

# 文章形式の出力による クロスドメインスロットフィリングの改善

大矢 一穂<sup>1</sup> 鶴岡 慶雅<sup>2</sup>

<sup>1</sup> 東京大学工学部 電子情報工学科 <sup>2</sup> 東京大学大学院 情報理工学系研究科  
{oya, tsuruoka}@logos.t.u-tokyo.ac.jp

## 概要

タスク指向対話システムのサブタスクの1つであるスロットフィリングにおいて、学習データと異なるドメインで推論を行うクロスドメインスロットフィリングが研究されている。先行研究では、BERT や BART といった事前学習済みモデルを利用した手法が提案されている。このうち、BART を用いた先行研究では、抜き出したエンティティをそのまま出力するように学習を行っていた。本研究では、出力が自然な文章になるようなテンプレートを用いることで先行研究の性能の改善を目指した。実験の結果、提案手法が、事前学習を用いた先行研究の性能を上回る結果となった。

## 1 はじめに

近年、人間のように会話を行うことのできる対話 AI が普及しつつある。こういった対話システムの研究は、タスク指向対話システムと非タスク指向対話システムの2つに大別される。このうち、タスク指向対話システムは、なんらかの目的を持ったユーザーと会話することで、その目的を達成することを補助する対話システムであるが、そのサブタスクとしてスロットフィリングがある。スロットフィリングとは、ユーザーの発話から必要な情報を抜き出すタスクである。スロットフィリングの例を図1に示す。スロットフィリングでは、抜き出さなくてはならない情報（スロット）の種類があらかじめ定義されており、図1の city, country, cuisine などがそれに相当する。そして、モデルはそれぞれのスロットに対応する語句（エンティティ）である “Falmouth” や “fast food restaurant” を発話から抜き出す。従来はドメインに特化した学習データを用いてモデルの学習を行っていた。しかし実際にタスク指向対話システムを構築しようとしたとき、ドメインによっては

Book a highly rated fast food restaurant in Falmouth.



Slot Type	Entity
city	Falmouth
country	×
cuisine	×
restaurant_name	×
restaurant_type	fast food restaurant

図1 スロットフィリングの例

ラベル付き学習データが用意できないことも珍しくなく、そのような状況でもスロットフィリングを行うことのできるモデルを学習する必要がある。

上述の課題を踏まえて、学習したいドメインとは異なるドメインの学習データを用いて学習を行うクロスドメインスロットフィリングが近年研究されている。クロスドメインスロットフィリングにおいては、ドメインによってスロットが異なるため、推論時には、学習データには存在しなかったスロットについて扱う必要がある。先行研究 [1, 2, 3] では、事前学習済みモデルをファインチューニングする手法が提案されており、以前の研究と比べて高い性能が報告されている。その1つである AISFG [3] という手法では、BART を使い、他の先行研究とは異なり seq2seq のアプローチをとっている。AISFG では、発話文やスロット説明を組み込んだ文章を入力として、エンティティの文字列のみを出力する方法をとっている。しかしこのデコード方法は、文を出力するように学習している BART の事前学習と乖離しており、事前学習で BART のデコーダ部分が獲得した言語的な知識を、タスクに活かすことができていない可能性があると考えた。

そこで本研究では、AISFG と同様に BART をベースとしながら、シンプルなテンプレートを用いて、エンティティだけでなくスロット説明やエンティ

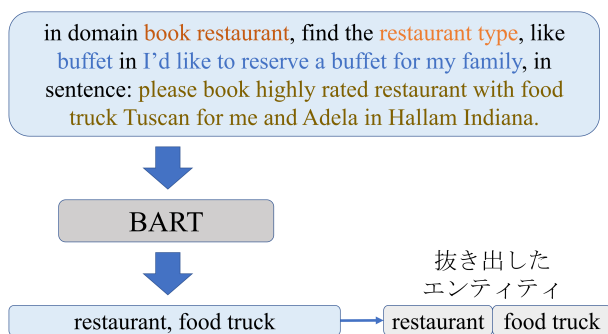


図2 AISFG のフレームワーク

ティ例もデコード系列に含める手法を提案する。

## 2 関連研究

Bapna ら [4] や Shah ら [5] は、スロットの説明文やエンティティの具体例を使って、スロットの意味的な情報をモデルに与える手法を提案した。

また、Liu ら [6] が提案した手法である Coach では、エンティティを見つける第一段階と、エンティティごとにスロットタイプで分類をする第二段階の2つの段階に処理を分けることでタスクの簡略化を図った。

近年の研究では、事前学習済みモデルをファインチューニングする手法が提案されている。Du ら [1] が提案した QASF や、Yu ら [2] が提案した RCSF では、スロットフィリングを質問応答タスクとして定式化し、事前学習済みモデルである BERT [7] をファインチューニングすることで、以前の手法よりも高い性能を達成した。

Yan ら [3] が提案した AISFG は、seq2seq の事前学習済みモデルである BART [8] をベースとした手法である。図 2 に AISFG のフレームワークを示す。この例は発話文である “please book highly rated restaurant with food truck Tuscan for me and Adela in Hallam Indiana.” から、restaurant\_type のスロットに対応するエンティティを抜き出す様子を表している。入力文には、ドメインを表している “book restaurant”，スロット説明である “restaurant type”，発話文の例である “I’d like to reserve a buffet for my family.” とその発話における restaurant\_type のエンティティである “buffet” を組み込んでいる。デコーダーは、抜き出した restaurant\_type のエンティティである “restaurant” と “food truck” をカンマ区切りで出力するように学習を行う。これは、複数のエンティティを発話中から抜き出した場合の例だが、一方

で、エンティティが1つである場合はそのエンティティがそのまま出力となり、スロットに対応するエンティティが存在しない場合についてはピリオドのみを出力する。

## 3 提案手法

### 3.1 定式化

スロットフィリングは、ドメイン  $d \in D$  において、発話文  $x$  が与えられたとき、あらかじめ定義されているスロットタイプ  $s \in S_d$  のそれぞれに対して、あてはまるスパンの集合  $y = \{y_1, y_2, \dots, y_i\}$  をを予測するタスクである。本研究では、入出力文のテンプレートである  $t_{in}$  と  $t_{out}$  を作成し、seq2seq タスクとして、スロットフィリングに取り組んだ。学習時は、発話文  $x$  とスロットタイプ  $s$  から作成した入力文  $t_{in}(x, s)$  と、 $s$  及び抜き出したエンティティの集合  $y$  から作成した出力文  $t_{out}(y, s)$  を用いて、入力文  $t_{in}(x, s)$  に対して  $t_{out}(y, s)$  を出力するように学習を行う。推論の際には、入力文  $t_{in}(x, s)$  に対する出力  $r$  が、 $r = t_{out}(y, s)$  の様式を満たしていると仮定した上で、出力文から  $y$  を取り出し、それをスロットタイプ  $s$  に対するエンティティの予測とする。もし  $r$  が  $t_{out}(y, s)$  の様式を満たしておらず  $y$  を取り出せなかった場合には、エンティティが存在しないと予測したものとして処理する。

### 3.2 エンティティ例の利用

先行研究 [5, 2, 3] では、スロットの情報をモデルに与える手段としてそのスロットに対応するエンティティの具体例を利用することで、性能が向上することが示されている。提案手法と先行研究の公正な比較のためにエンティティ例の利用に関する設定はそろえる必要がある。そこでエンティティ例の利用に関して以下の3つの設定で比較を行う。

**w/o example** エンティティ例を使用しない設定。

**exp × 2** 2つのエンティティ例を使用する設定で、RCSF [2] で採用されている手法である。

**exp + context** 1つのエンティティ例と、そのエンティティが含まれていた発話文の組み合わせを利用する設定で、AISFG [3] で使われている。

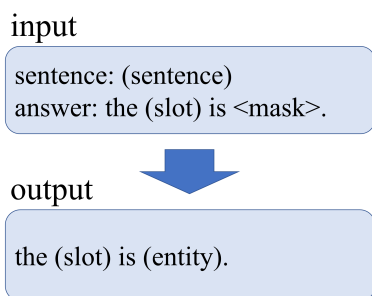


図3 w/o example のテンプレート

される。

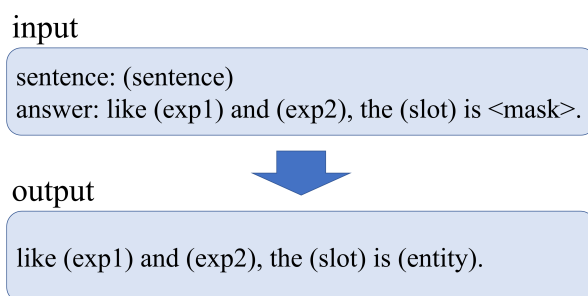


図5 exp + context のテンプレート

### 3.3 テンプレート

3つの設定のそれぞれに対して、テンプレートを作成した。図3は、w/o example のテンプレートであり、(sentence)には発話を、(slot)にはスロット説明を、(entity)には抜き出したエンティティを当てはめて、入出力文を作成する。ただし、スロットに対応するエンティティが発話中にない場合は(entity)に“unknown”と入れることにし、エンティティが複数の場合はカンマ区切りでエンティティを連結したものをいれるものとする。入力文のテンプレートは発話文と出力文のテンプレートをつなげて構成されており、このようにmaskトークンを用いて出力文の形式を入力として与えることで、テンプレート通りの出力文をモデルに出力させている。図4は、exp×2のテンプレートであり、(exp1)及び(exp2)に関しては抜き出したいスロットタイプのエンティティ例を入れる。図5はexp+contextのテンプレートであり、(exp)と(context)には、目的とするスロットタイプのエンティティ例とそのエンティティを抜き出したもととなった発話を当てはめる。

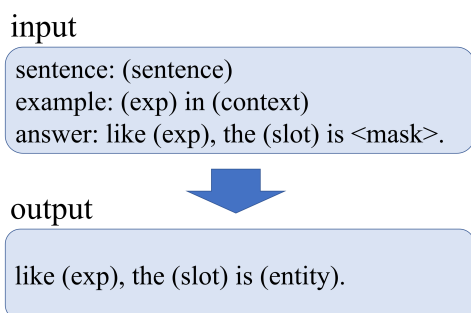


図4 exp×2のテンプレート

提案手法では、エンティティ例やスロット説明といった情報を入力文として与えるだけでなく、抜き出したエンティティと一緒に出力文にも含まれるように学習を行っており、それによりモデルの持つデコーダ由来の知識をタスクに活用できることが期待

## 4 実験

### 4.1 データセット

データセットは、先行研究[5, 6, 1, 2, 3]と同様にSNIPS[9]を用いた。SNIPSは、クラウドソーシングによって収集された発話理解のデータセットであり、7つのドメインにそれぞれ約2000の発話が収録されており、スロットは全部で39種類ある。実験では、7つのドメインのうち6つをソースドメインとして学習に用い、残りの1つのドメインをターゲットドメインとした。また、ターゲットドメインの発話のうち、500をバリデーションデータとし、その他の発話をテストデータとした。

### 4.2 ベースライン

ベースラインとして、以下の先行研究を、提案手法との比較に用いた。

**RCSF** Yuら[2]が提案したRCSFは、スロット説明から作った質問文を使って、抽出型質問応答タスクのようにクロスドメインスロットフィリングを行っている。事前学習済みモデルであるBERT-largeを質問応答用にファインチューニングしたものを、さらにファインチューニングして使用している。w/o exampleとexp×2の設定で提案手法と比較する。

**AISFG** AISFGは、2節で述べたようにseq2seqの手法であり、本研究のもととなった手法である。BART-baseをファインチューニングして利用している。w/o exampleとexp+contextの設定で比較する。

### 4.3 実験設定

公平な比較のためにベースライン手法とモデルのアーキテクチャをそろえて実験を行った。つまり、AISFGとの比較にはBART-baseを、RCSFとの比較にはBART-largeをファインチューニングして使用

した。エンティティ例や発話例についても、ベースラインモデルと同様のものを使用した。

実験は、無作為に選ばれた3つのランダムシードを用いて行い、その平均のスコアをそのモデルのスコアとした。

評価指標としてはF1を用いる。それぞれのドメインをターゲットドメインとしたときのF1スコアを算出し、その平均を最終的な評価に用いた。

## 4.4 実験結果

表1 提案手法 (BART-base) と AISFG の比較 (単位%)

setting	w/o exp		exp + context	
domain ↓ model →	AISFG	proposed	AISFG	proposed
AddToPlaylist	54.57	<b>58.51</b>	56.20	<b>61.12</b>
BookRestaurant	63.48	<b>69.01</b>	65.94	<b>66.38</b>
GetWeather	61.09	<b>71.94</b>	67.66	<b>75.28</b>
PlayMusic	<b>44.41</b>	44.01	50.12	<b>51.04</b>
RateBook	30.67	<b>33.81</b>	41.05	<b>42.62</b>
SearchCreativeWork	65.60	<b>66.24</b>	<b>67.46</b>	58.70
SearchScreeningEvent	30.33	<b>46.23</b>	35.05	<b>50.84</b>
Average F1	50.02	<b>55.68</b>	54.78	<b>58.00</b>

表2 提案手法 (BART-large) と RCSF の比較 (単位%)

setting	w/o exp		exp × 2	
domain ↓ model →	RCSF	proposed	RCSF	proposed
AddToPlaylist	<b>68.70</b>	59.80	<b>70.35</b>	61.18
BookRestaurant	63.49	<b>72.56</b>	72.68	<b>76.57</b>
GetWeather	65.36	<b>74.44</b>	<b>83.17</b>	82.34
PlayMusic	<b>53.51</b>	48.64	53.84	<b>57.64</b>
RateBook	36.51	<b>44.44</b>	50.08	<b>62.97</b>
SearchCreativeWork	<b>69.22</b>	66.49	66.59	<b>69.98</b>
SearchScreeningEvent	33.54	<b>48.33</b>	<b>65.81</b>	58.99
Average F1	55.76	<b>59.24</b>	66.08	<b>67.10</b>

表1及び表2に、ベースラインと提案手法を比較した結果を示す。

まず AISFG との比較では、w/o example と exp + context の2つの設定において、どちらも、7つのうち6つのドメインにおいて提案手法が上回る結果となった。平均F1スコアは、w/o example では5.66%、exp + context では3.22%の差で提案手法が上回っており、提案手法が AISFG よりも高い性能を達成している。

次に、RCSF との比較について、平均F1スコアではw/o example の設定で3.48%、exp × 2 の設定で1.02%の差で提案手法が上回る結果となっている。しかし、ドメインごとの結果を見てみると、どちらの設定においても、7つのうち4つのドメインでは提案手法が上回っているが、残り3つで RCSF が上回っており、ターゲットドメインや設定によって優

劣が異なる結果となった。

## 4.5 出力文の効果

提案手法の出力形式の効果を検証するために、exp × 2 の設定において、図6の3つの出力テンプレートで比較を行った。

input

sentence: (sentence)  
answer: like (exp1) and (exp2), the (slot) is <mask>.

output

simple (entity).  
slot the (slot) is (entity).  
slot + exp like (exp1) and (exp2), the (slot) is (entity).

図6 出力形式の比較

表3 出力文の形式による性能の比較 (単位%)

	simple	slot	slot + exp
Average F1	61.74	66.54	<b>67.10</b>

結果を表3に示す。AISFG の出力形式である simple と比べて、提案手法の出力形式である slot + exp は5.36%高いスコアになっており、提案手法の出力文の形式が、性能の向上に大きく寄与していることが示された。また、simple と slot の間には4.80%の差があり、スロット説明を出力文に組み込んだことが、性能の向上において特に重要であることが分かった。slot では、“is”という単語を介してスロット説明によるエンティティの条件付けを行っており、これによってデコーダのもつ言語的な知識を引き出したことが性能向上につながったのではないかと考えている。

## 5 おわりに

本研究では、クロスドメインスロットフィリングにおいて、seq2seq でエンティティを文章に組み込んで出力する手法を提案した。実験の結果、提案手法はベースライン手法よりも高い性能を達成した。また、抜き出したエンティティのみでなく、タスクの手がかりとなる情報を出力に組み込み、文章形式で出力させたことが、性能の向上に寄与していることが分かった。本研究の結果は、事前学習済みモデルをファインチューニングする際に、入出力文のフォーマットがモデルの性能に大きな影響を持ちうることを示している。

## 参考文献

- [1] Xinya Du, Luheng He, Qi Li, Dian Yu, Panupong Pasupat, and Yuan Zhang. QA-driven zero-shot slot filling with weak supervision pretraining. In **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)**, pp. 654–664, Online, August 2021. Association for Computational Linguistics.
- [2] Mengshi Yu, Jian Liu, Yufeng Chen, Jinan Xu, and Yujie Zhang. Cross-domain slot filling as machine reading comprehension. In Zhi-Hua Zhou, editor, **Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21**, pp. 3992–3998. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track.
- [3] Yang Yan, Junda Ye, Zhongbao Zhang, and Liwen Wang. AISFG: Abundant information slot filling generator. In **Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 4180–4187, Seattle, United States, July 2022. Association for Computational Linguistics.
- [4] Ankur Bapna, Gökhan Tür, Dilek Z. Hakkani-Tür, and Larry Heck. Towards zero-shot frame semantic parsing for domain scaling. In **INTERSPEECH**, 2017.
- [5] Darsh Shah, Raghav Gupta, Amir Fayazi, and Dilek Hakkani-Tur. Robust zero-shot cross-domain slot filling with example values. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 5484–5490, Florence, Italy, July 2019. Association for Computational Linguistics.
- [6] Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. Coach: A coarse-to-fine approach for cross-domain slot filling. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 19–25, Online, July 2020. Association for Computational Linguistics.
- [7] 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)**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [8] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics.
- [9] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut

Lavril, et al. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. **arXiv preprint arXiv:1805.10190**, 2018.



## A ハイパーパラメータ

学習に使用したハイパーパラメータを表 4 に示す.

表 4 ハイパーパラメータ

optimizer	Adam
batch size	16
learning rate	$5 \times 10^{-6}$
beam search size	2
patience	5

## B 評価指標

F1 スコアは以下のように計算される.

$TP$  = 正しく抜き出せたエンティティの数

$FN$  = 抜き出さずに見逃してしまった数

$FP$  = 誤った箇所を抜き出してしまった数

$$F1 = \frac{2TP}{2TP + FN + FP}$$

ここで正しく抜き出すとは、スロットとエンティティともに正しいことを指しており、エンティティが正しく抜き出せていてもスロットタイプが誤っていたり、また抜き出したエンティティが長すぎたり短すぎたりする場合は不正解 (FP) として数える.

## C SNIPS の補足

スロット説明は先行研究 [3] と同じものを用いたが、基本的にはスロット名のアンダーバーをスペースで置き換えたものになっている. 表 5 にそれ以外のスロット説明を示す.

表 5 スロット説明

スロット	スロット説明
timeRange	time range
party_size_number	number
poi	position
sort	type
party_size_description	person
geographic_poi	geographic position
condition_temperature	temperature
condition_description	weather
object_part_of_series_type	series
object_select	this current
object_location_type	location type

また, 表 6 に snips の構成も示しておく.

表 6 SNIPS の構成

ドメイン	スロット
AddToPlaylist	music_item playlist_owner entity_name playlist artist
BookRestaurant	city facility timeRange restaurant_name country cuisine restaurant_type served_dish party_size_number poi sort spatial_relation state party_size_description
GetWeather	city state timeRange current_location country spatial_relation geographic_poi condition_temperature condition_description
PlayMusic	genre music_item service year playlist album sort artist track
RateBook	object_part_of_series_type object_select rating_value object_name object_type rating_unit best_rating
SearchCreativeWork	object_name object_type
SearchScreeningEvent	timeRange movie_type object_location_type object_type location_name spatial_relation movie_name