

LLM による動的属性タグ生成に基づくヘルプデスク情報の構造化検索

和田拓也¹ 赤羽陵太¹ 堀浩史² 山口格² 傍示健太² 笹原康介² 植木一也¹

¹ 明星大学 情報学部, ² AMBL 株式会社

hiroshi.hori@ambl.co.jp, kazuya.ueki@meisei-u.ac.jp

概要

コールセンターやヘルプデスク業務における情報検索では、顧客の状況やデバイス環境など、文脈に依存した多岐にわたる属性情報を正確に考慮する必要がある。しかし、Retrieval-Augmented Generation (RAG) において、こうしたドメイン固有の厳密な属性を含む検索は、従来のベクトル類似度のみでは精度確保が困難である。本研究では、LLM を用いてテキストを動的に構造化し、属性タグに基づく検索を行う手法を提案する。また、本手法の評価基盤として、複雑な属性情報を含む擬似的なコールセンターログデータセットを構築した。本手法は、クエリから抽出した属性タグをデータベース側の構造化タグと照合することで、高精度なドキュメント特定を実現する。実験の結果、提案手法は Hit@1 においてベースラインを大幅に上回る精度を達成し、その有効性を示した。

1 はじめに

企業のコールセンターや社内ヘルプデスクにおいて、過去の対応履歴やマニュアルから適切な情報を迅速に検索することは、業務効率化や顧客満足度の向上に不可欠である。近年、こうした情報検索質問応答システムとして Retrieval-Augmented Generation (RAG) が活発に研究されており、質問文と回答をベクトル化し、コサイン類似度に基づく検索を行う手法が主流となっている。

しかし、ヘルプデスクのような複雑かつ文脈依存性の高いタスクにおいては、単なる意味的な類似検索だけでは不十分な場合が多い。例えば、「機種名」や「OS バージョン」、「エラーコード」といった厳密な属性情報が一致していなければ、解決策として機能しないためである。既存のベクトル検索では、質問の意図や文書の論理構造を十分に反映できず、

こうした属性の違いを厳密に区別できない課題がある。

そこで本研究では、文書の構造化情報を活用した新たな検索手法の有効性を検証する。具体的には、大規模言語モデル (LLM) を用いてコールセンターの擬似的な構造化データを生成し、既存手法のベクトル検索との比較実験を行った。データに明示的な構造と属性タグを与えることで、質問内容の階層性や属性情報をより精密に扱うことが可能となり、検索精度の向上が期待される。

2 関連研究

大規模言語モデル (LLM) は広範な知識を有しているが、学習データに含まれない最新情報や、企業内部のドメイン固有知識 (Private Knowledge) を正確に扱うことには限界がある。この課題に対し、Lewis ら [1] は、生成モデルに外部知識源からの検索機構を組み合わせる Retrieval-Augmented Generation (RAG) を提唱した。RAG は、パラメータ化されたメモリ (LLM) と非パラメータ化されたメモリ (検索インデックス) を結合することで、知識集約型タスクにおいて高い性能を示している。

RAG の検索フェーズにおいては、従来のキーワード検索に代わり、Dense Passage Retrieval (DPR) に代表されるベクトル検索が広く用いられている。Karpukhin ら [2] は、質問と文書を密なベクトル表現 (Dense Embeddings) に変換し、類似度計算を行うことで、単語の完全一致に依存しない文脈を考慮した検索が可能であることを示した。しかし、一般的なベクトル検索は意味的な類似性を捉えることには長けているものの、コールセンターのログデータに含まれるような「製品型番」や「エラーコード」といった厳密な属性情報の区別においては、必ずしも十分な精度を発揮できないという課題がある。文脈が類似していても対象となる属性が異なる場合、ベ

クトル空間上で近接してしまい、誤ったドキュメントを抽出するリスクが生じる。

一方で、非構造化テキストから特定の属性やエンティティを抽出するタスクにおいて、LLMの有用性が示されている。Agrawalら[3]は、LLMがわずかな例示を与えるだけで、医療文書から構造化データを高精度に抽出できることを明らかにした。このようにテキストデータの構造化自体は可能になりつつあるが、検索システムにおいて、クエリの文脈的な意味と、これらの厳密な属性情報をどのように統合して扱うかは自明ではない。特に、ドメイン知識が複雑に絡み合うタスクにおいて、LLMによる構造化能力を検索精度の向上に効果的に結びつける点については、依然として課題が残されている。

3 提案手法

本章では、構造化データを活用した検索システムの詳細について述べる。図1に提案手法の概要を示す。本手法は主に、擬似データの生成、および属性タグを用いた検索アルゴリズムの2段階で構成される。

3.1 擬似コールセンターデータの構築

本実験の評価にあたり、実運用環境を想定した情報システム部門の業務ログデータセットを構築した(データ構造の一例を付録の図A.1に示す)。データ生成にはOpenAIのGPT-5 mini¹⁾を用い、XML形式の構造化データとして出力した。

具体的には、架空の企業を設定し、その組織構造や業務フローに基づいたデータ生成を行った。各データには、テキストとしての問い合わせ内容に加え、検索精度向上に寄与する以下の属性タグ(ユーザー属性、デバイス情報、対応管理情報など)を付与した。

データの多様性と統一性を両立させるため、生成時には「ジャンル」と「サブトピック」を明示的に指定した。これにより、同一トピック内では類似した語彙や文脈を含みつつ、異なる属性値を持つデータ群が生成されるよう制御している。

3.2 構造化タグ検索手法

提案手法の中核は、ユーザーが入力する非構造化な自然言語クエリを、データベースのスキーマ(構造化定義)に基づいた構造化表現へと動的に変換する

1) <https://platform.openai.com/docs/models/gpt-5-mini>

プロセスにある。

本手法では、検索時に以下の手順でクエリの構造化をした。まず、ユーザーの質問文と、あらかじめ定義された属性タグのスキーマ(カテゴリ、優先度、デバイス情報などの定義)をLLMに入力する。LLMは質問文の意図を解釈し、その内容に最も合致すると予測される「検索用属性タグ」を自動生成する。例えば、「急に画面が映らなくなった」というクエリに対し、Category:Hardware, Status:Urgentといったタグを推論して生成する。

次に、この「生成された属性タグ」と、データベース内の各ドキュメントに付与されている「データベース側の構造化タグ」との間でマッチングをした。具体的には、生成タグ群とドキュメントタグ群のそれぞれを埋め込みモデルを用いてベクトル化し、コサイン類似度を算出する。これにより、キーワードの一致のみに依存する全文検索では捉えきれない、「緊急度」や「文脈的なカテゴリ」といった潜在的な属性情報を加味した、高精度な検索を実現する。

4 検証実験

4.1 検証条件

本研究で提案する構造化データを活用した検索手法の有効性を検証するため、既存手法との比較実験を行った。なお、本実験では提案手法の基礎的な検索能力を評価するため、構造化タグを用いた検索アルゴリズム単体での検証を行った。

比較対象としては、タグによる構造情報を破棄し、タグ内に含まれる質問および回答のテキスト文章を単一の文字列として結合し、ベクトル化する手法を採用した。実装にあたり、タグ生成モデルにはgpt-4.1²⁾を、ベクトル化モデルにはtext-embedding-3-large³⁾を使用した。

評価指標には、Hit@K, MRR (Mean Reciprocal Rank), および Average Rank (平均順位)を用いた。実験に用いる検索クエリは、図A.2のプロンプトで作成した。社内ヘルプデスクにおいて日常的に発生しうる問い合わせ内容を想定して生成した(生成されたクエリの例は図A.3を参照)。単なるキーワード検索との差異を明確にするため、部署名やデバイス情報などの属性条件を自然な口調で含むよう制約

2) <https://platform.openai.com/docs/models/gpt-4.1>

3) <https://platform.openai.com/docs/models/text-embedding-3-large>

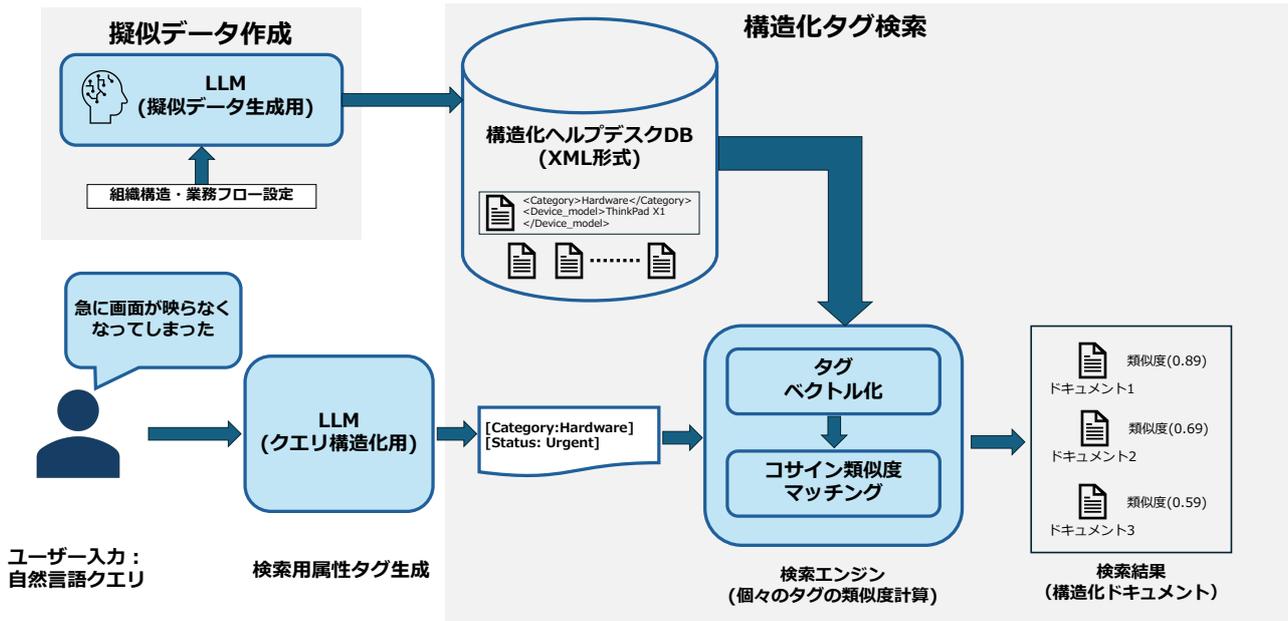


図 1 提案手法の概要図

を設け、実際の運用環境に近いデータセットを構築した。

4.2 検証結果

表 1 ベースライン（非構造）と提案手法（構造化）の検索精度比較

評価指標	ベースライン (非構造)	提案手法 (構造化)
Hit@1	0.6444	0.8889
Hit@3	0.8667	0.9167
Hit@5	0.9333	0.9306
Hit@10	1.0000	0.9722
MRR	0.7691	0.9119
AvgRank	1.9444	2.4444

表 1 に実験結果を示す。Hit@1 および Hit@3 において、提案手法はベースラインを大きく上回り、特に Hit@1 では約 24 ポイントの向上 (0.6444 から 0.8889) が見られた。これは、構造化タグを用いることで、ユーザーの意図に合致するドキュメントをピンポイントで上位に引き上げることができたためと考えられる。一方で、Hit@10 や Average Rank においてはベースラインがわずかに上回る結果となった。これは、提案手法がタグの一致を重視するため、タグ推論が外れた場合に順位が変動しやすい性質を持つ可能性を示唆している。しかし、MRR (0.9119) が示す通り、総合的なランキング品質としては提案手法が高い実用性を有しているといえる。

4.3 考察

高評価事例の分析 提案手法が有効に機能した事例として、表 2 に「勤怠システムのトラブル」に関する検索結果を示す。

この事例では、ユーザーの質問文の中に「Edge (ブラウザ名)」や「勤怠管理 SaaS (システム名)」といった具体的な属性情報が含まれている。ベースライン手法では、これらの単語が単なるテキストの一部として処理されるため、他の「Edge に関する別のトラブル」や「SaaS のログイン記事」などがノイズとして混入しやすい。

対して提案手法では、LLM が質問文からこれらの重要語句を認識し、Browser や System といったタグへ構造化して変換している。これにより、属性レベルでのフィルタリングが機能し、類似スコア 0.8411 という高い確信度で正解ドキュメントを特定できたと考えられる。

低評価事例の分析 一方で、表 3 に、検索精度が良くなかった事例を示す。

この事例では、ユーザーが「プリンタ管理」や「設定の全端末適用」について質問している。しかし、LLM は質問文の文脈（「全端末」などの表現）から、本来の検索対象ではない「ThinkPad X1 Carbon」という具体的な PC 機種名を誤って Model タグとして推論・生成してしまった。構造化検索はタグによる強力なフィルタリング効果を持つため、この誤った機種名タグが付与された時点で、本来ヒットすべ

表2 提案手法における高評価事例 (類似度: 0.8411)

項目	内容
事例	勤怠システムのセッションエラー
検索クエリ	ThinkPad X1 Carbon... Edgeで勤怠管理 SaaSに出勤打刻しようとする『セッションエラー(無効なセッション)』が表示されて打刻できません...
抽出タグ	Browser : Microsoft Edge Situation : セッションエラー (無効なセッション) が表示され打刻できない System : 勤怠管理 SaaS

き「プリンタ複合機」に関するドキュメントが検索範囲から除外されてしまったと考えられる。

また、Details タグにおいて、「Ricoh と Canon 両方の...」といった長文かつ複雑な文章が生成されている点も課題である。タグが長文化することでベクトル表現が拡散し、ピンポイントな属性マッチングという構造化データの利点が損なわれた可能性がある。このことから、構造化検索においては、タグ生成の誤りが検索結果の消失に直結するリスクがあり、LLM のハルシネーション抑制やタグ生成精度の向上が重要な課題であることが確認された。

表3 提案手法における低評価事例 (類似度: 0.6393)

項目	内容
事例	プリンタ設定 vs USB ドックの不具合
質問文	... 物流本部課長です。プリンタ管理をしています... 会社標準の両面・製本・中綴じステープル設定を全端末に適用・維持する手順は...
抽出タグ	Model : ThinkPad X1 Carbon Category : プリンタ設定 Details : Ricoh と Canon 両方のプリンタでブックレット+両面+中綴じステープル設定を全社で強制的に...

5 まとめと今後の課題

本研究では、コールセンターのログデータを対象に、LLM を用いて文書およびクエリを構造化し、属性タグを活用する新たな検索手法を提案した。評価実験の結果、提案手法は従来の非構造化データを用いたベクトル検索と比較して、Hit@1 および MRR において大幅な精度向上を達成し、ユーザーの意図を汲み取った高精度な検索が可能であることを示した。

今後の課題として、タグの重み付け手法の改善が挙げられる。現状の提案手法では、生成された各タグのベクトル類似度を単純平均しており、全てのタ

グを一律の重みで扱っている。しかし、実際の問い合わせにおいては、特定のタグ (例えば「機種名」や「エラーコード」) が他のタグ (例えば「拠点」) よりも解決において重要な意味を持つ場合が多い。

そこで今後は、Graph Neural Networks (GNN) の導入を検討する。タグとドキュメントの関係性をグラフ構造としてモデル化し、GNN を用いて各タグの重要度を学習させることで、クエリの特長や文脈に応じた動的かつ最適な重み付けを実現し、さらなる検索精度の向上を目指す。

参考文献

- [1] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In **Advances in Neural Information Processing Systems (NeurIPS)**, Vol. 33, pp. 9459–9474, 2020.
- [2] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau 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, 2020.
- [3] Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. Large language models are few-shot clinical information extractors. In **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 1998–2022, 2022.

A 付録

```
<kcsEntry>
  <question>パスワード変更後に OneDrive 同期停止。「サインインが必要」と繰り返される。ローカルファイルを保持したまま、クライアント側の資格情報リセットと再サインインを行う手順は？</question>
  <conditions>
    <user>
      <department>経営企画室</department>
      <location>東京本社</location>
    </user>
    <device>
      <model>ThinkPad X1 Carbon</model>
      <os>Windows 11 23H2</os>
    </device>
    <system>
      <name>Microsoft 365 - OneDrive</name>
    </system>
  </conditions>
  <answer>要点: ローカルファイルを保護しつつ、リンク解除・キャッシュリセット・資格情報削除を行う。
  手順:
  1. OneDrive リンク解除 (ファイルは保持)。
  2. コマンド「OneDrive.exe /reset」実行。
  3. 資格情報マネージャから OneDrive/Office 関連を削除。
  4. PC 再起動後に再サインイン。
  管理側: Azure AD で該当ユーザーのセッション取り消し (Revoke sessions) を実施。
</answer>
  <responseinfo>
    <category>OneDrive 同期</category>
    <priority>中</priority>
    <status>解決済み</status>
  </responseinfo>
</kcsEntry>
```

図 A.1 構築したデータセットの構造例 (複数カテゴリの抜粋)

あなたは日本語の社内 IT ヘルプデスクの教育データを作成するアシスタントです。与えられた KCS 記事 (質問・回答・条件情報) にもとづいて、実際のエンドユーザーが最初に問い合わせそなう自然な質問文を作ってください。必ず、可能な限り以下のような条件情報をクエリ文の中に自然な形で含めてください:

- ・ 部署名、役職
- ・ 勤務拠点 (例: 東京本社、ニューヨークオフィスなど)
- ・ 端末種別 (PC / iPhone / Android など)、機種名、OS バージョン
- ・ ブラウザ名 (例: Safari, Chrome など)
- ・ システム名 (例: Azure AD, Intune, VPN 名 など)

各質問文には、少なくとも 2 つ以上の条件情報が含まれるようにしてください。

その他の要件:

- ・ 社内ユーザーがチャット/メール/電話で伝えそうな口調で、丁寧だが長すぎない文にする
- ・ 状況説明や困りごとを含めてもよいが、KCS の回答全文をなぞらない
- ・ 「KCS」や「ナレッジ」などメタな単語は使わない
- ・ 同じ意味の言い換えになりすぎないように、バリエーションをつける

出力は JSON 配列 (文字列の配列) のみとし、余計なテキストは出力しないでください。

図 A.2 検証の際に使用した検索クエリ作成プロンプト

・ MacBook Air (M2)・Safari で社内ポータルを利用中、ヘッダーの検索ボックスがクリックできず透明になっています。拡張機能や設定の問題かとも思い色々試しましたが解決できません。対策をご教示ください。
・ ノート PC (Windows 11 23H2) から printsrv01.toa.co.jp 経由で Ricoh プリンタに印刷するとキューが詰まり、以降の印刷も全て止まってしまいます。プリンタ管理システムでドライバー配布状況を確認しましたが、v4 利用中です。恒久的に発生しないようにするにはどうしたらよいでしょうか？

図 A.3 生成された検索クエリの例