埋め込みモデルのドメイン適応における ラベルノイズに強い合成 QA の作成

成田 大起 [1]　蕭 喬仁 [1]　井本 稔也 [1]
[1] *Japan Digital Design* 株式会社
{taiki.narita,kyojin.syo,toshiya.imoto}@japan-d2.com

概要

ドメイン特化 RAG の構築では、合成 QA を用いた埋め込みモデルのファインチューニングが標準的である．しかし，企業の手続文書は類似した内容が複数箇所に記載されるため，QA 生成に用いたチャンクのみを単一の正解とする従来の手法ではラベルノイズが生じやすい．そこで，本研究では，複数箇所に正解が存在する合成 QA をフィルタリングする手法を提案する．具体的には，既存の埋め込みモデルを用いた検索結果に基づき，LLM を用いて他の正解箇所が存在するかを判定する．実務文書を用いた実験により，合成データを用いた評価において提案手法の有効性が示された．

1　はじめに

企業内に蓄積された手続文書には業務遂行に必須な内容が含まれているが，その膨大な量の中から必要な情報を迅速に取得することは，特に業務経験の浅い社員にとっては困難である．加えて，条件や状況に応じた似て非なる手順が規定されている場合も多く，正確な情報収集は負荷の高い業務となる．

そこで，RAG（Retrieval-Augmented Generation）[1, 2] を用いて，大規模言語モデル（LLM; Large Language Models）に外部知識として社内の手続文書を与えることで，社員からの質問に適切に応答し正確な業務知識に基づいた回答を生成する手法が着目されている [3, 4, 5]．これにより，膨大な文書群から人手による情報選別を行う時間的・精神的負担を軽減し，業務効率化への寄与が期待されている．

RAG の性能は Retriever に大きく依存しており，汎用的な Retriever ではドメイン固有の知識を十分に扱えないことが指摘されている [6, 7]．一方で，ドメインに特化した学習データを人手で構築することは困難であるため，対象ドメインの文書から合成データを生成し埋め込みモデルをファインチューニングする手法が一般的となっている [8, 9, 10]．

実運用では応答速度が重要であるため，再ランキングモデルを含む多段の検索構成は必ずしも適さない．そのため本研究では，埋め込みモデルによるベクトル検索のみを利用する構成に焦点をあて，手続文書に特化した埋め込みモデルの作成を目指す．

手続文書では適用条件，例外や補足事項が分散して，繰り返し記載される構造が一般的であり，その結果，合成された単一のクエリに対して複数の正解箇所が存在する場合がある．多くの埋め込みモデルのファインチューニングでは，単一の正解文書を正例，それ以外を負例とする対照学習を前提としている [11, 12]．その際，合成 QA 作成時に得られた正解箇所のみを正例として扱うと，本来正例であるチャンクが負例として扱われ，ノイズのあるラベルが付与される恐れがある．

そこで本研究では，作成された合成 QA のうち複数のチャンクが正解箇所として存在する可能性がある QA ペアをフィルタリングしたうえで，既存の埋め込みモデルをファインチューニングする手法を提案する．実験には実際に企業で用いられている日本語の手続文書を用い，提案手法の有効性を評価した．本研究の貢献は次の通りである．

- 手続文書に特化した合成 QA を作成し，既存の埋め込みモデルをファインチューニングすることで手続文書に特化した埋め込みモデルを作成する手法を提案した．
- 単一のクエリに対して複数の正解箇所が存在する場合を排除するフィルタリング手法を提案した．
- 実際に企業で用いられている社内手続文書を用いた実験により，提案手法の有効性を示した．

2 関連研究

2.1 合成 QA の作成

ドメイン特有の文章に対する検索精度を向上させるため，LLM を用いて合成データを生成し，既存の埋め込みモデルをファインチューニングする手法 [10] が提案されている．しかし，これらの研究ではあるクエリに対して単一の正解箇所のみが存在する合成 QA の生成を前提しており，対象文書の複数箇所に正解が存在する場合を考慮していない．

2.2 合成 QA のフィルタリング

合成データの選別手法として，Mao ら [13] は Generator の自己評価に基づいたフィルタリングを提案している．また，Dai ら [6] は合成 QA を既存の埋め込みモデルを用いて検索し，検索結果上位 1 件に正解箇所が存在しない場合に QA を削除する手法を提案している．

複数箇所に似た文書がある場合，自己評価に基づく選別 [13] ではラベルノイズの除去が不十分となる懸念がある．また，既存モデルを用いた選別 [6] では，本来学習するべき難しい QA まで除外されてしまう．そこで本研究では，Generator に依存せず検索結果における正解候補の重複に基づいて QA を選別することで，質の高い学習用クエリのみをファインチューニングで利用する手法を提案する．

3 手法

3.1 合成 QA の作成

埋め込みモデルのファインチューニングに使用するため，クエリ，正解，正解箇所のペアを LLM を用いて作成する．検索対象となる文書をページごとに分割した集合を $\mathcal{D} = \{d_1, d_2, \ldots, d_M\}$ とする．各ページ d_i はあらかじめ一定の長さで分割されたチャンクの集合 $\{s_{(i,1)}, s_{(i,2)}, \ldots, s_{(i,N_i)}\}$ で表される．ただし，N_i はページ d_i のチャンク数を表す．このとき，チャンク全体の集合を $\mathcal{S} = \{s_{(i,j)} | 1 \le i \le M, 1 \le j \le N_i\}$ とする．

図 1 に示すように，合成 QA の作成では，第一に各ページ d_i に対して LLM を用いてクエリ，正解と正解箇所 $(q_{(i,k)}, a_{(i,k)}, \mathbf{c}_{(i,k)})$ を $k = 1, 2, \ldots, K_i$ について作成する．ここで，$q_{(i,k)}$ はクエリ，$a_{(i,k)}$ は正解，$\mathbf{c}_{(i,k)}$ は 1 つ以上の正解の引用箇所，K_i はページ d_i に対して作成する QA ペア数を表す．本手法では，チャンク単位ではなくページを単位として合成 QA の作成を行うことで，チャンク単位では断片的になりがちな文脈を保持し，より整合性の取れた QA ペア $(q_{(i,k)}, a_{i,k}, \mathcal{S}_{(i,k)})$ を作成する．

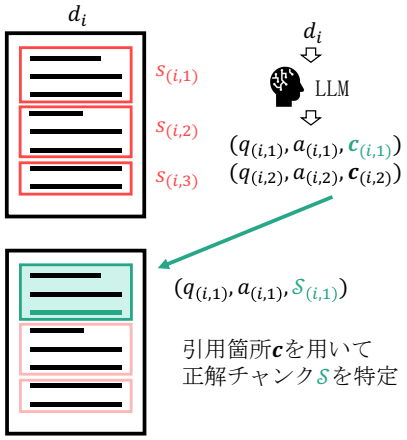

図 1 合成 QA の作成の流れ

引用箇所に基づいて正解チャンクを特定する．具体的には各引用箇所 $c_{(i,k,\ell)} \in \mathbf{c}_{(i,k)}$ に対して，ページ d_i 内の全チャンク $s_{(i,j)}$ とのレーベンシュタイン距離を計算し，

$$j^*_{(i,k,\ell)} = \arg\min_{1 \le j \le N_i} \mathrm{Lev}\left(c_{(i,k,\ell)}, s_{(i,j)}\right)$$

により正解チャンクを決定する．QA と正解箇所 $(q_{(i,k)}, a_{(i,k)}, \mathbf{c}_{(i,k)})$ に対する正解チャンク集合を

$$\mathcal{S}^*_{(i,k)} = \{s_{(i,j^*_{(i,k,\ell)})} \mid c_{(i,k,\ell)} \in \mathbf{c}_{(i,k)}\}$$

とし，$|\mathcal{S}^*_{(i,k)}| > 1$ の QA ペア $(q_{(i,k)}, a_{(i,k)}, \mathcal{S}^*_{(i,k)})$ は訓練，評価対象から除外する．

3.2 作成された合成 QA のフィルタリング

手続文書では，類似した事項が分散して記載される構造が一般的であり，異なる複数のチャンクの内容が類似する場合がある．そのため，あるクエリ q に対して複数のチャンク s が正解となり得るにもかかわらず，合成 QA 作成時に得られた単一の正解チャンクのみを正例として扱うと，本来正例であるチャンクが負例として扱われ，ノイズのあるラベルが付与される可能性がある．そこで，本研究では合成 QA 作成後に他の正解箇所があるかを LLM を用いて検証し，複数の正解チャンクが検出された QA ペアを除外する手法を提案する．

具体的には，既存の埋め込みモデルを用いたベクトル検索とキーワード検索を組み合わせた

表 1　類似したクエリの事例

クエリ	類似したクエリ	類似度
審査会議の主宰者は誰ですか？	審査会議の主宰者は誰ですか？	1.000
手数料比較シートにはどのようなデータが記載されていますか？	手数料比較シートに記載されている数値はどのような形式ですか？	0.980
外国債券の注文伝票を手書きで作成する場合，どのような手順が必要ですか？	外国債券の注文伝票を手書き作成する必要があるのはどのような場合ですか？	0.970
事後確認報告書の作成時に留意すべき事項について教えてください．	事後確認報告書はどのような場合に作成されるのですか？	0.960
会計用時価提供システムにおいて，扱者変更画面で新しい扱者を検索する方法を教えてください．	会計用時価提供システムを使用して送付先の扱者を変更する際，扱者の情報を検索する手順は何ですか？	0.951

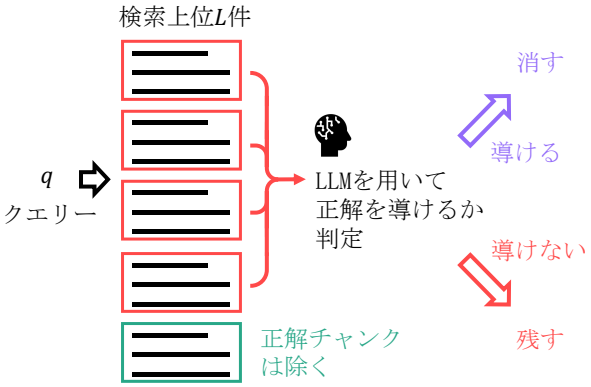

図 2　フィルタリング時の LLM を用いた判定

ハイブリッド検索により，チャンク集合 $\mathcal{S}$ からあるクエリ q に対する検索上位 L 個のチャンク $\mathcal{S}_{top_q} = \{s_{(i'_1,j'_1)}, s_{(i'_2,j'_2)}, \ldots, s_{(i'_L,j'_L)}\}$ を取得する．ここで，図 2 に示すように，合成 QA 生成時に特定された正解箇所 $\mathcal{S}^*_{(i,k)}$ に含まれるチャンクを除いた集合 $\mathcal{S}'_{top_q} = \mathcal{S}_{top_q} \setminus \mathcal{S}^*_{(i,k)}$ を作成する．次に $\mathcal{S}'_{top_q}$ 内の各チャンクがクエリ q に対する正解箇所であるかを LLM を用いて判定する．$\mathcal{S}'_{top_q}$ 内に正解箇所が存在すると判定された場合，その QA ペア $(q_{(i,k)}, a_{(i,k)}, \mathcal{S}^*_{(i,k)})$ を訓練，評価対象から除外する．

4　実験

4.1　データセット

実験では，実際に企業で用いられている社内手続文書を用いた．手続文書は PDF で提供されており，OCR 処理を行いテキスト化した上でページごとに分割した．さらに 3.1 節で述べたように，各ページの内容を 320 トークン単位でチャンク化した．社内文書全体のチャンク数は $|\mathcal{S}| = 79274$，1 チャンクあたりの平均文字数は約 395 文字であった．

合成 QA の作成には A.1 節，フィルタリングには A.2 節で示すプロンプトを用い，どちらも OpenAI の GPT-4o を用いた．ページあたりの作成する質問数 K は $\lceil N_i/3 \rceil$，すなわちページ内のチャンク数の 3 分の 1 とし，フィルタリング時に取得する検索結果 L は 5 とした．

4.2　類似クエリの削除によるリーク防止

合成 QA を用いた評価では，訓練データとテストデータの分割する際にリークが発生する恐れがある．手続文書では似ているチャンクが複数存在する構造のため，類似したクエリが異なるチャンクから生成される場合がある．

そこで，ruri-v3-310m[14] を用いて各クエリ q に対して埋め込みベクトルを計算し，コサイン類似度が閾値 0.97 以上であるクエリのペアを類似クエリとみなし，両方の QA ペアを削除した．閾値は，cos 類似度が 0.95 以上のクエリのペアを人手で確認し，0.97 未満ならば概ね同一質問とはみなされないことを確認したうえで設定した．表 1 に類似クエリのペアの事例を示す．結果として本実験では，21321 件の合成 QA が作成され，そのうち 13984 件が類似クエリの削除後に残った．

4.3　比較手法

提案手法の効果を評価するために以下の 4 種（No Filtering, Top 1, Top 5, Proposed）のフィルタリング手法を比較した．

- No Filtering: 訓練データに対して，フィルタリングを行わず類似クエリの削除のみを行う．
- Top k: Dai ら [6] の手法をもとに ruri-v3-310m でベクトル検索を行い，上位 k チャンクに正解チャンクが含まれない QA ペアを除外する（$k \in \{1, 5\}$）．
- Proposed: 提案手法．

表 2　フィルタリング前後の QA ペア数

フィルタリング手法	訓練データ	テストデータ
No Filtering	9788	4196
Top 1 [6]	3570	1531
Top 5	6542	2806
Proposed	5378	2304

表 3　合成 QA を用いた QA ペアに対する実験結果

手法	Recall@1	Recall@5	Recall@10
No Fine-tuning	0.424	0.674	0.759
No Filtering	0.485	0.734	0.813
Top 1 [6]	0.473	0.700	0.783
Top 5	0.450	0.720	0.796
Proposed	**0.506**	**0.751**	**0.816**

No Filtering の全ての QA ペアを 7:3 の割合で訓練，テスト用データに分割した後に，各フィルタリング手法を用いてフィルタリングを行った．最終的な QA ペアの数を表 2 に示す．なお，できるだけ正確な正解箇所が得られている合成 QA を用いて評価を行うため，最終評価では Proposed でフィルタリングされたテストデータのみを使用した．

4.4　実験設定

事前学習済みモデル ruri-v3-310m[14] をもとに合成 QA を用いたファインチューニングを行った．損失関数として Cached Multiple Negatives Ranking Loss[15, 16] を使用し，バッチサイズを 512，学習率を 9.0×10^{-5}，ステップ数を 100，アーリーストッピングを 10 とした．アーリーストッピングの評価には，訓練用の合成 QA のうち 20%を検証用データとした Recall@1 の平均値を用いた．なお，Recall@k は各クエリについて，上位 k 件の検索結果に正解チャンクが含まれる割合を表す．

最終的な評価は，LLM を用いてフィルタリングされた合成 QA の内テスト用とした 2304 ケースを用いて Recall@k（$k = 1, 5, 10$）を算出した．

5　結果と考察

5.1　合成 QA に対する評価

表 3 に合成 QA を用いた結果を示す．Proposed はすべての k において最も高い Recall を達成した．

手法ごとの差を評価するため，クエリ単位のブートストラップ（反復回数 $B = 10000$）により Recall@k の差分の 95%信頼区間を推定した．その

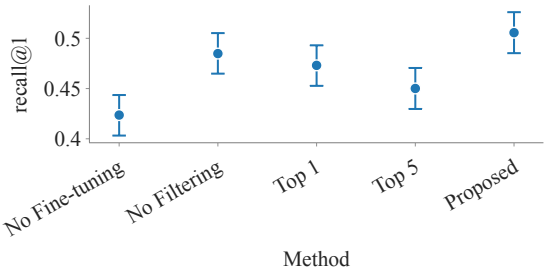

図 3　合成 QA における Recall@1 の差分分析

結果，Proposed は No Fine-tuning および Top 1/Top 5 に対して，Recall@1,5,10 で一貫して高い傾向が確認された．また，No Filtering と比較しても，Recall@1 および Recall@5 で改善が見られた．特に，Recall@1 の差分分析の結果を図 3 に示す．

表 3 に示すようにファインチューニングを行うことで全ての手法において Recall が向上している．このことから，先行研究と同じく手続文書に特化した埋め込みモデルが学習できていることが分かる．

また，Top 1 および Top 5 手法では，No Filtering 手法と比較して全ての k において一貫して Recall が低下した．これは，ファインチューニングしていないモデルで容易に解答できるクエリのみを用いてファインチューニングを行っても，検索性能の向上には十分ではないことを示唆している．

一方で，Proposed では，Top 1 および Top 5 と比較して全ての k において高い Recall を示した．提案手法では既存の埋め込みモデルでは容易に回答できないクエリも保持されており，より多様なクエリを用いてファインチューニングを行うことで，検索性能の向上に寄与していることを示唆している．

6　おわりに

本研究では合成 QA から複数の正解箇所を持つ QA ペアを削除することで，手続文書に特化した埋め込みモデルをファインチューニングする手法を提案した．実務で使用されている手続文書を用いた実験では，提案手法を用いることで他のフィルタリング手法と比較して最も高い Recall を達成し，手法の有効性を確認した．

今後は，複数箇所に正解が存在する場合に削除するのではなく，複数の正解箇所をすべて見つけたうえでファインチューニングを行う手法の検討をすすめる．また，正解箇所が単一ではなく複数のチャンクが必須となる合成 QA の作成手法についても検討を進めていきたい．

参考文献

[1] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. **Advances in neural information processing systems**, Vol. 33, pp. 9459–9474, 2020.

[2] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In **EMNLP (1)**, pp. 6769–6781, 2020.

[3] Jingyun Sun, Zhongze Luo, and Yang Li. A compliance checking framework based on retrieval augmented generation. In **Proceedings of the 31st International Conference on Computational Linguistics**, pp. 2603–2615, 2025.

[4] Dohyeon Lee, Jongyoon Kim, Jihyuk Kim, Seung-won Hwang, and Joonsuk Park. trag: Term-level retrieval-augmented generation for domain-adaptive retrieval. In **Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, pp. 6566–6578, 2025.

[5] Tianyang Zhang, Zhuoxuan Jiang, Shengguang Bai, Tianrui Zhang, Lin Lin, Yang Liu, and Jiawei Ren. RAG4ITOps: A supervised fine-tunable and comprehensive RAG framework for IT operations and maintenance. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina, editors, **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track**, pp. 738–754, Miami, Florida, US, November 2024. Association for Computational Linguistics.

[6] Zhuyun Dai, Vincent Y. Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. Promptagator: Few-shot dense retrieval from 8 examples. In **The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023**. OpenReview.net, 2023.

[7] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In Joaquin Vanschoren and Sai-Kit Yeung, editors, **Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual**, 2021.

[8] Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, **Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 2345–2360, Seattle, United States, July 2022. Association for Computational Linguistics.

[9] Luiz Henrique Bonifacio, Hugo Queiroz Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. Inpars: Unsupervised dataset generation for information retrieval. In Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai, editors, **SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022**, pp. 2387–2392. ACM, 2022.

[10] Haiyang Shen, Hang Yan, Zhongshi Xing, Mugeng Liu, Yue Li, Zhiyang Chen, Yuxiang Wang, Jiuzheng Wang, and Yun Ma. Ragsynth: Synthetic data for robust and faithful RAG component optimization. **CoRR**, Vol. abs/2505.10989, , 2025.

[11] Jaehee Kim, Yukyung Lee, and Pilsung Kang. A gradient accumulation method for dense retriever under memory constraint. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, **Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024**, 2024.

[12] João Coelho, Bruno Martins, João Magalhães, Jamie Callan, and Chenyan Xiong. Dwell in the beginning: How language models embed long documents for dense retrieval. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024 - Short Papers, Bangkok, Thailand, August 11-16, 2024**, pp. 370–377. Association for Computational Linguistics, 2024.

[13] Kelong Mao, Zheng Liu, Hongjin Qian, Fengran Mo, Chenlong Deng, and Zhicheng Dou. Rag-studio: Towards in-domain adaptation of retrieval augmented generation through self-alignment. In **Findings of the Association for Computational Linguistics: EMNLP 2024**, pp. 725–735, 2024.

[14] Hayato Tsukagoshi and Ryohei Sasano. Ruri: Japanese general text embeddings. **CoRR**, Vol. abs/2409.07737, , 2024.

[15] Matthew L. Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. Efficient natural language response suggestion for smart reply. **CoRR**, Vol. abs/1705.00652, , 2017.

[16] Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. Scaling deep contrastive learning batch size under memory limited setup. In Anna Rogers, Iacer Calixto, Ivan Vulic, Naomi Saphra, Nora Kassner, Oana-Maria Camburu, Trapit Bansal, and Vered Shwartz, editors, **Proceedings of the 6th Workshop on Representation Learning for NLP, RepL4NLP@ACL-IJCNLP 2021, Online, August 6, 2021**, pp. 316–321. Association for Computational Linguistics, 2021.

A 参考

A.1 合成 QA 作成のためのプロンプト

```
# タスク内容
- 手続き文書の抜粋が与えられるので、実際の社員が照会すると思われる質問と解答、解答の根拠となる引用箇所をそのまま抜粋したものの 3 つのペアを{num_pairs}件出力してください。
# 遵守してほしいルール
- 解答を生成する際にはあなたの事前知識を使わず、根拠となる文章の内容のみを参照してください。
- 質問、解答のペアを複数生成する際にはそれぞれのペアが独立して成立するようにしてください。質問に質問を重ねる場合は想定しておらず、それぞれの質問は別の人が行うことを想定しています。
- 解答は質問の内容に対して過不足なく答えてください
- 解答の根拠となる引用箇所を抜粋する際には、質問に関連する手続き文書の内容を一言一句も漏らさずに引用してください。
- 手続き文書にはただし書き（ただし や なお、注意といった言葉）の後に重要な補足情報が含まれていることが多いです。解答や引用箇所の生成にあたっては、ただし書きの内容も必ず参照してください。
- 質問、解答は日本語で出力してください。引用箇所は原文のまま出力してください。
## 質問に関するルール
- まず初めに対象文書から昊となる「手続き名／略称」「title」「重要用語（略語を含む）」をリストアップしてください。次に、得られた昊を用いて問を立ててください
- 質問は人間にとって理にかなった自然な文章としてください。
- 与えられた手続き文書のみから解答を生成できる質問としてください。
- 生成する質問には必ず参照している手続きの自然な略称を含めるようにし、どの分野の文書について質問を行っているのかわかるようにしてください。
- （人事）など手続きの略称を（）でくくって無理やり入れないでください。あくまでも自然に略称を含めるようにしてください。
- 質問を作成する際には、複数個所に同じ内容が書かれているような解答を作成しないように注意してください。一箇所にしか質問の答えが書かれていないような質問を作成するのが望ましいです。
- 可能な限り具体的な質問を聞くようにしてください。例えば、「注意点は？」「手続きの進め方は？」といった漠然とした質問は避けてください。
- 列挙を求める質問はやめてください。例えば、「必要な書類を教えてください。」「書くべき項目を教えて下さい。」といった質問は避けてください。
  - 必要書類を列挙するのではなく、ある単一の書類について必要かどうかを問う質問ならば許容されます。
- 質問を作成する際には、事前知識をもとにありそうな質問を考えても構いません。例えば、ある手続きに必要な文書が列挙されている際には運転免許証の提示が求められていなくても、運転免許証の提示が求められるかどうか質問しても構いません。
  - 解答を生成する際に事前知識を使わないように注意してください。引用箇所には必要な文章が列挙されている箇所を必ず含めてください。
- できるだけ少ない引用箇所で質問に答えられるようにしてください。質問の条件を絞って解答しやすくすることが望ましいです。
- 上記のルールをすべて満たした質問を作成してください。すべてのルールを満たした場合、書き出しは「〇〇の手続において」や「□□を行う際、」となるでしょう。
## 解答に関するルール
- 解答を生成する際には、根拠となる引用箇所のみから説明できる内容にしてください。
- 解答は根拠となる引用箇所から質問の解答に必要な情報を抜き出したうえで、要約して生成してください。
- 解答はできるだけ簡潔に行ってください。
## 引用箇所に関するルール
- 引用箇所を抜粋する際には、一言一句も漏らさず質問部分に解答するのに必要な箇所を抜粋してください。抜粋箇所は複数箇所でも 1 箇所でも構いません。
- 引用箇所の抜粋の中に質問に関係ない部分が途中含まれていても構いません。解答の根拠となる引用箇所を抜粋することが最優先です。
# 対象の手続きに関する説明
{base_prompt_per_procedure_type}
# 対象文書
{source_chunk_content}
```

A.2 合成 QA のフィルタリングに用いたプロンプト

```
# タスク内容
手続きに関する質問文、その期待される回答、そして文章が与えられます。 与えられた文章のみから、期待される回答と同等の内容を生成できるかどうかを、以下の 3 つのレベルで分類してください。
# 判断ルール
- 「Full」は、期待される回答の主要な内容が文章内に含まれており、質問に対して適切に回答できる場合に選択してください。
  - 重要: 質問が複数項目や列挙を求めている場合でも、期待される回答の一部のみが文章内に含まれている場合は「None」を選択してください。
- 「None」は、期待される回答に関連する情報が文章内に全く含まれていない場合、または部分的にしか含まれていない場合に選択してください。
  - 重要: 質問の前提条件が対象文書に含まれていない場合や、期待される回答の内容と対象文書で答えになると思われる部分の内容が異なる場合には「None」を選択してください。
  - 重要: 期待される回答の一部のみが文章内に含まれている場合も「None」を選択してください。
- 「Unknown」は、質問や回答の意図が不明瞭で分類が困難な場合に選択してください。
# 判断のポイント
期待される回答と文章内容の意味的な一致を重視してください
完全に同じ文言でなくても、同じ内容を表現していれば一致とみなします
部分的な一致は「None」として扱います
出力形式
分類結果のみを出力してください:
```