頑健な FAQ 検索に向けた Prompt-Tuning を用いた関連知識の生成

二宮 大空 [1]　邊土名 朝飛 [2]　友松 祐太 [1]
[1] 株式会社 AI Shift　　[2] 株式会社サイバーエージェント
{ninomiya_hirotaka, hentona_asahi, tomomatsu_yuta}@cyberagent.co.jp

概要

チャットボットが提供する機能の一つに，よくある質問集（FAQ: Frequently Asked Questions）を用いてユーザの質問に回答する FAQ 検索がある．FAQ 検索では，目的語が欠落しているなど，与えられる質問が不明瞭なことが多く，検索精度の低下の要因となり得る．そこで，我々は言語生成モデルを用いて質問の明確化を行い，FAQ 検索に活用する手法を提案する．質問の明確化のためにモデルが獲得すべき知識がドメインごとに異なることから，言語生成モデルの学習コストを抑えることができる Prompt-Tuning を用いた．チャットボット事業で収集したデータを用いた実験では 3 ドメインにおいて提案手法の有効性を検証した．

1　はじめに

カスタマーサポートの分野において注目を集めているチャットボットは，現在多くの企業においてユーザの課題解決を促すツールとして導入されている．そして，チャットボットが提供する機能の一つである FAQ 検索は現在盛んに研究が行われている [1][2].

FAQ 検索では，事前に定義した FAQ の中からユーザ質問との類似度が高い FAQ を選択する．この時，ユーザ質問と FAQ の間で表記が異なることが多く，質問の意味を考慮する必要があることから，我々は事前学習済み言語モデルを用いた検索モデルを構築している．特に，オープンドメイン質問応答や日本語クイズタスク JAQKET[3][4] において有効性が確認された Dense Passage Retrieval[5] の検索器（Retriever）を利用している．

FAQ 検索はオープンドメイン質問応答と異なり，検索対象となるテキスト集合が数十件から数百件程度の小規模な FAQ のデータベース（FAQDB）であるが，与えられるユーザ質問は比較的短く不明瞭な傾向にあり，回答が困難なタスクと考えられる．例えば，「入りたいのに入れません」というユーザ質問は「入りたい」に対する目的語が欠落しているため一般的には回答不能であるが，FAQ 検索の場合は検索対象が FAQDB に絞られるため，「マイページに入れません」といった質問とマッチすれば良いと推察できる．このことから，ユーザ質問を明確化するテキストを補足することができれば，より正確に回答を提示できる可能性がある．そこで，本研究では与えられたユーザ質問からそれを明確化するテキストを生成するモデルの学習を目指す．

ただし，FAQ 検索は複数のドメインで導入されることが多く，大規模な言語生成モデルをドメインごとに学習すると学習コストが膨大になる．そこで言語生成モデルの学習には Prompt-Tuning[6] を用いる．Prompt-Tuning では，質問の明確化に必要な情報を固定長の単語列としてモデルへの入力単語列の前方に付与し，その埋め込みベクトルを学習によって最適化する．これにより，それらの埋め込みベクトルを変更するだけで単一の言語生成モデルを複数のドメインに適応可能となるので，学習コストが抑えられる．

言語モデルの発展に伴い，多くのタスクにおいて事前学習済みモデルの Fine-Tuning が効果的であることが確認される一方で，事前学習済みモデルの大規模化により Fine-Tuning が困難な場合がある．そこで，GPT-3[7] では Prompt と呼ばれるテキストを与えることで，事前学習済みモデルの重みを更新せずに，広範囲のタスクを解くことが可能であることが確認された．さらに，Prompt に関する従来研究 [8][9] によると，Prompt を適切に設定することによって事前学習済みモデルを用いて事実や常識に関するテキストを生成することが可能であることが確認されたことから，Prompt が重要な役割を果

たすことがわかる．ただし，タスクの説明等に関する Prompt は人手で作成する必要があるため，事前学習済みモデルにとって最適な単語列であるとは限らない．そこで，先頭に固定長の単語列を付与し，それらに対する埋め込みベクトルを学習パラメタとして自動的に最適化する Prompt-Tuning が提案された [6][10]．これより，我々は質問の明確化に必要なドメインごとの知識を Prompt に埋め込むように学習できる可能性を考慮し，言語生成モデルを Prompt-Tuning する．

2 FAQ 検索

本研究で扱う FAQ 検索の処理を図 1 に示す．FAQ 検索とは，事前に定義された FAQDB から，ユーザ質問に最も適する FAQ を選択するタスクである．例えば図 1 のように，「銀行振り込みの自動更新について教えて欲しい」といったユーザ質問が入力された場合，システムは FAQDB を検索する．ここで FAQDB は (1) 拡張質問 Q'，(2)FAQ 質問 Q，(3)FAQ 回答 A を 1 組とする FAQ の集合である．通常，FAQ は質問と回答のペアで表されるが，質問の複数の言い換え表現として拡張質問 Q' を設定することで検索精度が向上することが知られている [11]．最終的に検索結果上位 1 件の拡張質問に対応する FAQ 回答をユーザに提示する．

検索モデルは Retriever と Reranker で構成される．Retriever は与えられたユーザ質問と検索対象となる拡張質問を個別にベクトルに変換する Bi-Encoder である．さらに，ユーザ質問と FAQ の単語間の Attention 計算が有効であると考え，Retriever の検索結果上位 K（実験では $K = 5$）件を BERT[12] をベースとする Reranker に与え，出力スコアを元にリランキングを行う．実験では，Retriever の検索結果と Reranker によるリランキングの結果を比較する．

学習時，Retriever はユーザ質問 q に対する Encoder の出力ベクトルと FAQ 質問 Q に対する Encoder の出力ベクトルの類似度を式 (1) により算出する．

$$\mathrm{sim}(q, Q) = E_q(q)^T E_Q(Q) \tag{1}$$

ここで E_q,E_Q はそれぞれユーザ質問 q と FAQ 質問 Q に対する Encoder を，^{T} は転置を表す．i 番目の事例において q_i をユーザ質問，Q_i^+ を正例の FAQ 質問，$Q_{i,j}^-$ を j 番目の負例の FAQ 質問，n を負例の FAQ 質問の総数，訓練データの件数を m とすると，訓練データは $D = \{\langle q_i, Q_i^+, Q_{i,1}^-, ..., Q_{i,n}^- \rangle\}_{i=1}^m$ と表さ

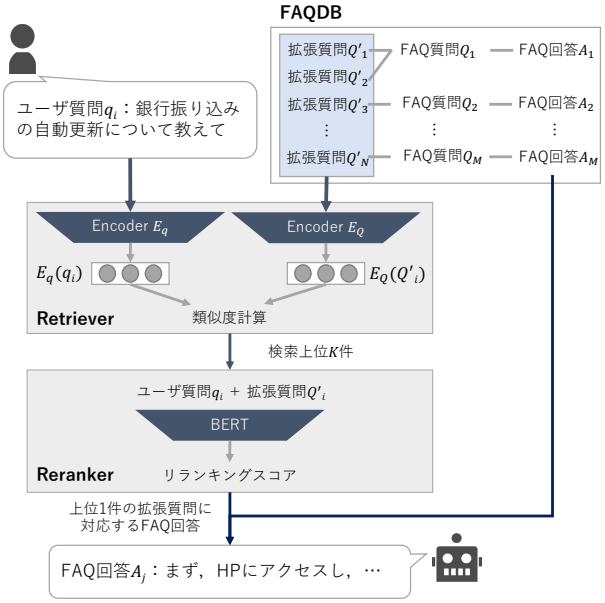

図 1 FAQ 検索の処理

れる．Retriever は式 (2) の損失関数が最小になるように学習される．ユーザ質問と正例の FAQ 質問が類似しているほど損失が小さくなり，負例の FAQ 質問との類似度が高いほど，式 (2) の損失は大きくなる．

$$L\left(q_i, Q_i^+, Q_{i,1}^-, ..., Q_{i,n}^-\right) = -\log\left(\frac{e^{\mathrm{sim}(q,Q_i^+)}}{e^{\mathrm{sim}(q,Q_i^+)} + \sum_{j=1}^n e^{\mathrm{sim}\left(q,Q_{i,j}^-\right)}}\right) \tag{2}$$

推論時はユーザの多様な言い回しに対応するために，FAQ 質問ではなく拡張質問を用いて検索を行う．つまり，拡張質問を Q' とすると式 (3) により類似度を計算し，最も類似度が高い拡張質問に対応する FAQ 回答を選択する．

$$\mathrm{sim}(q, Q') = E_q(q)^T E_Q(Q') \tag{3}$$

Reranker を用いる場合，Retriever の検索結果の上位 K 件から “[CLS]（ユーザ質問 q）[SEP]（拡張質問 Q'）[SEP]”という単語列を K 件作成し，それぞれ Reranker に与えてリランキングスコアを算出する．Reranker は BERT と全結合層 1 層からなり，ユーザ質問 q と拡張質問 Q' が正例のペアかどうかの二値分類を行い，softmax 関数の正例に対する出力値をリランキングスコアとする．

訓練データは Retriever と同様の訓練データを用いて正例を作成し，さらに FAQDB から答え以外の拡

張質問をランダムサンプリングすることで各事例に対して負例を 2 件作成する．

3 提案手法

不明瞭なことが多いユーザ質問に対して質問を明確化するようなテキスト（以降，関連知識）を付与することができれば検索精度が向上する可能性がある．そこで，我々は言語生成モデルを用いた関連知識の生成を提案する．生成されたテキストはユーザ質問に付与して Retriever もしくは Reranker に入力し，学習と推論に用いる．

3.1 Prompt-Tuning

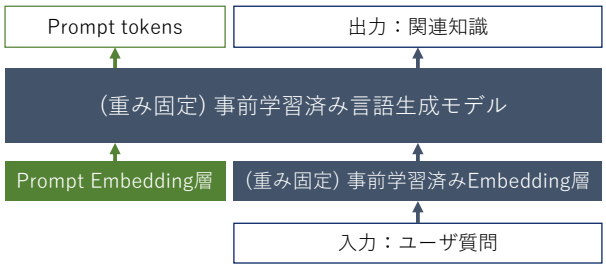

図 2 Prompt-Tuning の概要図

Prompt-Tuning では，入力単語列の前に固定長の単語列を付与し，そのドメインにおける関連知識を埋め込むこととする．学習時，言語生成モデルの重みは固定され，付与される単語列の埋め込みベクトルは学習によって最適化される．Prompt-Tuning に関する概要図を図 2 に示す．

Prompt-Tuning の学習方法として 2 通りを実験する．まず，FAQ 検索に関する従来研究 [11] では拡張質問と FAQ 質問の対応関係をモデルに学習させることで FAQ 検索システムを構築していることから，これを参考に拡張質問から FAQ 質問を生成するように言語生成モデルを学習させる（*GLM w/FAQDB*）．ただし，拡張質問と FAQ 質問は事前に定義されるテキストである一方で，ユーザ質問はチャットボット内でユーザが記述したテキストであるため，それらの間には表記や質問内容の粒度の違いが存在し，言語生成モデルがユーザ質問から関連知識を上手く生成できない可能性がある．そこで 2 つ目に，訓練データのユーザ質問からそれに対する正例の FAQ 質問を生成するように学習させる（*GLM w/TRAIN*）．

言語生成モデルとしては，公開されている日本語版の GPT-2[1] と GPT(1b)[2] を用いる．

1） https://huggingface.co/rinna/japanese-gpt2-medium
2） https://huggingface.co/rinna/japanese-gpt-1b

表 1 ドメイン別のデータセットの事例数

データ＼ドメイン	A	B	C
訓練	4,019	2,123	641
検証	502	265	80
評価	503	266	81

生成されたテキストはユーザ質問に付与する．Retriever の場合，ユーザ質問に対する EncoderE_q の入力単語列は，“[CLS]（ユーザ質問 q）[SEP]（生成されたテキスト）[SEP]”となる．一方で Reranker の場合， 入力単語列は “[CLS]（ユーザ質問 q）[SEP]（拡張質問 Q'）[SEP]（生成されたテキスト）[SEP]” となる．

実験では，同様のデータで GPT-2 を Fine-Tuning した場合 [13] と比較する．

4 実験

チャットボット[3]で収集したデータセットを用いて提案手法の有効性を検証する．データセットは 3 つのドメインからなり，ドメイン A,B,C と表記する．これらは互いに異なる事業で導入されたチャットボットから収集されたデータである．ドメインごとのデータセットの作成方法は従来研究 [14] に倣い，事例数は表 1 に示す．

4.1 実験設定

Retriever を構成する 2 つの Encoder と Reranker の重みの初期値には，公開されている日本語事前学習済み BERT[4] の重みを用いる．学習時のパラメタは付録 A に記載する．以降，Fine-Tuning された GPT-2，Prompt-Tuning された GPT-2，Prompt-Tuning された GPT(1b) をそれぞれ ftGPT-2, ptGPT-2, ptGPT(1b) と表記する．また，ベースラインとして利用する BM25 [15] では，FAQDB の拡張質問の集合から IDF 値をドメインごとに計算し，名詞と動詞原型のみ用いて算出する．Reranker の推論時は，言語生成モデルを用いずに Retriever による検索を行い，その検索結果の上位 K 件に対して Reranker によるリランキングを行う．評価指標には，検索結果の上位 1 件の精度である Top1 Accuracy を用いる．

4.2 実験結果

実験結果を表 2 に示す．Retriever と Reranker を比較すると，ドメイン A とドメイン B においては

3） https://www.ai-messenger.jp
4） https://huggingface.co/cl-tohoku/bert-base-japanese

表 2 実験結果 （Top1 Accuracy）

モデル＼ドメイン	A	B	C
BM25	28.6	33.8	**37.0**
Retriever	39.6	38.0	32.1
Reranker	**42.9**	43.2	30.9
GLM w/FAQDB			
Retriever w/ ftGPT-2	40.0	39.1	23.5
Retriever w/ ptGPT-2	38.0	40.2	28.4
Retriever w/ ptGPT(1b)	39.8	38.3	32.1
Reranker w/ ftGPT-2	37.4	39.5	27.2
Reranker w/ ptGPT-2	39.2	43.2	30.9
Reranker w/ ptGPT(1b)	39.8	**45.5**	27.2
GLM w/TRAIN			
Retriever w/ ftGPT-2	40.8	38.7	24.7
Retriever w/ ptGPT-2	38.2	38.3	30.9
Retriever w/ ptGPT(1b)	40.4	39.1	28.4
Reranker w/ ftGPT-2	40.0	41.4	25.9
Reranker w/ ptGPT-2	38.0	41.4	28.4
Reranker w/ ptGPT(1b)	36.0	42.1	30.9

Reranker の検索精度が高い値となった．これはユーザ質問と拡張質問を連結して Reranker で単語間の Attention 計算を行っており，この点がリランキングに効果的であったと考えられる．

言語生成モデルで生成されたテキストを付与したことに対する実験結果によると，ドメイン B では Reranker w/ptGPT(1b) が最も高いことから，Prompt-Tuning によって学習させた言語生成モデルが効果的な関連知識を生成することができたと考えられる．一方で，ドメイン A とドメイン C ではそれぞれ Reranker,BM25 が最も高い結果となった．したがって，言語生成モデルによる質問の明確化は全てのドメインにおいて有効であるとは限らないといえる．

次に言語生成モデルの学習手法 2 つそれぞれについて述べる．まず *GLM w/FAQDB* の場合，ドメイン B で ftGPT-2, ptGPT-2, ptGPT(1b) 全てにおいて検索精度が向上した一方で，Reranker の検索精度と比較するといずれも低い．したがって，今回の手法では質問の明確化よりも，単純に Reranker を利用することの方が効果的であるといえる．ただし，ドメイン B において Reranker は ptGPT(1b) のみ検索精度が向上した．これより，モデルサイズが生成される関連知識の質に影響する可能性があることがわかる．次に，*GLM w/TRAIN* の場合，ドメイン B で Retriever では ftGPT-2, ptGPT-2, ptGPT(1b) の全てで検索精度が向上した一方で，Reranker では全て検索精度が低下した．これは Reranker と言語生成モデルの訓練データが同じであり，生成した関連知識が Reranker にとって不要であったと考えられる．

4.3 今後の展望

実験結果によると，ドメイン C では BM25 が Retriever と Reranker を上回った．FAQ 検索ではドメイン次第で効果的な手法が異なることが経験的にわかっているため，BM25 のような表層形を用いた検索と Retriever や Reranker のような機械学習モデルを用いた検索を上手く組み合わせることが重要である．さらに，今回扱った FAQ 検索特有の問題である質問の不明瞭さへの対処として質問の明確化を行うことで頑健な手法を検討したい．

また，実験結果から ftGPT-2 と ptGPT-2 による学習手法の比較と，ptGPT-2 と ptGPT(1b) によるモデルサイズの比較を行った場合でも一貫した傾向は確認されなかった．モデルサイズと関連知識の質に関して今後の調査を検討中である．

生成されたテキストを確認すると，いずれのモデルの場合においても入力単語列がそのまま出力される場合や意味の通らない単語列が生成される場合が確認された．言語生成モデルによる安定した関連知識の生成が今後の課題である．

5 おわりに

本研究では，FAQ 検索におけるユーザ質問の不明瞭さへの対処として，言語生成モデルで質問の明確化を行い，特定のドメインにおいては検索精度が向上することを示した．今回実験したドメインにおいては Fine-Tuning が困難である大規模な言語生成モデルを Prompt-Tuning することで関連知識の生成が可能であることがわかった．しかし，実験中の全てのドメインにおける一貫した検索精度の向上は確認されなかった．今後は安定した関連知識の生成を検討しつつ，より頑健な FAQ 検索手法の構築を目指す．

謝辞

本論文の作成にあたりご協力いただきました，株式会社 AI Shift の杉山雅和氏，戸田隆道氏，東佑樹氏，下山翔氏にこの場を借りて厚く御礼申し上げます．

参考文献

[1] Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. Faq retrieval using query-question similarity and bert-based query-answer relevance. In **Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval**, pp. 1113–1116, 2019.

[2] Yosi Mass, Boaz Carmeli, Haggai Roitman, and David Konopnicki. Unsupervised FAQ retrieval with question generation and BERT. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 807–812, Online, July 2020. Association for Computational Linguistics.

[3] 鈴木正敏, 鈴木潤, 松田耕史, 西田京介, 井之上直也. Jaqket: クイズを題材にした日本語 qa データセットの構築. 言語処理学会第 26 回年次大会, pp. 237–240, 2020.

[4] 加藤拓真, 宮脇峻平, 西田京介, 鈴木潤. オープンドメイン qa における dpr の有効性検証. 言語処理学会第 26 回年次大会, pp. 237–240, 2020.

[5] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wentau Yih. Dense passage retrieval for open-domain question answering. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics.

[6] Guanghui Qin and Jason Eisner. Learning how to ask: Querying LMs with mixtures of soft prompts. In **Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 5203–5212, Online, June 2021. Association for Computational Linguistics.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. **Advances in neural information processing systems**, Vol. 33, pp. 1877–1901, 2020.

[8] Fabio Petroni, Tim Rocktäschel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases? **CoRR**, Vol. abs/1909.01066, , 2019.

[9] Joshua Feldman, Joe Davison, and Alexander M. Rush. Commonsense knowledge mining from pretrained models. **CoRR**, Vol. abs/1909.00505, , 2019.

[10] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[11] Haytham Assem, Sourav Dutta, and Edward Burgin. DTAFA: Decoupled training architecture for efficient FAQ retrieval. In **Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue**, pp. 423–430, Singapore and Online, July 2021. Association for Computational Linguistics.

[12] 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.

[13] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. **OpenAI blog**, Vol. 1, No. 8, p. 9, 2019.

[14] 二宮大空, 邊土名朝飛, 杉山雅和, 戸田隆道, 友松祐太. チャットボット事業における dense retriever を用いた zero-shot faq 検索. 第 96 回言語・音声理解と対話処理研究会 (第 13 回対話システムシンポジウム), 2022.

[15] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. **Foundations and Trends® in Information Retrieval**, Vol. 3, No. 4, pp. 333–389, 2009.

A　学習時のパラメタ

Retriever と Reranker の Fine-Tuning 時の学習率は 1×10^{-5}（warmup rate=10%）とし，Optimizer は Adam を用いる．Retriever の Fine-Tuning 時の Epoch は 10，バッチサイズは 64 とする．Reranker の Fine-Tuning 時の Epoch は 2，バッチサイズは 32 とし，Cross Entropy Loss を最小化するように学習させる．Retriever と Reranker の入力最大長はそれぞれ 50 トークン，100 トークンであり，超過する単語列は削除される．

言語生成モデルである GPT-2 と GPT(1b) は，Prompt-Tuning と Fine-Tuning 両方において学習率は 1×10^{-5}，バッチサイズは 8，Dropout 率は 0.1，Optimizer は Adam を用いる．学習時の Epoch はドメインによって異なり，ドメイン {A,B,C} はそれぞれ *GLM w/FAQDB* の場合 {30，30，100} とし，*GLM w/TRAIN* の場合 {3，5，10} とする．　Prompt-Tuning に関しては Prompt に対する埋め込みベクトルのみ更新され，事前学習済みの言語生成モデルの重みは固定する．Prompt-Tuning 時の付与される単語列は 20 トークンとする．実装には huggingface を用いており，Prompt に対する正解ラベルは-100 を設定することで関連知識と無関係な損失が計算されないように設定している．