旅行情報サイトのレビューを用いた抽象的な要求に対する根拠付き推薦文の生成

叶内 晨 *[1]　根石 将人[2]　林部 祐太[1]　岡崎 直観[3]

[1] 株式会社リクルート Megagon Labs, Tokyo　[2] 東京大学　[3] 東京工業大学

1 はじめに

コールセンターなどの対話型サービスで，自然言語処理技術を利用した機械による対話システムの導入が進んでいる．しかしながら，言語タスクの難しさや現在の対話システムの限界から，未だ人手が欠かせない状況である．例えば，旅行情報サイトにおける対話型の宿推薦サービスでは，ユーザが提示する要求に合わせてオペレータが宿を推薦する必要がある．この際，ユーザは日程や人数などの具体的な要求だけでなく，「子連れに優しい宿が良い」などの抽象的な要求[1]を提示することがある [1]．具体的な要求はルールとして使うことで条件に合致した宿の検索が可能だが，抽象的な要求は曖昧性が高く対処が難しい．抽象的な要求に対してオペレータは，宿の推薦と同時に「こちらの宿は，キッズスペースもあり子連れにオススメです」と推薦根拠も提示することが多い．このような抽象的な要求に対する推薦根拠の提示は，推薦の説得力を増すだけでなく，ユーザのより具体的な要求を引き出すことを可能にする．

本研究では，抽象的な要求に対して根拠を含んだ推薦文を提示する対話システムの実現を目指す．具体的には，宿推薦のための対話システムを対象として，抽象的な要求と推薦対象の宿のレビューデータが与えられた状況下で，(1) レビュー本文から推薦根拠を含む文を抽出し，(2) 得られた根拠文を推薦文へ言い換える．

まず，根拠を含む文の抽出のために，旅行情報サイトのレビューデータに対して，そのタイトルの一部を抽象的な要求とみなし，レビュー本文の各文が要求に対応する根拠を含むかをクラウドソーシングを用いてアノテーションし，根拠文判定データセットを構築する．次に，推薦文言い換えのために，クラウドソーシングを用いて，上記で得た根拠文を推薦文として適切な文へ言い換え，推薦文言い換えデータセットを構築する．最後に，以上の2つからなる根拠説明データセットを利用することで，要求と推薦対象の宿のレビューを入力として根拠を含む推薦文を出力するシステムを構築する．実験では，根拠文判定と推薦文言い換えを段階的に行うモデルと両方を一括で行うモデルの2種類を比較した．結果として，より精度の高い段階的モデルでは，BLEU スコア 41.54 で要求に対する根拠付きの推薦文を生成できることを示した．本研究で構築したデータセットは公開予定である．

2 根拠説明データセットの構築

本節では，ユーザからの抽象的な要求に対して，根拠を含んだ推薦文を生成するためのデータセット構築の手順を説明する．本研究では，旅行情報サイトじゃらん net[2] の宿に関するレビューデータのタイトルと本文を利用する．レビュー本文（以下，「本文」という）には，ユーザの実体験に基づく宿のサービスの描写やそれに対する感想が記述され，一方でレビュータイトル（以下，「タイトル」という）には，その中でも特に満足した点が一言で要約して記述されやすい．この要約記述されたタイトルは，対話型サービスにおいてユーザが提示する「子連れに優しい宿」などの抽象的な要求と表現が類似することがある．また，それらのタイトルを抽象的な要求とみなした場合，その本文には要求に対する推薦根拠が記述されていると捉えられる．そこで本研究では，宿のレビューデータについてタイトルの一部を抽象的な要求とみなし，本文からその要求に対応する根拠を含む文を抽出し，さらにその根拠文を推薦文へ言い換える．

データセット構築のための全体のパイプラインを図 1 に示す．処理は次の 3 段階である．

1. **レビュータイトルから抽象的な要求を収集**
   「宿」を表す表現を含むなどのルールにより，抽象的な要求とみなせるタイトルを収集する．
2. **根拠文判定（クラウドソーシングを利用）**
   収集した要求をタイトルにもつレビューについて，本文の各文がその要求に対する根拠を含むか判定する．
3. **推薦文言い換え（クラウドソーシングを利用）**
   要求の根拠を含むと判定された文を，その要求をもつユーザに対する根拠付きの推薦文へと言い換える．

*shin187nlp@megagon.ai

[1] 本研究おいて，抽象的な要求とは具体的な商品や体験・サービスを指定しない要求のことを指す．

[2] https://www.jalan.net/

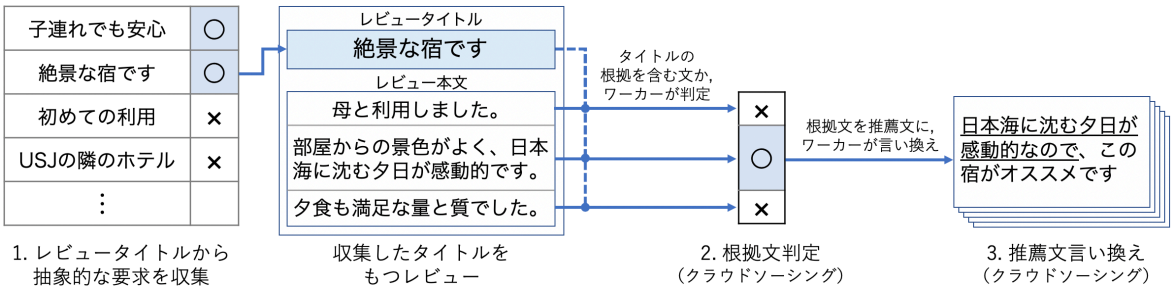

図 1: 根拠説明データセット構築のパイプライン

表 1: 収集した抽象的な要求の例と収集数・利用数

カテゴリ	例 拡張前	例 拡張後	収集数 拡張前	収集数 拡張後	利用数
綺麗	綺麗なホテル	綺麗です	15k	71k	4.6k
ゆっくり	寛げる旅館	寛げました	8k	80k	4.6k
接客	親切な宿	対応が親切	10k	143k	4.6k
便利	便利なホテル	観光に便利	4k	113k	3.6k
子連れ	子供に優しい宿	子供に優しい	3k	81k	3.6k
景色	絶景のホテル	絶景でした	1k	34k	3.1k
食事	美味しい宿	美味しい朝食	1k	145k	3.1k
コスト	コスパ良い宿	コスパ最高	5k	89k	3.1k
グッド	最高の宿	最高でした	33k	278k	3.6k
その他	昭和な宿	昭和の雰囲気	19k	297k	16.0k

2.1 抽象的な要求の収集

本研究では，タイトルの一部が抽象的な要求と類似することに着目し，タイトルを選別することで抽象的な要求を収集する．実際のタイトルには，本研究が対象とする「子連れに優しい宿」のような表現以外にも，「お世話になりました」や「初めての利用」など，抽象的な要求とはみなせないノイズも多い．そこで，宿に関する要求を低ノイズかつ網羅的に収集するため，「宿・ホテル」などの宿を表す表現を含むタイトルを抽出した．さらに，「USJ の隣のホテル」のような具体的な名所を含むタイトルなどを，追加のルール[3]で排除した．

収集したデータに偏りがある可能性を考慮して，SudachiPy(v0.3.4)[4] による形態素解析結果の代表表記を用いて表記揺れを吸収し，頻度上位のものから人手によって確認することで，要求をカテゴリ[5]分けした．結果，カテゴリ間の収集数の偏りだけでなく，いくつかのカテゴリのデータが予想に反して少ないことが分かった．この原因として，宿を表す表現に連結されにくい要求[6]の存在が確認されたため，抽出したタイトルから宿を表す表現を除いたものと同じ内容語を含むタイトルも抽出対象[7]とし，収集する要求表現を拡張した．

表 1 に収集した抽象的な要求の例を示す．拡張によって，収集数を大幅に増加し，データの偏りを軽減できた．また一方で要求表現のバリエーションも増加した．利用数には，本研究においてデータセット構築のために利用したレビュー数を示す．

表 2: 根拠文の数とレビュー数

根拠文の数	レビュー数	
なし	24,034	(48.1%)
1 文	21,675	(43.4%)
2 文	3,464	(7.0%)
3 文以上	827	(1.7%)

2.2 根拠文判定データセットの構築

要求に対する根拠文抽出のために，Yahoo!クラウドソーシング[8]を用いてアノテーションをした．この際，タスクを 2 段階のシンプルなサブタスクに分けることで，ワーカーが直感的に作業できるようにした．

まず 2.1 節で収集したレビューを対象とし，タイトルと関係する文を本文から抽出した．具体的には，タイトルおよび文分割した本文すべてを提示し，**「タイトル」であることがわかる文にチェックしてください**，という質問により，本文の各文がタイトルに関係するか判定した．

次にタイトルに関係すると判定された文に対して，タイトルに対応する根拠が含まれるか判定した．具体的には，タイトルと対象の本文 1 文のみを提示し，**「タイトル」なぜなら〜だから，と言える根拠を本文は含みますか**，という質問により，タイトルに対応する根拠が文に含まれているか判定した[9]．

1 段階目の関係文判定タスクは，表 1 の利用数に示した 5 万レビューを対象とし，2 段階目の根拠文判定タスクは，関係文判定タスクにおいてワーカー 5 人のうち 3 人以上がタイトルに関係すると答えた 45,190 文を対象とした．品質向上のためにアノテーションは 1 件につき 5 人で行い，また各タスクにおいて答えを用意したチェック問題を設け，これに正解したワーカーのデータのみ採用した．根拠文判定タスクにおいて，ワーカー 5 人のう

[3]ルールは以下．固有名詞を含まない．「宿」以外の内容語を 1 語以上含む．句点を含まない．3 文字以上 15 文字以下など．
[4]https://github.com/WorksApplications/SudachiPy
[5]データの比率調整とエラー分析のために，著者が適当に作成した．
[6]例えば，「美味しい宿」より「ご飯が美味しい」と記述されやすい．
[7]例えば，綺麗な宿→綺麗 (宿を除く) →綺麗です (収集対象を拡張)
[8]https://crowdsourcing.yahoo.co.jp/
[9]2 文以上の組み合わせにより根拠として意味を成す場合も考えられるが，タスク簡単化のために単一文毎にアノテーションした．

表 3: ワーカー間の推薦文の一致数

一致数	文の数	
3 人以上一致	7,646	(24.4%)
2 人一致	13,850	(44.2%)
一致なし	6,438	(20.5%)
根拠なし報告	1,155	(3.7%)
ネガティブ報告	2,262	(7.2%)

表 4: 根拠文予測の F1 スコア

	ロジスティック回帰	BERT
F1	47.92	74.07

ち 3 人以上が根拠を含むと答えた文を根拠文とした．

表 2 に，レビュー毎の要求に対する根拠文の数の分布を示す．約半数のレビューは，タイトルに記述された要求に対する根拠が本文内に含まれていることが分かる．

2.3 推薦文言い換えデータセットの構築

2.2 節で根拠を含むと判定された各文を対象とし，クラウドソーシングを用いて，タイトルに記述された要求をもつユーザに対する根拠付きの推薦文へと言い換えた．タイトルと 2.2 節で根拠文と判定された文を提示し，**「タイトル」の宿を探している人に対して「〜なのでこの宿がオススメです」と根拠付きの説明をしてください**，というタスクを設計した．作業前の事前の説明として，必要に応じた元の根拠文の修正は認めつつ，可能な限り元の表現を用いて作文することを指示した．例外として，要求の感情表現がネガティブで推薦文として相応しくない場合と，与えられた文に根拠が記述されていない場合は，別途報告するよう指示した．

本タスクは 2.2 節でワーカー 5 人のうち 3 人以上が根拠を含むと答えた 31,351 文を対象とした．品質向上のため，1 文につき 5 人がアノテーションし，また要求がネガティブであるチェック問題を設定し，これを適切に報告したワーカーのアノテーションデータのみ採用した．チェック問題は通過しているものの，何も入力されていないなどの問題がある推薦文は，ルールによるフィルタリングを行い，残ったデータを後続の実験に利用した．

表 3 に，作成した推薦文における 5 人のワーカーの一致数を示す．言い換え時に，元の文を抽出するだけで「ので」に接続できる場合，推薦文が一致する傾向があった．

3 実験

2 節で構築したデータセットを用いて，(1) 要求と各文が与えられた時の根拠文の予測モデル，(2) 要求とその根拠文が与えられた時の推薦文の生成モデル，(3) 要求と本文が与えられた時にそれらを一括で行うモデルを構築した．まず，根拠文予測と推薦文生成について個別に評価し，その後，それらを組み合わせたモデルと両方を一括で行うモデルの 2 種類を比較評価した．

3.1 根拠文予測タスク

実験設定 2.2 節で構築したデータセットのうち根拠文と判定された文を正例，それ以外の文を負例とし，要求の根拠が含まれる文かを予測する二値分類問題とした．データはレビュー単位でランダムに 8 : 1 : 1 の比率で分割し，学習データ : 検証データ : テストデータ = 40,000 件（202k 文）: 5,000 件（25k 文）: 5,000 件（25k 文）とした．以降の実験でもすべて同じ分割のデータを用いる．

分類器にはロジスティック回帰[10]と BERT[2] を利用した．ロジスティック回帰の素性には，タイトルと本文をそれぞれ SentencePiece[11]で分割後，tf-idf ベクトルと word2vec(CBOW)[12]の平均ベクトルをそれぞれ作成し，各ベクトルやタイトルと本文の差分ベクトルなどを利用した．BERT は，SentencePiece で分割後，データセットと同一情報源の大規模なレビューデータで pre-train した．fine-tuning 時は，入力として，[CLS] トークン，要求，[SEP] トークン，本文の順に連結したものを与え，[CLS] トークンに対応する最終層の上に 1 層の出力層を接続したモデルで，根拠文かどうかの二値分類問題を解いた．

精度 各分類器による予測結果を表 4 に示す．BERT による分類結果の F1 スコアは 74.07 となり，ロジスティック回帰をおよそ 26 点上回った．

3.2 根拠文からの推薦文言い換えタスク

実験設定 根拠文から推薦文への言い換えを実験した．2.3 節で構築したデータのうち，対応する推薦文が 3 文以上あるデータのみを用いた．根拠文毎に複数ワーカーの推薦文があるが，それらすべてを学習データとして扱った．

文生成モデルは，FAIRSEQ[13]の実装を利用し， Luong らの LSTM に基づくモデル [3] と，Vaswani らの Transformer[4] を用いた．前処理として推薦文の共通文末である「この宿がオススメです」という表現を削除後，SentencePiece で分割した．モデルの入力には，要求，[SEP] トークン，根拠文の順に連結したものを用いた．

評価には，SentencePiece による分割結果に対する BLEU を測定した．BLEU 測定時は，複数のワーカーが作成した推薦文から 1 つをランダムに抽出したものを参照訳として用いた．

精度 根拠文入力時の推薦文生成の BLEU スコアを表 5 に示す．「言い換えなし」は入力をそのまま出力した場合である．両モデルとも BLEU は非常に高く，また LSTM モデルは Transformer を 7 点ほど上回った．

[10] https://scikit-learn.org
[11] https://github.com/google/sentencepiece
[12] https://radimrehurek.com/gensim/models/word2vec.html
[13] https://github.com/pytorch/fairseq

表 5: 根拠文入力時の推薦文生成の精度

手法	BLEU
言い換えなし	49.17
Luong らの LSTM	67.10
Vaswani らの Transformer	59.50

表 6: レビュー入力時の推薦文生成の精度

手法	BLEU	F1
段階的モデル (BERT → LSTM)	41.54	62.21
一括モデル (LSTM)	18.20	46.74

表 7: レビュー入力時の推薦文生成の例

タイトル	本文	正解の推薦文（参照文）	段階的モデル	一括モデル
見晴らしが最高でした	初日はチェックイン前にプレミアム観光券を購入してから，東山動物園にイケメンゴリラのシャバーニに会いに行き，翌日はリニア・鉄道館に行って満喫しました．お部屋は名古屋城が綺麗に見えるところで，子供達も大喜びでした．....	[お部屋は名古屋城が綺麗に見えるところなので],[お部屋は名古屋城が綺麗に見えるところで，子供達も大喜びでしたので]	名古屋城が綺麗に見えるところで，子供達も大喜びでしたので	お部屋は名古屋城が綺麗に見えるので
わんこ連れで満足!	夫婦 2 人と父，わんこ 2 匹でお世話になりました．.... お部屋はゆったりしていて，大きな窓からは海が一望でき，とっても満足できました．.... ペット可能ホテルはまだまだがっかりする所が多い中，こちらのホテルはお薦めです．....	[ペット可能ホテルはまだまだがっかりする所が多い中，こちらのホテルはお薦めですので],[ペットと楽しめるので]	ペット可能ホテルはとても便利なので	お部屋はゆったりしていて，大きな窓からは海が一望でき，とっても満足できましたので
子連れにオススメの旅館	一歳四カ月の子供がいるので，夕食は部屋出しなのと貸切温泉があるプランに惹かれて宿泊しました．.... 敷き布団が年季が入っている感じで薄かったのでもっと厚いと良かったかな．温泉や接客は良く，子供にも優しく接して貰えて良かったです．	[子供にも優しく接してくれるので],[温泉や接客は良く，子供にも優しく接して貰えてたので]	温泉や接客は良く，子供にも優しく接して貰えたので	プールや接客は良く，子供にも優しいので

3.3 要求とレビュー入力時の推薦文生成タスク

実験設定 要求と本文 1 件を入力とし，根拠付きの推薦文を生成する実験（3.1 節と 3.2 節を一括で行う実験）を行った．要求に対する根拠が本文になく推薦文がないレビューも存在するため，その場合には推薦文ではなく，根拠がないことを示す特殊トークンの出力を正解とした．

モデルには，3.1 節と 3.2 節を組み合わせて段階的に処理する，段階的モデルと，Luong らの LSTM を用いて end2end で両方を一括で処置する，一括モデルを用いた．段階的モデルは，まず，BERT で本文の各文が要求に対する根拠文であるか予測後，根拠文があった場合のみ予測時のスコアが最大の 1 文を入力として，推薦文を生成した．一括モデルでは，根拠に対する推薦文があるレビューに対しては推薦文を出力し，それ以外に対しては根拠がないことを示す特殊トークンを出力するように学習した．一括モデルの入力には，要求，[SEP] トークン，本文の順に連結したものを用いた．評価には，3.2 節と同様の BLEU と，マクロ平均の F1 スコア[14]を用いた．

精度 表 6 に，両モデルの精度を示す．段階的モデルは BLEU スコアが 41.54 となり，一括モデルよりも 23 点高い．一括モデルは根拠がないと過剰に予測してしまう傾向があった．F1 スコアは BLEU に比べて根拠がないことを正解した時の加点が大きいく，段階的モデルでは 62.21 となり，一括モデルを 15.5 点上回った．

表 7 にモデルの出力例を示す．1 例目は，どちらのモデルも生成に成功している．2 例目は，一括モデルにおいて，生成文のみを見ると自然な推薦文だが，今回の要求に対する根拠としては適切でない．「わんこ連れ」が低頻度な要求表現であり，うまく認識できなかったことが原因と考えられる．3 例目は，一括モデルにおいて，「温泉」が「プール」として出力されてしまい，事実と異なった説明をしている．

[14]SQuAD[5] の評価と同様に，参照訳と生成結果の bag-of-tokens の重複を F1 スコアで計算する．ただし，根拠文がないレビューに対しては，根拠がないと答えた場合にスコア 1, それ以外は 0 とする．

4 おわりに

本研究では，抽象的な要求に対する根拠付きの推薦文を提示するため，根拠説明データセットとそのシステムを構築し，評価した．近年，感情分析や推薦において根拠情報は注目されており [6, 7, 8, 9, 10, 11]，Zhao らは，楽曲の推薦における自然言語文の推薦根拠提示がユーザのクリック率を大幅に向上することを示している [12]．今後の展望としては，Zhao らのような実システムへの導入を考えている．また，本研究では，システムに入力する宿のレビューは与えられるものとして扱ったが，この選定を自動で行うことなどが課題である．

参考文献

[1] Yuliang Li, Aaron Feng, Jinfeng Li, Saran Mumick, Alon Halevy, Vivian Li, and Wang-Chiew Tan. Subjective databases. *Proc. VLDB Endow.*, 2019.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[3] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[5] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2016.

[6] Ying Chen, Sophia Yat Mei Lee, Shoushan Li, and Chu-Ren Huang. Emotion cause detection with linguistic constructions. In *COLING*, 2010.

[7] Lin Gui, Dongyin Wu, Ruifeng Xu, Qin Lu, and Yu Zhou. Event-driven emotion cause extraction with corpus construction. In *EMNLP*, 2016.

[8] Evgeny Kim and Roman Klinger. Who feels what and why? annotation of a literature corpus with semantic roles of emotions. In *COLING*, 2018.

[9] Haijing Liu, Yang Gao, Pin Lv, Mengxue Li, Shiqiang Geng, Minglan Li, and Hao Wang. Using argument-based features to predict and analyse review helpfulness. In *EMNLP*, 2017.

[10] Koji Murakami, Eric Nichols, Suguru Matsuyoshi, Asuka Sumida, Shouko Masuda, Kentaro Inui, and Yuji Matumoto. Statement map: assisting information crediblity analysis by visualizing arguments. In *WICOW*, 2009.

[11] Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain yourself! leveraging language models for commonsense reasoning. In *ACL*, 2019.

[12] Guoshuai Zhao, Hao Fu, Ruihua Song, Tetsuya Sakai, Zhongxia Chen, Xing Xie, and Xueming Qian. Personalized reason generation for explainable song recommendation. *ACM Trans. Intell. Syst. Technol.*, 2019.