Title: A Comprehensive Evaluation of LLM Phenotyping Using Retrieval-Augmented Generation (RAG): Insights for RAG Optimization

Authors: Heekyong Park, PhD[1], Martin Rees[1], Nils Kruger, MD[2,3], Kenshiro Fuse, MD[4], Victor M. Castro, MS[1], Vivian Gainer, MS[1], Nich Wattanasin, MS[1], Barbara Benoit[1], Kavishwar B. Wagholikar, MBBS, PhD[1,3,5], Shawn N. Murphy, MD, PhD[1,3,5,6]

Affiliations: [1]Center of AI and Biomedical Informatics for the Learning Healthcare System, Mass General Brigham, Somerville, MA, USA;

[2]Division of Pharmacoepidemiology and Pharmacoeconomics, Department of Medicine, Brigham and Women's Hospital;

[3]Harvard Medical School;

[4]Harvard T.H. Chan School of Public Health;

[5]Laboratory of Computer Science, Massachusetts General Hospital;

[6]Department of Neurology, Massachusetts General Hospital, Boston, MA, USA

Corresponding author:

Heekyong Park, Ph.D.

Mass General Brigham

399 Revolution Drive,

Somerville, MA 02145, USA

hpark25@mgb.org

Keywords: LLM, Retrieval-Augmented Generation, Phenotyping, Evaluation, Optimization

ABSTRACT

Objective

ICD codes are commonly used to filter patient cohorts but may not accurately reflect disease presence. Furthermore, many health problems are recorded in unstructured clinical notes, complicating cohort discovery from EHR data. Existing computed phenotyping methods have limitations in identifying evolving disease patterns and incomplete modeling. This study explores the potential of LLMs, by evaluating GPT-4o's type II diabetes mellitus (T2DM) phenotyping ability using Retrieval-Augmented Generation (RAG).

Methods

A RAG system was built, leveraging 275 patients' entire notes. We performed total 336 experiments to study the sensitivity of RAG to various chunk sizes, the number of chunks, and prompts across seven embedding models. Then the effectiveness of GPT-4o in T2DM phenotyping was assessed using optimized RAG configurations, comparing with ICD code and PheNorm phenotype performance. Token usage was also evaluated.

Results

The results show that GPT-4o with optimized RAG significantly outperformed ICD-10 and PheNorm in sensitivity, NPV, and F1, although PPV and specificity need improvement. When used with general embedding models or a zero-shot prompt, the results showed better sensitivity, NPV, and F1-scores, while domain-specific models and a few-shot prompt excelled in specificity and PPV. Furthermore, RAG optimization allowed lower-ranked embedding models achieve reliable performance. Gte-Qwen2-1.5B-instruct and GatorTronS provided the highest performance in specific evaluation metrics at a substantially lower cost.

Conclusion

Optimized RAG configurations significantly enhanced key performance metrics compared to existing methods. This study provides valuable insights into optimal configurations and cost-effective embedding model choices, while identifying limitations such as ranking issues and contextual misinterpretation by LLM.

INTRODUCTION

International Classification of Diseases (ICD) codes are frequently used to filter patient cohorts for clinical research. However, as ICD codes are mainly recorded for billing purposes, they often do not accurately represent the patient's actual health conditions. Consequently, investigators are required to perform chart reviews to confirm target patients screened by ICD codes. This process is laborious and time consuming. In addition, many health problems are recorded only in unstructured notes. These challenges highlight the need for derived phenotypes not just based on coded data.

Mass General Brigham (MGB) has been servicing computed phenotypes to internal researchers through the *Research Patient Data Registry (RPDR)*[1,2] and the *Biobank Portal*[3]. These phenotypes were developed by various natural language processing and/or machine learning methods[4-10]. One limitation of current phenotyping methods is maintaining up-to-date performance, as cohort patterns such as medications evolve over time. Additionally, not all initially developed phenotypes were accurate resulting in some failing to meet service standards. The recent advancement of *Large Language Model (LLM)* technologies has the potential to address these issues. Recent studies have reported the use of LLMs to enhance phenotyping[11-16].

Retrieval-Augmented Generation (RAG)[17] is an effective framework that enhances LLM performance by integrating relevant results retrieved from local database in the LLM prompts. In the medical field, this approach has become popular[18-29]. The quality of RAG output is sensitive to its hyperparameters[30-33], but only few studies report RAG optimization effort in medical domain[18,34-36]. There are studies to evaluate RAG hyperparameters[37-41], but most focus on individual component, such as embedding model or LLM. This overlooks the interactive nature of RAG, where performance depends on multiple hyperparameter settings. In this paper, we aim to evaluate LLM's phenotyping ability using RAG and investigate impacts of RAG hyperparameters for optimization, focused on *type II diabetes mellitus (T2DM)* phenotyping.

METHODS

Data description

The dataset consists of 300 patients selected based on T2DM ICD codes (see Supplementary Material S1), whose T2DM diagnosis was validated by physicians in a previous study[42].We divided the patients into training and test sets of 100 and 200 patients respectively. Ambiguous cases were filtered out, resulting in 90 training and 185 test patients with 170,678 clinical notes. This split aims to enhance generalizability and reduce resources and time for vectorization. We used the patients' entire notes available from the *MGB RPDR Note Repository*, without restricting note types or time period.

Retrieval-Augmented Generation environment

A RAG system was built to use LLM for phenotyping. We chose RAG since it enables getting LLM answers focused on a specific patient in our local patient repository, reducing hallucinations. It also provides an effective way to process enormous number of patients' lifelong data, by augmenting retrieved results. The notes of the 275 T2DM patients were loaded from the RPDR Note Repository and segmented into chunks using fixed size overlapping sliding window method. They were vectorized by selected embedding models and stored into the PostgreSQL vector database.

Relevant chunks were retrieved by a query based on cosine similarity. A preset number of retrieved chunks or their parent document (*pdoc*) formed the context and sent to GPT-4o. For pdoc, 300-character long chunks (c300, we will use *"c"* + *length* format to describe fixed-length character chunks in the subsequent descriptions) were used for retrieval. GPT-4o was tasked to generate a binary answer for history of T2DM diagnosis and an explanation for its answer. To protect patient data, MGB-dedicated *Azure OpenAI*'s GPT-4o and *text-embedding-3-large* were used. All the other embedding models and RAG components were built and run locally, within the MGB network.

Experimental design

We examined *GPT-4o*'s phenotyping performance using RAG framework. The performance was measured on seven embedding model environments. We configured different RAG hyperparameter values optimized for each embedding model, as a uniform configuration could lead to biased results due to the varying optimal settings for different models. Figure 1 describes overall study design. It consists of prompt engineering, RAG optimization, and evaluation process. The training data was

used for developing prompts and for finding optimal hyperparameters of the seven embedding models. The test patient set was used for phenotyping evaluation.

First, Zero-shot prompt (P_{imprv}) was developed by analyzing false cases of the initial prompt ($P_{initial}$) run. For RAG Optimization, we explored interactive impacts of prompts, chunk size, number of chunks, and embedding models on RAG performance, using the P_{imprv} and $P_{initial}$. We selected one or two best performing RAG configurations for each seven embedding models and used for evaluation. The errors found in the final configurations were reviewed by physicians and the analysis results were used to build a few-shot prompt. The few-shot prompt was curated in collaboration with a software engineer and physicians.

In the evaluation phase, we assessed GPT-4o's T2DM phenotyping capability on the optimized experimental settings. The baseline prompt ($P_{initial}$) and the enhanced prompts (P_{imprv} and $P_{fewshot}$) were tested. Then the results were compared with ICD codes and existing PheNorm phenotyping method. In addition, cost-effectiveness of the LLM phenotyping was reviewed by token usage analysis.

Zero-shot prompt engineering

The initial prompt, $P_{initial}$, consisting of a direct question and straightforward output instruction (see Supplementary Material S2) was tested on Table 1(a) configurations, using OpenAI's text-embedding-3-large embedding model. False positive (FP) and false negative (FN) cases were reviewed by a software engineer, and insights were incorporated into the new prompt, P_{imprv}.

Table 1. Experimental design: RAG configurations

(a) Preliminary experimental RAG configurations for zero-shot prompt engineering

Prompt	Initial prompt ($P_{initial}$)							
Embedding model	Text-embedding-3-large							
Chunking method	**chunk size**	**number of chunks**						
	1,000 (characters)		5	10	20	40	50	100
	pdoc*	3	5	10				

* pdoc uses whole parent-document of the retrieved chunk, which retrieval was performed on 300-character long chunks

(b) RAG configurations for single model RAG sensitivity analysis

Prompt	Initial prompt ($P_{initial}$)										
	Improved zero-shot prompt (P_{imprv})										
Embedding model	Text-embedding-3-large										
Chunking method	**chunk size**	**number of chunks**									
	300	5	10	20	40	50	100	200	300	400	500
	500	5	10	20	40	50	100	200	300		
	1,000	5	10	20	40	50	100	200	300		
	pdoc	5	10								

(c) RAG configurations for secondary RAG sensitivity analysis

Prompt	Improved zero-shot prompt (P_{imprv})													
Embedding model	BioBERT													
	BiomedBERT													
	Clinical-Longformer													
	GatorTronS													
	Gte-Qwen2-1.5B-instruct													
	Medical-T5-Large													
	Text-embedding-3-large													
Chunking method	chunk size	number of chunks												
	300	5	10	20	40	50	100	200	300	400	500	600	700	800
	500	5	10	20	40	50	100	200	300					
	1,000	5	10	20	40	50	100	200	300					
	2,000	5	10	20	40	50	100	200						
	3,000	5	10	20	40	50	100							
	pdoc	5	10											

(d) RAG configurations for T2DM phenotype validation

(i) Prompts

Prompt	Initial prompt ($P_{initial}$)
	Improved zero-shot prompt (P_{imprv})
	Few-shot prompt ($P_{fewshot}$)

(ii) Optimal 'embedding model – chunking method' combinations

Embedding model	**Chunk size**	**Number of chunks**
BioBERT	300	800
BiomedBERT	300	600
BiomedBERT	pdoc	5
Clinical-Longformer	pdoc	10
GatorTronS	2,000	200
Gte-Qwen2-1.5B-instruct	2,000	20
Medical-T5-Large	1,000	200
Text-embedding-3-large	2,000	200

RAG sensitivity analysis for optimization

Single embedding model analysis: Prompt + Context chunks

A comparative analysis of P_{imprv} with $P_{initial}$ was conducted across various chunking combinations to assess the impact of prompt engineering and its relationship with chunking methods (Table 1(b)). Using text-embedding-3-large, prompts were tested on small (c300), medium (c500), and large (c1000) chunks, as well as pdoc settings. The number of retrieved chunks ranged from 3 to 500 per run. Sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV), and F1-score were evaluated.

Secondary analysis: Embedding models + Context chunks

The above experimental design was expanded to various embedding models and additional chunk configurations (Table 1(c)). Seven embedding models (Table 2) were selected based on dimension size, base model, domain relevance, licensing, performance, and popularity. Due to the PostgreSQL limitation, we selected models with dimension sizes 2,000 or less. The models were categorized into small (dimension size = 512), medium (768), and large (>=1,000) for analysis. Chunk configurations were examined up to 3,000 characters and pdoc, incrementing the number of chunks until reaching GPT-4o's maximum context size. For each embedding model, P_{imprv} was tested on 44 different "chunk size + number of retrieved chunks" configurations. The top five configurations were selected for further analysis based on F1-score, a balanced metric of PPV and sensitivity. Sensitivity, specificity, PPV, NPV, F1-score, and token usage were evaluated. In addition, optimal RAG configurations of each embedding model were identified based on F1-score.

Few-shot prompt engineering

FP and FN from the optimal RAG configurations were reviewed by two physicians. Reviewers analyzed context chunks and their metadata, LLM-generated answer and reason, and original patient notes. For each patient, the final T2DM classification (YES / NO), reviewer's comments, key chunk IDs, human error (YES / NO), and LLM pipeline errors were documented.

Table 2. Experimental design: vector embedding models

Embedding model	Dimension size	Dimension size category	Base model	(Bio-) medical	General	Commercial	Open source
text-embedding-3-large	2,000*	large	Unknown		v	v	
gte-Qwen2-1.5B-instruct[49]	1,536	large	GTE (General Text Embedding)[49]		v		v
GatorTronS[50]	1,024	large	BERT[51]	v			v
BiomedBERT[52]	768	medium	BERT	v			v
BioBERT[53]	768	medium	BERT	v			v
Clinical-Longformer[54]	768	medium	Longformer[55]	v			v
Medical-T5-Large	512	small	T5[56]	v			v

* The dimension size of the text-embedding-3-large embedding model is originally 3,072. Due to the dimensional limitation of the PostgreSQL vector database in our RAG system, reduced-dimension version was used.

A software engineer curated a few-shot prompt, $P_{fewshot}$, using the context misinterpretation instances (Supplementary Material S2). Five, the most informative, chunks with gold labels were incorporated on top of P_{imprv}. If a selected chunk is an entire note (pdoc), it was refined to include only the minimal segment with reviewer-provided keywords. Physicians confirmed that each example contained enough evidence to support the answer. No direct patient identifiable information was included.

T2DM phenotyping evaluation

To assess GPT-4o's effectiveness in T2DM phenotyping using RAG, $P_{initial}$, P_{imprv}, $P_{fewshot}$ were tested across optimized RAG configurations with test patients (Table 1(d)). The PPV of the LLM was compared to the precision of ICD T2DM codes. We also compared sensitivity, specificity, PPV, NPV, and the F1-score to PheNorm[4], MGB's main phenotyping method. RPDR T2DM PheNorm data was obtained, which version was backed up by the month of chart review.

Results

Zero-shot prompt engineering results

Error analysis revealed that LLM's mis-conclusions mainly stemmed from a lack of comprehension regarding the target task, which is phenotyping patient data for research use. For example, the LLM incorrectly responded "YES" when instances where T2DM was indicated solely as a family history. In addition, the LLM answered "NO" for resolved T2DM, although "YES" was expected for our phenotyping goal.

Misinterpretation of structural information also resulted in erroneous conclusions. We use clinical notes converted to text format, which include structural information that may be difficult for LLM to comprehend. For example, LLM recognized "?" in "?Type 2 diabetes mellitus" as a question mark, which "?" was originally a bullet point in the *Patient Active Problem List* section. As a result, LLM misinterpreted it as *suspected* or a *question*. Similarly, "T2DM" in the *Diagnosis* column of a *Medical History* table was not identified correctly in relation to its section title and the column header. It was

challenging because “T2DM” was surrounded by numerous spaces intended to fill vacant cell spaces and followed by an incomplete term due to word segmentation prompted by constrained cell spaces.

In some cases, crucial evidence was ranked low, affecting results when fewer chunks were used, due to the exclusion of essential information from the context. Interestingly, information density also seemed to impact results. For example, a clear, the only evidence in the 43rd-ranked chunk led to different answers: the LLM answered “YES” with 50 chunks but “NO” with 100 chunks. These insights are incorporated into the improved prompt P_{imprv} (Supplementary Material S2).

RAG sensitivity analysis results

Impact of Prompt + Context chunks

The results of $P_{initial}$ with P_{imprv} on a fixed embedding model are described in Figure 2. The two prompts showed distinct trends in the highest-performing configurations. $P_{initial}$ performed best with smaller chunks on text-embedding-3-large. Specifically, within the $P_{initial}$ results, all eleven c300 settings achieved the highest performance rankings across all metrics. The performance rankings of the other chunk sizes were mixed, with c500 notably occupying the lowest ranks.

In contrast, with P_{imprv}, all top ten highest F1-scores, nine out of top ten sensitivities and NPVs were achieved using c1000 and pdoc. The highest rank achieved using c300 were 13th in F1-score, 22nd in sensitivity, and 22nd in NPV. For specificity and PPV, the majority of the top ten performance were associated with c1000. Overall, when used with c1000 or pdoc, P_{imprv} demonstrated better performance than $P_{initial}$, whereas $P_{initial}$ outperformed P_{imprv} with smaller chunk sizes.

Impact of Embedding models + Context chunks

The five highest F1-score configurations of each embedding model, along with their token usage and performance are summarized in Table 3. In Figure 3, regarding F1-score, the two large, general embedding models demonstrated exceptional performance, securing the highest eight ranks. The highest F1-score was 0.9701 by text-embedding-3-largeand and gte-Qwen2-1.5B-instruct. These were

Table 3. Secondary RAG sensitivity experimental results: results of top five F1-score configurations

This table describes secondary RAG sensitivity experimental results, which configurations are selected by five highest F1-score results of each tested embedding model.

Embedding model	Dimension	Dim. category	Chunk size	Num chunks	Avg. num tokens	Sensitivity	Specificity	PPV	NPV	F1
BioBERT	768	medium	300	500	27,926.87	0.9254	0.9565	0.9841	0.8148	0.9538
			300	600	33,643.51	0.9254	0.9565	0.9841	0.8148	0.9538
			300	700	39,363.18	0.9254	0.9565	0.9841	0.8148	0.9538
			300	800	44,996.34	0.9403	0.9565	0.9844	0.8462	0.9618
			1,000	200	36,336.22	0.9254	0.9565	0.9841	0.8148	0.9538
BiomedBERT	768	medium	300	400	23,463.41	0.9104	0.9565	0.9839	0.7857	0.9457
			300	600	35,503.87	0.9104	1.0000	1.0000	0.7931	0.9531
			300	700	41,527.22	0.9104	1.0000	1.0000	0.7931	0.9531
			pdoc	5	20,512.05	0.9403	0.9545	0.9844	0.8400	0.9618
			pdoc	10	37,920.24	0.9403	0.8636	0.9545	0.8261	0.9474
Clinical-Longformer	768	medium	300	300	15,484.71	0.8788	0.9565	0.9831	0.7333	0.9280
			300	500	26,638.08	0.8788	1.0000	1.0000	0.7419	0.9355
			300	700	38,172.99	0.8939	0.9130	0.9672	0.7500	0.9291
			500	300	25,721.04	0.8955	0.9565	0.9836	0.7586	0.9375
			pdoc	10	33,268.39	0.9254	0.8696	0.9538	0.8000	0.9394
GatorTronS	1,024	large	300	400	21,981.61	0.8955	0.9565	0.9836	0.7586	0.9375
			2,000	50	16,188.38	0.8806	1.0000	1.0000	0.7419	0.9365
			2,000	200	68,568.66	0.9403	0.9130	0.9692	0.8400	0.9545
			pdoc	5	25,883.82	0.9403	0.8696	0.9545	0.8333	0.9474
			pdoc	10	51,061.01	0.9385	0.8261	0.9385	0.8261	0.9385
gte-Qwen2-1.5B-instruct	1,536	large	2,000	20	7,129.71	0.9701	0.9091	0.9701	0.9091	0.9701
			2,000	30	10,772.46	0.9701	0.9091	0.9701	0.9091	0.9701

			2,000	40	14,425.85	0.9552	0.9091	0.9697	0.8696	0.9624
			2,000	50	18,167.26	0.9701	0.9091	0.9701	0.9091	0.9701
			2,000	100	36,967.94	0.9701	0.8636	0.9559	0.9048	0.9630
Medical-T5-Large	512	small	300	500	24,270.56	0.9104	0.9565	0.9839	0.7857	0.9457
			300	600	29,322.37	0.8955	1.0000	1.0000	0.7667	0.9449
			300	700	34,387.11	0.9254	0.9565	0.9841	0.8148	0.9538
			300	800	39,453.08	0.9104	1.0000	1.0000	0.7931	0.9531
			1,000	200	26,758.58	0.9403	0.9565	0.9844	0.8462	0.9618
text-embedding-3-large	2,000	large	1,000	30	6,323.22	0.9701	0.8261	0.9420	0.9048	0.9559
			1,000	40	8,445.69	0.9701	0.8261	0.9420	0.9048	0.9559
			1,000	200	42,627.19	0.9701	0.8696	0.9559	0.9091	0.9630
			1,000	300	63,832.84	0.9701	0.8696	0.9559	0.9091	0.9630
			2,000	200	77,070.05	0.9701	0.9130	0.9701	0.9130	0.9701

followed by small and medium size (bio-)medical-specific models, BioBERT, Medical-T5-Large, and BiomedBERT, all achieving 0.9618. GatorTronS, despite its large dimension size, did not perform better than the three domain-specific models. The lowest performance was 0.928 by Clinical-Longformer.

The top ten highest sensitivity and NPV were achieved by the two general models. The highest sensitivity was 0.9701 by both text-embedding-3-large and gte-Qwen2-1.5B-instruct, and the highest NPV was 0.9130 by text-embedding-3-large. BioBERT was the best-performing medical-specific model in sensitivity (0.9403) and NPV (0.8462). In contrast, (bio-)medical-specific models excelled in specificity and PPV. BiomedBERT, Medical-T5-Large, GatorTronS, and Clinical-Longformer achieved up to 1.0000 in both metrics. The best general embedding model ranked 20th by text-embedding-3-large, demonstrating 0.9130 and 0.9701, respectively.

The optimal chunk size and number combination varied per embedding model. Large embedding models predominantly performed best with larger chunks (c1000 or longer). Especially, all top five gte-Qwen2-1.5B-instruct configurations used c2000 chunks. In contrast, a small model, Medical-T5-Large mostly performed well with short chunks, using a large number of them (c300, at least 500 chunks). There was no clear pattern for medium models. BioBERT performed well with short chunks, but BiomedBERT and Clinical-Longformer showed inconsistent patterns. None of the c3000 configurations ranked among the top five (Figure 4).

As described in Figure 5, in medium and small size embedding models, all top five performing configurations used certain token range. However, Figure 3 shows that a large embedding model, text-embedding-3-large tended to perform better in F1-score with more tokens. The best F1-score with text-embedding-3-large used up to 77,070.05 tokens. In contrast, most gte-Qwen2-1.5B-instruct, another large size embedding model, worked great with fewer than 20,000 tokens. Both text-embedding-3-large and gte-Qwen2-1.5B-instruct reached the highest F1-scores, but the latter did so with less than 10% of the tokens. GatorTronS, a large (bio-)medical embedding model, was the most efficient for specificity and PPV, using 16,188.38 tokens on average.

We finalized optimal configurations (Table 1(d-ii)) to test GPT-4o's phenotyping performance on the test patient set. For BiomedBERT, both the best (pdoc, 5) and the second-best (c300, 600) F1-score configurations were chosen, since the former had one patient failed to get an LLM response due to context length limitation.

Few-shot prompt engineering result

We analyzed 43 FP and FN results from the optimal RAG configurations, observed across 18 unique patients. Two human errors, four indeterminate, and twelve patients with LLM pipeline issues were identified. LLM pipeline errors had two primary reasons. First, RAG failed to retrieve key chunks despite explicit evidence (3 patients). It was observed in limited embedding models: Clinical-Longformer, BioBERT, and the two BiomedBERT. Second, GPT-4o misinterpreted from properly retrieved context (9 patients).

In context misinterpretation cases, GPT-4o often missed to capture key evidence like T2DM-specific medication (e.g., liraglutide, which prescribed only for T2DM) or other DM types (e.g., Type 1 DM). Sometimes, GPT-4o was unable to rule out T2DM despite successfully identifying other types of DM (e.g., Type 1 DM, steroid-induced DM, cystic fibrosis related diabetes). Other cases involved phenotyping based on surface-level textual information. These DM patients did not have direct mentions of the type or T2DM-specific treatments, but clinicians typically interpreted these patients as having T2DM based on holistic reviews.

For the few-shot prompt, we selected two positive (gold label "YES") and three negative examples from misinterpretation cases. Positive examples featured T2DM medications without specifying the type, while negative examples included chunks explicitly indicating other DM types.

T2DM phenotyping evaluation results

Table 4 and Figure 6 show GPT-4o's T2DM phenotyping results using $P_{initial}$, P_{imprv}, and $P_{fewshot}$. The highest sensitivity (0.9517) was achieved by gte-Qwen2-1.5B-instruct using both $P_{initial}$ and P_{imprv}, as well as by BioBERT utilizing P_{imprv}. The use of BioBERT with P_{imprv} also contributed to the best NPV

Table 4. T2DM phenotype validation results

Prompt	Embedding model	Chunk size	Num chunks	Avg. num tokens	Sensitivity	Specificity	PPV	NPV	F1
$P_{initial}$	BioBERT	300	800	44,131.36	0.9172	0.8500	0.9568	0.7391	0.9366
	BiomedBERT	300	600	35,062.25	0.8759	0.8250	0.9478	0.6471	0.9104
	BiomedBERT	pdoc	5	19,025.18	0.8828	0.7692	0.9343	0.6383	0.9078
	Clinical-Longformer	pdoc	10	34,682.29	0.9310	0.7179	0.9247	0.7368	0.9278
	GatorTronS	2,000	200	65,111.79	0.9172	0.7500	0.9301	0.7143	0.9236
	gte-Qwen2-1.5B-instruct	2,000	20	7,371.85	0.9517	0.7500	0.9324	0.8108	0.9420
	Medical-T5-Large	1,000	200	25,967.30	0.8690	0.7750	0.9333	0.6200	0.9000
	text-embedding-3-large	2,000	200	73,890.35	0.9236	0.8462	0.9568	0.7500	0.9399
P_{imprv}	BioBERT	300	800	44,131.36	0.9517	0.8000	0.9452	0.8205	0.9485
	BiomedBERT	300	600	35,062.25	0.9034	0.7750	0.9357	0.6889	0.9193
	BiomedBERT	pdoc	5	19,025.18	0.8828	0.7949	0.9412	0.6458	0.9110
	Clinical-Longformer	pdoc	10	34,682.29	0.9379	0.7692	0.9379	0.7692	0.9379
	GatorTronS	2,000	200	65,111.79	0.9379	0.7250	0.9252	0.7632	0.9315
	gte-Qwen2-1.5B-instruct	2,000	20	7,371.85	0.9517	0.7750	0.9388	0.8158	0.9452
	Medical-T5-Large	1,000	200	25,967.30	0.9034	0.7750	0.9357	0.6889	0.9193
	text-embedding-3-large	2,000	200	73,890.35	0.9097	0.8205	0.9493	0.7111	0.9291
$P_{fewshot}$	BioBERT	300	800	65,111.79	0.9034	0.8000	0.9424	0.6957	0.9225
	BiomedBERT	300	600	44,131.36	0.8414	0.8750	0.9606	0.6034	0.8971
	BiomedBERT	pdoc	5	73,890.35	0.8690	0.7949	0.9403	0.6200	0.9032
	Clinical-Longformer	pdoc	10	19,025.18	0.9241	0.7949	0.9437	0.7381	0.9338
	GatorTronS	2,000	200	25,967.30	0.8207	0.7750	0.9297	0.5439	0.8718
	gte-Qwen2-1.5B-instruct	2,000	20	7,371.85	0.9167	0.8000	0.9429	0.7273	0.9296
	Medical-T5-Large	1,000	200	35,062.25	0.8483	0.8250	0.9462	0.6000	0.8945
	text-embedding-3-large	2,000	200	34,682.29	0.8750	0.7949	0.9403	0.6327	0.9065

(0.8205) and F1-score (0.9485). The highest specificity (0.8750) and PPV (0.9606) were attained by BiomedBERT (c300, 600) when using $P_{fewshot}$. Regarding token usage, some configurations achieved superior performance with significantly fewer resources. For example, while the highest sensitivity, NPV, and F1-score needed over 40,000 tokens on average with BioBERT, gte-Qwen2-1.5B-instruct achieved the same or similar performance using less than 10,000 tokens.

The zero-shot prompt and the few-shot presented different impacts on the evaluation metrics. P_{imprv} boosted overall performance in sensitivity, NPV, and F1-score, except text-embedding-3-large model, compared to the initial prompt, $P_{initial}$. The largest performance improvement observed with P_{imprv} was a 0.0814 increase in NPV, when paired with BioBERT. For specificity and PPV, it did not demonstrate notable improvements. Only about a half of the experimented configurations had performance enhancements. Moreover, the effects were mostly minor, except for Clinical-Longformer which showed 0.0513 improvement in specificity.

In contrast, $P_{fewshot}$ made improvements in specificity and PPV in majority of the configurations. NPV and F1-score were lower than the baseline, except for the Clinical-Longformer model. No embedding models had improvement in sensitivity. Clinical-Longformer exhibited the most improved result, enhancing 0.0769 in specificity. In addition, it was the only model that made improvements in NPV and F1-score by using $P_{fewshot}$. Interestingly, the two BiomedBERT configurations used in this evaluation initially had similar sensitivity, NPV, and F1-score.

The sensitivity, specificity, PPV, and NPV, and F1 value of the PheNorm were 0.6000, 0.9000, 0.9560, 0.3830, 0.7373. RAG phenotyping significantly outperformed PheNorm in sensitivity, NPV, and F1. Sensitivity improved by up to 0.3517, NPV by 0.4375, and F1-score up to 0.2112. However, only text-embedding-3-large and BioBERT with $P_{initial}$ and BiomedBERT (c300, 600) using $P_{fewshot}$ exhibited minor improvements (all less than 0.005) in PPV. Specificity in all RAG configurations was lower than PheNorm, with gaps from 0.0250 up to 0.1821. These results were attributed to the low cut-off value setting (PheNorm score >= 0.5). The precision of ICD code was 0.7838, and all RAG PPV values improved by 0.1409 to 0.1768.

Discussion

We evaluated GPT-4o's T2DM phenotyping performance using optimized RAG configurations. A previous study[13] also used LLMs for T2DM phenotyping, where LLMs generated phenotyping SQL algorithms. Our results presented superior performance compared to their best result by GPT-4, with even the lowest PPV of our optimal configurations exceeding theirs by 40.5% points. This shows the value of using RAG.

However, the nave application of RAG does not guarantee high performance. While we initially focused on the high-performing configurations in the RAG sensitivity analysis, in the entire experimental settings, the variability between the highest and the lowest performance was significant, reaching up to 0.7910 gap in the Clinical-Longformer's sensitivity. Our in-depth analysis provides valuable insights into the effective use of RAG for phenotyping. We investigated interactions of multi-hyperparameters on RAG performance, leveraging prompts developed in collaboration with a software engineer and clinicians. Furthermore, the analysis is grounded in real-world, clinician-validated patient data, encompassing all notes collected from our institution's multi-hospital system.

The single embedding model RAG sensitivity analysis highlights the different impact of chunk sizes on a prompt. P_{imprv} excelled with larger chunks, while $P_{initial}$ demonstrated superior results with smaller chunks. It is likely due to P_{imprv}'s refined phenotyping task and instructions for handling structural information, which demand more context. It suggests that both context size and prompt should be carefully considered for optimal results.

In the secondary RAG sensitivity analysis, there were notable differences between general embedding models and domain-specific embedding models. The two general embedding models outperformed (bio-)medical-specific models in sensitivity, NPV, and F1. This observation could be partly attributed to our general embedding model selection strategy, which prioritized established high performance. Text-embedding-3-large was from *OpenAI*, a commercial service, and gte-Qwen2-1.5B-instruct held the top-rank, under the 2,000-dimensional constraint, on the *Massive Text Embedding Benchmark (MTEB) Leaderboard*[43] for both overall performance and retrieval at the time of this study. Another

potential contributing factor could be their larger dimension size. However, further investigations are needed, particularly given that GatorTronS did not outperform the other (bio-)medical models.

In contrast, (bio-)medical models showed superior results in specificity and PPV. Furthermore, although the top performance ranks in some metrics were achieved by the general models, (bio-)medical domain-specific models demonstrated comparably reliable performance by configuring proper hyperparameter values (i.e., chunk size and the number of chunks). These findings align with our previous study on ischemic stroke[44], which compared text-embedding-3-large and BiomedBERT for a phenotyping task.

The top five performance rankings present clusters of optimal hyperparameter values for some embedding models. With P_{imprv}, larger chunks generally benefited large embedding models. However, for certain medical-specific models, smaller chunks were more effective. In such cases, significantly higher number of chunks than larger chunks were required. RAG configurations using c3000 chunks were not included in the top list. Notably, five out of the 35 top-performing configurations used pdoc, which size is mostly larger than c3000, but the context was built based on 300-character length chunk retrieval results. Token usage analysis indicated that some models could achieve similar results with significantly lower cost, which is crucial for large-scale phenotyping tasks. Gte-Qwen2-1.5B-instruct was the most cost-effective in F1, sensitivity, and NPV, while GatorTronS was the most efficient for specificity and PPV.

FP and FN analysis showed that LLM needed specific goal and task definitions to perform the nuanced and purpose-specific nature of medical tasks. It also uncovered GPT-4o's limitations in applying domain knowledge, such as recognizing important medications or ruling out target disease based on identified evidence. We approached to solve it with few-shot prompt, which improved specificity and PPV, but it decreased sensitivity, NPV, and F1-score. This highlights the complexity of few-shot prompt engineering[45-48]. Further investigations are needed to improve the few-shot prompt. Notably, one patient consistently produced false results across all tested configurations,

which was later attributed to human error. This suggests the potential of RAG as a useful support to help prevent errors in chart reviews.

Another significant finding was that chunks containing key evidence were sometimes low-ranked and thus excluded from the context. Effective retrieval and ranking methods should be explored. Furthermore, this study is limited that the RAG optimization insights were derived based on a single LLM and a single disease domain. Future work will include evaluating additional RAG hyperparameters and expanding phenotypes to gain deeper insights.

Conclusion

This study explored LLM phenotyping efficacy by using GPT-4o and a T2DM use case. We evaluated GPT-4o on the optimized RAG environment with seven embedding models. Various prompts, developed by a software engineer and clinicians, were used for evaluation. In summary, GPT-4o surpassed ICD codes and PheNorm in key metrics, though PPV and specificity need improvement compared to PheNorm. Overall, the results were promising, but tuning RAG hyperparameters with consideration of resource usage and prompt engineering would be the key for successful applications.

The main contribution of this study is the systematic investigation of RAG hyperparameters and their inter-relationships using real patient data, providing insights into optimal RAG configurations and cost-effective embedding model choices. We found that the two tested general embedding models perform better than domain-specific models in sensitivity, NPV, and F1. Conversely, (bio-)medical embedding models were more effective in specificity and PPV. Prompt design also had a crucial impact: zero-shot prompt was more effective in sensitivity, NPV, and F1, while few-shot prompt did better in specificity and PPV.

Moreover, our results indicated that even embedding models that are not the top ranks can achieve reliable performance by hyperparameter tuning. The optimal chunk size varied depending on embedding model and prompt. There was a tendency that smaller models required substantial number

of chunks to reach high performance. In addition, token usage analysis demonstrated that some models, like gte-Qwen2-1.5B-instruct and GatorTronS, can provide similar results with significantly lower costs.

This research identified areas requiring improvements through error analysis, such as contextual misinterpretation by the LLM and chunk ranking problem. The limitations of our study are that we tested LLM using GPT-4o only and the scope of phenotyping targeted just one disease domain. Future studies will expand hyperparameter investigation into diverse phenotypes.

ACKNOWLEDGMENTS

This study was approved by the Institutional Review Board (IRB) of Mass General Brigham (2020P000060). The Mass General Brigham IRB approved for a waiver of patient informed consent.

COMPETING INTERESTS

The authors have no conflicts of interest to disclose.

REFERENCES

1. Murphy SN, Gainer V, Chueh HC. A visual interface designed for novice users to find research patient cohorts in a large biomedical database. AMIA Annu Symp Proc 2003;2003:489-93. (In eng).

2. Nalichowski R, Keogh D, Chueh HC, Murphy SN. Calculating the benefits of a Research Patient Data Repository. AMIA Annu Symp Proc 2006;2006:1044. (In eng).

3. Castro VM, Gainer V, Wattanasin N, et al. The Mass General Brigham Biobank Portal: an i2b2-based data repository linking disparate and high-dimensional patient data to support multimodal analytics. J Am Med Inform Assoc 2022;29(4):643-651. (In eng). DOI: 10.1093/jamia/ocab264.

4. Yu S, Ma Y, Gronsbell J, et al. Enabling phenotypic big data with PheNorm. J Am Med Inform Assoc 2018;25(1):54-60. (In eng). DOI: 10.1093/jamia/ocx111.

5. Denny JC, Bastarache L, Ritchie MD, et al. Systematic comparison of phenome-wide association study of electronic medical record data and genome-wide association study data. Nat Biotechnol 2013;31(12):1102-10. (In eng). DOI: 10.1038/nbt.2749.

6. McCoy TH, Jr., Yu S, Hart KL, et al. High Throughput Phenotyping for Dimensional Psychopathology in Electronic Health Records. Biol Psychiatry 2018;83(12):997-1004. (In eng). DOI: 10.1016/j.biopsych.2018.01.011.

7. Gainer VS, Cagan A, Castro VM, et al. The Biobank Portal for Partners Personalized Medicine: A Query Tool for Working with Consented Biobank Samples, Genotypes, and Phenotypes Using i2b2. J Pers Med 2016;6(1) (In eng). DOI: 10.3390/jpm6010011.

8. Liao KP, Cai T, Savova GK, et al. Development of phenotype algorithms using electronic medical records and incorporating natural language processing. Bmj 2015;350:h1885. (In eng). DOI: 10.1136/bmj.h1885.

9. Klann JG, Mendis M, Phillips LC, et al. Taking advantage of continuity of care documents to populate a research repository. J Am Med Inform Assoc 2015;22(2):370-9. (In eng). DOI: 10.1136/amiajnl-2014-003040.

10. Castro VM, Apperson WK, Gainer VS, et al. Evaluation of matched control algorithms in EHR-based phenotyping studies: a case study of inflammatory bowel disease comorbidities. J Biomed Inform 2014;52:105-11. (In eng). DOI: 10.1016/j.jbi.2014.08.012.

11. Alsentzer E, Rasmussen MJ, Fontoura R, et al. Zero-shot interpretable phenotyping of postpartum hemorrhage using large language models. npj Digital Medicine 2023;6(1):212. DOI: 10.1038/s41746-023-00957-x.

12. Bhattarai K, Oh IY, Sierra JM, et al. Leveraging GPT-4 for identifying cancer phenotypes in electronic health records: a performance comparison between GPT-4, GPT-3.5-turbo, Flan-T5,

Llama-3-8B, and spaCy's rule-based and machine learning-based methods. JAMIA Open 2024;7(3):ooae060. (In eng). DOI: 10.1093/jamiaopen/ooae060.

13. Yan C, Ong HH, Grabowska ME, et al. Large language models facilitate the generation of electronic health record phenotyping algorithms. Journal of the American Medical Informatics Association 2024;31(9):1994-2001. DOI: 10.1093/jamia/ocae072.

14. Garcia BT, Westerfield L, Yelemali P, et al. Improving Automated Deep Phenotyping Through Large Language Models Using Retrieval Augmented Generation. medRxiv 2024 (In eng). DOI: 10.1101/2024.12.01.24318253.

15. Groza T, Caufield H, Gration D, et al. An evaluation of GPT models for phenotype concept recognition. BMC Medical Informatics and Decision Making 2024;24(1):30. DOI: 10.1186/s12911-024-02439-w.

16. Vasisht Shankar S, Thangaraj P, Adejumo P, Khera R. Abstract 4145446: Scalable Phenotyping of Heart Failure Across Multicenter, Non-Interoperable Health Systems Using Retrieval-Augmented Generation and Large Language Models. Circulation 2024;150(Suppl_1):A4145446-A4145446. DOI: doi:10.1161/circ.150.suppl_1.4145446.

17. Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. Proceedings of the 34th International Conference on Neural Information Processing Systems. Vancouver, BC, Canada: Curran Associates Inc.; 2020:Article 793.

18. Unlu O, Shin J, Mailly CJ, et al. Retrieval-Augmented Generation–Enabled GPT-4 for Clinical Trial Screening. NEJM AI 2024:AIoa2400181.

19. Wang C, Ong J, Wang C, Ong H, Cheng R, Ong D. Potential for GPT Technology to Optimize Future Clinical Decision-Making Using Retrieval-Augmented Generation. Ann Biomed Eng 2024;52(5):1115-1118. (In eng). DOI: 10.1007/s10439-023-03327-6.

20. Zhu Y, Ren C, Wang Z, et al. EMERGE: Integrating RAG for Improved Multimodal EHR Predictive Modeling. arXiv preprint arXiv:240600036 2024.

21. Jeong M, Sohn J, Sung M, Kang J. Improving medical reasoning through retrieval and self-reflection with retrieval-augmented large language models. Bioinformatics 2024;40(Supplement_1):i119-i129. (In eng). DOI: 10.1093/bioinformatics/btae238.

22. Ge J, Sun S, Owens J, et al. Development of a liver disease-specific large language model chat interface using retrieval-augmented generation. Hepatology 2024 (In eng). DOI: 10.1097/hep.0000000000000834.

23. Murugan M, Yuan B, Venner E, et al. Empowering personalized pharmacogenomics with generative AI solutions. J Am Med Inform Assoc 2024;31(6):1356-1366. (In eng). DOI: 10.1093/jamia/ocae039.

24. Alkhalaf M, Yu P, Yin M, Deng C. Applying generative AI with retrieval augmented generation to summarize and extract key clinical information from electronic health records. J Biomed Inform 2024;156:104662. (In eng). DOI: 10.1016/j.jbi.2024.104662.

25. Abdullahi T, Mercurio L, Singh R, Eickhoff C. Retrieval-Based Diagnostic Decision Support: Mixed Methods Study. JMIR Med Inform 2024;12:e50209. (In eng). DOI: 10.2196/50209.

26. Miao J, Thongprayoon C, Suppadungsuk S, Garcia Valencia OA, Cheungpasitporn W. Integrating Retrieval-Augmented Generation with Large Language Models in Nephrology: Advancing Practical Applications. Medicina (Kaunas) 2024;60(3) (In eng). DOI: 10.3390/medicina60030445.

27. Yazaki M, Maki S, Furuya T, et al. Emergency Patient Triage Improvement through a Retrieval-Augmented Generation Enhanced Large-Scale Language Model. Prehosp Emerg Care 2024:1-7. (In eng). DOI: 10.1080/10903127.2024.2374400.

28. Wu L, Xu J, Thakkar S, et al. A framework enabling LLMs into regulatory environment for transparency and trustworthiness and its application to drug labeling document. Regul Toxicol Pharmacol 2024;149:105613. (In eng). DOI: 10.1016/j.yrtph.2024.105613.

29. Toro S, Anagnostopoulos AV, Bello SM, et al. Dynamic Retrieval Augmented Generation of Ontologies using Artificial Intelligence (DRAGON-AI). J Biomed Semantics 2024;15(1):19. (In eng). DOI: 10.1186/s13326-024-00320-3.

30. Barnett S, Kurniawan S, Thudumu S, Brannelly Z, Abdelrazek M. Seven Failure Points When Engineering a Retrieval Augmented Generation System. Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI. Lisbon, Portugal: Association for Computing Machinery; 2024:194–199.

31. Fu J, Qin X, Yang F, et al. AutoRAG-HP: Automatic Online Hyper-Parameter Tuning for Retrieval-Augmented Generation. Miami, Florida, USA: Association for Computational Linguistics; 2024:3875-3891.

32. Barker M, Bell A, Thomas E, Carr J, Andrews T, Bhatt U. Faster, Cheaper, Better: Multi-Objective Hyperparameter Optimization for LLM and RAG Systems. arXiv preprint arXiv:250218635 2025.

33. Juvekar K, Purwar A. Introducing a new hyper-parameter for RAG: Context Window Utilization. arXiv preprint arXiv:240719794 2024.

34. Zakka C, Shad R, Chaurasia A, et al. Almanac - Retrieval-Augmented Language Models for Clinical Medicine. Nejm ai 2024;1(2) (In eng). DOI: 10.1056/aioa2300068.

35. Giuffrè M, Kresevic S, Pugliese N, You K, Shung DL. Optimizing large language models in digestive disease: strategies and challenges to improve clinical outcomes. Liver Int 2024 (In eng). DOI: 10.1111/liv.15974.

36. Xiong G, Jin Q, Wang X, Zhang M, lu Z, Zhang A. Improving Retrieval-Augmented Generation in Medicine with Iterative Follow-up Questions. 2024. DOI: 10.48550/arXiv.2408.00727.

37. Bora A, Cuayáhuitl H. Systematic Analysis of Retrieval-Augmented Generation-Based LLMs for Medical Chatbot Applications. Machine Learning and Knowledge Extraction 2024;6(4):2355-2374. (https://www.mdpi.com/2504-4990/6/4/116).

38. Xiong G, Jin Q, Lu Z, Zhang A. Benchmarking Retrieval-Augmented Generation for Medicine. Bangkok, Thailand: Association for Computational Linguistics; 2024:6233-6251.

39. Excoffier J-B, Roehr T, Figueroa A, Papaioannou J-M, Bressem KK, Ortala M. Generalist embedding models are better at short-context clinical semantic search than specialized embedding models. ArXiv 2024;abs/2401.01943.

40. Elgedawy R, Srinivasan S, Danciu I. Dynamic Q&A of Clinical Documents with Large Language Models. arXiv preprint arXiv:240110733 2024.

41. Si Y, Wang J, Xu H, Roberts K. Enhancing clinical concept extraction with contextual embeddings. J Am Med Inform Assoc 2019;26(11):1297-1304. (In eng). DOI: 10.1093/jamia/ocz096.

42. Ge T, Irvin MR, Patki A, et al. Development and validation of a trans-ancestry polygenic risk score for type 2 diabetes in diverse populations. Genome Medicine 2022;14(1):70. DOI: 10.1186/s13073-022-01074-2.

43. Enevoldsen K, Chung I, Kerboua I, et al. Mmteb: Massive multilingual text embedding benchmark. arXiv preprint arXiv:250213595 2025.

44. Park HR, Martin; Hsieh, Yichuan Grace; Wattanasin, Nich; Harris, Allan J.; Gainer, Vivian; McShane, Thomas; Wagholikar, Kavishwar; Murphy, Shawn;. Optimizing Retrieval-Augmented Generation (RAG) for Retrospective Ischemic Stroke Identification: A Comparative Study of Embedding Models and Retrieved Chunks. AMIA Informatics Summit. Pittsburgh2025.

45. Zhao Z, Wallace E, Feng S, Klein D, Singh S. Calibrate before use: Improving few-shot performance of language models. International conference on machine learning: PMLR; 2021:12697-12706.

46. Gao T, Fisch A, Chen D. Making Pre-trained Language Models Better Few-shot Learners. Online: Association for Computational Linguistics; 2021:3816-3830.

47. Schick T, Schütze H. True Few-Shot Learning with Prompts—A Real-World Perspective. Transactions of the Association for Computational Linguistics 2022;10:716-731. DOI: 10.1162/tacl_a_00485.

48. Ge Y, Guo Y, Das S, Al-Garadi MA, Sarker A. Few-shot learning for medical text: A review of advances, trends, and opportunities. J Biomed Inform 2023;144:104458. (In eng). DOI: 10.1016/j.jbi.2023.104458.

49. Li Z, Zhang X, Zhang Y, Long D, Xie P, Zhang M. Towards general text embeddings with multi-stage contrastive learning. arXiv preprint arXiv:230803281 2023.

50. Peng C, Yang X, Chen A, et al. A study of generative large language model for medical research and healthcare. npj Digital Medicine 2023;6(1):210. DOI: 10.1038/s41746-023-00958-w.

51. Devlin J, Chang M-W, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)2019:4171-4186.

52. Gu Y, Tinn R, Cheng H, et al. Domain-specific language model pretraining for biomedical natural language processing. ACM Transactions on Computing for Healthcare (HEALTH) 2021;3(1):1-23.

53. Lee J, Yoon W, Kim S, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 2020;36(4):1234-1240.

54. Li Y, Wehbe RM, Ahmad FS, Wang H, Luo Y. A comparative study of pretrained language models for long clinical text. J Am Med Inform Assoc 2023;30(2):340-347. (In eng). DOI: 10.1093/jamia/ocac225.

55. Beltagy I, Peters ME, Cohan A. Longformer: The long-document transformer. arXiv preprint arXiv:200405150 2020.

56. Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research 2020;21(140):1-67.

Figure 1. Overall Experimental Design

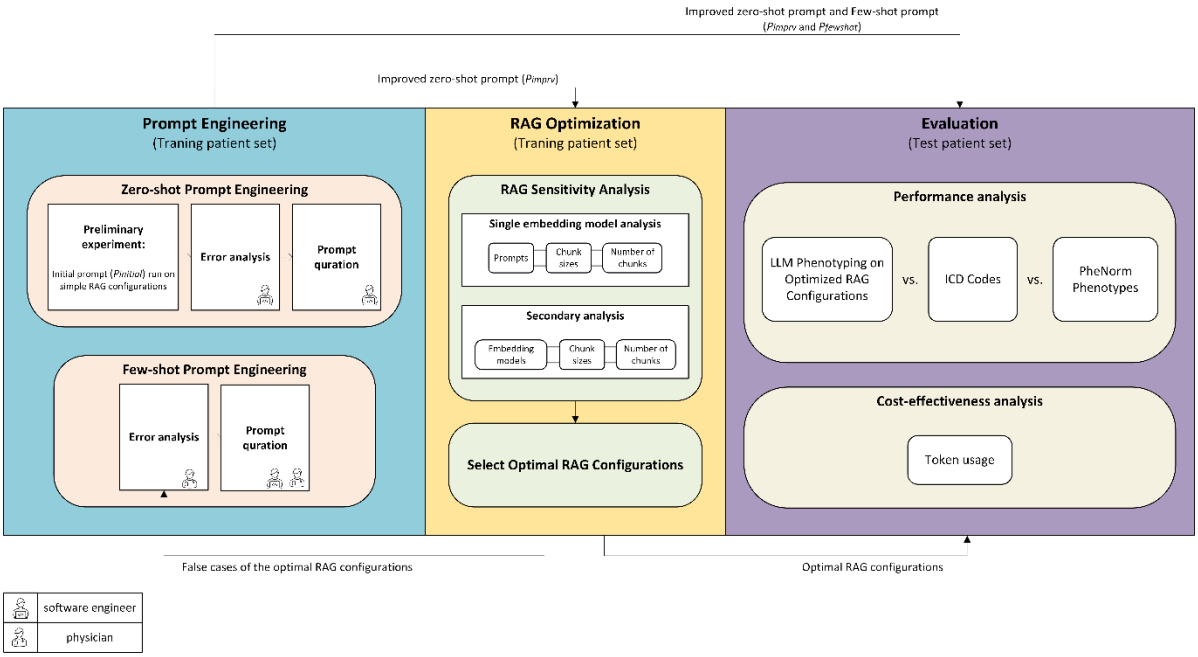


Figure 2. Single model RAG sensitivity experimental results

This figure visualizes single model RAG sensitivity experimental results. The experiment measured performance of GPT-4o's T2DM phenotype identification on text-embedding-3-large embedding model. Using chunks with size 300, 500, and 1,000 characters, and parent documents of 300-character long chunk retrieval results were examined, incrementing the number of chunks up to 500. RAG configurations with chunk sizes 500 characters and longer used fewer than 500 chunks due to the GPT-4o's context size limitation. Each column (a) ~ (d) demonstrates sensitivity, specificity, PPV, NPV, and F1-score results grouped by chunk size used. Note that the y-axis scale is different per evaluation metric to highlight the difference between the initial prompt ($P_{initial}$) and the zero-shot improved prompt (P_{imprv}).

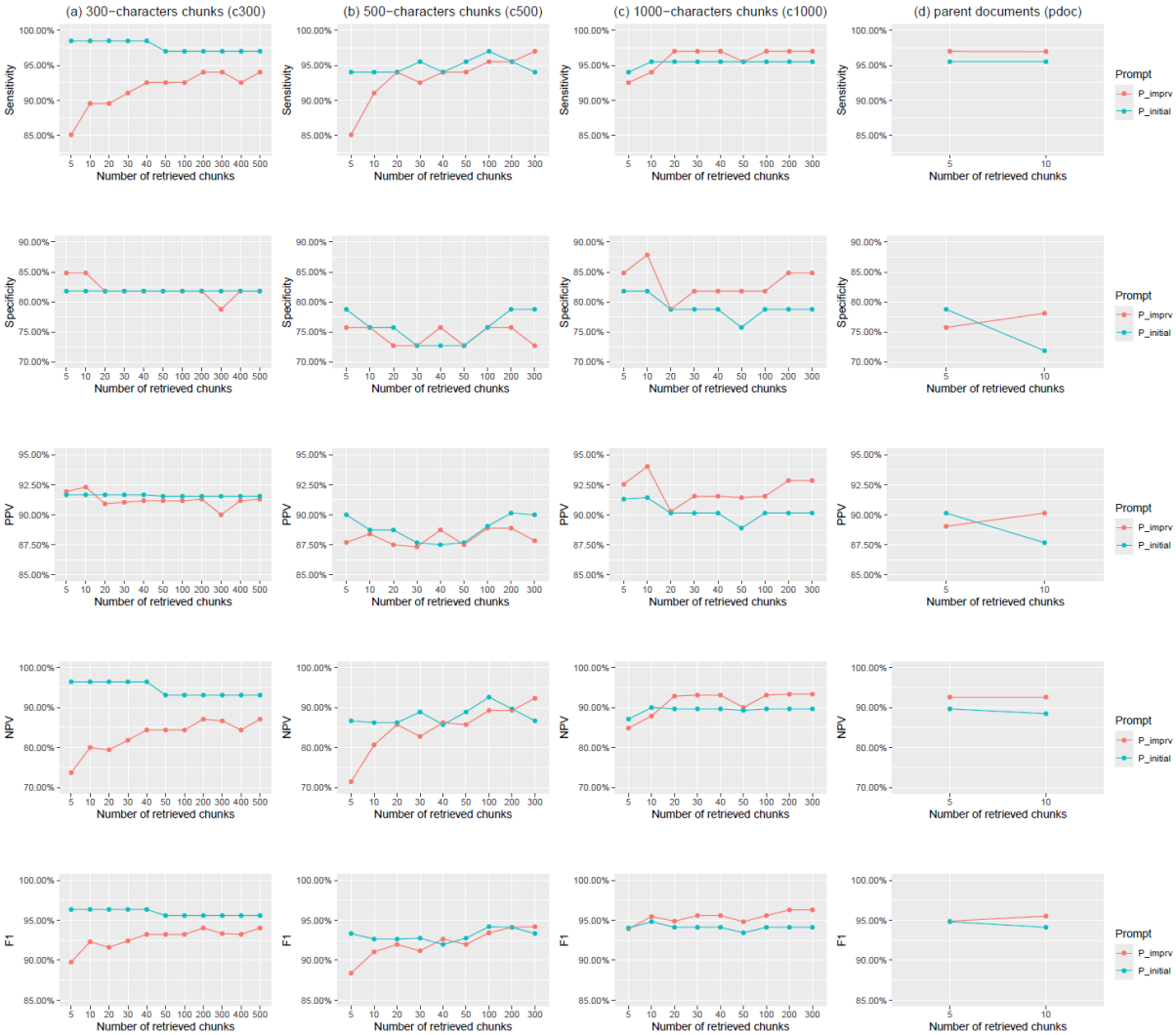


Figure 3. Sensitivity analysis results: performance and average number of tokens used

These demonstrate the average number of tokens used per patient (average number of tokens) used within the top five F1-score RAG configurations identified in the sensitivity analysis. (b) ~ (f) visualize the relationship between evaluation metrics and token usage.

Figure 4. Sensitivity analysis results: top five F1-score RAG configurations of the seven embedding models

This figure demonstrates secondary RAG sensitivity analysis results, visualizing the top five RAG configurations ranked by F1-score, for each embedding model. (b) illustrates F1-score performance based on the interaction among chunk size, the number of retrieved chunks, and the embedding models used.

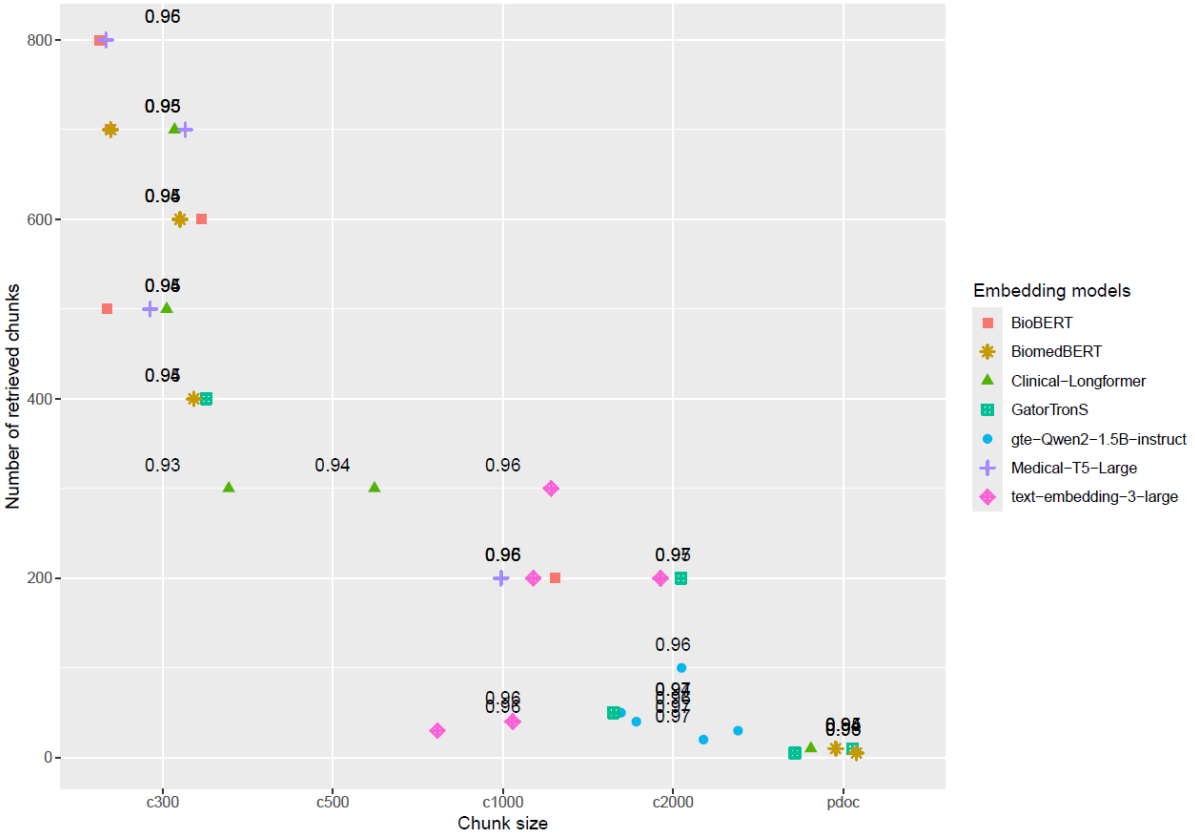


Figure 5. Sensitivity analysis results: F1-score and token usage by embedding model size

This figure shows token usage grouped by embedding model dimension (small: 512, medium: 768, large: >= 1,000).

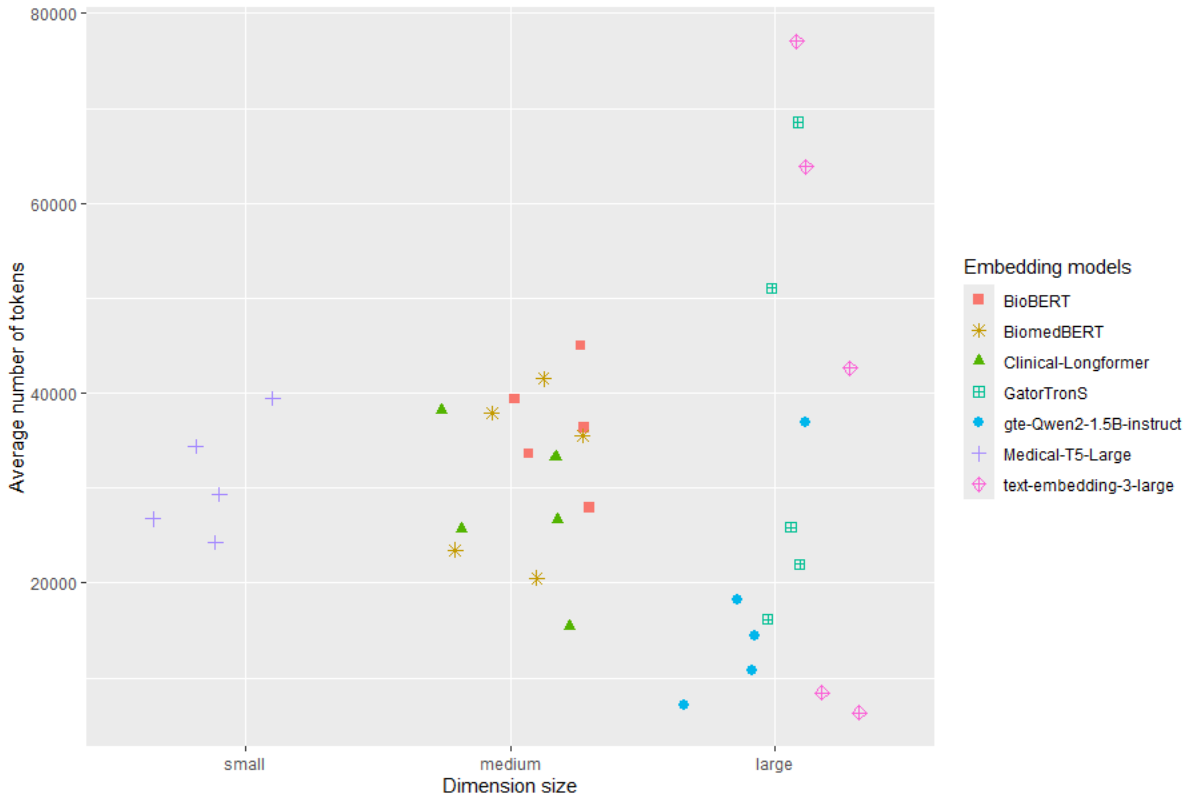
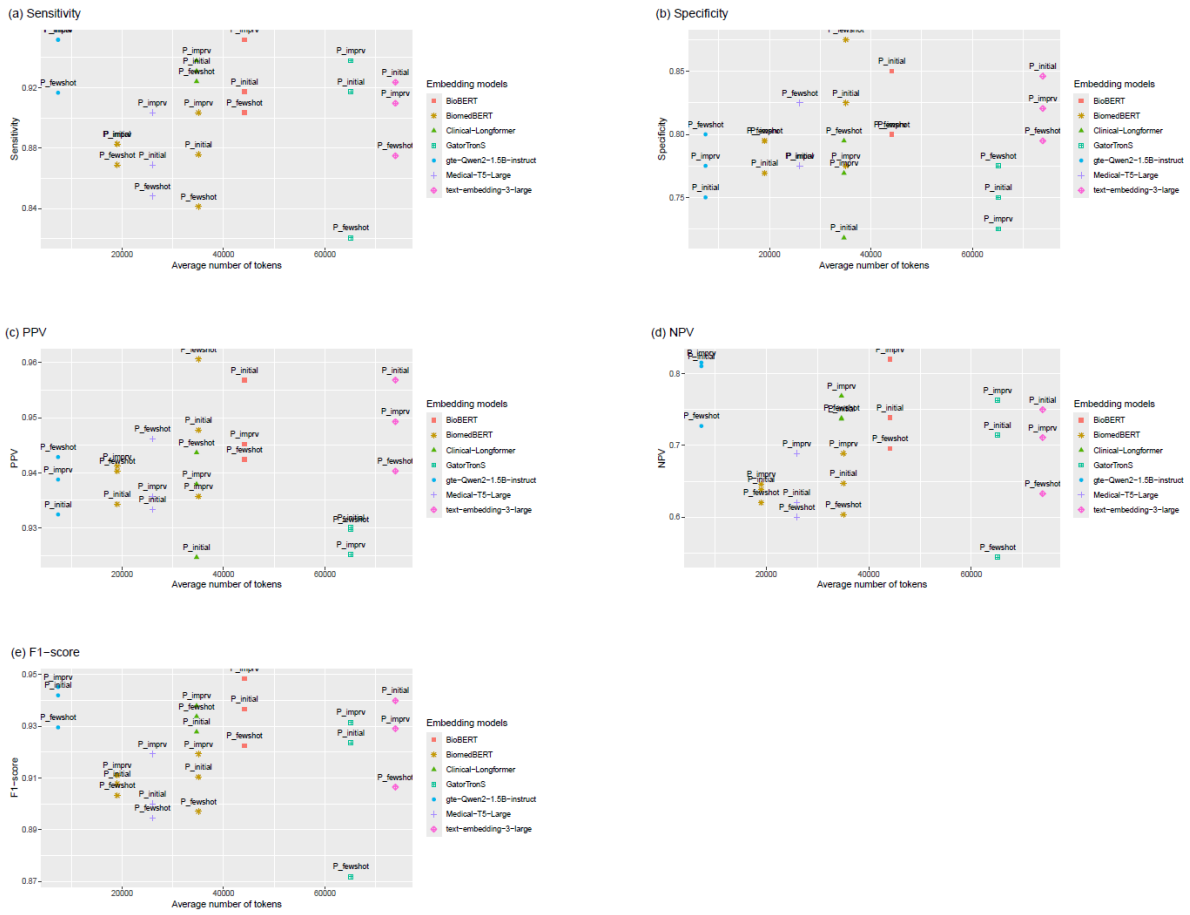


Figure 6. T2DM phenotyping validation results

These are visualizations of optimal RAG's T2DM phenotyping results validated on test patient dataset. The graphs highlight performance and cost aspects of the test settings. To focus on the differences, each figure uses different y-axis scale. The *average number of tokens* means average number of tokens used per patient. $P_{_initial}$, $P_{_imprv}$, and $P_{_fewshot}$ stand for the initial prompt, improved zero-shot prompt, and few-shot prompt used in this experiment.