曖昧な要求と気の利いた応答を含む対話コーパスの収集と分類

田中 翔平 [1,3], 吉野 幸一郎 [2,1,3], 須藤 克仁 [1,3], 中村 哲 [1,3]
{tanaka.shohei.tj7, sudoh, s-nakamura}@is.naist.jp, koichiro.yoshino@riken.jp
[1] 奈良先端科学技術大学院大学
[2] 理化学研究所 ロボティクスプロジェクト
[3] 理化学研究所 革新知能統合研究センター

1 はじめに

タスク指向型対話システムとは，ユーザの要求に対してあらかじめ定義されたシステムの API などを用いて応答するシステムであり，スマートスピーカーやデジタルサイネージなどの形で実社会へと普及しつつある．しかしこれまで研究・実用化されたきたタスク指向型対話システム [1, 2] は，ユーザ発話の中に明確に要求が含まれていることを前提としたものが多く，ユーザの要求が曖昧な場合に，適切な応答を生成することが難しい．

一方で，人間のコンシェルジュやガイドなどは，曖昧なユーザ発話に対しても気の利いた応答を行う．例えば，ユーザが「ここの景色は綺麗だね」と言った場合に，「写真を撮りましょうか？」などと応答することができる．このように，特定の機能を要求しておらず，またそもそも何らかの機能を要求しているかどうかが曖昧なユーザ発話に対しても，ユーザが望むであろう行動が可能なシステムを構築することが本研究の目的である．そこでまず，ユーザ発話と，それに対する対話エージェントの気の利いた応答を含むコーパスを収集した．

ユーザとシステムの対話を想定したコーパスの収集方法としては，2 人の被験者をユーザ役とシステム役に割り当てて対話してもらう方法が一般的である [3, 4]．だが，全ての曖昧なユーザ発話に対して常に気が効いた応答を返すことは，人間であっても難しい．また，実際に気の利いた応答が認定できたとしても，システムが実行可能な行動はシステム自身に定義されている API などの機能に制約され，その応答を返すことが現実的ではない場合も多い．そこで本研究では，システム側の応答 70 種類をシステムが利用可能な API に基づいてあらかじめ定義し，各応答が気が利いているとみなせるような先行発話を，クラウドワーカーに入力してもらうことでコーパスを収集した．これは，API などに制約されるシステム応答に応じてこれらが有効な文脈を収集する方が，広範な文脈に対して現実的な気の利いた応答候補を収集できると考えたためである．

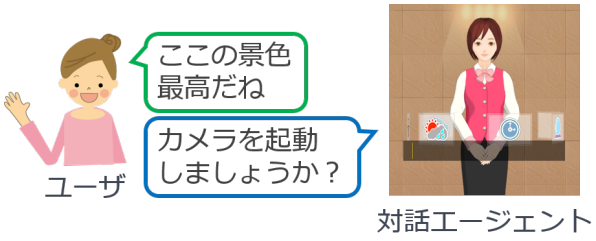

図 1 気の利いた対話例

こうして収集したコーパスに対し，ユーザの曖昧な要求に対応する気の利いた応答を分類するモデルを構築した．具体的には，Wikipedia などのコーパスを用いて事前学習を行った BERT[5] を用いて，ユーザ発話を対応するシステム応答へと分類する分類器を学習させた．学習した分類器を自動評価した結果，55%以上のユーザ発話を対応するシステム応答へと分類できることが判明した．また 83%以上のユーザ発話に関して，対応するシステム応答を上位 5 位以内へと分類できることが判明した．

2 データセット構築

ユーザの要求が曖昧な発話と，そうした発話に対するシステムの気の利いた応答を含むコーパスを収集する．本節ではクラウドソーシングを用いた収集方法について説明する．

本研究で収集するコーパスは，観光案内のドメインにおいてユーザとスマートフォンアプリケーション上の対話エージェントとの対話を想定したもので

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

表 1　対話エージェントの機能とカテゴリーのリスト

機能	カテゴリー	カテゴリー数
スポット検索	遊園地, 公園, スポーツ施設, 体験施設, お土産, 動物園, 水族館, 植物園, 観光案内所, ショッピングモール, 温泉, 寺院, 神社, 城, 自然・風景, 美術館, 博物館, 着物レンタル, 紅葉, 桜, 人力車, 駅, バス停, 休憩所, WiFi スポット, 静かなところ, 綺麗なところ, 楽しいところ, 広いところ, 眺めが良いところ	30
レストラン検索	カフェ, 抹茶, かき氷, 和菓子, 洋菓子, カレー, おばんざい, 豆腐料理, パン屋, ファーストフード, 麺類, 鍋料理, 丼もの・揚げ物, 肉料理, 寿司・魚料理, 粉もの, 京料理, 中華, イタリアン, フレンチ, 子供向けレストラン・ファミレス, 懐石料理, 精進料理, ベジタリアン向けレストラン, 居酒屋・バー, レストラン街, 朝食, 価格帯が安いレストラン, 価格帯が普通なレストラン, 価格帯が高いレストラン	30
アプリ起動	カメラ, 写真, 天気, 音楽, 乗り換え, メッセージ, 電話, アラーム, ブラウザ, 地図	10

表 2　収集したコーパスの統計情報

機能	平均ユーザ発話長	発話数
スポット検索	13.44(±4.69)	11,670
レストラン検索	14.08(±4.82)	11,670
アプリ起動	13.08(±4.65)	3,890
合計	13.66(±4.76)	27,230

ある．対話は全て一問一答形式であり，ユーザは要求が曖昧な発話や独話を行い，対話エージェントはそのユーザ発話に対して気の利いた応答を返す．図 1 にユーザと対話エージェントの対話例を示す．ここでユーザの「ここの景色最高だね」という発話は必ずしも特定の機能に対する要求というわけではない．これに対して，対話エージェントが「カメラを起動しましょうか？」という気の利いた応答を返し，実際のカメラ機能を起動できるようにする．

図 1 のような対話を収集する方法として，クラウドソーシングなどを利用して，2 人のワーカーにユーザ役と対話エージェント役に分かれて対話してもらう WoZ 対話が考えられる．しかし，先に述べたように，意図の曖昧なユーザ発話に対して常に気の利いた応答を返すことは，人間にとっても難しいタスクであり，一般的な WoZ 対話では期待する気の利いた応答候補を収集することが難しい．また，対話エージェントがスマートフォン上などで稼働するアプリケーションとして実運用されることを想定すると，その応答はアプリケーションが利用可能な API の機能に紐付いている必要がある．すなわち，対話エージェントにとって可能な気の利いた応答の範囲はユーザ発話の範囲と比較して限定的であり，対話エージェントの行動空間をあらかじめ定義した方がコーパスの質を担保しやすい．これらのあらかじめ定義されたエージェントの行動カテゴリーに対して，広範なユーザの先行発話をワーカーに入力してもらう方法により収集する．

本研究では対話エージェントの機能をスポット検索，レストラン検索，アプリ起動の 3 つに大きく分けて定義した．定義した機能のリストを表 1 に示す．各機能はそれぞれ細分化されたカテゴリーを持ち，コーパス中の対話エージェントの応答はこれらのカテゴリーに紐付いて生成される．定義したカテゴリーは全部で 70 種類である．スポット検索は特定のカテゴリーのスポットを検索する機能であり，「近隣の美術館を検索しましょうか？」といった応答としてユーザに提示される．レストラン検索は特定のカテゴリーのレストランを検索する機能であり，「近隣のかき氷を検索しましょうか？」といった応答としてユーザに提示される．アプリ起動は特定のアプリケーションを起動する機能であり，「カメラを起動しましょうか？」といった応答としてユーザに提示される．

定義した対話エージェントの応答カテゴリーに基づき，日本語コーパスをクラウドワークス[1)]を用いて収集した[2)]．収集したコーパスの統計情報を表 2 に，コーパス中の発話例を表 3 に示す．表 3 より，定義された応答を気が効いているとみなせるような，要求が曖昧なユーザ発話を収集できていることがわかる．収集した 27,230 ユーザ発話を含むコーパスを，学習データ : 検証データ : テストデータ = 24,430 : 1,400 : 1,400 の割合で分割した．ここで，各データに含まれる各カテゴリーの割合が同一になるようにコーパスを分割した．

1）　https://crowdworks.jp/
2）　収集画面や方法の詳細は A 付録に示す．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

表 3　コーパス中のユーザ発話例

ユーザ発話（クラウドソーシングを用いて収集）	システム応答（あらかじめ定義）
汗をかいて気持ち悪い	周辺の温泉を検索しましょうか？
歩くと時間がかかるな.	周辺のバス停を検索しましょうか？
子供達がお腹が減っちゃって駄々こねてるね.	周辺の子供向けレストラン・ファミレスを検索しましょうか？
最近は和食が多くて少し飽きてきたんですよね.	周辺の肉料理を検索しましょうか？
いい景色だな	カメラを起動しましょうか？
皆にホテルの部屋番号を伝えないと.	メッセージを起動しましょうか？

表 4　使用した事前学習済み BERT モデル

モデル名	公開元	モデルサイズ [5]	学習コーパス	トークナイザー	語彙サイズ
Kurohashi Kurohashi L	京都大学 黒橋研究室 [8]	BASE LARGE	日本語 Wikipedia 全記事	Juman++[6] & BPE[7]	32,000
NICT NICT BPE	NICT[9]	BASE	日本語 Wikipedia 全記事	MeCab[10] MeCab & BPE	100,000 32,000
hottoSNS	hottolink[11]	BASE	SNS 中の 85,925,384 投稿	sentencepiece[12]	32,000

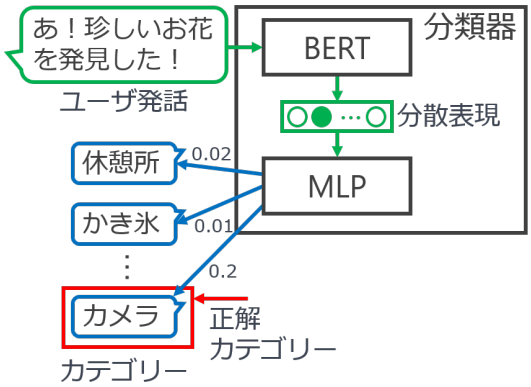

図 2　ユーザ発話の分類タスクとモデル

表 5　ユーザ発話の分類結果

モデル	Acc. (%)	R@5 (%)	MRR
Kurohashi	58.36	**87.14**	**0.71**
Kurohashi L	57.21	86.21	0.70
NICT	55.50	83.93	0.68
NICT BPE	**58.50**	86.50	**0.71**
hottoSNS	55.14	84.36	0.68

表 6　トークナイズ後の未知語の割合

モデル	未知語の割合 (%)
Kurohashi (L)	0.44
NICT	2.55
NICT BPE	0.85
hottoSNS	8.73

表 7　ユーザ発話の分類を間違えやすいカテゴリー

順位	正解カテゴリー	誤り数
1	自然・風景	17.00(±2.65)
2	寺院	14.60(±1.67)
3	公園	13.60(±1.52)
4	豆腐料理	13.40(±1.14)
5	ブラウザ	13.20(±1.10)

3　ユーザ発話分類モデル

前節で収集したコーパスを用いて，ユーザ発話を対応する応答のカテゴリーへと分類するモデルを構築・評価した．

3.1　分類モデル

図 2 にユーザ発話の分類タスク及びモデルの概要を示す．分類モデルは入力されたユーザ発話を，そのユーザ発話に紐付けられた対話エージェントの正解応答カテゴリーへと分類する．具体的な手続きとしては，BERT[5] によってユーザ発話を分散表現へと変換した後，1 層の Multi Layer Perceptron (MLP) によって各カテゴリーに対応する確率値 p を算出し，全 70 カテゴリーのうち最も高い確率値を持つカテゴリーを予測カテゴリーとして出力する．モデルの学習には Softmax Cross Entropy 損失関数を用いる．

3.2　実験

本実験では BERT モデルとして，Wikipedia 記事などを対象に事前学習を行った 5 つの日本語 BERT モデル [8, 9, 11] を用いた．表 4 に使用した事前学習済み BERT モデルの一覧を示す．各モデルは隠れ層の大きさなどのモデルサイズ [5] や学習に用いたコーパス，文をトークンに分割するためのトークナ

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

表 8　正解カテゴリーと誤って分類したカテゴリー

ユーザ発話例	正解カテゴリー	予測カテゴリー
自然に癒やされたいな	自然・風景	植物園
心を清めたいな.	寺院	神社
混雑していたから，少し気分が悪くなってきた.	公園	静かなところ
京都ならではの料理ってなんだろう	豆腐料理	おばんざい
○○には行ったことないな.	ブラウザ	地図

イザーなどが異なっている．

構築した分類モデルの性能を，テストデータを用いて評価した結果を表 5 に示す．R@5 (Recall @ 5) は，分類モデルが出力した正解カテゴリーの順位が，上位 5 位以内に含まれている割合である．MRR ($0 < MRR \leq 1$) は Mean Reciprocal Rank の略であり，次式の通り算出される．

$$MRR = \frac{1}{|U|} \sum_{u \in U} \frac{1}{rank_u} \quad (1)$$

ここで $rank_u$ はユーザ発話 u に対応する正解カテゴリーについて，分類モデルが出力した順位を意味し，U はテストデータに含まれるユーザ発話の集合である．Acc. (Accuracy), MRR ともに数値が大きいほど，分類モデルの性能が高いことを意味する．

表 5 を見ると，どのモデルも 55%以上のユーザ発話に対応する正解カテゴリーを予測できていることがわかる．また R@5 の数値より，どのモデルも 83%以上のユーザ発話に対応する正解カテゴリーの順位を上位 5 位以内に予測できていることがわかる．全体の傾向として，事前学習に SNS コーパスを用いた BERT よりも，Wikipedia コーパスを用いた BERT の方が性能が高い．ここで，学習データ中のユーザ発話をトークナイズした後の，モデルごとの未知語の割合を表 6 に示す．表 6 より，SNS コーパスから構築したトークナイザーを用いた場合よりも，Wikipedia コーパスから構築したトークナイザーを用いた場合の方が，未知語の割合が低いことがわかる．すなわち SNS コーパスよりも，Wikipedia コーパスの方が本研究にて対象とした観光ドメインに関連するテキストが多く含まれていたため，事前学習に Wikipedia コーパスを用いた BERT の方が性能が高かったと考えられる．

さらに，どのカテゴリー応答に先行するユーザ発話の分類を間違えやすいのかを調査した結果を表 7 に示す．誤り数は全モデルの平均を算出したものを使用している．表 7 を見ると，正解カテゴリーが自然・風景である場合の先行発話の分類を最も多く間違えており，こうした先行発話を正解カテゴリーに分類することが難しいことがわかる．

ただし，詳細な分類結果を見ると，分類誤りとされているものが実際には分類誤りでない場合がある．正解カテゴリーと誤って分類したカテゴリーの例を表 8 を示す．表 8 を見ると，これらのユーザ発話は正解カテゴリーの他に，予測カテゴリーの応答を用いたとしても誤りというわけではない．つまり，これらの応答カテゴリーに対応するユーザ発話はそもそも複数の応答候補を取り得る，マルチラベル問題であり，ユーザ発話と応答カテゴリーを一対一で対応付ける今回の収集方法は，こうした対応を収集できていない可能性が示唆される．

4　おわりに

本論文では，対話エージェントの機能に着目し，ユーザの曖昧な要求と対話エージェントの気の利いた応答を含むコーパスを収集した．具体的な方法として，API に基づきあらかじめ定義された対話エージェントの応答に対し，その応答が気が利いているとみなせるような先行発話をクラウドワーカーに入力してもらった．さらに，収集したコーパスを用いて，曖昧なユーザ発話を対応する気の利いたシステム応答のカテゴリーへと分類する分類器を構築した．構築した分類器を評価した結果，55%以上の正解カテゴリーを 1 位に，83%以上の正解カテゴリーを 5 位以内にランク付けできることが判明した．

一方で，曖昧なユーザ発話は複数のシステム応答カテゴリーと紐付き得ることが示唆された．今後はこうした複数の意図として解釈し得るユーザ発話に対して，マルチラベリングなどの手法を適用することを検討する．

謝辞

本研究にご協力いただいた理化学研究所革新知能統合研究センター観光情報解析チームの皆様に感謝いたします．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

参考文献

[1] Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. Mem2Seq: Effectively Incorporating Knowledge Bases into End-to-End Task-Oriented Dialog Systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1468–1478, 2018.

[2] Andrea Vanzo, Emanuele Bastianelli, and Oliver Lemon. Hierarchical multi-task natural language understanding for cross-domain conversational ai: Hermit nlu. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 254–263, Stockholm, Sweden, September 2019. Association for Computational Linguistics.

[3] 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*, pp. 5016–5026, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[4] Dongyeop Kang, Anusha Balakrishnan, Pararth Shah, Paul Crook, Y-Lan Boureau, and Jason Weston. Recommendation as a communication game: Self-supervised bot-play for goal-oriented dialogue. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1951–1961, Hong Kong, China, November 2019. Association for Computational Linguistics.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 18th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186, 2019.

[6] Arseny Tolmachev, Daisuke Kawahara, and Sadao Kurohashi. Juman++: A morphological analysis toolkit for scriptio continua. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 54–59, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[7] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.

[8] 柴田知秀, 河原大輔, 黒橋禎夫. BERT による日本語構文解析の精度向上. 言語処理学会 第 25 回年次大会 発表論文集 (ANLP), pp. 205–208, 2019.

[9] National Institute of Information and Communications Technology (NICT). NICT BERT (https://alaginrc.nict.go.jp/nict-bert/index.html). 2020.

[10] Taku Kudo. MeCab (https://taku910.github.io/mecab/). 2006.

[11] Takeshi Sakaki, Sakae Mizuki, and Naoyuki Gunji. Bert pre-trained model trained on large-scale japanese social media corpus. 2019.

[12] Taku Kudo and John Richardson. SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

A　付録

図 A.1　コーパス収集におけるインストラクションおよび入力フォーム．

図 A.1 はコーパス収集におけるインストラクションおよび入力フォームの一例である．本研究で収集するユーザ発話は要求が曖昧である必要があるため，良くない例として「周辺の休憩所を検索して.」という要求が明確な発話を挙げ，そのような発話は入力しないよう注記している．1 人のワーカーごとにそれぞれ異なる 10 カテゴリーに対してユーザ発話を入力してもらった．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).