オンプレミス型生成 AI による電子カルテ要約支援チャットボットの試行的開発
Gemma3 ＋ RAG による閉域運用設計とその評価

梶原　晃[1]
池田治彦[2]

0．要約

本研究は、医療情報を外部送信せず院内 LAN で運用可能な電子カルテ要約支援チャットボットを、オンプレミス環境で試行的に開発・評価したものである[3]。具体的には、Windows 11 Pro + GPU 搭載 PC 上で Gemma3 をコンテナ稼働させ、電子カルテダミーデータ（5 年以上、約 1.8 万字、200 件超）を RAG で検索し、根拠に基づく要約生成と質疑応答システムを実装した。その結果、短文指示により 500 ～ 1000 字の要約を短時間で生成し、追加質問にも短時間で応答した。臨床経験を有する専門職による定性的評価により臨床現場における情報把握の迅速化に寄与する実用的な精度を有していることが確認された一方、期間・キーワード等の抽出条件制御、および CPU/ メモリ処理の最適化が課題であり、基盤の Linux 化等を今後検討することとなった。

1．背景

医療現場では、診療録、検査結果、看護記録、サマリー等の医療文書が日々大量に生成され、臨床判断に必要な情報を短時間で抽出・整理する作業が常態化している。特に入退院や転科、救急受入れ、カンファレンス準備などの局面では、過去経過の把握と要点の要約が不可欠である一方、医師・看護師等の専門職が限られた時間で膨大な記録を参照する負担は大きい。こうした背景から、医療文書の要約や要点抽出を支援する情報技術の導入が期待されている。

1　久留米大学　文学部情報社会学科・医学部医療経営研究センター（兼担）
2　株式会社ライクブルー代表取締役
3　オンプレミス型生成 AI 導入による病院内業務の効率化については、梶原（2025）で基本的な論点整理を行った。

しかし、生成 AI をクラウドで利用する場合、入力データが院外へ送信される可能性や第三者環境での処理・保管に伴うリスクが懸念となる。電子カルテ情報は極めて機微性が高く、個人情報保護や院内規程、委託先管理等の観点から、外部送信を前提とする運用には制約が生じやすい。結果として、院内ネットワークに閉じた形で AI を稼働させ、データの所在とアクセス制御を病院側が直接管理できるオンプレミス型の需要が高まっているのである。

一方で、生成 AI は流暢な出力を生成できる反面、事実に基づかない内容をもっともらしく提示する「ハルシネーション」を生じ得る。医療領域では、誤った要約や不正確な推論が診療の安全性に影響しうるため、単に自然な文章を生成するだけでは不十分であり、根拠を明確にした出力統制が重要である。そこで注目されるのが、参照すべき文書を検索し、その検索結果を根拠として回答・要約を生成する RAG（Retrieval-Augmented Generation）である。RAG は、院内の電子カルテ記録等に基づいて生成範囲を限定し、モデルの内部知識への過度な依存を抑制できる点で、医療で求められる説明可能性と安全性の要請に適合しやすい。本研究は以上の課題意識に立ち、閉域オンプレミス環境で RAG を用いた要約支援チャットボットを試作し、実装可能性と課題を検討するものである。

２．目的

本研究の目的は、医療機関が保有する機微性の高い診療情報を院外へ送信することなく、院内ネットワークに閉じた形で生成 AI を活用し、電子カルテ情報の参照・要約業務を支援し得る技術基盤を試行的に構築し、その実装可能性と課題を明らかにすることにある。本目的は以下の二段階で定義する。

第一の目的は、外部ネットワーク（インターネット）からは論理的および物理的に隔離された院内専用 LAN 環境において大規模言語モデル（LLM）を、実運用を想定した形で稼働させる実装可能性を確認することである。具体的には、外部通信を前提としない構成で、LLM の推論処理、アプリケーション層、データ検索層を組み合わせ、院内端末からの利用を成立させる。加えて、処理時間や資源消費（CPU、GPU、メモリ、ストレージ）、ネットワーク要件、導入手順の再現性を把握し、病院内で自律的に維持可能なシステム像を描くことを狙う。ここでは「院内データを院内で処理する」ことを技術要件として具体化し、閉域運用に必要な前提条件と構成要素を整理する。

第二の目的は、電子カルテ要約支援という具体的ユースケースに対し、当該構成がどの程度有効であるかを検証するとともに、実装・運用の両面から課題を抽出することである。技術面では、長期にわたる記録から要点を抽出する際の検索・抽出条件（期間、キーワード、重要度等）の設計、生成結果の安定性、応答時間、ハルシネーション抑制といった論

点で評価する。さらに運用面では、利用者が短い指示で実務的な要約を得るためのプロンプト設計やテンプレート化、評価・精査体制（医療専門職による確認）、アクセス制御や監査可能性、継続的改善の手順を検討し、院内導入に向けた実務的な要件を明確化する。以上を通じて、閉域オンプレミス型生成 AI による要約支援チャットボットが医療現場の負担軽減に寄与し得るか、またそのために必要な技術的・組織的条件は何かを示すことを本研究の目的とする。

３．対象データと前提条件

本研究で RAG の検索対象としたデータは、電子カルテシステム上に用意したダミーデータである。実運用データを直接扱う前段階として、院内情報の構造や記載様式を模したテキスト群を参照基盤に設定し、ユーザ入力に応じて関連箇所を検索・抽出した上で、LLM にコンテキストとして付与し要約・回答を生成する流れを構成した。ダミーデータの規模は、200 以上のレコード（データ行）と約１万８千字のテキストからなり、診察日付は５年以上にわたる長期間の記録を含む設定である。これにより、長期経過を含むカルテ要約という実務に近い負荷条件を、個人情報を含まない形で再現し、要約品質と検索設計上の課題（期間限定やキーワード抽出の必要性等）を検討可能とした。

前提条件として、本システムは院内 LAN 環境のみに接続し、LLM はローカル内で処理を完結させ、外部との通信は一切行わない閉域構成とした。LAN 内から接続した院内端末はチャットボットを利用可能であり、医療情報の院外送信を伴わずに生成 AI の有用性を検証できる設計である。また、導入の思想としては、まず研究・教育用途などリスクを制御しやすい領域から試行し、評価と改善を重ねた上で、より実務に近い運用へ段階的に拡張する方針を採るものである。

４．システム設計

本研究のシステム設計は「既存設備の活用」と「閉域での安全運用」を両立させつつ、電子カルテ要約支援を RAG で実現することを基本方針としたものである。以下に構成要素を示す。

４．１　ハード／ソフト構成

ハードウェアは、既存 PC を基盤に OS を Windows 11 Pro とし、RAM32GB、AMD Ryzen 9 5900X（12-Core）を搭載した。GPU は NVIDIA GeForce RTX 5090 へ換装し、電源ユニットも更新することで推論処理の余力を確保した。

ソフトウェア面では、LLM として Gemma3 を採用した [4]。Gemma3 はマルチモーダル対応であり、最大 128,000 トークンの長文入力を扱え、多言語対応にも優れることから、長期経過を含む医療文書を対象とした要約・対話処理に適すると判断した。

4．2　LLM 実行基盤（コンテナ化）

LLM は Windows 上でコンテナとして稼働させ、同一マシン内に構築したチャットボットアプリケーションから呼び出す方式とした。これにより、LLM 実行環境とアプリケーション層を論理的に分離し、将来的なモデル差替えや運用更新の影響範囲を限定しつつ、外部通信を行わない閉域構成を担保した。

4．3　RAG 設計（処理フロー）

生成結果の信頼性向上のため、検索拡張生成（RAG）を中核に据えた。処理フローは、（1）ユーザが要約・質問を入力し、（2）入力内容に基づいて電子カルテのダミーデータから関連箇所を検索、（3）抽出テキストを LLM のコンテキストとして付与、（4）LLM が検索結果を根拠として回答・要約を生成、の 4 段階で構成した。RAG の採用により、LLM が内部知識のみで回答することを抑制し、院内データに基づく生成を促す設計とした。

4．4　プロンプト設計・チューニング

医療現場では、出力の「流暢さ」よりも「根拠整合性」と「一貫性」が重要である。このため、プロンプト設計原則として、(a) 検索結果に含まれる情報のみを用いて回答する、(b) 不足情報がある場合は推測せず不足を明示する、(c) 主訴・経過・所見等の医療文書として自然な構成を意識する、の 3 点を明文化した。さらに、短文命令（例：「要約」）や追加質問に対しても品質を維持するため、生成プロセス全体を通じてプロンプト調整（チューニング）を反復し、ばらつき低減を図った。

これまで得た知見として、品質安定化には出力形式・記載順序・情報範囲を具体的に指示すること、略語・単位・日付形式等の表記ルールをプロンプト内で標準化することが有効である。また、診療科ごとにテンプレートを用意し選択可能とすることで、現場ニーズへの適合を高め得る。必要に応じて推論過程の出力をログとして分離保存する設計も検討対象となる。

4　Gemma3 は、Google が 2025 年 3 月に発表した Gemini 2.0 技術を基盤とする第 3 世代の軽量・高性能なオープン AI モデル。

４．５　ネットワーク／セキュリティ設計

ネットワークは院内 LAN のみに接続し、インターネットを含む外部ネットワークとの通信は一切行わない。LLM、検索データ、生成結果はローカル環境内で完結して処理され、LAN 内端末からブラウザ経由でチャットボットに接続できる構成とした。これにより、医療情報を外部送信せずに組織内で共有・活用可能とした。

本研究の設計思想として、セキュリティは後付けの制約ではなく導入の前提条件（by design）として位置づけており、導入段階から技術的・組織的対策を包括的に設計すべきであると考える。具体的には、外部アクセス遮断を基本としつつ、通信暗号化、認証・認可によるアクセス制御、さらに法令・指針の更新を踏まえた継続的見直しと監査対応・改善の枠組みを、運用設計として組み込むことが重要となる。

５．試行的作成プロセス

本章では、オンプレミス型生成 AI を活用した電子カルテ要約支援チャットボットの試行的作成プロセスを、他施設・他担当者でも追試可能な粒度で記述する。全工程は「閉域運用を前提とした要件定義」から開始し、「実行基盤」「RAG パイプライン」「プロンプト設計」「試験入力」「評価」を経て、プロンプトと評価を往復する改善サイクルとして収束させる構成である。

（工程１）要件整理（閉域・ダミー・要約タスク）

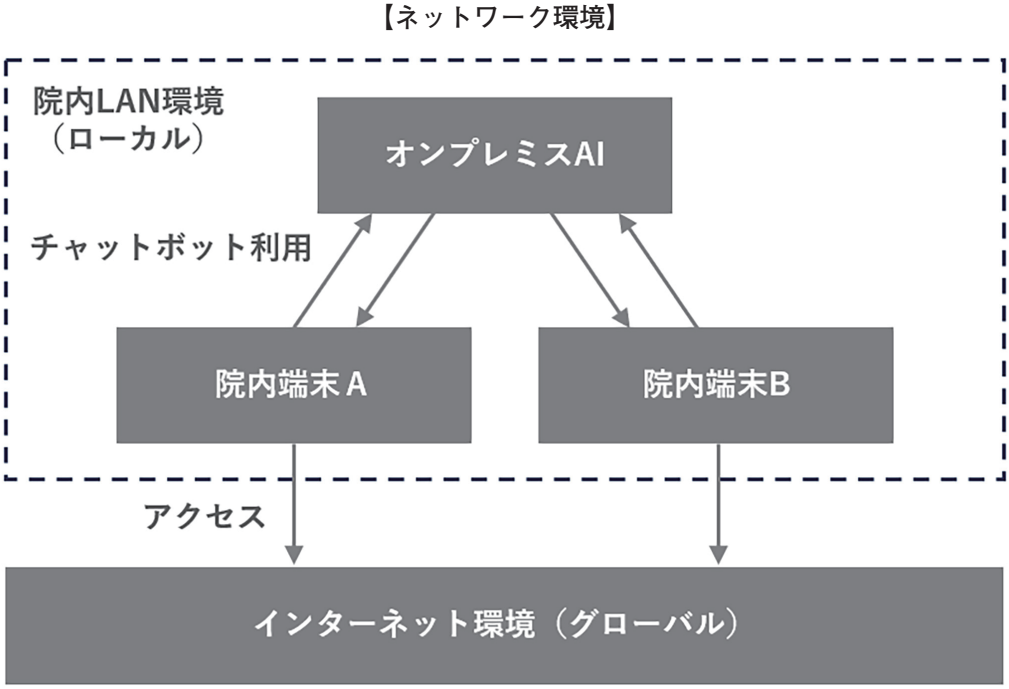
最初に、臨床情報の院外送信を行わないことを必須要件として定義した。すなわち、院内 LAN のみに接続し、インターネットを含む外部通信を遮断した閉域環境で、生成・検索・ログ保存までを完結させる。次に、実データを用いず電子カルテのダミーデータを対象とすることを明確化し、個人情報リスクを回避しつつ、記載様式や長期経過の参照という実務負荷に近い条件を再現する方針とした。タスクは「短文命令でカルテ要約を作成し、追質問に対して要点を補足する」ことに設定し、要約の想定出力長（例：500 ～ 1000 字）と許容応答時間（例：概ね 1 分以内）を暫定目標として置いた。

【カルテ要約 AI チャットボットの応答プロセス】

（工程２）実行基盤構築（GPU 換装、コンテナ準備）

次に、既存 PC を用いてオンプレミス推論が成立するかを検証するため、OS を

Windows 11 Pro とし、CPU（Ryzen 9 5900X）・RAM（32GB）を前提に GPU を高性能品へ換装した。ここでの狙いは、推論処理を GPU に寄せ、応答時間を実用域に近づけることである。LLM 実行環境はコンテナとして構築し、モデル周辺（依存ライブラリ、ランタイム、設定）をアプリ層から切り離した。これにより、(a) 環境差異の吸収、(b) モデル差替え時の影響局所化、(c) 閉域での一貫した実行状態の維持、を可能にした。コンテナ稼働後、ローカル推論の疎通（モデルのロード、簡単な対話応答、メモリ消費の確認）をチェックし、常時稼働が可能な状態まで整備した。



（工程３）RAG パイプライン構築（データ整形→検索→投入）

生成 AI の安全性・説明可能性を高めるため、RAG を中核に据えた。まずダミーデータを「検索可能な単位」に整形する。具体的には、レコード単位（例：日付、診療イベント、所見、検査、処方など）に分割し、検索時に参照すべきメタ情報（日時、カテゴリ、キーワード）を付与する。次に、ユーザ入力を受けて関連レコードを検索し、抽出テキストをコンテキストとして LLM に投入する流れを実装した。処理は「入力→検索→抽出→根拠ベース生成」の４段階とし、LLM には“検索結果が根拠である”ことを明示する。これにより、モデル内部知識に依存した推測的生成を抑制し、回答の範囲を院内データに拘束する設計とした。

【AI 要約チャットボットの構成】

（工程４）プロンプトテンプレート設計→チューニング反復

実務での安定稼働には、プロンプトを場当たり的に調整するのではなく、テンプレートとして固定し、改善を差分管理することが重要である。そこで、設計原則として①検索結果のみを用いる、②不足は推測しない（不足を明示する）、③医療文書として自然な構造（例：主訴／現病歴／経過／検査／評価／方針）でまとめる、をテンプレート内に明文化した。さらに、出力形式（見出し、箇条書き可否、字数目安）、記載順序、対象範囲（全期間か直近〇か月か等）を指示として固定し、表記（略語、単位、日付）を標準化した。診療科ごとの差異が大きい場合には、科別テンプレートの用意を想定し、最低限の共通骨格と可変部（評価観点、重要項目）を分離する設計とした。チューニングは、同一入力に対する出力のばらつき、不要な推測の混入、長文化・短文化の偏りを観察し、テンプレートの語尾・禁止事項・優先順位表現を段階的に修正する方法で実施した。

（工程５）試験入力（「要約」等の短文命令＋追加質問）

試験は、利用者の操作負担を最小化する観点から、短文命令（例：「要約」）でも一定品質を保てるかを中心に行った。次に、追加質問（例：「直近の検査異常は何か」「治療方針の変遷は」等）を投入し、（a）検索が追従して根拠が更新されるか、（b）同一患者経過の整合性が保たれるか、（c）回答がカルテ記載から逸脱しないか、を確認した。ここで重要なのは、追加質問を“会話の流れ”として処理する一方、根拠は毎回 RAG で再取得させ、前回出力の誤りが連鎖しないようにする点である。

（工程６）評価（時間・内容妥当性）

評価は定量と定性の二本立てとした。定量評価では、要約生成および追加質問の応答時間、要約文字量、検索件数やコンテキスト量を記録し、性能ボトルネック（CPU ／メモリ／ I/O）を推定した。定性評価として臨床経験を有する医師による出力内容の妥当性検証を行い、事実整合性、重要事項の抜け、表現の自然さ、誤解を招く推測の有無を観点として判定した。併せて、長期データを一括要約した際に「期間やキーワードによる抽出条

件がないと要点が平均化する」などの課題を整理し、改善要求として明文化した。

改善サイクル（工程（4）↔（6））

本試行の中心は、プロンプトテンプレート（工程４）と評価（工程６）を往復する改善サイクルである。評価で抽出された問題（推測混入、重要情報の欠落、要約構造の崩れ、抽出範囲の曖昧さ、応答遅延）を、①テンプレート修正（禁止・優先・形式の強化）、②検索条件の追加（期間指定、キーワード、カテゴリ重み付け）、③運用ルール整備（利用目的、確認手順、ログ管理）に分解し、技術的・運用的介入の容易な項目から優先的に適用して、段階的に精度向上を図った。各改修後は同一試験入力セットで再評価し、改善の有無を比較する。これにより、技術面（RAG･性能･出力統制）と運用面（検証体制･ルール・継続改善）を一体として整備し、閉域オンプレミスに適した実装プロセスを再現可能な形で確立することを目指した。

６．評価方法

本研究における評価は、（１）性能（応答時間・出力量）と（２）内容妥当性（臨床文書としての品質）の二軸で実施した。

まず性能評価では、同一条件下で「要約」指示および追加質問を入力し、ユーザ入力から最終出力が得られるまでの所要時間を計測した。評価対象データは、５年以上の記録を含む電子カルテのダミーデータ（約 1.8 万字、200 件超）であり、長期経過を一括して参照する負荷条件を前提とした。要約タスクは短文命令（例：「要約」）で起動し、生成された要約の文字量を記録した。性能指標としては、要約の応答時間を概ね１分程度、追加質問に対する応答時間も１分以内を目標水準とし、実測値がこの範囲に収まるかを確認した。併せて、要約長については実務で可読な分量として 500 ～ 1000 字を目安に設定し、指示のばらつきに対する出力安定性（過度な長文化・短文化の発生有無）を観察した。これらの計測は、閉域オンプレミス環境での実用性を判断するための基礎データとして位置づけた。

内容妥当性の評価は、医療専門職による精査を中心とした枠組みを採用した。具体的には､臨床経験を有する医師が生成された要約および回答を確認し､電子カルテ記載（ダミーデータ）との整合性と臨床文書としての適切性を評価した。評価観点は、①正確性（事実誤認、取り違え、推測の混入がないか）、②網羅性（重要なイベント・検査・治療方針の抜けがないか）、③表現（専門用語の適切さ、誤解を招く曖昧表現の有無、簡潔性）、④構造（時系列整理、見出し構成、要点の優先順位付け）を基本項目として明文化し、各観点に基づき総合判断を行った。さらに、追加質問への応答については、質問意図に沿った根拠提示の一貫性、前回答との矛盾の有無、検索結果に依拠した回答になっているかを補助

観点として確認した。以上の枠組みにより、生成結果が「実務レベルで十分」と判断できるかを評価し、同時に改善が必要な論点（抽出条件制御や出力統制の強化等）を抽出することとした。

7．結果

本章では、閉域オンプレミス環境における電子カルテ要約支援チャットボットの試行結果を、成果と課題に分けて示す。対象は電子カルテのダミーデータ（５年以上の記録、約 1.8 万字、200 件超）であり、RAG により関連記録を検索した上で Gemma3 が要約・対話応答を生成する構成で評価した。

成果として、院内 LAN に限定し外部通信を行わない閉域構成において、要約および追加質問への対話応答が一連の流れとして成立した。短文命令（例：「要約」）で要約生成が起動し、500 ～ 1000 字程度の要約を概ね１分で出力でき、同一データに対する追加質問にも概ね１分以内で応答した。本試行で確認された『１分程度での要約生成』という性能は、先行事例（那須赤十字病院）で示された『１症例あたり約 50% の時間削減』を達成するための十分な技術的裏付けとなる[5]。また、生成内容については臨床経験を有する医師による出力内容の妥当性検証を行い、正確性・網羅性・表現・構造の観点から、情報把握の迅速化と診療の質を維持するための補助ツールとしての実用性が示唆された。以上より、機微情報を外部送信せずに生成 AI を活用するという要請に対し、オンプレミス RAG チャットボットが現実的な選択肢となり得る可能性を確認した。

一方、主要課題も明確となった。第一に、抽出内容の選択に関する課題である。長期間（５年以上）の記録を一括で要約すると、重要事項が平均化され、臨床上の焦点（直近の状態変化、重要イベント、治療方針転換点等）が埋もれやすい。これは要約アルゴリズムが全期間を均等に扱ってしまうことによる「焦点の欠如」による現象である。モデルのコンテキスト長（128k トークン）は５年分のデータ（約 1.5 万トークン相当）を完全に収容可能であるため、これはリソース不足ではない。むしろ、RAG による検索結果を LLM がフラットに（時間的優先順位なしに）処理してしまうため、プロンプトに時間軸の重み付けが含まれていないという論理設計上の課題であると考えられる。要約対象の期間指定、特定の病名･薬剤･検査値等に基づくキーワード指定、診療イベントの重み付けなど、検索・抽出条件をユーザが制御できる設計を考慮するとともに、要約戦略（プロンプトやフィルタリング）の検討も不可欠である。

第二に、性能面では CPU/ メモリがボトルネックとなる場面が観察された。これは、GPU 推論自体には余力がある一方で、データ検索･整形、コンテキスト生成、アプリケー

5　梶原（2025）

ション処理などチャットボット稼働時におけるテキスト出力の逐次処理（トークナイズやストリーミング生成）および RAG のベクトル検索プロセスが Windows 環境下でのリソース競合を引き起こしている可能性を示唆している。こうした「推論以外」の負荷が相対的に支配的となり、特に Windows 環境を介した運用（Windows 11 上のコンテナ実行によるオーバーヘッド等）が全体の効率に影響し得る。このため、基盤の最適化（処理分担、メモリ管理、I/O 効率化）に加え、必要に応じて Linux ベースでの運用形態も含めた再設計が、実務導入に向けた次段階の検討課題となった。

８．考察

８．１　有効性

本試行で示された最大の示唆は、短文指示（例：「要約」）のみで一定の品質をもつ要約が生成され、追加質問にも即時に追従できる点にある。臨床現場では、患者背景の把握、当直引継ぎ、救急受入れ、カンファレンス準備など、限られた時間で過去記録を横断的に参照する局面が多い。従来は、記録を時系列に読み解き、要点を手作業で抽出する負担が大きかったが、要約と対話を組み合わせることで「まず全体像を俯瞰し、必要箇所のみ深掘りする」という情報参照様式へ転換できる可能性がある。加えて、閉域オンプレミスで外部送信を伴わずに運用できることは、クラウド利用に慎重な医療機関でも導入検討を前進させる要因となる。もっとも、生成物は診療判断の代替ではなく参照支援であり、最終確認は原記録に立ち戻る運用を前提とすべきである。

８．２　技術的制約と改善余地

第一の制約は、長期間データを一括要約した際に重要情報が平均化され、臨床的な焦点が曖昧になり得る点である。これは、RAG による検索結果を LLM がフラットに（時間的優先順位なしに）処理してしまうことに起因する。その改善案として、(a) 期間指定 UI（直近 7 日／ 30 日／入院期間／任意期間）、(b) キーワード抽出（病名いし、薬剤、手技、検査異常、転帰イベント等の自動候補提示と選択）、(c) 重要度付け（「転科･手術･抗菌薬変更・急変・検査異常」を優先表示するルール、または診療科別の重み付け）を組み合わせ、検索・抽出段階で要約の材料を制御する設計が有効である。これにより、要約の一貫性を保ちつつ、利用者の意図（直近経過／治療方針変遷／検査トレンド等）に沿った要点提示が可能となる。

第二の制約は Windows OS を介した仮想化環境におけるリソース競合である。GPU 推論自体に余力があっても、検索・整形・コンテキスト生成など推論以外が CPU/ メモリに集中し、全体応答時間を押し上げてしまう。推論以外の前処理（検索・整形）が全体の応答時間に支配的な影響を及ぼしているからである。対策として、CPU 強化とメモリ増

設に加え、(a) Linux ベースでの直接運用によるオーバーヘッド低減、(b) 検索基盤と推論基盤の分離（処理分散、スケールアウト）、(c) キャッシュ戦略（頻出クエリや直近要約の再利用）、(d) コンテキスト圧縮（重要文抽出、重複削除）を検討すべきである。閉域運用を維持しつつ、負荷特性に合わせて「推論を速くする」だけでなく「前処理を軽くする」設計が鍵となる。

8.3 運用・ガバナンス

医療における生成 AI 導入は、技術実装と同程度に運用設計が重要である。倫理面では、バイアスや差別的表現、特定属性への不適切な推論が出力されないよう、禁止事項の明文化、テンプレート化、監査ログの整備を行う必要がある。教育研究利用から開始する場合でも、利用目的の限定、対象データの管理（ダミー・匿名化・アクセス権）、結果の取り扱い（臨床意思決定への直接利用禁止、引用ルール）を手続として整備することが望ましい。さらに、個人情報保護や医療情報システム安全管理等の関連法令・指針は更新され得るため、継続的モニタリングと運用ルールの改訂を「導入後の作業」ではなく「運用設計の一部」として組み込むべきである。すなわち、責任者、変更管理、定期点検、インシデント対応、利用者教育までを含むガバナンス枠組みを先に定義することが、閉域オンプレミスの利点（データ主権・統制可能性）を実効化する。

さらに、オンプレミス型は外部ネットワーク遮断時にも稼働可能であり、災害時の医療情報管理インフラとしての可用性（BCP）も有している点は、病院経営における導入価値がさらに多角的に評価されるプラス面ともなろう。

8.4 経済性

経済性は、初期導入コストとランニングコストを分けて整理する必要がある。導入コストには GPU サーバ（または高性能 GPU 搭載 PC）、ストレージ、冗長化（バックアップ、予備機）、院内ネットワーク・認証基盤との接続、導入支援（設計・構築・検証）が含まれる。ランニングコストには保守、電力・冷却、モデル・ソフトウェア更新、プロンプト／テンプレートの継続的チューニング、人材育成と運用工数が含まれる。効果測定は、要約・参照に要する時間削減を起点に、(a) 削減時間×人件費による金額換算、(b) 業務滞留の解消（当直引継ぎの短縮、カンファレンス準備時間の低減等）、(c) 品質面の副次効果（見落とし防止の支援、標準化）を指標化し、TCO（総保有コスト）と ROI（投資対効果）で評価する枠組みが妥当である。特にオンプレミスは初期投資が先行しやすいため、まずは効果測定がしやすい業務から適用し、利用量と改善サイクルに応じて段階的に拡張する戦略が、費用対効果の不確実性を低減する。

9．今後の課題と展望

本試行では、電子カルテダミーデータを対象に、閉域オンプレミス環境で RAG チャットボットが成立することを示した一方、臨床実装に向けては段階的に解くべき課題を明らかにした。第一に、実データ想定の評価設計である。個人情報保護と安全管理を前提に、匿名化・仮名化の範囲、アクセス権限、ログ保全、監査手続を整えた上で、実データに近い条件で正確性・網羅性・再現性を評価する枠組みが必要となる。評価観点は、本研究で用いた正確性・網羅性・表現・構造に加え、臨床上の重要イベントの拾い上げ、誤解を招く要約の抑制、利用者間の一貫性など、より高い安全要求に対応した指標へ拡張すべきである。

第二に、抽出条件の高度化への課題である。長期記録の要約では重要事項が平均化し得るため、期間指定、キーワード指定、診療イベントの重要度付け、診療科別テンプレートなどを組み合わせ、利用目的に応じて「何を要約するか」を制御できる設計が求められる。将来的には、検査値の推移や治療方針変更点などを半自動で検出し、要約の骨格を先に提示する機能も検討対象となる。

第三に、性能最適化の問題である。推論以外（検索・整形・コンテキスト生成）の負荷が全体の応答時間を左右するため、処理の分離・キャッシュ・コンテキスト圧縮、ならびに実行環境の最適化（CPU/ メモリ強化、Linux ベースでの運用、必要に応じた分散構成）を段階的に進める必要がある。

以上を踏まえ、展望としては、教育・研究用途から臨床実装へ至るロードマップを明確化することが重要である。初期導入は、効果測定が容易でリスクを制御しやすい業務（例：カンファレンス準備、退院サマリーの下書き支援、研究データ整理等）に限定し、時間削減や品質安定化を指標として ROI を検証する。その後、ガバナンス（責任体制、監査、改訂手順）と技術成熟（抽出制御・性能・テンプレート）の両輪を整えながら、限定的な臨床ユースケースへ拡張し、最終的には電子カルテ参照支援の標準機能として活用するとともに、オンプレミス型音声入力ツールとの連携による診察記録の自動構造化（SOAP 形式[6]等）などへの発展など、院内に定着させる段階的展開が現実的である。こうして、単なる要約ツールから、診療プロセス全体の自動化を支援する診療支援基盤への拡張が期待される。

10. 結論

本研究では、医療情報を外部送信しない「閉域オンプレミス」環境において、RAG（検

6　SOAP 形式とは、医療・看護・介護現場で患者情報を「S（主観的情報）」「O（客観的情報）」「A（評価）」「P（計画）」の 4 項目で構造化して記録する手法のこと。

索拡張生成）とプロンプト統制を組み合わせた電子カルテ要約支援チャットボットを試行的に開発し、その成立性と課題を検討した。電子カルテのダミーデータを検索対象とし、入力に応じて関連記録を抽出して LLM（Gemma3）へ根拠として付与することで、短文指示による要約生成と追加質問への対話応答が実用的な応答時間で実現可能であることを確認した。すなわち、「閉域オンプレ× RAG ×プロンプト統制」により、医療文書要約の支援機能を院内で完結して提供できる可能性が示唆された。

一方で、臨床実装に向けた主要課題も明確化された。第一に、長期記録の一括要約では重要情報が平均化しやすく、期間・キーワード等による抽出条件の制御設計が不可欠となる点である。第二に、GPU 推論以外の処理が CPU/ メモリ負荷となり得るため、検索・整形処理を含めた資源ボトルネックの解消と実行環境の最適化が求められる点である。第三に、倫理・法令・監査を含む運用ガバナンスを導入初期から設計し、継続的改善の枠組みとして組み込む必要がある点である。以上より、本試行はオンプレミス生成 AI の実装可能性を示すと同時に、実務導入に必要な技術要件と運用要件を具体化する基礎知見を提供したといえよう。

（参考文献）

梶原晃（2025）「オンプレミス型生成 AI 導入による病院内業務の効率化に関する研究」『医療経営と病院管理』第 7 号 31-46pp.