Title: Evaluation of T2DM Phenotyping Using Optimized Retrieval-Augmented Generation (RAG) and the Impact of Embedding Model, Context, and Prompt

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

Identification of patient cohorts from EHRs is challenging because ICD codes primarily serve billing and may misrepresent disease status, while key information is buried in unstructured notes. Existing computed phenotyping methods also have limitations in maintenance and incomplete modeling. We evaluated GPT-4o's type II diabetes mellitus (T2DM) phenotyping ability using optimized Retrieval-Augmented Generation (RAG).

Methods

We built a RAG pipeline and clinical notes were loaded for 275 patients screened by T2DM ICD codes. We optimized chunk size and top-k across seven embedding models, testing 308 RAG configurations using training patients. Prompts (zero-shot and few-shot) were developed via error analysis. GPT-4o's phenotyping performance was evaluated against ICD codes and PheNorm, within the optimized RAG framework. Token usage and sensitivity to key hyperparameters were also assessed.

Results

GPT-4o with optimized RAG significantly outperformed ICD in precision (PPV: 0.940), and PheNorm in sensitivity (0.902), NPV (0.697), and F1 (0.920), while PPV was slightly lower and specificity (0.791) needs improvement compared to PheNorm. General embedding models and zero-shot prompt presented better sensitivity, NPV, and F1-scores, while domain-specific models and a few-shot prompt excelled in specificity and PPV. Optimization enabled lower-ranked embedding models to achieve comparably good performance to the highest ones. Gte-Qwen2-1.5B-instruct and GatorTronS provided the highest token-efficiency in specific metrics. Error analysis revealed contextual misinterpretation and ranking issues.

Conclusion

GPT-4 using optimized RAG showed superior in T2DM phenotyping in key metrics. This study provides valuable insights into practical guidance of using RAG, while identifying limitations in errors LLM reasoning and retrieval ranking.

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 the diagnosis, which is laborious and time consuming. In addition, many health problems are recorded only in clinical notes. These challenges highlight the need for derived phenotypes not just based on coded data.

Mass General Brigham (MGB) has been providing computed phenotypes to internal researchers through the *Research Patient Data Registry (RPDR)* [3 4] and the *Biobank Portal* [5]. These phenotypes were developed by various natural language processing and/or machine learning methods [6-12]. One challenge of current phenotyping methods is maintaining up-to-date performance, as cohort patterns evolve over time (e.g., use of medications). Furthermore, in the process of phenotype development with the existing approaches, some disease domains fail to reach the performance threshold required for deployment. 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 [14-19].

Retrieval-Augmented Generation (RAG) [21] is an effective framework for enhancing LLM performance. By incorporating relevant information retrieved from a local vector database into a LLM's context, it enables efficient handling of large-scale patient data with improved accuracy and reduced hallucinations. Recent studies show the use of RAG in the medical field, such as phenotyping [17], clinical trial eligibility screening [22], information extraction and summarization [24], decision support [25 26], personalized medicine [27], and clinical Q&A [28 29]. For example, [24] report that RAG-based summarization produced significant improvement than traditional generative methods.

Despite its promise, RAG performance is sensitive to its numerous hyperparameters, such as embedding models, chunk size, top-k retrieved chunks [33-36]. Initial efforts to address RAG optimization in the medical domain are found in several studies [29 37-40]. However, they are mostly

focused on individual factors, such as comparing embedding models or evaluating different LLMs, without considering the interactive effects among multiple parameters [37].

In this paper, we aim to evaluate LLM's phenotyping ability using optimized RAG, focused on type II diabetes mellitus (T2DM). Specifically, we investigate GPT-4o's performance, optimized by multiple RAG hyperparameters, against ICD codes and an established phenotyping method, PheNorm [6]. We further investigate the collective impact of key RAG hyperparameters and prompt engineering on phenotyping performance and cost-effectiveness, providing practical insights into building LLM-based phenotyping pipelines.

METHODS

Overall study design

The T2DM phenotyping performance of GPT-4o was evaluated within a RAG framework, using clinical notes from patients whose T2DM status was previously validated by physicians. We examined the performance across seven embedding models using three prompt strategies: a baseline prompt without prompt engineering, a zero-shot prompt, and a few-shot prompt. For each embedding model, an individually optimized RAG configuration, using adjusted chunk size and top-k, was implemented to minimize potential bias. The results were compared with ICD codes and the existing PheNorm phenotyping method, and the cost-effectiveness of the embedding models was investigated by token usage. Furthermore, we analyzed RAG sensitivity to key parameters, including embedding models, chunk sizes, top-k, and prompts. The overall process is illustrated in Figure 1.

Data description

The dataset consists of 300 Biobank Portal patients with T2DM ICD codes, whose T2DM diagnosis was validated by physicians in a previous study [41]. As described in [42], due to the low expected prevalence of a phenotype in the EHR population, which would significantly limit the PPV, the patients were randomly selected from the pre-screened patient set by ICD codes. Data floor (e.g., at least one note) was also applied, to ensure minimum amount of records. The T2DM ICD code set and the data floor is described in Appendix A.

We divided the patients into training and test sets of 100 and 200 patients respectively. Among the T2DM diagnosis Yes / No / Indeterminate gold labels, indeterminate cases were filtered out, resulting in 90 training and 185 test patients. Training patients were used for prompt engineering and RAG optimization, and test patients for phenotyping evaluation. We used all patient notes available from the *MGB RPDR Note Repository*, without restricting note types or time period. Total 170,678 clinical notes were loaded into our RAG system.

Retrieval-Augmented Generation (RAG) framework and implementation

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 (top-k) 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 locally, within the MGB network.

Embedding models

Embedding models influence retrieval quality and context representation, which are critical for RAG performance. Seven embedding models (Table 1) 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.

Table 1. Experimental design: vector embedding models

Embedding model	Dimension size	Dimension size category	Base model	(Bio-) medical	General	Commercial	Open source
text-embedding-3-large	3,072 (2,000 after reduction)*	large	Unknown		v	v	
gte-Qwen2-1.5B-instruct [1]	1,536	large	GTE (General Text Embedding) [1]		v		v
GatorTronS [2]	1024	large	BERT [13]	v			v
BiomedBERT [20]	768	medium	BERT	v			v
BioBERT [23]	768	medium	BERT	v			v
Clinical-Longformer [30]	768	medium	Longformer [31]	v			v
Medical-T5-Large	512	small	T5[32]	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.

Prompt engineering

We developed a zero-shot prompt and a few-shot prompt based on the error analysis of RAG output. For zero-shot prompt engineering, we began with a baseline prompt ($P_{initial}$) that consists of a direct question and straightforward output instruction: *"Has the patient been diagnosed with type 2 diabetes? Answer YES or NO. JSON formatted response: {"Answer" : "YES | NO", "Reason": "Reason Here"}"*. $P_{initial}$ was tested across simple configurations (Table 2 (a)) using the *OpenAI*'s *text-embedding-3-large* embedding model. A software engineer reviewed the False Positive (FP) and False Negative (FN) cases, and the insights were incorporated into a new prompt iteratively. The refined prompt was tested again in the same RAG environment to confirm performance improvements, resulting in finalized zero-shot prompt, P_{imprv}. This improved P_{imprv} was used for RAG optimization process.

For few-shot prompt engineering, two physicians reviewed FP and FN cases produced from the optimization process which uses seven embedding models and P_{imprv} (details in the next section). We narrowed the scope into the errors produced by optimal RAG configurations. The reviewers analyzed context chunks and their metadata, LLM-generated answer and reason, and original patient notes. They documented the final T2DM classification (YES / NO), reviewer's comments, key chunk IDs, and the evaluation of retrieval failure (i.e. failure to retrieve relevant evidence present in the notes) and context misinterpretation (i.e., misinterpretation of correctly retrieved relevant chunks) and human error (YES / NO).

A software engineer then curated a few-shot prompt, $P_{fewshot}$, using the context misinterpretation instances. 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. All prompts used in this study are available in Appendix B.

Table 2. Configurations used for zero-shot prompt engineering and RAG optimization

(a) RAG configurations used for zero-shot prompt engineering

Prompt	Baseline prompt ($P_{initial}$)							
Embedding model	Text-embedding-3-large							
Chunking method	**chunk size**	**number of chunks**						
	1000 (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 used for RAG optimization

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					
	1000	5	10	20	40	50	100	200	300					
	2000	5	10	20	40	50	100	200						
	3000	5	10	20	40	50	100							
	pdoc	5	10											

RAG optimization

We optimized our RAG system by systematically testing the combined configurations of various embedding models, chunk sizes and the number of top-k chunks retrieved (Table 2(b)). For each embedding model, P_{imprv} was tested across 44 different "chunk size + top-k" configurations on the training patient set. Chunks sizes varied from small (c300) to larger chunks (up to c3,000) and entire documents (pdoc), and their top-k was incrementally increased until reaching GPT-4o's maximum context window. A configuration with the highest F1-score (a balanced metric of positive predictive value (PPV) and sensitivity) was then selected as optimal. If the selected configuration failed to process any patient due to a context length limitation, the next best-performing configuration without error was also chosen for that embedding model.

Evaluation

To assess GPT-4o's performance in T2DM phenotyping using RAG, $P_{initial}$, P_{imprv}, $P_{fewshot}$ were tested across optimized RAG configurations using the test patients. The LLM's PPV was compared to the precision of ICD T2DM codes. Sensitivity, specificity, PPV, negative predictive value (NPV), and the F1-score to PheNorm [6], MGB's primary phenotyping method. RPDR T2DM PheNorm data was obtained, which version was backed up by the month of chart review. To evaluate embedding models' efficiency in terms of cost, we conducted token usage analysis by calculating average number of tokens used per patient for each optimal configuration.

Furthermore, a RAG sensitivity analysis was performed. First, we analyzed the collective impact of an embedding model, chunk size, and top-k. For each embedding model, the top five best performing results obtained during the RAG optimization process were investigated. Sensitivity, specificity, PPV, NPV, F1-score, and token usage were used to discover relationships among the RAG hyperparameters. Second, we explored the combined effects of a prompt and an embedding model by analyzing the phenotype validation results.

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. Our source note repository provides 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 during the RAG retrieval process. It caused false result 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, 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 optimization results

The highest F1-score configurations for seven embedding models were selected. For BiomedBERT, the second top-performing configuration was chosen additionally, since the best one that used whole parent documents (pdoc, 5) failed to process one patient due to the context length limitation. Total eight RAG configurations were identified as optimal and summarized in Table 3.

Table 3. Optimal RAG configurations

Embedding model	Chunk size	Number of chunks
BioBERT	300	800
BiomedBERT	300	600
BiomedBERT	pdoc	5
Clinical-Longformer	pdoc	10
GatorTronS	2000	200
Gte-Qwen2-1.5B-instruct	2000	20
Medical-T5-Large	1000	200
Text-embedding-3-large	2000	200

Few-shot prompt engineering results

We analyzed 43 FP and FN results produced by P_{imprv} within the optimal RAG configurations, across 18 unique patients. Two human errors, four indeterminate, three retrieval failures, and nine context

misinterpretations were identified. Retrieval failure cases were observed in specific embedding models: Clinical-Longformer, BioBERT, and BiomedBERT.

Regarding context misinterpretation, GPT-4o often failed 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.

The key findings were used as few-shot examples. 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. $P_{fewshot}$ was built by incorporating them to the P_{imprv} (see Appendix B-3).

T2DM phenotyping evaluation results using the optimized RAG configurations

Phenotyping T2DM using the three prompts on the optimized RAG configurations for seven embedding models achieved a sensitivity of 0.902 (95% CI: 0.887–0.917), specificity of 0.791 (95% CI: 0.775–0.807), PPV of 0.940 (95% CI: 0.936–0.944), NPV of 0.697 (95% CI: 0.665–0.728), and F1-score of 0.920 (95% CI: 0.912–0.928). The PPV demonstrates a superior performance compared to the precision of T2DM ICD codes (0.7838). Furthermore, the optimized RAG outperformed PheNorm in sensitivity (PheNorm's sensitivity: 0.6000), NPV (0.3830), and F1-score (0.7373). In contrast, specificity (0.9000) and PPV (0.9560) were higher in PheNorm.

Figure 2 visualizes individual configuration performance and token usage at the prompt – embedding model levels (detail values are provided in Appendix C). 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 achieved the best NPV (0.8205) and F1-score (0.9485). The highest specificity (0.8750) and PPV (0.9606) were attained by BiomedBERT (c300, 600) when using $P_{fewshot}$.

GPT-4o significantly outperformed PheNorm in sensitivity, NPV, and F1 types when used with optimized RAG setting regardless of prompt types. Sensitivity improved by up to 0.3517, NPV by 0.4375, and F1-score by 0.2112, at most. For PPV, most of the configurations presented lower PPV performance than the PheNorm, but the difference was minor, not exceeding 0.0313. Specificity in all configurations was lower than PheNorm, with gaps from 0.0250 up to 0.1821. These results are attributed to the RPDR PheNorm phenotyping cut-off value setting.

Among all tested embedding models, gte-Qwen2-1.5B-instruct used the least tokens to construct an optimized RAG environment. Notably, gte-Qwen2-1.5B-instruct achieved the highest or similar performance in sensitivity, NPV, and F1-score using 7,371.85 tokens, while BioBERT used 40,000 tokens. The embedding model which used the most tokens for optimization was text-embedding-3-large, requiring 73,890.35 tokens per patient in average.

RAG sensitivity analysis results

Impact of embedding model + chunk size + top-k

The performance and token usage of the top five F1-score configurations for each embedding model were analyzed to investigate the collective impacts of embedding model, chunk size, and top-k. Figure 3 visualizes the performance and token usage of the evaluated configurations. It shows that the two large, generic embedding models consistently achieve exceptional performance in sensitivity, NPV, and F1-score. The top ten highest sensitivity and NPV and the top eight F1-score were achieved by text-embedding-3-largeand and gte-Qwen2-1.5B-instruct. In contrast, certain biomedical-specific models excelled in specificity and PPV. BiomedBERT, Medical-T5-Large, GatorTronS, and Clinical-Longformer achieved up to 1.0000 in both metrics, while the best general embedding model (text-embedding-3-large) ranked 20th, demonstrating specificity of 0.9130 and PPV of 0.9701.

Notably, when optimized, small and medium-sized biomedical models – BioBERT, Medical-T5-Large, and BiomedBERT – achieved F-scores of 0.9618, differing by less than 0.01 from the best-performing model (0.9701), indicating comparable performance. 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 optimal chunk size and top-k combination varied per embedding model (Figure 4). 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, all (bio-) medical embedding models performed well with short chunks (c300), but required larger number of chunks (at least 300 chunks). While the large (bio-) medical model, GatorTronS, mostly outperformed with large chunks, the small- and medium- sized (bio-) medical embedding models tended to perform well with smaller chunks. Some medium-sized models, BiomedBERT and Clinical-Longformer, showed high performance in more diverse settings: either with small chunks or pdoc. None of the c3000 configurations ranked among the top five.Figure 5 illustrates the average token count per patient for each configuration, categorized by embedding dimensions. All small- and medium-sized embedding models used a narrow token range, whereas large embedding models demonstrated considerable variability. As shown in Figure 3, some large embedding models achieved equivalent or superior performance using a substantially lower number of tokens. For example, text-embedding-3-large tended to perform better in F1-score with increasing token counts, reaching its peak performance at an average of 77,070.05 tokens, while most gte-Qwen2-1.5B-instruct maintained strong performance with fewer than 20,000 tokens. Furthermore, both models achieved the highest F1-scores, but gte-Qwen2-1.5B-instruct accomplished this using less than 10% of the tokens. For specificity and PPV, GatorTronS achieved the best performance using the lowest token counts among all models (see Appendix D for detailed token counts).

Impact of prompt + embedding model

The zero-shot prompt and the few-shot prompt presented different impacts on the evaluation metrics. In Figure 2, P_{imprv} boosted performance in sensitivity, NPV, and F1-score compared to the baseline prompt, except text-embedding-3-large model. The largest performance improvement observed was a 0.0814 increase in NPV, when paired with BioBERT. However, in gte-Qwen2-1.5B-instruct and BiomedBERT (pdoc, 5), the effect of zero-shot prompt engineering was minor. For specificity and PPV, P_{imprv} did not demonstrate notable association. Only approximately half of the experimented configurations had performance enhancements, which effects were mostly minor. One exception was Clinical-Longformer, exhibiting a more significant 0.0513 improvement in specificity

In contrast, $P_{fewshot}$ made improvements in specificity and PPV across a majority of the configurations. However, NPV and F1-score were lower than the baseline, except for the Clinical-Longformer model. No embedding models had improvement in sensitivity. Clinical-Longformer demonstrated 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 $P_{fewshot}$.

Discussion

We evaluated GPT-4o's T2DM phenotyping performance using optimized RAG configurations across seven embedding models. The results show a strong capability of GPT-4o for T2DM phenotyping task. However, naïve application of RAG would not guarantee high performance consistently. In the entire RAG experimental results, not limited to the top-ranked RAG configurations, the variability between the highest and the lowest performance was significant. The maximum performance difference within an embedding model reached up to 0.7910 gap in the Clinical-Longformer's sensitivity (see Appendix E). In this study, by analyzing the collective impact of prompt, context, and embedding models on RAG performance, we provide insights for setting up an effective RAG configuration in phenotyping. These findings are valuable as the analysis was grounded in real-world, clinician-validated patient data, encompassing all notes collected from our institution's multi-hospital system.

In the 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 was the top-ranked model, under the 2,000-dimensional constraint, in the *Massive Text Embedding Benchmark (MTEB) Leaderboard [43]* for both overall and retrieval performances 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, which dimension size is the largest in the tested domain-specific embedding models, did not outperform the others.

In contrast, (bio-) medical models showed superior results in specificity and PPV. Furthermore, although the top performance in some metrics were achieved by the general models, (bio-) medical specific models demonstrated comparably reliable performance by configuring proper hyperparameter values (i.e., chunk size and the top-k). These findings align with our previous study on ischemic stroke[44], which compared text-embedding-3-large and BiomedBERT for a phenotyping task. This suggests that, to maximize LLM's capability using RAG system, an embedding model should be selected based on use cases (i.e., considering prioritized metrics).

The RAG sensitivity analysis presented clusters of optimal hyperparameter values for each embedding model. When used with P_{imprv}, larger chunks generally benefited large embedding models. However, smaller chunks were more effective in small- and medium-sized embedding models. In addition, we found that all (bio-) medical specific embedding models produced high performance with small-size chunks, if sufficient number of chunks are provided as a context. Notably, RAG configurations using c3000 chunks were not included in the top-performing configuration list, while five of them used pdoc, which size is mostly larger than c3000. Moreover, it revealed that some models could achieve similar results using significantly less number of tokens, 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.

Error 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 tried to solve it with few-shot prompt. This approach was successful in improving specificity and PPV, but 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. 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.

This paper highlights the potential of LLMs to enhance the phenotyping process to improve phenotyping process and provide useful insights into their application, considering performance and cost-effectiveness. We plan to incorporate LLM technology into our institutional phenotyping pipeline. However, 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 expanding phenotyping performance evaluation to additional disease domains and evaluating additional RAG hyperparameters to gain deeper insights.

Conclusion

This study explored LLM's T2DM phenotyping efficacy, within a RAG framework, to advance phenotype identification using real-world clinical notes. We evaluated GPT-4o on the optimized RAG environment, combined with seven embedding models. For RAG optimization, the collective impacts of embedding models, chunk size, top-k, and various prompt engineering approaches were examined and optimal configurations were selected based on F1-score. 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 careful hyperparameter optimization with consideration of token usage and prompt refinement would be the key for reliable phenotyping.

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 observed that the two tested general-purpose embedding models exhibited superior performance in sensitivity, NPV, and F1-score, while domain-specific models excelled in specificity and PPV, highlighting the importance of aligning model choice with task priorities. Prompt design also had a crucial impact: zero-shot prompt was more effective in sensitivity, NPV, and F1, while few-shot prompt outperformed in specificity and PPV.

Results further indicate that even embedding models that are not the top- ranked can achieve comparable performance by hyperparameter tuning. Optimal chunk size varied by embedding model and prompt, and smaller models tended to require substantial number of chunks to reach high performance. Token usage analysis demonstrated that certain models, such as gte-Qwen2-1.5B-instruct and GatorTronS, can deliver similar outcomes at significantly lower costs, which is critical for scalability in large cohorts.

Error analysis revealed limitations in LLM reasoning and retrieval ranking, suggesting needs for improved prompt design and retrieval strategies. This study is limited to a single disease domain and one LLM. Future studies will expand hyperparameter investigation into diverse phenotypes. Ultimately, integrating optimized LLM-driven phenotyping into institutional pipelines could significantly reduce manual chart review burden and enhance cohort identification for clinical research.

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.

Declaration of Competing Interests

The authors have no conflicts of interest to disclose.

REFERENCES

1. Li Z, Zhang X, Zhang Y, Long D, Xie P, Zhang M. Towards general text embeddings with multi-stage contrastive learning. arXiv preprint arXiv:2308.03281 2023.
2. 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.
3. 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.
4. Nalichowski R, Keogh D, Chueh HC, Murphy SN. Calculating the benefits of a Research Patient Data Repository. AMIA Annu Symp Proc 2006;**2006**:1044.
5. 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):643-51 doi: 10.1093/jamia/ocab264.
6. Yu S, Ma Y, Gronsbell J, et al. Enabling phenotypic big data with PheNorm. J Am Med Inform Assoc 2018;**25**(1):54-60 doi: 10.1093/jamia/ocx111.
7. 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 doi: 10.1038/nbt.2749.
8. 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 doi: 10.1016/j.biopsych.2018.01.011 [published Online First: 20180226].
9. 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) doi: 10.3390/jpm6010011 [published Online First: 20160226].
10. 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 doi: 10.1136/bmj.h1885 [published Online First: 20150424].
11. 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 doi: 10.1136/amiajnl-2014-003040 [published Online First: 20141028].
12. 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 doi: 10.1016/j.jbi.2014.08.012 [published Online First: 20140906].
13. 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.
14. 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.
15. 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 doi: 10.1093/jamiaopen/ooae060 [published Online First: 20240703].
16. 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.
17. Garcia BT, Westerfield L, Yelemali P, et al. Improving Automated Deep Phenotyping Through Large Language Models Using Retrieval Augmented Generation. medRxiv 2024 doi: 10.1101/2024.12.01.24318253 [published Online First: 20241202].

18. 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.
19. 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-A46 doi: doi:10.1161/circ.150.suppl_1.4145446.
20. 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.
21. 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.
22. Unlu O, Shin J, Mailly CJ, et al. Retrieval-Augmented Generation–Enabled GPT-4 for Clinical Trial Screening. NEJM AI 2024:AIoa2400181.
23. Lee J, Yoon W, Kim S, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 2020;**36**(4):1234-40.
24. Mohammad A, Ping Y, Mengyang Y, Chao D. Applying generative AI with retrieval augmented generation to summarize and extract key clinical information from electronic health records. Journal of Biomedical Informatics 2024;**156**:104662 doi: https://doi.org/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 doi: 10.2196/50209 [published Online First: 20240619].
26. 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):1115-18 doi: 10.1007/s10439-023-03327-6 [published Online First: 20230802].
27. Murugan M, Yuan B, Venner E, et al. Empowering personalized pharmacogenomics with generative AI solutions. J Am Med Inform Assoc 2024;**31**(6):1356-66 doi: 10.1093/jamia/ocae039.
28. Zakka C, Shad R, Chaurasia A, et al. Almanac - Retrieval-Augmented Language Models for Clinical Medicine. Nejm ai 2024;**1**(2) doi: 10.1056/aioa2300068 [published Online First: 20240125].
29. Elgedawy R, Srinivasan S, Danciu I. Dynamic Q&A of Clinical Documents with Large Language Models. arXiv preprint arXiv:2401.10733 2024.
30. 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):340-47 doi: 10.1093/jamia/ocac225.
31. Beltagy I, Peters ME, Cohan A. Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150 2020.
32. 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.
33. 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, 2024:194–99.
34. AutoRAG-HP: Automatic Online Hyper-Parameter Tuning for Retrieval-Augmented Generation; 2024 November; Miami, Florida, USA. Association for Computational Linguistics.
35. 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:2502.18635 2025.
36. Juvekar K, Purwar A. Introducing a new hyper-parameter for RAG: Context Window Utilization. arXiv preprint arXiv:2407.19794 2024.

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):2355-74.
38. Benchmarking Retrieval-Augmented Generation for Medicine; 2024 August; Bangkok, Thailand. Association for Computational Linguistics.
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. Si Y, Wang J, Xu H, Roberts K. Enhancing clinical concept extraction with contextual embeddings. J Am Med Inform Assoc 2019;**26**(11):1297-304 doi: 10.1093/jamia/ocz096.
41. 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.
42. Liao KP, Cai T, Savova GK, et al. Development of phenotype algorithms using electronic medical records and incorporating natural language processing. bmj 2015;**350**.
43. Enevoldsen K, Chung I, Kerboua I, et al. Mmteb: Massive multilingual text embedding benchmark. arXiv preprint arXiv:2502.13595 2025.
44. Optimizing Retrieval-Augmented Generation (RAG) for Retrospective Ischemic Stroke Identification: A Comparative Study of Embedding Models and Retrieved Chunks. AMIA Informatics Summit; 2025; Pittsburgh.
45. Calibrate before use: Improving few-shot performance of language models. International conference on machine learning; 2021. PMLR.
46. Making Pre-trained Language Models Better Few-shot Learners; 2021 August; Online. Association for Computational Linguistics.
47. Schick T, Schütze H. True Few-Shot Learning with Prompts—A Real-World Perspective. Transactions of the Association for Computational Linguistics 2022;**10**:716-31 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 doi: 10.1016/j.jbi.2023.104458 [published Online First: 20230723].

Figure 1. Overall experimental design

This figure demonstrates overall experiments. First, zero-shot prompt was developed by incorporating error analysis results from an initial baseline prompt and simplified RAG settings. Using the improved zero-shot prompt, RAG optimization process explored total 308 combinations of embedding model, chunk size, and the number of retrieved chunks (top-k) by systematically incrementing the size of chunk and top-k for each embedding model. The optimal configurations for seven embedding models were selected based on F1-score. Few-shot prompt engineering was performed by analyzing false positive and false negative cases produced during RAG optimization, leading to clinician-verified examples for enhanced prompting. Then, GPT-4o's T2DM phenotyping performance was evaluated using the optimized configurations and the three prompt strategies (baseline, zero-shot, and few-shot) against ICD codes and PheNorm. To assess cost-effectiveness, token usage analysis was performed. Finally, a RAG sensitivity analysis examined the collective influence of embedding model, chunk size, top-k, and prompt choice on performance and token efficiency.

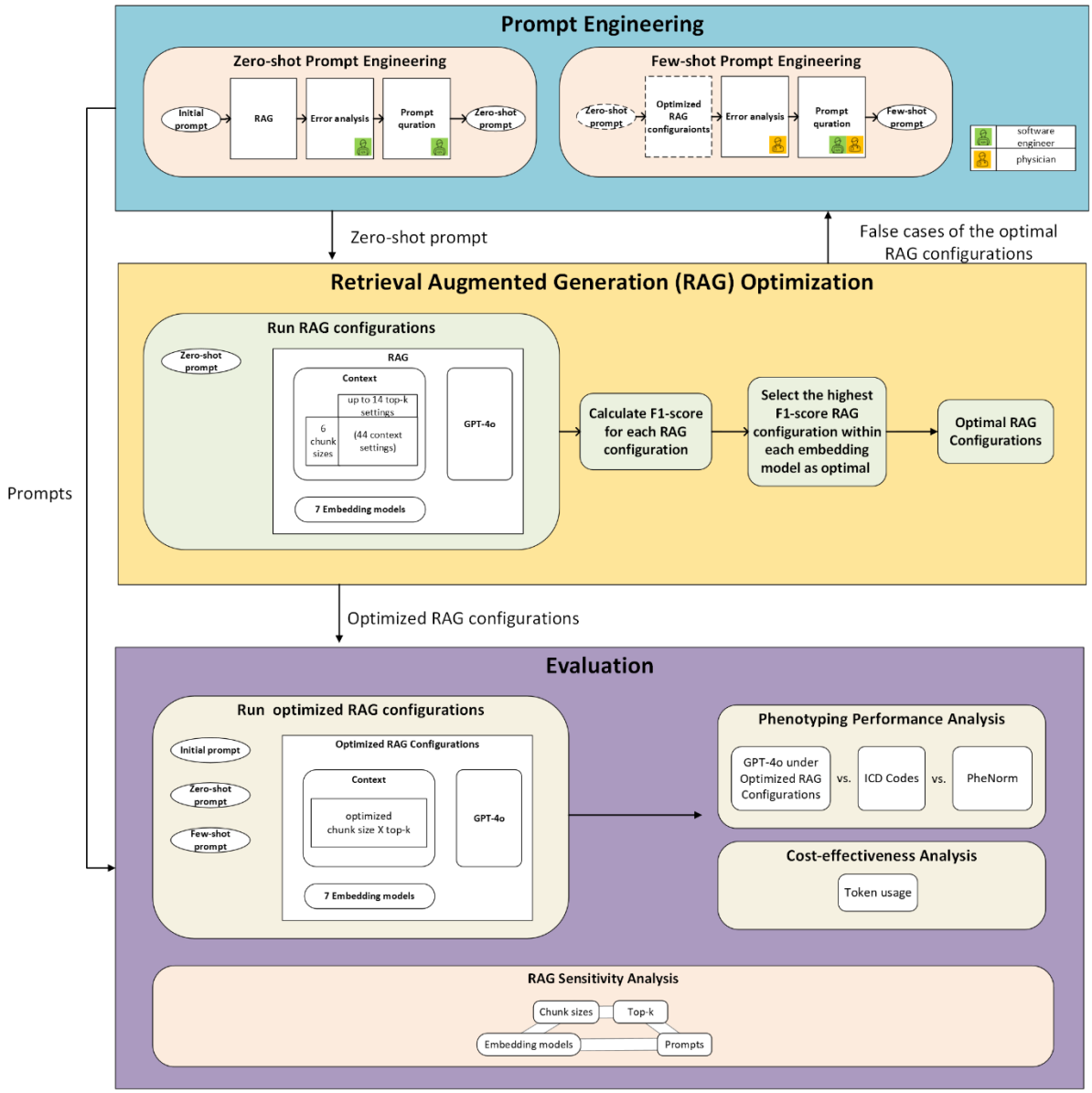

Prompt Engineering
Zero-shot Prompt Engineering
Initial prompt
RAG
Error analysis
Prompt quration
Zero-shot prompt
Few-shot Prompt Engineering
Zero-shot prompt
Optimized RAG configuraionts
Error analysis
Prompt quration
Few-shot prompt
software engineer
physician
Zero-shot prompt
False cases of the optimal RAG configurations
Retrieval Augmented Generation (RAG) Optimization
Run RAG configurations
Zero-shot prompt
RAG
Context
up to 14 top-k settings
6 chunk sizes
(44 context settings)
GPT-4o
7 Embedding models
Calculate F1-score for each RAG configuration
Select the highest F1-score RAG configuration within each embedding model as optimal
Optimal RAG Configurations
Prompts
Optimized RAG configurations
Evaluation
Run optimized RAG configurations
Initial prompt
Zero-shot prompt
Few-shot prompt
Optimized RAG Configurations
Context
optimized chunk size X top-k
GPT-4o
7 Embedding models
Phenotyping Performance Analysis
GPT-4o under Optimized RAG Configurations
vs.
ICD Codes
vs.
PheNorm
Cost-effectiveness Analysis
Token usage
RAG Sensitivity Analysis
Chunk sizes
Top-k
Embedding models
Prompts

Figure 2. 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 baseline prompt, improved zero-shot prompt, and few-shot prompt used in this experiment.

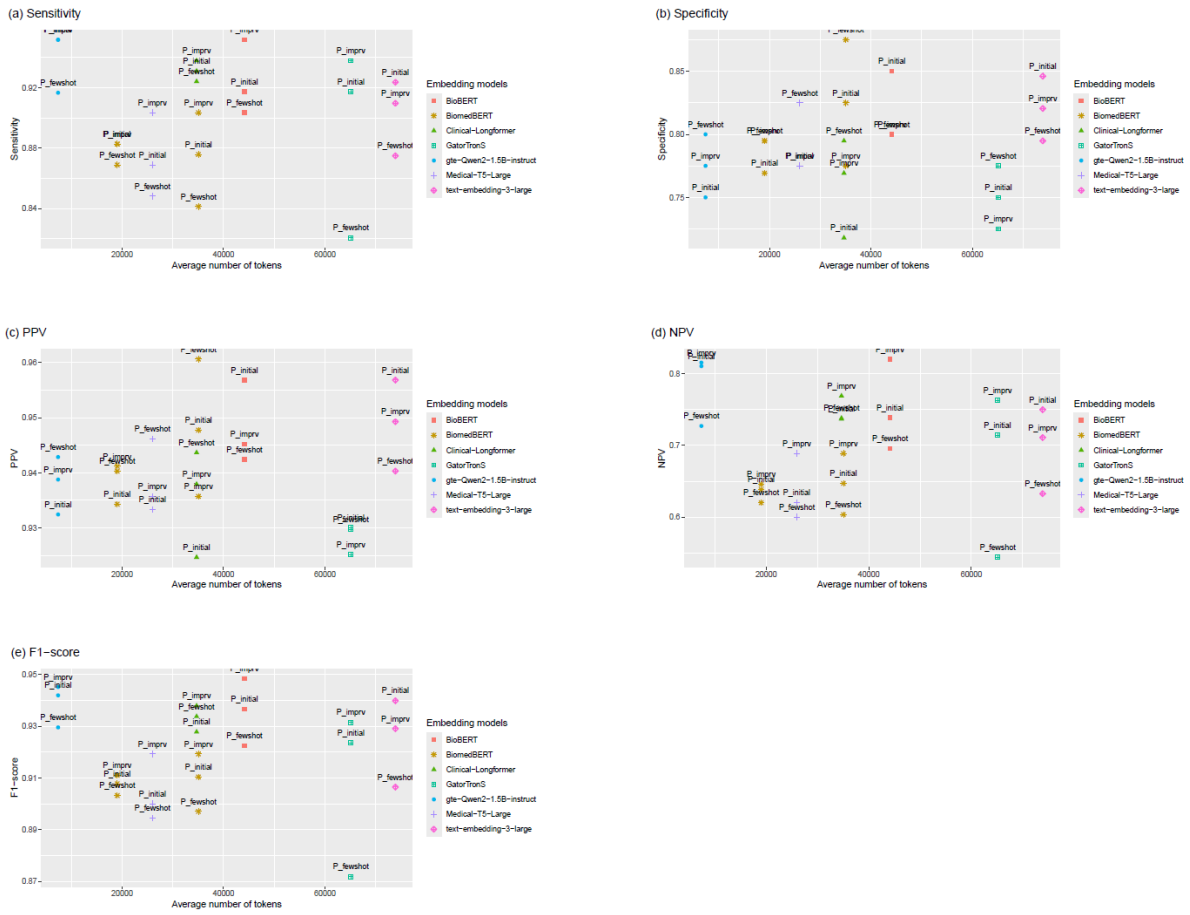
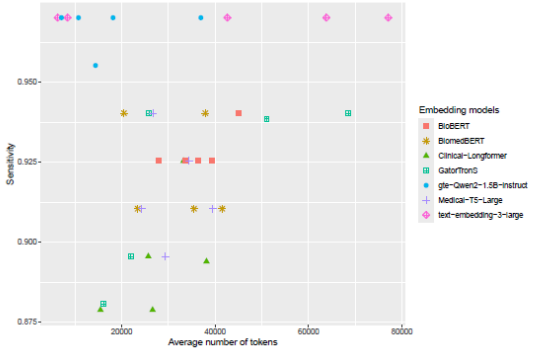


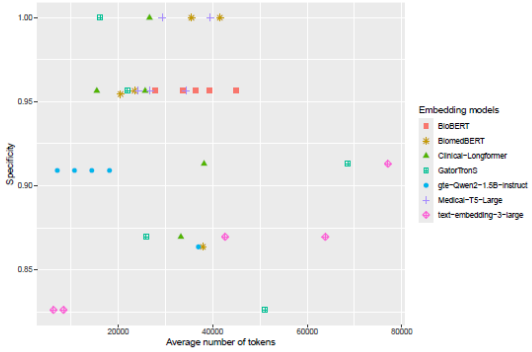
Figure 3. Sensitivity analysis results: performance and average number of tokens used

The top five F1-scoring configurations for each embedding model were analyzed to assess the combined effects of chunk size, retrieval depth (*top-k*), and embedding choice on performance and the average number of tokens used per patient. Large general-purpose models (text-embedding-3-large, gte-Qwen2-1.5B-instruct) achieved the highest sensitivity, NPV, and F1-scores, while biomedical models excelled in specificity and PPV. Optimal chunk-size and top-k varied by model, and token usage ranged widely - from fewer than 20,000 tokens for gte-Qwen2-1.5B-instruct to over 70,000 for text-embedding-3-large.

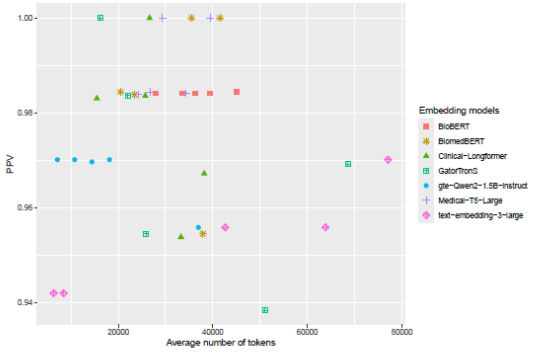
(a) Sensitivity and token usage



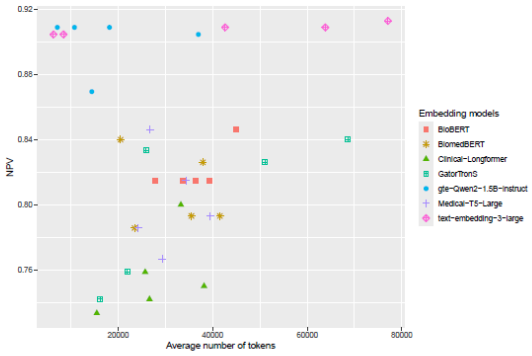
(b) Specificity and token usage



(c) PPV and token usage



(d) NPV and token usage



(e) F1-score and token usage

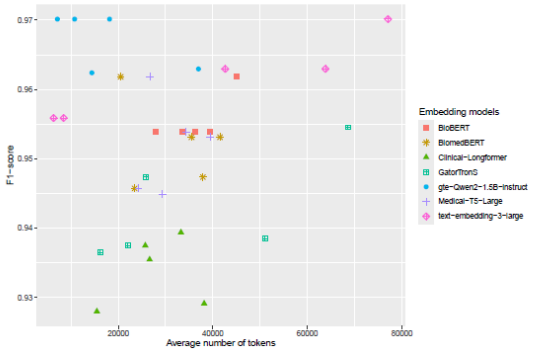


Figure 4. Sensitivity analysis results: top five F1-score RAG configurations of the seven embedding models

This figure illustrates F1-score values of the top five configurations for each embedding model, highlighting how chunk size and the number of retrieve chunks (top-k) influence performance per embedding model. Large general-purpose models performed best with longer chunks (c1000–c2000), while small and medium biomedical models favored short chunks (c300) and higher top-k.

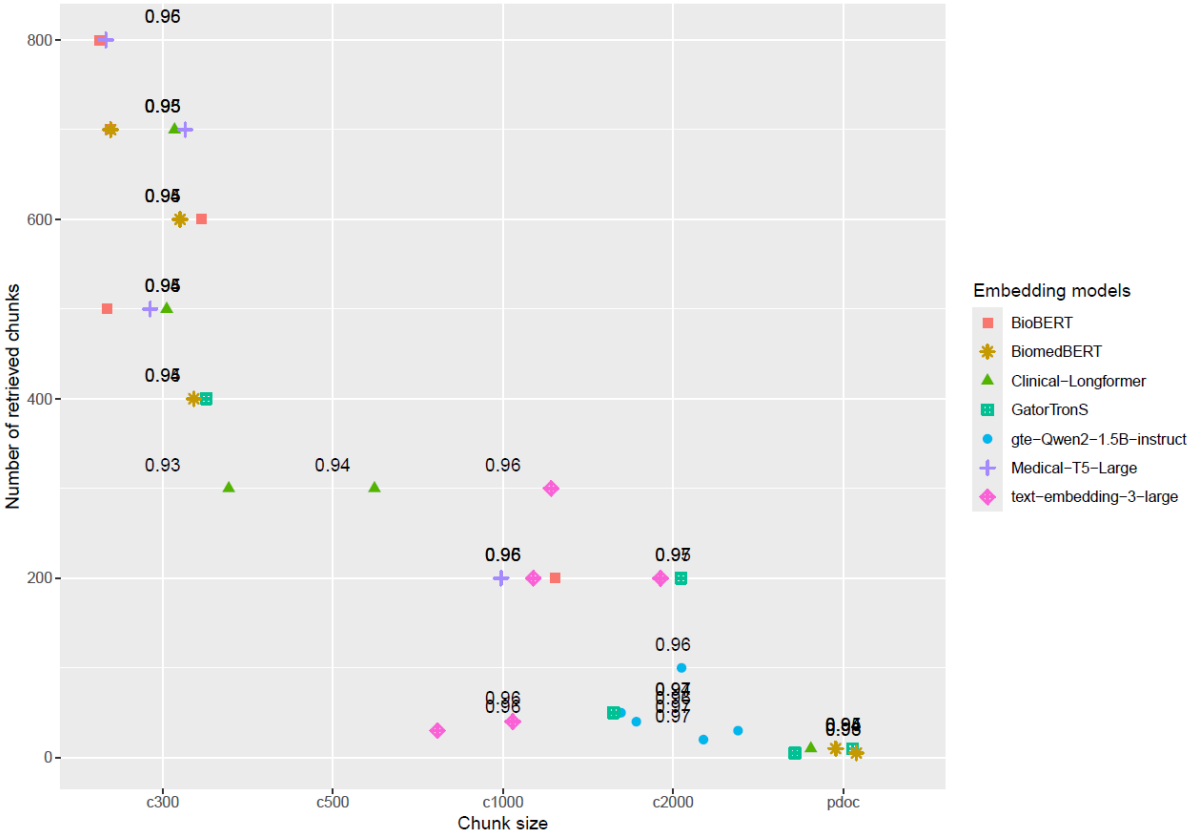
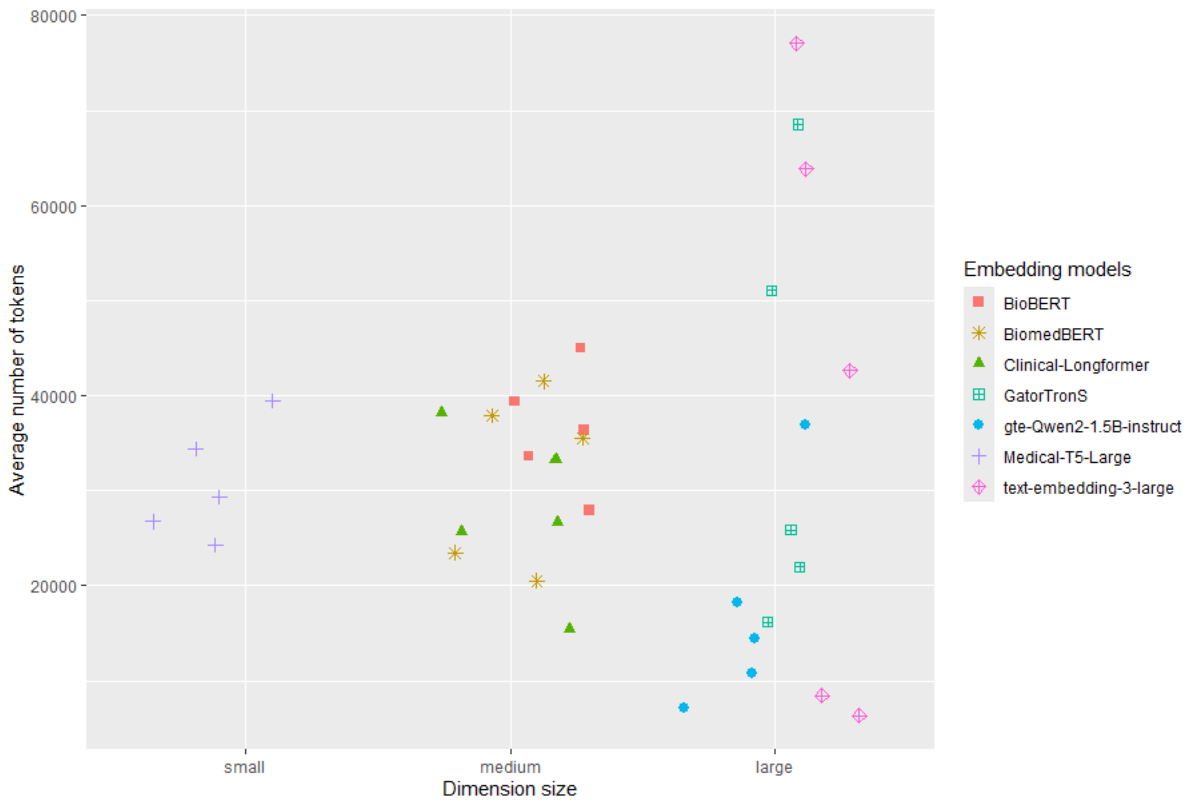


Figure 5. Sensitivity analysis results: F1-score and token usage by embedding model size

This figure highlights token usage across embedding models grouped by embedding model dimension size (small: 512, medium: 768, large: >= 1,000). Small and medium biomedical models exhibited a narrow token range, while large models showed substantial variability.