

AutoEXP: 検索クエリに含まれる固有表現の充足性に基づく自律的な論文検索

高橋拓誠 谷口友紀 朱晨成 大熊智子
旭化成株式会社

{takahashi.tkr, taniguchi.tcr, zhu.cd, okuma.td}@om.asahi-kasei.co.jp

概要

本稿では、検索意図を網羅的に反映した論文集合を獲得する自律的な検索手法 AutoEXP を提案する。検索クエリに含まれる固有表現は、検索意図を示す重要な要素であるという観測に基づき、固有表現を中心とした論文集合の選択・評価・再実行の方策生成を大規模言語モデルを用いたマルチエージェントにより実現する。自然言語指示に基づく論文検索タスクにおいて、AutoEXP は自ら検索結果を評価し、自律的に検索を洗練することで、密ベクトル検索と比較して網羅性が大幅に向上することを示した。

1 はじめに

学術論文の検索は、研究における重要なタスクであるものの、論文の投稿数は年々増加する傾向¹⁾にあり、調査負荷の増加が深刻な課題である。さらに、複雑なクエリへの対応や包括的な検索が要求されるため、依然として難易度が高いタスクである。

複雑な条件に対して柔軟なマッチングを実現するため、自然文のクエリに基づき密ベクトル検索を行うアプローチが多く提案されている [1, 2, 3, 4, 5]。また、大規模言語モデル (Large Language Model; LLM) の高い言語理解・推論能力を活かし、検索の一部を自律化することで、より網羅的な検索を実現する手法が提案されている [6, 7, 8, 9, 10]。しかしながら、既存の手法は、検索クエリに対してより関連する文書集合を得るための方法論であり、**検索の目的や条件からなる情報要求 (以下、検索意図) に対する文書集合の充足性²⁾**は十分に考慮されていない。

本研究では、**検索クエリに含まれる固有表現は検索意図を反映する重要な要素であるという観測 (2 節) に基づき、検索クエリと文書集合の充足性を**

1) https://arxiv.org/stats/monthly_submissions
2) 検索意図を構成する複数の条件や構成要素が、検索結果の文書集合においてどのくらい網羅されているかを示す指標。

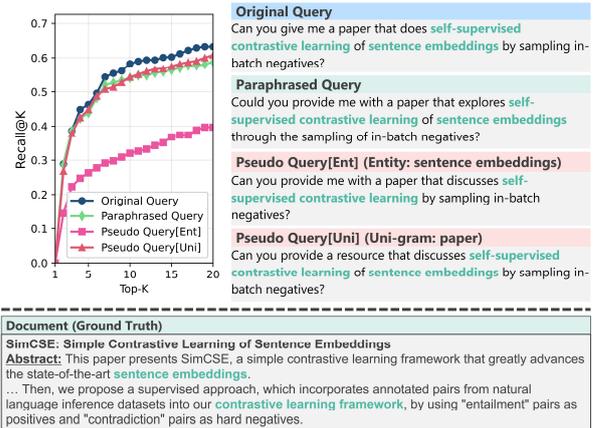


図 1 元クエリおよび特定の語彙を削除または換言したクエリによる論文検索の性能の推移と具体的なクエリの例 (上図)。元クエリに対する適合文書の例 (下図)。

評価しながら、自律的に論文を探索する検索手法 AutoEXP を提案する。AutoEXP は、検索クエリから抽出した固有表現に基づき、(1) 検索クエリに対する文書集合の充足性評価、(2) 評価結果に基づく再検索の方策生成、(3) 生成された方策に基づく検索クエリの洗練、を充足性を満たすまで繰り返し実行する。本論文の貢献は、以下の 3 点である。

- 検索クエリに含まれる固有表現は、検索意図を反映する重要な要素であることを示す (2 節)。
- 検索意図の充足性を評価しながら自律的に探索する論文検索 AutoEXP を提案する (3 節)。
- 自律的な検索により、密ベクトル検索と比較して網羅性が大幅に向上することを示す (4 節)。

2 事前分析

クエリ側からの観測: 検索クエリに含まれる固有表現が、適合文書を検索するために特に重要な情報を含むのであれば、以下の性質が成り立つ。

性質 1 固有表現を検索クエリから削除すると、検索性能が大幅に低下する。

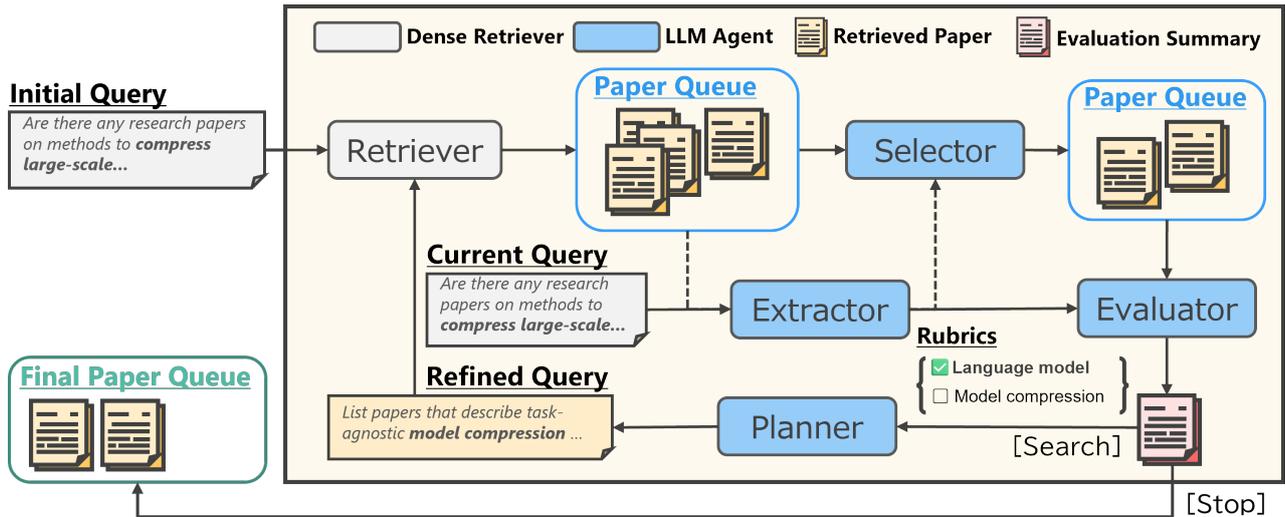


図2 AutoEXPの概要. Evaluatorの評価において、論文集合 (Paper Queue) が評価項目 (Rubrics) を充足すると判定されるか、指定回数に到達するまで繰り返し実行する。繰り返し処理終了時の論文集合を最終的な検索結果として出力する。

性質 2 固有表現を維持したまま検索クエリを換言しても、検索性能はほとんど変化しない。

計算科学の論文を対象とする検索評価のデータセット LitSearch [11] を用いて検証した。図1の評価結果から、元の検索クエリ (Original Query) に対する検索性能と比較して、元クエリから固有表現を削除したクエリ (Pseudo Query[Ent]³⁾ に対する検索性能が大幅に低下することがわかる (性質 1)。一方で、固有表現を維持しつつ検索クエリに摂動を与えても、検索性能はほとんど変化しない (性質 2)。具体的には、元クエリの単純な換言 (Paraphrased Query⁴⁾) と元クエリから固有表現以外の単語ユニグラムを削除したクエリ (Pseudo Query[Uni]³⁾) を用いても、検索結果にほとんど影響を与えないことが示された。したがって、**検索クエリに含まれる固有表現は検索意図を反映する重要な要素**である。

文書側からの観測: 図1に検索クエリに対する適合文書の例を示す。適合文書は、検索クエリに含まれる固有表現をもれなく含むことが確認できる。この性質は、統計的にも顕著に表れていることから (付録 A)、文書中の固有表現は適合文書を識別するうえで重要な評価基準になりうる。

3 AutoEXP: 自律的な論文検索

本研究では、検索クエリから抽出した固有表現に基づき、論文の選択・評価・再実行の方策生成を繰り返し実行する自律型検索 **AutoEXP** を提案する。

3) 語彙を削除した後の文法破綻を避けるため、GPT-4o-mini による文法チェックおよび修正を適用した。

4) GPT-4o-mini を用いて言い換え文を生成した。

図2に提案モデルの概要を示す。AutoEXPは、検索クエリ q から密ベクトル検索 Retriever を用いて候補論文 $\tilde{P} = \{p_1, \dots, p_N\}$ を獲得し、マルチエージェント $\mathcal{A} = \{\text{Selector}, \text{Extractor}, \text{Evaluator}, \text{Planner}\}$ により自律的に検索を拡張・洗練する。以降では、各エージェントの具体的な処理について説明する。

3.1 Selector

検索クエリ q 、タスク指示 I_{Selector} に基づき、論文 p_i が検索クエリ q の適合文書か示すラベルを生成する。分類精度向上のため、論文 p_i に含まれるラベル \tilde{l}_j のエンティティ \tilde{e}_j から構成される固有表現 $\tilde{s}_j = (\tilde{e}_j, \tilde{l}_j)$ のリスト $\tilde{S}_{p_i} = \{\tilde{s}_1, \dots, \tilde{s}_M\}$ を Extractor で生成し、Selector への入力に追加する。形式的には、 $\tilde{o} = \text{Selector}([q; I_{\text{Selector}}; p_i; \tilde{S}_{p_i}])$ により各論文を分類する。なお、 $\tilde{o} \in \{\text{[Select]}, \text{[Drop]}\}$ であり、 $\tilde{o} = \text{[Select]}$ の論文のみ論文集合 P に追加する。

3.2 Extractor

事前分析で得た観測 (2節) に基づき、本研究では検索クエリに含まれる固有表現を検索で満たすべき評価項目 (Rubrics) とみなす。そこで、文脈内学習 (In-context Learning; ICL) による固有表現抽出 [12] に基づき、クエリ q に含まれる固有表現 $\tilde{S}_q = \{\tilde{s}_1, \dots, \tilde{s}_L\}$ を生成する。具体的には、 $\tilde{S}_q = \text{Extractor}([q; I_{\text{Extractor}}; D])$ を実行する。なお、 $D = \{d_1, \dots, d_k\}$ であり、 d_i はテキスト x_i と正解の固有表現 S_i から構成される少量の抽出事例である。抽出事例 D の詳細は付録 C を参照されたい。

表1 BroadおよびSpecificにおける論文検索の性能評価。R@KはRecall@Kを表す。

	Broad				Specific			
	R@5	R@10	R@15	R@20	R@5	R@10	R@15	R@20
text-embedding-3-large								
Retriever	0.4671	0.5639	0.5933	0.6277	0.6109	0.6957	0.7443	0.7692
+ QE (GPT-4.1)	0.4994	0.5735	0.6071	0.6297	0.6312	0.7059	0.7534	0.7817
AutoEXP (GPT-4.1-nano)	0.5262	0.5735	0.5826	0.5912	0.6437	0.6787	0.6855	0.6889
AutoEXP (GPT-4.1-mini)	0.5219	0.5908	0.6127	0.6320	0.6776	0.7364	0.7500	0.7557
AutoEXP (GPT-4.1)	0.5671	0.6273	0.6385	0.6611	0.6731	0.7443	0.7658	0.7692
Qwen3-Embedding-0.6B								
Retriever	0.4895	0.5628	0.5940	0.6069	0.6391	0.6946	0.7353	0.7432
+ QE (GPT-4.1)	0.4905	0.5809	0.6275	0.6275	0.6584	0.7229	0.7692	0.7805
AutoEXP (GPT-4.1-nano)	0.5185	0.5520	0.5746	0.5940	0.6210	0.6403	0.6448	0.6471
AutoEXP (GPT-4.1-mini)	0.5432	0.5692	0.6144	0.6176	0.6867	0.7149	0.7161	0.7229
AutoEXP (GPT-4.1)	0.5561	0.5951	0.6370	0.6370	0.7002	0.7262	0.7432	0.7455

3.3 Evaluator

Evaluatorは、(1) Selectorで選別した論文集合 P が検索クエリ q の評価項目 \tilde{S}_q を満たすか評価し(**Document Evaluation**), (2) 評価結果に基づき再検索の必要性和次に着目すべき評価項目を含めた評価レポートを生成する(**Report Generation**).

Document Evaluation (DE). 論文集合 P に含まれる論文 p_i について、クエリ q に対する充足性を評価する。具体的には、 $S_{p_i}^* = \text{Evaluator}_{\text{DE}}([p_i; I_{\text{Evaluator}}; \tilde{S}_q])$ により、論文 p_i が満たす評価項目の系列 $S_{p_i}^*$ を生成する。この操作を P に含まれるすべての論文に行うことで、各論文が満たす評価項目の系列集合 $S_p^* = \{S_{p_1}^*, \dots, S_{p_n}^*\}$ を得る。

Report Generation (RG). DEの評価結果 S_p^* に基づき、論文集合 P が評価項目 \tilde{S}_q を満たすか総合評価し、評価レポート r 、次の行動 $a \in \{\text{[Search]}, \text{[Stop]}\}$ 、次に着目する評価項目 s^* を生成する。具体的には、 $z = \text{Evaluator}_{\text{RG}}([P; I_{\text{Evaluator}}; \tilde{S}_q; S_p^*])$ を実行する。なお、 $z = (r, a, s^*)$ である。最終的に、 $a = \text{[Search]}$ の場合は再検索用のクエリを生成するため、Planner(3.4節)を呼び出す。一方で、 $a = \text{[Stop]}$ または一連の検索処理が指定回数に到達した場合は、論文集合 P を最終的な検索結果として出力する。

3.4 Planner

3.3節で得た評価レポート r と次に着目する評価項目 s^* に基づき、検索クエリ q と論文集合 P のギャップを埋めるため、 s^* に着目した再検索クエリ q^* を生成する。具体的には、 $q^* = \text{Planner}([r; I_{\text{Planner}}; s^*; q])$ により再検索のためのクエリ q^* を生成する。最終的に、クエリ q^* を用いて一連の処理を再実行する。

4 実験

4.1 実験設定

データセット. 計算機科学の論文を対象とした情報検索のデータセット LitSearch[11]を用いて評価する。幅広い論文が該当する設定(**Broad**)と具体的で少数の論文しか該当しない設定(**Specific**)に分けて評価した。また、Extractor(3.2節)の少量事例のデータセットとして SciER [13]を利用した。対象の固有表現ラベルは、**Dataset**, **Method**, **Task**の3種類である。データセットの詳細は付録Bを参照されたい。
ベースライン. 密ベクトル検索(Retriever)を構築するための埋め込みモデルとして、text-embedding-3-largeとQwen3-Embedding-0.6B [14]を採用した。さらに、GPT-4.1を用いたクエリ拡張に基づく検索(Retriever+QE)を比較手法として用意した。

マルチエージェント. GPT-4.1-nano, GPT-4.1-mini, GPT-4.1を各エージェントのLLMとして採用した。各モデルの詳細は付録Cを参照されたい。

4.2 評価結果

AutoEXP vs. ベースライン. 表1より、AutoEXPはRetrieverと比較して大幅な性能向上を示した。特に、text-embedding-3-largeの検索上位においては、最大10.0%の性能向上を達成した。さらに、クエリ拡張(+QE)と比較して、AutoEXPはほとんどの場合において、より網羅性の高い検索結果を示した。検索クエリに対する論文集合の充足性に基づき、自律的に検索プロセスを改善することで、単純なクエリ拡張よりも大きな改善効果を得ることができる。

Broad vs. Specific. 表1より、AutoEXPはBroadの

表 2 アブレーション実験による検索性能の評価。全ての評価データ (Broad および Specific) を対象に評価した。

	R@5	R@10	R@15	R@20
Retriever	0.5735	0.6615	0.7051	0.7325
AutoEXP (mini)	0.6372	0.6986	0.7143	0.7236
w/o Selector	0.5735	0.6648	0.7118	0.7409
w/o Extractor	0.6238	0.6819	0.6976	0.7076
w/o Evaluator	0.5802	0.6444	0.6833	0.7093
w/o Planner	0.6112	0.6643	0.6767	0.6808

設定においてより改善効果大きいことがわかる。これは 4.1 節で示したように、Broad はより幅広い論文が該当することから、検索クエリの充足性を考慮することが一層重要であるためだと推察される。

埋め込み表現に対するロバスト性。 AutoEXP は、text-embedding-3-large および Qwen3-Embedding-0.6B において、Retriever を大きく上回る性能を示した (表 1)。したがって、AutoEXP の効果は、特定の埋め込み表現に依存せず一貫して得られる。

LLM の性能が検索に与える影響。 表 1 より、LLM の性能に応じて全体的な検索の性能が大きく変化することがわかる。特に、GPT-4.1-nano の性能向上 (全体で最大 4.60%) に対して、GPT-4.1 は最大 8.11% の性能向上を達成した。AutoEXP の効果を最大化するうえで、LLM の性能は重要な要素といえる。

4.3 アブレーション実験

AutoEXP (GPT-4.1-mini) を対象として、各エージェントの効果を評価する。なお、埋め込みモデルは、text-embedding-3-large を用いて評価した。

表 2 より、すべてのエージェントが性能向上に貢献していることがわかる。特に、検索した論文集合を評価することなく指定回数まで検索を繰り返す場合 (w/o Evaluator)、検索性能が大幅に低下することから、**検索クエリに対する文書集合の充足性を評価したうえで再検索することは非常に有益である**といえる。さらに、検索したすべての論文を逐次追加する場合 (w/o Selector)、検索上位の性能を著しく損なうことが示された。自律的な論文検索において、適切な文書選択が重要な役割を持つことは、先行研究 [10] の報告とも一致する。

4.4 自律的な繰り返し検索の効果

本節では、AutoEXP の自律的な反復検索により網羅性が向上するか検証する。なお、埋め込みモデルは text-embedding-3-large、ベースラインは Retriever、提案モデルは AutoEXP (GPT-4.1) を用いて LitSearch

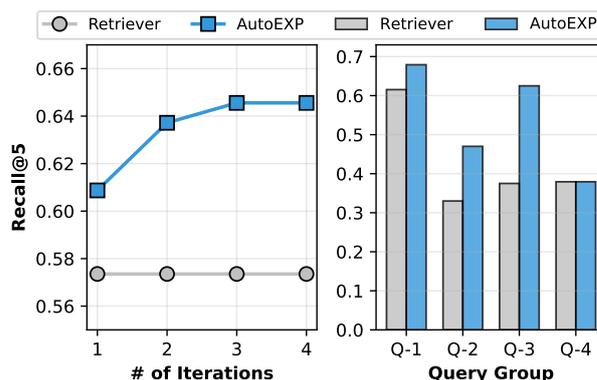


図 3 繰り返し回数の増加に伴う検索性能の推移 (左図)。実際に AutoEXP で要した検索回数 N' に基づきクエリ Q を分割した各グループ ($Q-N'$) の検索性能 (右図)。なお、全グループで Retriever は 1 回だけ検索を実行する。

の全データ (Broad と Specific) を対象に評価した。

繰り返し回数ごとの性能評価。 図 3(左図) に繰り返し回数ごとの検索性能の推移を示す。AutoEXP は、自律的な繰り返し検索により網羅性が向上することがわかる。さらに、検索回数が 1 回の場合でも Retriever と比較して高い性能を示すことから、Selector における適合文書の選択や Evaluator の充足判定が効果的に機能していると推察される。

クエリごとの性能評価。 図 3(右図) より、AutoEXP は反復検索が必要なクエリ (Q-2, Q-3) について、大幅な性能向上を示した。一方で、反復回数が上限に達した場合 (Q-4)、AutoEXP による改善は見られず、Retriever と AutoEXP のいずれも低い水準であった。検索結果が検索意図を充足しないことを正しく判定しつつも、有効な方策を立案できておらず、今後 Planner を改良する必要性が示唆される。また、自律的な反復検索におけるクエリの改善と検索結果の具体的な事例については、付録 D を参照されたい。

5 おわりに

本研究では、検索クエリに含まれる固有表現は検索意図を反映するという性質に基づき、クエリと文書集合の充足性を評価しながら自律的に論文を検索する AutoEXP を提案した。検索した論文集合の充足性評価に基づき再検索を繰り返すことで、AutoEXP は密ベクトル検索やクエリ拡張と比較して、検索の網羅性を大幅に向上することを示した。

今後の課題として、4.4 節で示したように Planner の改善が挙げられる。また、検索のフロー全体を通して、各エージェントの全体最適化を行うことで、さらなる効果を得られるか検証する予定である。

参考文献

- [1] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wentaoh Yih. Dense passage retrieval for open-domain question answering. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing**, pp. 6769–6781, 2020.
- [2] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. **Transactions on Machine Learning Research**, 2022.
- [3] Yibin Lei, Liang Ding, Yu Cao, Changtong Zan, Andrew Yates, and Dacheng Tao. Unsupervised dense retrieval with relevance-aware contrastive pre-training. In **Findings of the Association for Computational Linguistics: ACL 2023**, pp. 10932–10940, 2023.
- [4] SeongKu Kang, Yunyi Zhang, Pengcheng Jiang, Dongha Lee, Jiawei Han, and Hwanjo Yu. Taxonomy-guided semantic indexing for academic paper search. In **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing**, pp. 7169–7184, 2024.
- [5] Yunyi Zhang, Ruozhen Yang, Siqi Jiao, SeongKu Kang, and Jiawei Han. Scientific paper retrieval with LLM-guided semantic-based ranking. In **Findings of the Association for Computational Linguistics: EMNLP 2025**, pp. 2049–2060, 2025.
- [6] Sotaro Takeshita, Tornike Tsereteli, and Simone Paolo Ponzetto. GenGO ultra: an LLM-powered ACL paper explorer. In **Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)**, pp. 242–251, 2025.
- [7] Jiazhan Feng, Chongyang Tao, Xiubo Geng, Tao Shen, Can Xu, Guodong Long, Dongyan Zhao, and Daxin Jiang. Synergistic interplay between search and large language models for information retrieval. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 9571–9583, 2024.
- [8] Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Min zhang, Qingsong Wen, Wei Ye, Shikun Zhang, and Yue Zhang. Autosurvey: Large language models can automatically write surveys. In **The Thirty-eighth Annual Conference on Neural Information Processing Systems**, 2024.
- [9] Ori Press, Andreas Hochlehnert, Ameya Prabhu, Vishaal Udandarao, Ofir Press, and Matthias Bethge. CiteME: Can language models accurately cite scientific claims? In **The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track**, 2024.
- [10] Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, and Weinan E. PaSa: An LLM agent for comprehensive academic paper search. In **Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 11663–11679, 2025.
- [11] Anirudh Ajith, Mengzhou Xia, Alexis Chevalier, Tanya Goyal, Danqi Chen, and Tianyu Gao. LitSearch: A retrieval benchmark for scientific literature search. In **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing**, pp. 15068–15083, 2024.
- [12] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, Guoyin Wang, and Chen Guo. GPT-NER: Named entity recognition via large language models. In **Findings of the Association for Computational Linguistics: NAACL 2025**, pp. 4257–4275, 2025.
- [13] Qi Zhang, Zhijia Chen, Huitong Pan, Cornelia Caragea, Longin Jan Latecki, and Eduard Dragut. SciER: An entity and relation extraction dataset for datasets, methods, and tasks in scientific documents. In **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing**, pp. 13083–13100, 2024.
- [14] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. **arXiv preprint arXiv:2506.05176**, 2025.

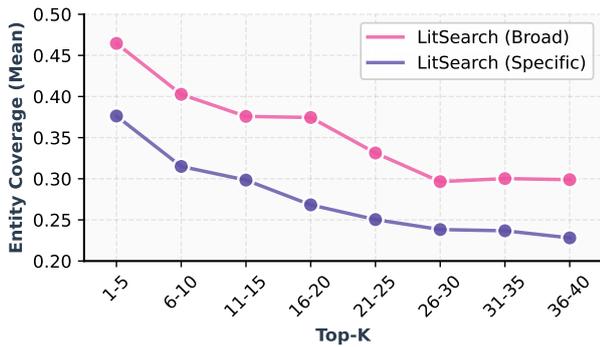


図4 検索クエリに含まれる固有表現が検索文書に含まれる割合。

表3 LitSearch [11] の統計量.

	クエリ数	適合文書数/クエリ	対象文書数
Broad	155	1.17	64,183
Specific	442	1.04	64,183

A 検索文書が含む固有表現の分析

検索クエリに含まれる固有表現を表す語彙やフレーズが、検索で得られた文書集合にどのくらい含まれているか評価した。各固有表現に関する記載が文書中に存在するか否かについては、GPT-4o-miniを用いて判定した。図4より、検索結果が上位の文書は、検索クエリに含まれる固有表現を表す語彙やフレーズが多く含まれる傾向を示した。適合文書は検索上位に含まれる確率が高いことから、検索クエリに含まれる固有表現をより多く含意する傾向にあるといえる。

B データセットの詳細

論文検索のデータセット. 本稿では、LitSearch[11]を用いて実験および分析を実施した。LitSearchは機械学習および自然言語処理の論文(ICLR, ACL)を対象とした論文検索を評価するためのデータセットである。表3に統計量を示す。

固有表現抽出のデータセット. 3.2節より、ExtractorではICLによる拡張を実施した。本稿では、科学論文からの情報抽出を目的としたデータセットであるSciER [13]をICLの少量事例として利用した。表4にデータセットの統計量を示す。

C モデルの設定

共通のハイパーパラメータ. マルチエージェント d のすべてのモデルにおいて、温度 $T = 0$, $\text{top-}p = 1.0$ に設定した。出力するトークン数は2048を上限と

表4 SciER [13] の統計量.

	Train	Dev	Test
Dataset	3,220	269	370
Method	11,424	1,549	1,890
Task	3,397	416	688

表5 自律的な繰り返し検索に基づく検索クエリの改善と検索結果の例。強調表示はDatasetとMethodを示す。

Retriever	
Query	Are there any examples of using dense phrase retrieval systems in the automatic curation of entity dictionaries ?
Title	Learning Dense Representations of Phrases at Scale
AutoEXP (# of iterations = 2)	
Query	What are the methods and evaluations for automatic curation in knowledge base and entity dictionary creation , with a focus on the use of dense phrase retrieval and dense representation techniques? ...
Title	Automatic Creation of Named Entity Recognition Datasets by Querying Phrase Representations

して生成した。AutoEXPにおける繰り返し検索の上限数は4回に設定した。

Retriever. 検索クエリおよび論文を分散表現に変換しコサイン類似度で上位 K 件の文書を取得する。なお、論文はタイトルおよび概要を検索対象とした。

Retriever+QE. 元クエリの文意を維持しながら異なる表現や構造を持つ拡張クエリ $Q' = \{q'_1, \dots, q'_n\}$ をGPT-4.1により生成した。なお、提案手法と公平に比較するため $n = 3$ に設定した。最終的に、元クエリ q および拡張クエリ Q' を用いて検索を行い、コサイン類似度の上位 K 件の文書を取得し評価した。

Extractor. ICLにおける少量事例 $D = \{d_1, \dots, d_k\}$ は、検索クエリ q に最も類似する事例をSciER [13]の学習データから5件サンプリングして活用した。具体的には、text-embedding-3-largeを用いてクエリと事例を分散表現に変換し、Retrieverで検索した。

D 事例分析

表5に自律的な検索クエリの改善と検索結果の事例を示す。Retrieverのクエリは、Datasetに関する研究に焦点を当てつつも、実際の検索結果においてはMethodに関する論文が出力された。一方、AutoEXPのクエリは、Datasetに関する記述をより詳細に記述することで、密ベクトル表現(Method)を用いた固有表現抽出のデータセットの自動構築(Dataset)に関する適合文書を得ることができた。