OPAC等レガシーな検索システムに対する大規模言語モデル技術の適用可能性について

小野 亘[†1,a)]

概要：
本稿は、GPT のような大規模言語モデル（LLM）の技術の進展に伴い、図書館の蔵書検索（OPAC）のようなレガシーな検索システムに対して、GPT のような大規模言語モデルの技術が、検索質問の生成、検索式への変換、意味を考慮した検索、結果の表示と適合性の評価という検索課程のそれぞれに対して、適用できることを示した。また、OPAC 自体が LLM に対しての情報基盤となり得ることを検討した。

キーワード：大規模言語モデル, 生成 AI, 大学図書館, 蔵書検索, OPAC

The applicability of Large Language Model(LLM) techniques to legacy information retrieval systems such as OPAC

Wataru, ONO[†1,a)]

Abstract: This paper shows that with advances in large language model (LLM) techniques such as GPT, they can be applied to legacy retrieval systems such as the Library's Online Public Access Catalogue (OPAC) for the following tasks: generating search questions, transforming them into search query, semantically aware retrieval, search results and evaluating their suitability. It was shown that it could be applied to each of the search processes. It was also investigated that the OPAC itself can serve as an information infrastructure for LLM.

Keywords: large language model, LLM, generative AI, University Libraries, Online Public Access Catalogue, OPAC

1. はじめに

本稿では、図書館の蔵書検索 (Online Public Access Catalog: OPAC) のようなレガシーな検索システムに対して、生成 AI や大規模言語モデル (Large Language Model: LLM) の技術がどのように適用できるかを考察する。なお、システム的な実装や、実際の LLM を使った具体的なテストなどは行っておらず、文献調査に基づいたエッセイであることをあらかじめお断りしておく。

2. 結論

OPAC 等のレガシーな検索システムにおける、検索質問の生成、検索式への変換、意味を考慮した検索、結果の表示と適合性の評価という検索課程のそれぞれに対して、LLM を適用することができることが分かった。LLM を適用することによって、自然言語による問い合わせを行い、問い合わせの意味、回答の意味を考慮した検索を行うことや、ユーザにより分かりやすい順位で回答を行うことができ、また、単に結果のリストを表示するのみでなく、自然言語による直接的な回答も参照文献つきで可能である。OPAC 等への LLM 技術の適用は、既存の OPAC 等の検索システム自体は大きく変更することなく行える部分も大きい。また、既存の OPAC 自体が LLM に対してメタデータを提供する基盤となることが重要である。そのためには、より LLM に可読な、あるいは埋め込み可能、学習可能なメタデータに改善していく必要がある。また、これらは、従来ともすれば、検索式の作成や検索語の選び方などに重点が

[†1] 現在，人間文化研究機構 本部事務局
Presently with Headquarter Office, National Institutes for the Humanities
[a)] ono@nihu.jp

おかれていた図書館的な情報リテラシーの考え方に変更を迫るものである。

3. OPAC

本稿でいうレガシーな検索システムとは、転置インデクスを、ブール演算に基づいて検索するタイプの、現在、OPAC で一般的に見られるシステムを想定している。一般に、OPAC のような検索システムでは、ユーザーはテキスト形式で検索システムに問い合わせ、検索システムは、これらのクエリをデータベースと照合し、通常は、データベースのインデクスと入力した検索語が合致した結果を返す。

一方、検索エンジンに代表される一般的な検索システムは、一見別物のようには見える部分もあるが、データ入力、インデクス、検索、順位付けという段階に変わりがあるわけではない。いわゆる検索エンジンは、OPAC のような検索システムに較べて、より最新の技術が適用されていることが多い。それ以上に、OPAC は蔵書としてきっちり管理されたデータベースから、網羅的に漏れなく検索する、というメンタルモデルに基づいているのに対し、検索エンジンは、ウェブ全体という漠然とした検索対象に対して得られる膨大な検索結果から、質問にあっていると思われる順番に提示する、というメンタルモデルに基づいている、という違いが大きいと考えている。

言い換えれば、OPAC は、データベース全体がある図書館の蔵書全体と（原則）一致しており、そこから、検索によってどのくらい網羅的に検索できたかを示す再現率が 100%であることが想定されてきた。言い換えれば、ある図書館の蔵書にその図書があるはずならば、検索できなければおかしい、という想定がある。検索エンジンの場合、再現率は、理論的には計算は可能かもしれないが、なにが検索対象の全体かを把握することは難しい。

3.1 LLM 技術の必要性

しかし、研究や学習にとって、本質的に、一つの図書館の蔵書だけで資料が足りるということはありえない。検索エンジン以前にも、冊子体目録やカード目録の交換などの工夫は行われてきたが、Google 検索が登場した 1998 年からすでに 20 年以上が過ぎ、他の図書館はもちろん、論文やさまざまデータ、必ずしも学術資料でないものも含めて検索することが当然となっている。情報オーバーロード [1] や、情報爆発と言われて久しいが、“1981 年に比べ現在は、世界で発表される論文量は約 4.0 倍になっており、世界で行われる研究活動は一貫して量的拡大傾向にあ”[2] り、分野によってはその分野内に絞った論文であっても、再現率のもととなる文書全体の母数の把握は難しく、ある程度は把握できたとしても、検索結果の一覧は膨大になる傾向があり、その中から、自分が必要とする文献を抽出することは容易ではない。

従って、学術的な情報検索においても、膨大な論文、データやドキュメントから、必要とするものを検索するため、LLM の高度な能力の適用が模索されており、LLM の活用によって情報検索（Information Retrieval: IR）も大きく変わりつつある。LLM を IR に統合することで、例えば、IR はよりパーソナライズされ、ユーザの質問の意図に対して、直接的な回答を得られるようになることが期待されている。また、LLM の時代に従来の IR が必要かどうかということも、大きな課題となっている。[3] [4]

しかし、現状の LLM の弱点は、ハルシネーション（幻覚）や最新の情報、特定のドメインに関する知識が不足していることなどがあげられ、ChatGPT などの生成 AI をそのままでは検索エンジンや文献検索に使用することはできない。

3.2 検索過程

OPAC 等の検索システムでは、ユーザが自分で指定したキーワードを指定するシステムが主流である。OPAC で検索を行う過程は、検索質問の作成、検索式への変換、検索式中の検索語と蓄積情報中の索引語（インデクス）との照合、結果の適合性の評価などからなっている [5]。それらの過程のうち、LLM 技術を適用できるのは、以下の 4 つの部分が考えられる。

(1) 検索質問の作成
  (a) ユーザが入力したキーワードをより適切なキーワードに書き直す（クエリの書き直し）
  (b) ユーザーは自然文、またはキーワード以外の入力を行い、それを適切なキーワードに書き直す（クエリの生成）
  (c) 会話形式により、検索を進める（対話型検索）
(2) 検索式への変換
(3) 検索語とインデクスの照合： 転置インデックスからベクトル化インデックスによる高密度検索へ (意味を考慮した検索)
(4) 結果の表示と、適合性の評価：
  (a) 再順位付け（リランク）
  (b) 検索結果と結果の評価

本稿では、それぞれの過程について、LLM の活用方法を考えてみる。

4. 検索質問の作成

4.1 クエリの書き直し

ユーザは、必ずしも自身の情報要求に対する適切なキーワードを生成できるわけではない。自身の要求を適切に言語化できていない場合や、対象となるデータベースのインデクスにあるキーワードを完全に把握することは不可能なため、何がキーワードとして適切かは分からない。そのため、キーワードの上位語や下位語、関連する語への言い換

え、基本的なブール演算子、前方一致などの演算子を工夫することなどが必要であり、OPAC を使いこなすための図書館情報学的な情報リテラシーの教育、トレーニングが必要とされてきた。

LLM は幅広い知識を持ち、さまざまな概念や情報を利用できるため、LLM を使うことによって、ユーザが最初に指定したキーワードから、より関連性の高いキーワードを生成できる可能性がある。一方、LLM は、過去のある時点のテキストデータによって学習が行われているため、最新の知識や、特定のドメインの知識はなく、必ずしも OPAC に存在するキーワードを学習しているとは限らない。そのため、ファインチューニング、検索拡張生成 (Retrieval Augmented Generation: RAG)、ナレッジグラフやベクトルストア（ベクトルデータベースともいう）を使った方法などが提案されている。

4.2 自然文からのクエリの生成

LLM は、言語の意味や文脈を効果的に捉えることができるため、自然文であっても、ユーザの検索要求の意図を理解できる可能性が高い。自然文の質問から検索要求の意図を読み取り、そこからキーワードを抽出（特徴語抽出、トピック抽出、固有表現抽出などの手法がある）することが可能となる。

加えて、質問以外からもキーワードの抽出は可能である。例えば、あるドキュメント（論文など）を入力として、そこから適切なキーワードを抽出するなどは、関連文献の検索などには効果的な手法であろう。

4.3 対話型検索

いわゆるチャット形式による対話型検索では、ユーザーと検索システムの動的なやり取りが行われ、システムはユーザーのクエリに応答し、情報ニーズを明確にするための対話が行われる。現在の OPAC 等においては、ユーザが検索結果を確認しながら、上位語や下位語や関連する語で検索しなおしたり、キーワードを追加したり、ファセットで絞り込んだりすることが、ほぼ必須の作業だが、ChatGPT のような対話型インターフェイスにより、より適切な検索結果にたどり着くことが可能になるだろう。

4.4 検索式への変換

先述の自然文からのクエリの生成の応用とも言えるが、LLM を使って、自然文をデータベース問い合わせ言語である SQL に変換する Text-to-SQL という手法がある。これを応用することによって、自然言語の質問文から、より精緻にデータベースに問い合わせることができるのではないか。

5. 意味を考慮した検索

5.1 RAG

LLM の本質は、単語の（正確にはトークンあるいはチャンクごとの）条件付確率に基づいて、それっぽい文章を生成しているだけであり、事実が回答されるとは限らない。

宮尾は、LLM について、

事実を述べるように設計されてはいない

- 人間が書いたかのような文章を生成することで、ある程度事実を述べることができる（言語モデルの中に知識が組み込まれている）
- 学習データに含まれる知識はニューラルネットワークのパラメータとして埋め込まれており、確実に引き出せるわけではない
- 現状の原理（のみ）では、事実性を保証することは困難

と述べている [6]。元の学習データに必要な事実が含まれているとは限らないし、元の学習データに必要な事実が含まれていたとしても、それがそのまま引き出せるとは限らない。これが、LLM の大きな課題である幻覚（ハルシネーション）の一つの原因である。

それらの課題について、様々な改善策、回避策などが提案されており、実用的な実装も出始めている。LLM 自体を、検索として使うには、必要な事実を学習させ、学習したデータを適切に引き出す必要がある。OPAC に適用するとすれば、ある OPAC のデータ全体を学習し、その検索語に対する請求記号なり資産番号なりが返ってくることが必要である [10] [11]。

一方で、たとえば図書館ごとに OPAC のデータ全体を学習した LLM を構築することは、現時点では現実的ではない。そこで、LLM に必要な情報を学習させるには、LoRA（Low-Rank Adaptation: 低ランク近似）などによるファインチューニングや RAG（Retrieval-augmented generation: 検索拡張生成）という手法があるが [12]、図書館ごとにファインチューニングを行うことも、あまり現実的ではないとすれば（LLM 生成の高速化、省エネ化、低コスト化は今後進むと考えられるので、将来的には可能となる可能性は高いが）、Augmented Language Models の一種である RAG による生成が当面有望と考えられる (RAG では幻覚を減らすことはできない、という見解もある [13])。

RAG は、”大規模な言語モデルの出力を最適化するプロセスです。そのため、応答を生成する前に、トレーニングデータソース以外の信頼できる知識ベースを参照します。” [7] という手法で、一般的には、”プロンプトに含まれる情報に基づき，検索エンジンや類似性を判断できるベクトルデータベースを使って，関連する知識を参照可能にする” [8][9]。知識ベース（ナレッジベース）へのアクセス方法

は、上述のベクトルデータベースを用いるもののほか、さまざまな手法があり、RAG という手法を使うことによって、既存の OPAC などを LLM の知識ベースとして使うことも可能となる。より具体的な活用イメージとしては、生成 AI に質問し、生成された内容を使って OPAC などを検索し、その結果を OPAC へのリンクなどがついた引用文献リストとして返すことなどが想定される。

また、LLM が、RAG を用いて OPAC から情報を取得することは可能だが、より効率的、効果的に情報を取り出すためには、さらなる機械可読化あるいは構造化された書誌データを提供することが必要である。メタデータを Linked Data 化する試みとしては、Linky MARC [14] や BIBFRAME [15] [16] が存在する。

さらに、先述の Text-to-SQL 類似の手法として、LLM を用いてテキストから、ナレッジベースに対するクエリ言語 SPARQL を生成する手法もある [17]。

ティム・バーナーズ＝リーの提唱したいわゆるセマンティックウェブは、情報の意味（セマンティクス）を計算機で扱うための試みと言える。セマンティックウェブを構成する要素の一つに RDF（Resource Description Framework）があり（ティム・バーナーズ＝リーは、RDF の出発点の一つが図書館のカード目録だ、といっているのは興味深い）[18]、RDF をナレッジグラフとして、ベクトル化あるいは埋め込みを行うことによって、OPAC の書誌データが持つ「意味」、また加えてユーザーのクエリの意味を LLM に伝えることができる。

5.2 セマンティック検索

LLM の功績の一つは、意味の分散表現、計算意味論という形での限定的な意味ではあるが、計算機で、言葉の「意味」の解析をに可能としたことだろう [19]。ここでいう「意味」は、情報と情報の関係性と言い換えてもよい。これによって、単純なクエリの単語と転置インデックスとの照合を超えた検索を行うことが可能となる。LLM を使って、データを数値のベクトルに変換し、このベクトル間の距離や類似性を計算することで検索を行う技術を、セマンティック検索やベクトル検索と言う。分散表現自体は、Word2Vec などで既に行われており、ベクトル検索自体も LLM 以前からあるが、LLM によって、精度も上がり、実用的になってきた。[21] [22]

セマンティック検索の利点は、以下のように言われている。[20]

(1) 精度と関連性: ユーザーの意図とコンテキストを理解することにより、関連性の高い結果を提供
(2) 自然言語理解: 複雑なクエリを理解し、自然言語による対話を行う
(3) あいまいさの排除: あいまいなクエリを解決し、ユーザーの行動とコンテキストに基づいて正確な結果を提供
(4) パーソナライゼーション: ユーザーの行動から学習して結果をカスタマイズし、時間の経過とともに関連性を向上させることができる

OPAC のデータをベクトル化したり、ベクトル埋め込み（Embedding、埋め込み表現）とすることによって、セマンティック検索を実現することによって、上記の利点を活かすことが可能となる。

一方、OPAC のデータ自体にセマンティック検索を導入することは、根本的なアーキテクチャの変更を伴うため、ハードルも高い。既存の OPAC のデータを、ベクトル埋め込みを行うための基盤のデータベースとして使っていく方法は、前節の RAG とともに検討していく必要があるだろう。

6. 結果の表示と、適合性の評価

6.1 再順位付け（リランク）

OPAC に LLM 技術を適用する方法の一つとして、検索結果に対する処理がある。reranker または reranking と言われる手法では、検索結果について、元のクエリとの関連性を、LLM を使って計算し、再ランク付けし、検索結果の表示順を調整できる。

検索結果の表示順として、タイトルや出版年月等の降順・昇順のほか、関連性が用いられている。関連性の算出手法としては、クエリの単語に基づく TF-IDF や Okapi BM25 などが使われているが、LLM を使うことによって、クエリが自然文であっても、クエリの意味と検索結果の意味との関連性を計算し、表示順を調整することができる。

表示順における関連度は、網羅的に漏れなく検索する、あるいは検索語に対応した検索結果、というメンタルモデルがある場合、仕組みがブラックボックスで、なぜヒットしているか分かりにくい、と言われていたこともあったが、クエリが自然文であったり、必ずしも検索対象にない単語がクエリに使われ、また、検索対象であるメタデータや全文のデータも大規模化、リッチになってきていることから、単純に上記のメンタルモデルでは、検索が難しくなってきており、LLM を使うことによって、関連度の精度を挙げることは必須である。

6.2 検索結果と結果の評価

LLM を使うことによって、単に検索結果のリストをランキング順に表示することを超えて、質問に対して直接的な回答文を提示することも可能である [23]。この場合、Microsoft Bing AI や Google Gemini のように、書誌情報の典拠とともに、回答文が表示できるため、既存の OPAC が有効に活用できる。先述の RAG によって取得した参照情報を組み込むことで、LLM のハルシネーションの改善

にも有効である [24]。

7. おわりに

本稿では、OPAC のようなレガシーな検索システムに対して、生成系 AI と言われる LLM の技術がどのように適用できるかを検討した。OPAC への LLM 技術の適用は、既存の OPAC の検索システム自体は大きく変更することなく行える部分も多い。図書館が、OPAC という形で、メタデータ情報 を整備してきたことは、まったく無駄にならず、LLM 技術を使った、質問応答などの基盤として活用することができる。

もちろん、さらに LLM 可読な、あるいは埋め込み可能で、学習可能なメタデータに改善していく必要もあるが、図書館の機械化、と言われていた時代から数えれば、図書館が半世紀以上にわたって構築してきた電子的な書誌データが、生成 AI や LLM にとっても重要なデータ基盤になる。

また、LLM が OPAC にも活用されることによって、従来、ともすれば、検索式の作成や検索語の選び方などに重点が置かれていた図書館的な情報リテラシーの考え方を問い直していくと考えている。

参考文献

[1] “情報オーバーロード.” Wikipedia.org, https://ja.wikipedia.org/wiki/%E6%83%85%E5%A0%B1%E3%82%AA%E3%83%BC%E3%83%90%E3%83%BC%E3%83%AD%E3%83%BC%E3%83%89. Accessed 13 Apr. 2024.

[2] “科学技術指標 2020・Html 版 — 科学技術・学術政策研究所 (NISTEP).” Nistep.go.jp, 2020, https://www.nistep.go.jp/sti_indicator/2020/RM295_41.html. Accessed 13 Apr. 2024.

[3] Zhu, Yutao, et al. “Large Language Models for Information Retrieval: A Survey.” ArXiv.org, 2023, https://arxiv.org/abs/2308.07107. Accessed 13 Apr. 2024. なお、以下に簡単な紹介がある。“情報検索のための大規模言語モデル.” Qiita, 18 Sept. 2023, https://qiita.com/wonox/items/d0483b495129f722d0dc. Accessed 24 Mar. 2024.

[4] Ai, Qingyao, et al. “Information Retrieval Meets Large Language Models: A Strategic Report from Chinese IR Community.” AI Open, vol. 4, Elsevier BV, Jan. 2023, pp. 80–90, https://doi.org/10.1016/j.aiopen.2023.08.001. Accessed 4 Nov. 2023.

[5] 図書館情報学用語辞典 第 5 版. “検索過程 (けんさくかてい) とは？ 意味や使い方 - コトバンク.” コトバンク, 2014, https://kotobank.jp/word/%E6%A4%9C%E7%B4%A2%E9%81%8E%E7%A8%8B-1702624. Accessed 13 Apr. 2024.

[6] 宮尾 祐介. “大規模言語モデルの原理と可能性”, https://edx.nii.ac.jp/lecture/20230512-04. Accessed 13 Apr. 2024.

[7] “RAG とは何ですか? - 検索拡張生成の説明 - AWS.” Amazon Web Services, Inc., 2023, https://aws.amazon.com/jp/what-is/retrieval-augmented-generation/. Accessed 17 Mar. 2024.

[8] ”「2.1.5 データサイエンスに基づく問題解決」に関するコラム — 生成 AI — “情報基礎教育ポータルサイト.”, 2024, https://fie.tsuyama-ct.ac.jp/. Accessed 17 Mar. 2024.”

[9] CiNii の結果をベクトルデータベースに取り込む実験をしたことがある。“CiNii の DocumentLoader からベクトルデータベース Chroma にロードし近傍探索してみる（附 Embedding Model の比較）.” Qiita, 8 Jan. 2024, https://qiita.com/wonox/items/6d90674e7664a5cd3ce2. Accessed 21 Apr. 2024.

[10] Sun, Weiwei, et al. “Learning to Tokenize for Generative Retrieval.” ArXiv.org, 2023, https://arxiv.org/abs/2304.04171. Accessed 17 Dec. 2023.

[11] Li, Yongqi, et al. “Learning to Rank in Generative Retrieval.” ArXiv.org, 2023, https://arxiv.org/abs/2306.15222. Accessed 17 Dec. 2023.

[12] Ovadia, Oded, et al. “Fine-Tuning or Retrieval? Comparing Knowledge Injection in LLMs.” ArXiv.org, 2023, https://arxiv.org/abs/2312.05934. Accessed 30 Dec. 2023.

[13] Freedom Preetham. “RAGs Do Not Reduce Hallucinations in LLMs — Math Deep Dive” Feb, 2024, 16 Feb. 2024, https://medium.com/autonomous-agents/rag-does-not-reduce-hallucinations-in-llms-math-deep-dive- Accessed 21 Apr. 2024.

[14] WALLIS, Richard. ”MARC and beyond: our three Linked Data choices.” Paper presented at: IFLA WLIC 2018 – Kuala Lumpur, Malaysia - Transform Libraries, Transform Societies in Session 113 - Information Technology.2018, https://library.ifla.org/id/eprint/2124/. Accessed 21 Apr. 2024.

[15] “Overview of the BIBFRAME 2.0 Model (BIBFRAME - Bibliographic Framework Initiative, Library of Congress).” Loc.gov, 2016, https://www.loc.gov/bibframe/docs/bibframe2-model.html. Accessed 17 Mar. 2024.

[16] “Guides: Penn Libraries Linked Data Framework: Appendix: Linked Data and Other Formats.” Upenn.edu, 2024, https://guides.library.upenn.edu/c.php?g=1278641&p=9424728. Accessed 17 Mar. 2024.

[17] 江上 周作, and 福田 賢一郎. “大規模言語モデルを用いた SPARQL クエリ生成の予備的実験.” 人工知能学会第二種研究会資料, vol. 2023, no. SWO-060, 一般社団法人 人工知能学会, Aug. 2023, p. 04, https://doi.org/10.11517/jsaisigtwo.2023.SWO-060_04. Accessed 4 Nov. 2023.

[18] Berners-Lee, Tim. “The Semantic Web.” W3.org, 2024, https://www.w3.org/2000/Talks/0906-xmlweb-tbl/text.htm. Accessed 24 Mar. 2024.

[19] 次田瞬. “意味がわかる AI 入門 : 自然言語処理をめぐる哲学の挑戦.” 筑摩書房, 2023.11.

[20] Kumar, Selva. “Semantic Search with ElasticSearch - GoPenAI.” Medium, GoPenAI, 26 Sept. 2023, https://blog.gopenai.com/semantic-search-with-elasticsearch-1dab248d116f. Accessed 14 Apr. 2024.

[21] 高橋 和輝, “Cognitive Search を使ったベクトル検索のメリットとは？ ChatGPT システムと連携したデモで解説！”, 4 Dec. 2023, https://cloud.nissho-ele.co.jp/blog/cognitive-search-vector-search/. Accessed 24 Mar. 2024.

[22] 以下は、ベクトル検索の先駆的な紹介である。岸田 和明. “情報検索の発展過程と新たな動き (情報検索の新潮流).” 情報の科学と技術, vol. 50, no. 1, 一般社団法人 情報科学技術協会, 2000, pp. 3–8, https://doi.org/10.18919/jkg.50.1_3. Accessed 22 Oct. 2023.

[23] LLM を使って検索結果の要約を行う実験をしたことがある。 “CiNii の検索結果を LLM でまとめる.”

Qiita, 10 Dec. 2023, https://qiita.com/wonox/items/76b5ffffc5ee31dbda64. Accessed 21 Apr. 2024.

[24] Shi, Xiang, et al. "Know Where to Go: Make LLM a Relevant, Responsible, and Trustworthy Searcher." ArXiv.org, 2023, https://arxiv.org/abs/2310.12443. Accessed 4 Nov. 2023.