適応型負例選択を用いた対照学習による回答検索

伊東 秀夫
株式会社リコー デジタル戦略部 デジタル技術開発センター 言語 AI 開発室
hideo.itoh@jp.ricoh.com

概要

質問応答における RAG では，質問文と回答が必ずしも類似関係にないため，回答を含む文脈（チャンク）が検索できない場合がある．我々はこの問題を，質問文から回答をピンポイントに検索する機能（回答検索）を用いて従来の検索結果をリランキングすることで解決する．具体的には生成言語モデルによる質問文のエンコード結果から得られる質問ベクトルを検索キーとし，同じく検索対象のエンコード結果から得られるベクトル列に対し最大内積検索（MIPS）を行う．これらベクトルは次トークン予測用に最適化されているため，MIPS 用に変換する必要がある．このベクトル変換の対照学習を効果的にするため，適応型負例選択と呼ぶ提案手法を用いる．実験では，回答検索によるリランキングにより一貫して検索精度が向上した．

1 はじめに

質問応答において外部知識に基づいて言語モデルに回答させる枠組みとして RAG [1] がある．RAG における課題の１つとして，質問文と回答が必ずしも類似関係にないため，回答に必要な知識を含む文脈（チャンク等）が，質問文との類似検索では得られない場合があることが指摘され，この課題に対していくつかの解決方法が提案されている [2][3][4].

我々は上述の課題に対し，質問文から回答をピンポイントに検索する機能（回答検索と呼ぶ）を加えることで，質問文を検索条件とする従来の検索手法を補うアプローチを提案する．

具体的には図 1 に示すように，質問応答を担う生成言語モデル（LM）を用いて質問文をエンコードして得られる最上位層の出力ベクトル群の内，質問文の末尾トークンに対応する出力ベクトルを検索キーとする（質問ベクトルと呼ぶ）．

検索対象は，外部知識に相当するテキスト群を LM でエンコードして得られる最上位層の出力ベクトル群であり，質問ベクトルとの最大内積検索（MIPS）によりトークン単位の検索を行う．

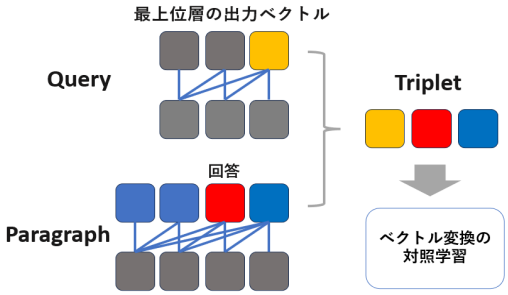

図 1 回答検索の概要図

ただし，これら質問ベクトルおよび検索対象ベクトル群は，次トークン予測 (NTP）のために最適化されたものであるから，検索前に MIPS 用のベクトルに変換する必要がある．このベクトル変換は，図中に示した３つ組（トリプレット）を用いた対照学習により得るが，この学習を効果的に行うために，適応型負例選択と呼ぶ手法を提案する．

回答検索の結果を RAG におけるリランキングに用いることで，質問文全体をキーとした検索と相補的に作用し，検索精度が向上することを実験により示す．

2 関連研究

質問文と回答が必ずしも類似関係にないことへの対策として以下の提案がある．

検索専用のエンコーダーを用いる手法 質問文と外部知識のエンコーダーを分けた上で，両者の埋め込みが近くなるように学習するデュアルエンコーダー手法 [5] や，単一のエンコーダーにおいて特殊トークンにより質問文用と外部知識用の埋め込みを切り分ける手法（タスクタイプ別埋め込み）[4] が提案されている．

これらは質問応答用の LM とは別に，検索（RAG）専用のエンコーダーが必要となり，それらの事前学習やドメイン適応のコストが運用上の問題となる．

回答生成を行う手法　質問応答用の LM に回答候補を生成させて，質問文とともに検索条件として用いる手法（HyDE）[3] も提案されている．

本提案と同じく，検索用と質問応答のための LM を兼用することが可能だが，自己回帰予測による回答生成のために応答が遅延する問題がある．

3　提案手法

提案手法ではテキストを Transformer 型の生成言語モデル（causal language model）でエンコードして得られる最上位層の出力ベクトルが中心となるので，これらを “状態ベクトル” と略称する．

3.1　回答検索

回答検索の検索キーは，図 1 を用いて概説したように，質問文の末尾トークンに対応する状態ベクトル（質問ベクトル）である．回答生成において質問ベクトルは次トークン，つまり回答の開始トークン予測のための唯一の情報源であることから，回答の検索キーとして妥当と考える．

回答検索の検索対象は，外部知識を表すテキスト（トークン列）から得られる状態ベクトル列である．各検索対象ベクトルにはその前方文脈とともに，回答パートか否かの手がかりになる情報がエンコードされていると考える．

最大内積検索（MIPS）により，検索対象ベクトル列中から，質問ベクトルとの内積値順に上位 N 件のトークン位置を取得し，それらが回答または回答を含むチャンク（パッセージ等）にヒットする割合として回答検索の性能を定める．

3.2　ベクトル変換の対照学習

質問ベクトルおよび検索対象ベクトル群を MIPS 用のベクトルに変換するため，変換行列 $\boldsymbol{M} \in R^{O \times I}$ を用いて式 1 に示す線形変換を行う．ここで I, O は変換前後の次元数であり $\boldsymbol{h}, \boldsymbol{v}$ は変換前後のベクトルである．

$$\boldsymbol{v} = \boldsymbol{M}\boldsymbol{h} \tag{1}$$

入力次元数 I は状態ベクトルの次元数となるが，出力次元数 O を小さくすることで，検索精度を損なわない程度に学習・推論を効率化できる．

実装としてはバイアス項なしの線形層と，正則化のためのドロップアウト層を線形層の前に設ける．

変換行列のパラメタ（行列要素）を訓練データから収集した トリプレット $(\boldsymbol{q}, \boldsymbol{p}, \boldsymbol{n})$ の集合を用いて対照学習する．ここで $\boldsymbol{q}, \boldsymbol{p}, \boldsymbol{n}$ は以下のトークンに対応する状態ベクトルである．

- $\boldsymbol{q}$　質問文の末尾トークンのベクトル
- $\boldsymbol{p}$　検索対象中の回答開始トークンのベクトル
- $\boldsymbol{n}$　検索対象中の回答外のトークンのベクトル

対照学習ではトリプレット集合をランダムにバッチに分割し，バッチ内の $\boldsymbol{q}, \boldsymbol{p}, \boldsymbol{n}$ に対するベクトル変換後，InfoNCE [6] として知られる式 2 に示す損失関数 L を用いて学習する．

$$L = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp\left(\boldsymbol{Q}_i \cdot \boldsymbol{P}_i\right)}{\exp\left(\boldsymbol{Q}_i \cdot \boldsymbol{P}_i\right) + \sum_{k=1}^{B} \exp\left(\boldsymbol{Q}_i \cdot \boldsymbol{N}_k\right)} \tag{2}$$

ここで B はバッチ内のトリプレット数，$\boldsymbol{Q}_i, \boldsymbol{P}_i, \boldsymbol{N}_k$ はベクトル変換後の $\boldsymbol{q}_i, \boldsymbol{p}_i, \boldsymbol{n}_k$ を表す．なお，式 2 の計算における桁溢れを防ぐため exp 内の内積値は出力ベクトルの次元数 O で除算しスケーリングする．

3.3　適応型負例選択

トリプレット内に限らずバッチ内の負例を用いる手法はバッチ内負例（IBNeg）と呼ばれ [7]，負例の選択方法として以下がある [8][9].

- 式 2 のようにバッチ内の全負例を選択（all）
- $\boldsymbol{Q}_i \cdot \boldsymbol{N}_k > \boldsymbol{Q}_i \cdot \boldsymbol{P}_i$ となる負例のみ（semi-hard）
- $\boldsymbol{Q}_i \cdot \boldsymbol{N}_k$ が最大となる負例のみ（max-hard）

しかしこれら従来法には以下の問題点がある．

- 負例選択の範囲をランダムなバッチ分割に任せるのは非効率
- 学習が進むにつれ困難な負例（semi-hard）が得られ難くなる

これらの問題点を解消するために，適応型負例選択と呼ぶ以下の手法を提案する．

- 従来の負例選択方法でバッチ学習する
- 学習後にトリプレットの負例 $\boldsymbol{n}_i$ をバッチ内の max-hard 負例 $\boldsymbol{n}_k (k = \underset{j}{\operatorname{argmax}}\, \boldsymbol{Q}_i \cdot \boldsymbol{N}_j)$ に置換

これにより，エポックが進むにつれて，より困難な負例がトリプレット毎に着実に収集されてゆく．

3.4　リランキング

回答検索の検索単位はトークンであるため，従来の文書やチャンクを単位とする検索（文書検索と総称する）と比べ，スケーラビリティ（検索対象の増大に対する耐性）に乏しい．一方，文書検索の結果に対し回答を含む文書の検索順位を上げるリランキ

ングに応用することで，質問内容の検索を回答内容の検索で補う作用が期待できる．

リランキングは以下の手順で行う．

1. 質問文に対して質問ベクトル $\boldsymbol{q}$ のベクトル変換結果 $\boldsymbol{Q} = \boldsymbol{M}\boldsymbol{q}$ を求める．（$\boldsymbol{M}$ は式 1 の変換行列）
2. 文書検索結果の上位 N 件の文書 D_i $(i = 1..N)$ 毎にトークン列に対応する状態ベクトル列 $\boldsymbol{H}_i$ を求める．ここで $\boldsymbol{H}_i$ は各トークンの状態ベクトルを列とする行列を表す．
3. ベクトル変換 $\boldsymbol{V}_i = \boldsymbol{M}\boldsymbol{H}_i$ を行い，$\boldsymbol{Q}$ と $\boldsymbol{V}_i$ から最大内積値 $R_i = max(\boldsymbol{Q}^T\boldsymbol{V}_i)$ を得る．
4. D_i $(i = 1..N)$ を式 3 で求めるスコア RS_i の降順にリランキングする．ここで S_i は D_i の文書検索スコア，定数 α は調整パラメタである．

$$RS_i = \alpha R_i + S_i \tag{3}$$

現状，リランキングにはクロスエンコーダ [10] を用いるのが標準的であるが，本提案手法では文書毎のベクトル変換済み状態ベクトル列を MIPS 用データベースに予め登録しておくことで，より高速なリランキングを実現できる．

4 実験

4.1 使用データ

表 1 に示す機械読解用テストデータを用いる．いずれも質問対象となるパラグラフ毎に質問文と回答が記述されている．回答毎にパラグラフ内の開始終了位置（文字単位）が付されており，パラグラフのトークン分割結果と照合することで，回答パートに相当するトークン列を取得できる．

表 1 使用データ

データ名	種別	質問	パラグラフ
SQuAD [11]	train	86821	19035
	dev	5928	1204
JSQuAD [12]	train	62859	15652
	valid	4442	1145
DDQA [13]	train	17905	8968
	test	1045	1042

4.2 使用言語モデル

パラメタ数 3B 〜 4B で入力幅 4k トークン以上 の英語・日本語対応モデルとして表 2 に示す生成言語モデルを選択した．本論文内での略称と共に示す．

表 2 使用言語モデル

略称	モデル名
llama3.2	llama3.2-3B-Instruct
phi3.5	phi-3.5-mini-instruct
qwen2.5	Qwen2.5-3B-Instruct
nemotron	Nemotron-Mini-4B-Instruct
llm-jp	llm-jp-3-3.7b-instruct

4.3 適応型負例選択と回答検索の実験

使用言語モデル毎に，表 1 に示した各使用データの train データからトリプレット $(\boldsymbol{q}, \boldsymbol{p}, \boldsymbol{n})$ を抽出し，ベクトル変換行列を対照学習する（節 3.2 参照）．

トリプレット抽出では質問毎に最大 5 つの負例を用いた（つまりトリプレット総数は質問数の約 5 倍）．ベクトル変換の出力次元数は 512，対照学習のバッチサイズは 32，エポック数は 120 とした．

対照学習におけるバッチ内負例の選択法として以下を設定する．

1. バッチ内の全負例を使用
2. semi-hard 負例を使用
3. max-hard 負例を使用
4. 上記 1 かつ適応型負例選択を適用
5. 上記 2 かつ適合型負例選択を適用
6. 上記 3 かつ適合型負例選択を適用

上記 1 〜 6 の対照学習によるベクトル変換を用いて，以下に示す回答検索の性能を比較する．

- 表 1 の dev, valid, test をテストデータとする．
- テストデータの全パラグラフに対し生成言語モデルで求めた状態ベクトル列をベクトル変換して検索対象とする．
- テストデータの質問文に対し生成言語モデルで求めた質問ベクトルをベクトル変換し，検索対象の全ベクトル列との内積を計算する．
- パラグラフ毎に最大の内積値を検索スコアとし，検索スコア順に上位 N 件のパラグラフ群を検索結果とする．
- 質問の回答を含むパラグラフ（すなわち機械読解において質問対象とされたパラグラフ 1 件）を正解とし，検索性能を評価する

表 3 に言語モデル llama3.2 および llm-jp を用いた SQuAD データに対する回答検索の性能を示す．表中で N 列は上述のバッチ内負例選択法 1 〜 6，MAP は平均適合率，R@N は再現率（Recall@N）を表す．

表 3 SQuAD における回答検索性能

	llama3.2			llm-jp		
N	MAP	R@1	R@10	MAP	R@1	R@10
1	13.0	5.6	28.5	35.9	50.4	63.2
2	27.7	16.7	50.8	47.5	61.9	73.0
3	14.2	6.8	29.5	36.8	50.7	63.6
4	12.6	5.8	26.1	56.8	71.1	80.5
5	**45.8**	**34.3**	**69.3**	**58.6**	**72.1**	**81.4**
6	43.9	31.4	67.4	56.6	70.8	80.6

バッチ内負例選択法については，llama3.2 と llm-jp の両ケースにおいて，選択法 5 すなわちバッチ内の semi-hard 負例を用い，かつ適合型負例選択を適用した場合がベスト性能となった．適合型負例選択を行わない従来法 1〜3 との性能差は顕著である．

回答検索の性能は llm-jp の方が llama3.2 より高かった．この要因は llma3.2 の方が短いトークンに分割するため，トークン当たりの情報量が少なくなる点にあると考える．すなわちトークンを検索単位とする回答検索では，同程度のパラメタ数であればトークン長が長く語彙数が多い言語モデルの方が高性能になる傾向がある．

4.4 リランキングの実験

表 1 のテストデータ（dev, valid, test）のパラグラフ集合を検索対象として BM25 [14] の検索スコアを用いた検索結果の上位 10 件に対して節 3.4 で述べた回答検索によるリランキングを行った場合の効果を測定する．

BM25 による検索では，各言語モデルのトークン分割結果に沿って検索/索引タームの設定，および文書長の計数等を行った．

回答検索に用いるベクトル変換行列は，バッチ内負例選択法 5 で対照学習し，リランキングの調整パラメタ α は表 1 の train データを用いた試行実験を通して，言語モデル毎に下表 4 の通り定めた．パラメタ α の適正値は言語モデルの状態ベクトルのノルムのオーダーに依存すると考えられる．

表 4 リランキングの調整パラメタ α

言語モデル	α 値
llama3.2, phi3.5	0.10
qwen2.5, nemotron, llm-jp	0.05

表 5 にリランキング前後の MAP（平均適合率）と再現率（Recall@1）を示す．全てのデータと言語モデルの組み合わせにおいて，一貫してリランキング

表 5 リランキング前後の再現率の変化

		前		後	
データ	言語モデル	MAP	R@1	MAP	R@1
SQuAD	llama3.2	81.9	74.7	85.2	78.9
	phi3.5	82.1	74.8	85.5	79.3
	nemotron	81.2	74.0	84.7	77.8
	qwen2.5	81.5	74.4	85.0	78.4
	llm-jp	82.4	75.4	86.3	80.5
JSQuAD	llama3.2	90.8	86.9	91.1	87.4
	phi3.5	87.7	82.6	89.0	84.3
	nemotron	89.6	85.6	90.6	86.8
	qwen2.5	90.3	86.1	91.3	87.5
	llm-jp	89.1	84.8	90.8	87.0
DDQA	llama3.2	73.0	65.6	73.4	65.8
	phi3.5	62.1	53.8	63.2	54.8
	nemotron	70.8	63.3	74.5	67.2
	qwen2.5	74.6	67.7	76.4	69.8
	llm-jp	67.3	60.4	71.3	65.4

により MAP と Recall@1 が向上した．

前掲の表 3 に示した回答検索単独での検索性能（MAP, R@1）は，BM25 による文書検索の性能に比べ低いが，リランキングに用いることで，文書検索と相補的に働くことがわかる．この効果は llama3.2 のように回答検索の性能が低い場合でも見られた．

5 おわりに

生成言語モデルを用いた回答検索，および，そのためのベクトル変換の対照学習を効果的にする適応型負例選択を提案した．

回答検索の実験から，適応型負例選択としては semi-hard 負例を用いたバッチ内負例選択と組み合わせた場合，およびトークン長が長く語彙数が多い言語モデルを用いた場合に検索性能が高いことがわかった．リランキングへの応用実験からは，質問内容の検索に回答内容の検索を加えることで，質問文と回答は必ずしも類似関係にない，という RAG の課題を解決する見込みを得た．

なお，適応型負例選択は対照学習一般に適用できる技術であるため，トリプレットデータを用いた埋め込み学習などへの応用も期待できる．今後の研究課題としたい．

参考文献

[1] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wentau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. **CoRR**, Vol. abs/2005.11401, , 2020.

[2] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

[3] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels, 2022.

[4] Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernandez Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftekhar Naim. Gecko: Versatile text embeddings distilled from large language models, 2024.

[5] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 6086–6096, Florence, Italy, July 2019. Association for Computational Linguistics.

[6] Evgenia Rusak, Patrik Reizinger, Attila Juhos, Oliver Bringmann, Roland S. Zimmermann, and Wieland Brendel. Infonce: Identifying the gap between theory and practice, 2024.

[7] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wentau Yih. Dense passage retrieval for open-domain question answering. **CoRR**, Vol. abs/2004.04906, , 2020.

[8] Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. Learning dense representations for entity retrieval. In Mohit Bansal and Aline Villavicencio, editors, **Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)**, pp. 528–537, Hong Kong, China, November 2019. Association for Computational Linguistics.

[9] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. **CoRR**, Vol. abs/1703.07737, , 2017.

[10] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. **CoRR**, Vol. abs/1908.10084, , 2019.

[11] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. **CoRR**, Vol. abs/1606.05250, , 2016.

[12] Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. JGLUE: Japanese general language understanding evaluation. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, editors, **Proceedings of the Thirteenth Language Resources and Evaluation Conference**, pp. 2957–2966, Marseille, France, June 2022. European Language Resources Association.

[13] Norio Takahashi, Tomohide Shibata, Daisuke Kawahara, and Sadao Kurohashi. Machine comprehension improves domain-specific Japanese predicate-argument structure analysis. In Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen, editors, **Proceedings of the 2nd Workshop on Machine Reading for Question Answering**, pp. 98–104, Hong Kong, China, November 2019. Association for Computational Linguistics.

[14] Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. **Found. Trends Inf. Retr.**, Vol. 3, pp. 333–389, 2009.

A　使用データの詳細

いずれも SQuAD 2.0 形式の json ファイルで提供されている．実験では回答不能フラグが付されている質問データは除外した．パラグラフ長（平均トークン数）は llm-jp のトークナイザを用いて計測した値である．

データ名	言語	種別	質問数	パラグラフ数	文書数	パラグラフ長
SQuAD (v2.0) [11]	英語	train	86821	19035	442	154
		dev	5928	1204	35	165
JSQuAD (v1.1) [12]	日本語	train	62859	15652	710	101
		valid	4442	1145	59	97
DDQA (RC-QA v1.0) [13]	日本語	train	17905	8968	1	72
		test	1045	1042	1	74

A.1　データ例（SQuAD データ より）

パラグラフ：Even before the Norman Conquest of England, the Normans had come into contact with Wales. Edward the Confessor had set up the aforementioned Ralph as earl of Hereford and charged him with defending the Marches and warring with the Welsh. In these original ventures, the Normans failed to make any headway into Wales.

質問：Who was Ralph in charge of being at war with?

回答：the Welsh

B　使用言語モデルの詳細

略称	hidden_size	vocab_size	URL
llama3.2	3072	128256	https://huggingface.co/meta-llama/Llama-3.2-3B
phi3.5	3072	32064	https://huggingface.co/microsoft/Phi-3.5-mini-instruct
qwen2.5	2048	151936	https://huggingface.co/Qwen/Qwen2.5-3B-Instruct
nemotron	3072	256000	https://huggingface.co/nvidia/Nemotron-Mini-4B-Instruct
llm-jp	3072	99584	https://huggingface.co/llm-jp/llm-jp-3-3.7b-instruct

C　BM25 による文書検索

以下の検索スコアを用いて質問 Q に対し文書（パラグラフ）D をランキング検索した．検索語，索引語は各言語モデルが定めるトークンとし，調整パラメタ k_1, b の値は一律に各々 2.0, 0.75 とした．

$$score(D,Q) = \sum_{q \in Q} \log\left(1 + \frac{N}{df(q)}\right) \cdot \frac{f(q,D) \cdot (k_1 + 1)}{f(q,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)}$$

where

q	質問中の検索語
$df(q)$	文書頻度
N	総文書数
$f(q,D)$	文書内頻度
$\|D\|$	文書長
$avgdl$	平均文書長