“GOODBYE”は，対話の途中で抽出することはできないので精度が向上しなかった．また，インテント推定での誤りの多くは，最初の発話で異なるインテントを選択する誤りとインテントを維持すべきところで変更してしまう誤りであった．維持すべきところで変更してしまう誤りに関しては，前のターンのインテントをインテント推定の入力に用いることで解決すると期待できる．

Requested Slot F1 に関しては大きな変化が見られない．つまり，要求スロットは現在のターンのユーザ発話の情報に大きく依存し，対話履歴では推定に必要な情報量があまり増加しない．

ここまでの結果から，対話履歴をシステムの対話行為を用いて抽出することで性能が上がることを確認できた．今回は対話行為を1つだけ選択したが，複数の対話行為を用いることで更なる性能の向上が期待できる．

表 6.2 従来手法との比較

	Active Intent Accuracy	Requested Slot F1	Average Goal Accuracy	Joint Goal Accuracy	学習 時間
ベースライン (直近 2 発話 入力)	0.940	0.944	0.816	0.502	15.3(h)
提案手法 (OFFER)	0.946	0.947	0.845	0.565	20.7(h)
直近 3 発話入力	0.943	0.947	0.815	0.508	20.9(h)
直近 4 発話入力	0.974	0.948	0.844	0.553	26.3(h)

続いて，従来の対話履歴の使用法との比較を行った．今回は提案手法として，他の対話行為より総合的に優れた結果を残した“OFFER”を用いた提案手法と，直近の2発話を入力としたモデル，3発話を入力としたモデル，4発話を入力としたモデルとを比較した．その結果を表6.2に示す．ベースラインと提案手法の結果はミニバッチサイズを下げたことで変化している．直近3発話入力ではベースラインと比べて少ししか性能が向上していない．原因としては，1つ前のユーザ発話に含まれる既に推定済みの情報が現在のターンのユーザ発話に含まれる新しい情報の推定の妨げとなったことと，直

近3発話では対話の流れを捉えるには情報不足であったことが考えられる。Joint Goal Accuracy に関しては、“OFFER”を用いた提案手法が直近3発話入力より5.7%高い数値を示している。同じ発話数でも性能が向上していることから、“OFFER”を用いた履歴抽出がスロット値推定に必要な情報を抽出できていることが分かる。また、提案手法は直近4発話入力より Joint Goal Accuracy が1.2%高い数値を示している。これは、提案手法が対話によって6発話前や8発話前のスロット値も推定可能だからである。提案手法は3発話入力で直近4発話入力以上の結果を示していることから、対話状態追跡に重要な発話だけを抽出して重要ではない発話を取り除いているといえる。また、提案手法は直近4発話入力に比べて学習時間が5.4時間短く、計算量の削減にも成功している。しかし、Active Intent Accuracy では、直近4発話入力の方が提案手法に比べて2.8%高い数値を示している。インテント推定は対話の流れを理解する必要があるため、連続した発話を入力する方が良い。そのため、提案手法のように過去の発話を抽出したものをを用いると対話が断片的になり、連続した発話を入力とする従来手法より対話の流れを理解しにくいのだと考えられる。

第7章

むすび

本研究では，対話状態追跡における対話履歴の使い方として，対話行為タグを用いて重要な対話履歴を抽出する手法を提案した．DSTC8-Track4 で提供されたベースラインモデルと比較した結果，ユーザにスロット値候補を提案する“OFFER”という対話行為タグを用いたもので Joint Goal Accuracy が 7.5% 向上した．直近の数発話を履歴として入力する従来手法と“OFFER”タグを用いた提案手法とを比較した結果，提案手法がモデルの学習時間を減らすのと同時に，性能を向上させることが可能であることを示した．

今回の実験では“OFFER”を持つ発話のみを抽出した場合が最高の結果を示したが，他の対話行為と組み合わせて複数の対話行為を用いることで更なる性能の向上が期待できる．ゆえに，各対話行為タグが各ドメインにどのように影響するのかの調査を行い，対話行為タグの最適な組み合わせを見つけたい．

今回の手法は人間が選択的にどの対話行為タグを持つ発話に注目するかを決めていた．この手法でも性能は向上するが，ドメインごとに注目すべき発話が異なる場合や他のデータセットの場合で重要な対話行為タグや対話履歴が変化するため対応ができない．そのため，機械側で対話履歴中のどの発話に注目すべきかを推定可能な対話状態追跡モデルを検討したい．

謝辞

本研究を進めるにあたり，適切なご指導とご助言をいただきました李晃伸教授，酒向慎司准教授に深く感謝をいたします。また，共同で研究を行ってくださった李研究室博士前期課程 2 年田中涼太氏，尾関晃英氏，同研究室博士前期課程 1 年加藤修悟氏，研究環境の維持を含め，様々なご協力をいただいた同研究室博士前期課程 2 年森凜太郎氏をはじめ，李研究室の方々，また共に研究を進めた同期の方々に心より感謝をいたします。最後に，本学で研究をする機会を与えてくださった両親と祖父母に感謝をいたします。

参考文献

- [1] Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, 2017.
- [2] Bing Liu and Ian Lane. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. In *Proceedings of Interspeech 2017*, pages 2506–2510, 2017.
- [3] Seokhwan Kim, Michel Galley, Chulaka Gunasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, Minlie Huang, Luis Lastras, Jonathan K. Kummerfeld, Walter S. Lasecki, Chiori Hori, Anoop Cherian, Tim K. Marks, Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, and Raghav Gupta. The eighth dialog system technology challenge. In *Third NeurIPS workshop on Conversational AI: “Today’s Practice and Tomorrow’s Potential”*, 2019.
- [4] Antoine Raux, Brian Langner, Dan Bohus, Alan W Black, and Maxine Eskenazi. Let’s go public! taking a spoken dialog system to the real world. In *Proceedings of INTERSPEECH 2005*, pages 885–888, 2005.
- [5] 大浦 圭一郎, 山本 大介, 内匠 逸, 李 晃伸, and 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 (<特集>音声対話システムの実用化に向けて). *人工知能学会誌*, 28(1):60–67, 2013.
- [6] Kai Sun, Qizhe Xie, and Kai Yu. Recurrent polynomial network for dialogue state tracking. *arXiv preprint arXiv:1507.03934*, 2015.
- [7] Steve Young. Using pomdps for dialog management. In *2006 IEEE Spoken Language Technology Workshop*, pages 8–13, 2006.
- [8] Steve Young, Milica Gasic, Blaise Thomson, and Jason Williams. Pomdp-based

- statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101:1160–1179, 2013.
- [9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [10] Matthew Henderson, Blaise Thomson, and Jason D. Williams. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, 2014.
- [11] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, 2018.
- [12] Abhinav Rastogi, Raghav Gupta, and Dilek Hakkani-Tur. Multi-task learning for joint language understanding and dialogue state tracking. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 376–384, 2018.
- [13] Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, 2019.
- [14] Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 264–273, 2019.
- [15] “dstc8-track4 のベースラインモデル”. https://github.com/google-research/google-research/tree/master/schema_guided_dst.
- [16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

- [17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. 2017.
- [18] Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset, 2019.
- [19] “abci”. <https://abci.ai/ja/>.
- [20] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2014.

付録

A.1 各サービスのスキーマ

表 A.1: 各スキーマのロットとインテント

サービス	スロット	インテント
Alarm_1	['alarm_time', 'alarm_name', 'new_alarm_time', 'new_alarm_name']	['GetAlarms', 'AddAlarm']
Banks_1	['account_type', 'recipient_account_type', 'balance', 'amount', 'recipient_account_name']	['CheckBalance', 'TransferMoney']
Banks_2	['account_type', 'recipient_account_type', 'account_balance', 'transfer_amount', 'recipient_name', 'transfer_time']	['CheckBalance', 'TransferMoney']
Buses_1	['from_location', 'to_location', 'from_station', 'to_station', 'leaving_date', 'leaving_time', 'fare', 'travelers', 'transfers']	['FindBus', 'BuyBusTicket']
Buses_2	['origin', 'destination', 'origin_station_name', 'destination_station_name', 'departure_date', 'price', 'departure_time', 'group_size', 'fare_type']	['FindBus', 'BuyBusTicket']
Calendar_1	['event_date', 'event_time', 'event_location', 'event_name', 'available_start_time', 'available_end_time']	['GetEvents', 'GetAvailableTime', 'AddEvent']

表は次ページに続く

前ページからの続き

サービス	スロット	インテント
Events_1	['category', 'subcategory', 'event_name', 'date', 'time', 'number_of_seats', 'city_of_event', 'event_location', 'address_of_location']	['FindEvents', 'BuyEventTickets']
Events_2	['event_type', 'category', 'event_name', 'date', 'time', 'number_of_tickets', 'city', 'venue', 'venue_address']	['FindEvents', 'GetEventDates', 'BuyEventTickets']
Flights_1	['passengers', 'seating_class', 'origin_city', 'destination_city', 'origin_airport', 'destination_airport', 'departure_date', 'return_date', 'number_stops', 'outbound_departure_time', 'outbound_arrival_time', 'inbound_arrival_time', 'inbound_departure_time', 'price', 'refundable', 'airlines']	['SearchOnewayFlight', 'SearchRoundtripFlights', 'ReserveOnewayFlight', 'ReserveRoundtripFlights']
Flights_2	['passengers', 'seating_class', 'origin', 'destination', 'origin_airport', 'destination_airport', 'departure_date', 'return_date', 'number_stops', 'outbound_departure_time', 'outbound_arrival_time', 'inbound_arrival_time', 'inbound_departure_time', 'fare', 'is_redeye', 'airlines']	['SearchOnewayFlight', 'SearchRoundtripFlights']

表は次ページに続く

前ページからの続き

サービス	スロット	インテント
Flights_3	['passengers', 'flight_class', 'origin_city', 'destination_city', 'origin_airport_name', 'destination_airport_name', 'departure_date', 'return_date', 'number_stops', 'outbound_departure_time', 'outbound_arrival_time', 'inbound_arrival_time', 'inbound_departure_time', 'price', 'number_checked_bags', 'airlines', 'arrives_next_day']	['SearchOnewayFlight', 'SearchRoundtripFlights']
Homes_1	['area', 'address', 'property_name', 'phone_number', 'furnished', 'pets_allowed', 'rent', 'visit_date', 'number_of_beds', 'number_of_baths']	['FindApartment', 'ScheduleVisit']
Hotels_1	['destination', 'number_of_rooms', 'check_in_date', 'number_of_days', 'star_rating', 'hotel_name', 'street_address', 'phone_number', 'price_per_night', 'has_wifi']	['ReserveHotel', 'SearchHotel']
Hotels_2	['where_to', 'number_of_adults', 'check_in_date', 'check_out_date', 'rating', 'address', 'phone_number', 'total_price', 'has_laundry_service']	['BookHouse', 'SearchHouse']
Hotels_3	['location', 'number_of_rooms', 'check_in_date', 'check_out_date', 'average_rating', 'hotel_name', 'street_address', 'phone_number', 'price', 'pets_welcome']	['ReserveHotel', 'SearchHotel']

表は次ページに続く

前ページからの続き

サービス	スロット	インテント
Hotels_4	['location', 'number_of_rooms', 'check_in_date', 'stay_length', 'star_rating', 'place_name', 'street_address', 'phone_number', 'price_per_night', 'smoking_allowed']	['ReserveHotel', 'SearchHotel']
Media_1	['title', 'genre', 'subtitles', 'directed_by']	['FindMovies', 'PlayMovie']
Media_2	['movie_name', 'genre', 'subtitle_language', 'director', 'actors', 'price']	['FindMovies', 'RentMovie']
Movies_1	['price', 'number_of_tickets', 'show_type', 'theater_name', 'show_time', 'show_date', 'genre', 'street_address', 'location', 'movie_name']	['BuyMovieTickets', 'FindMovies', 'GetTimesForMovie']
Movies_2	['title', 'genre', 'aggregate_rating', 'starring', 'director']	['FindMovies']
Music_1	['song_name', 'artist', 'album', 'genre', 'year', 'playback_device']	['LookupSong', 'PlaySong']
Music_2	['song_name', 'artist', 'album', 'genre', 'playback_device']	['LookupMusic', 'PlayMedia']
RentalCars_1	['type', 'car_name', 'pickup_location', 'pickup_date', 'pickup_time', 'pickup_city', 'dropoff_date', 'total_price']	['GetCarsAvailable', 'ReserveCar']
RentalCars_2	['car_type', 'car_name', 'pickup_location', 'pickup_date', 'pickup_time', 'pickup_city', 'dropoff_date', 'total_price']	['GetCarsAvailable', 'ReserveCar']

表は次ページに続く

前ページからの続き

サービス	スロット	インテント
Restaurants_1	['restaurant_name', 'date', 'time', 'serves_alcohol', 'has_live_music', 'phone_number', 'street_address', 'party_size', 'price_range', 'city', 'cuisine']	['ReserveRestaurant', 'FindRestaurants']
Restaurants_2	['restaurant_name', 'date', 'time', 'has_seating_outdoors', 'has_vegetarian_options', 'phone_number', 'rating', 'address', 'number_of_seats', 'price_range', 'location', 'category']	['ReserveRestaurant', 'FindRestaurants']
RideSharing_1	['destination', 'shared_ride', 'ride_fare', 'approximate_ride_duration', 'number_of_riders']	['GetRide']
RideSharing_2	['destination', 'ride_type', 'ride_fare', 'wait_time', 'number_of_seats']	['GetRide']
Services_1	['stylist_name', 'phone_number', 'average_rating', 'is_unisex', 'street_address', 'city', 'appointment_date', 'appointment_time']	['BookAppointment', 'FindProvider']
Services_2	['dentist_name', 'phone_number', 'address', 'city', 'appointment_date', 'appointment_time', 'offers_cosmetic_services']	['BookAppointment', 'FindProvider']
Services_3	['doctor_name', 'phone_number', 'average_rating', 'street_address', 'city', 'appointment_date', 'appointment_time', 'type']	['BookAppointment', 'FindProvider']

表は次ページに続く

前ページからの続き

サービス	スロット	インテント
Services_4	['therapist_name', 'phone_number', 'address', 'city', 'appointment_date', 'appointment_time', 'type']	['BookAppointment', 'FindProvider']
Travel_1	['location', 'attraction_name', 'category', 'phone_number', 'free_entry', 'good_for_kids']	['FindAttractions']
Weather_1	['precipitation', 'humidity', 'wind', 'temperature', 'city', 'date']	['GetWeather']

これで終了