Biomedical Text Normalization through Generative Modeling

Authors: Jacob S. Berkowitz[1], Yasaman Fatapour[1], Apoorva Srinivasan[1], Jose Miguel Acitores Cortina[1], Nicholas P Tatonetti[1†]

[1]: Department of Computational Biomedicine – Cedars-Sinai

[†]: Corresponding Author

Keywords: retrieval-augmented generation, clinical text normalization, large language models, GPT-4, prompt engineering

Word count, excluding title page, abstract, references, figures and tables: 2756

ABSTRACT

Objective: Around 80% of electronic health record (EHR) data consists of unstructured medical language text. By its nature, this text is flexible and inconsistent, making it challenging to use for clinical trial matching, decision support, and predictive modeling. In this study, we develop and assess text normalization pipelines built using large-language models.

Materials and Methods: Here, we evaluated four LLM-based normalization strategies: Zero-Shot Recall, Prompt Recall, Semantic Search, and Retrieval-Augmented Generation (RAG) and one baseline, TF-IDF based String Matching. We compared normalization performance across two datasets of condition terms mapped to SNOMED, one tailored to oncology, and one covering a wide range of medical conditions. Additionally, we benchmarked our models against the TAC 2017 drug label annotations, which normalizes terms to the Medical Dictionary for Regulatory Activities (MedDRA) Preferred Terms.

Results: RAG, which effectively combines Prompt Recall and Semantic Search, was the most effective, accurately identifying the correct term 88.31% of the time for the domain-specific dataset and 79.97% for the broader dataset. Our model achieved a micro F1 score of 88.01 on task 4 of the TAC2017 conference, surpassing all other models without relying on the provided training data.

Discussion: These findings demonstrate the potential of LLMs in medical text normalization. We find that retrieval-focused approaches overcome traditional LLM limitations for this task.

Conclusion: Large language models combined with retrieval-augmented generation should be explored for text normalization of biomedical free text.

INTRODUCTION

The global transition to digital clinical records has led to a drastic increase in electronically stored unstructured data, making up around 80%(1) of the information within healthcare. This data has the potential to improve patient care, deepen our understanding of diseases, and support research. However, the different ways medical terms are used in these records make it difficult to extract meaningful insights.(2) For instance, clinical notes often use ambiguous shorthand, inconsistent structure, and domain-specific vocabularies.(3)

Text normalization is the mapping of natural language expressions of medical concepts into a standardized vocabulary.(4) Traditionally, methods have relied upon some form of string matching. These systems follow strict patterns, which can misinterpret or oversimplify clinical language.(5) This results in lost critical information and reduces data reliability for clinical decisions and research.

Semantic embedding methods such as word2vec(6) can transform words into numerical representations, capturing their contextual meanings.(7) This conversion enables computers to understand and process natural language by identifying semantic relationships between words, even when their usage varies. Quantifiable measures such as cosine similarity are used to return mapped terms that are semantically similar to the natural language term.(8) While improving on the potential for misinterpretation in string matching, there are issues when the semantic space becomes denser. As the number of terms increases the embedding space may not capture enough of the

context. The result is that the most contextually suitable word may not always appear as the top result.

Transformer-powered Large Language Models (LLMs) such as OpenAI’s GPT-4 can help with this by effectively processing and understanding complex language patterns. The transformer architecture processes data more efficiently than earlier methods, allowing it to extract meaningful insights from expansive datasets.(9) Beyond simply recognizing language structure, modern LLMs retain a human-inspired decision-making process.(10) In the case of text normalization, this added logic can help select the most relevant terms.(11)

A recent work by Wang et al.(12) has shown that fine-tuning LLMs with domain-specific corpora can significantly improve their performance in medical text normalization tasks. While effective, this approach is often task-specific and computationally expensive, limiting its scalability and generalizability across different medical domains. Shin et al.(13) found that while fine-tuning can yield high performance, in many cases, using a language model while engineering an informative prompt can record comparable performance while being more resource efficient. Prompt engineering involves the strategic formulation of input text that leverages the model's training and capabilities.(14) For instance, instead of using vague or ambiguous language, prompts are crafted to be precise, incorporating specific keywords or phrases that are known to trigger the most accurate responses from the model.

Within prompt engineering, retrieval-augmented generation (RAG) is an approach to enhance model adaptability with minimal added computation.(15) RAG enables systems to extend beyond their inherent knowledge, using external databases to source relevant context when generating text responses. Retrieval-focused methodology has shown promising results in enhancing the quality and applicability of LLM outputs across diverse applications.(16–18)

In this study, we develop and evaluate several methodological approaches to text normalization, aligning medical terms with those provided by Systematized Nomenclature of Medicine—Clinical Terms (SNOMED CT).(19) For validation, we tested our approach on Task 4 of the TAC 2017 Adverse Drug Reaction Extraction from Drug Labels Track.(20) Overall, we found that a RAG-based approach performs the strongest on both tasks.

METHODS

Dataset Descriptions

Domain Specific

We created a specialized dataset focusing on oncology text normalization from the OMOP (Observational Medical Outcomes Partnership) database at our institution. This dataset consists of 106 SNOMED CTs descendants of the concept "Malignant neoplastic disease," each linked to specific ICD-10 billing codes. These codes are commonly utilized in our facility and observed in the records of over 1,000 patients each. We used SQL Server to pull data securely from our institutional database.

We then used GPT-4 (version=gpt-4-0125-preview, temperature=0, max new tokens=256), to generate synonyms for each oncology term to simulate the terminology encountered in real-world clinical environments. We fed GPT-4 the original terms and asked it to produce exactly 10 synonyms for each, focusing on obtaining synonyms that are semantically identical to the original terms but are linguistically diverse.

Cross-Domain

To evaluate the robustness of our text normalization models across a wider spectrum of medical terminology, we generated a second dataset comprising a broad array of terms. From the approximately 40,000 terms that have appeared at least once in our institutional billing codes, we randomly selected 750 non-domain-specific terms. Similarly to the domain specific dataset, we used GPT-4 to create 10 synonyms for each term.

Evaluation Metrics: In our evaluation of the methods for normalizing medical terms to their respective SNOMED CT codes, we primarily focused on the accuracy of each method. In addition to accuracy, we monitored the number of API calls made to GPT-4 for each method, serving as an indicator of computational efficiency. As another metric of efficiency, we considered the average number of tokens per API call. This metric reflects the efficiency and cost of the communication with the API, as API pricing is based on the number of tokens processed in each call.

TAC2017 Dataset Description

To validate our results, we used the TAC2017 Adverse Drug Reaction (ADR) dataset.(20) This dataset includes a normalization task for adverse event terms mentioned on drug labels to the standardized Medical Dictionary for Regulatory Activities (MedDRA) vocabulary. The dataset includes a set of 200 annotated drug labels with identified adverse reaction terms and their MedDRA mappings. This task is in line with our study's goal of evaluating text normalization methods, as it necessitates the precise mapping of varied linguistic expressions to a standardized medical terminology system. We used MedDRA v18.1 and augmented our normalization process with the UMLS Terminology Services to assist in mapping the extracted terms to their corresponding MedDRA Preferred Terms (PTs).

Evaluation Metrics: In our evaluation of models for TAC2017 Task 4, we looked at three key metrics: Precision, Recall, and F1 Score, each calculated at both the micro and

macro levels. Micro metrics aggregate true positives, false positives, and false negatives across all classes, treating each instance equally, beneficial for datasets with class imbalances. Macro metrics calculate the metric for each class individually and then average them, treating each class equally regardless of size.

Approaches

We present four frameworks to evaluate the capability of LLMs to normalize text. These approaches are visualized in Figure 1. Additionally, we compare an established string-matching technique to serve as a baseline.

TF-IDF String Matching

TF-IDF (Term Frequency-Inverse Document Frequency) String Matching(21) is a method that leverages the statistical properties of terms within a document corpus to determine their importance and relevance. It is calculated by multiplying the term frequency (TF) of a word in a document by its inverse document frequency (IDF) across the entire corpus. This helps in emphasizing terms that are important within a specific document while downplaying common terms that appear frequently across many documents.

In our implementation, each term and synonym are transformed into a TF-IDF vector using the TfidfVectorizer from the sklearn library. To find the closest match for each synonym, we compute the cosine similarity between the TF-IDF vector of the synonym

and the TF-IDF vectors of all SNOMED CT entries. The SNOMED CT with the highest cosine similarity is considered the closest match.

This provides a straightforward and interpretable way to normalize text by leveraging the statistical properties of the terms and serves as a baseline for comparing the performance of more advanced techniques.

Zero Shot Prompting

Zero-shot learning(22) is a concept in machine learning where a model makes predictions on data it has never seen during training, effectively requiring it to generalize from previous knowledge without any task-specific fine-tuning. LLMs are trained on diverse objectives that require understanding the relationships between words, sentences, and broader discourse to generate coherent and contextually appropriate text making them especially suited to this strategy. To assess GPT-4's latent pre-trained knowledge of SNOMED codes, we used a zero-shot prompting strategy. For each term in our datasets, we constructed a prompt that asked the model to provide the SNOMED CT code for a given synonym. (Supplemental Table 2). The model's responses are then compared to the actual SNOMED CT codes to calculate accuracy. This method was applied to both the domain-specific dataset, which focused on "Malignant neoplastic disease" descendants, and the broader dataset, which included a random sample of terms from commonly used institutional billing codes.

Prompt Recall

In the Prompt Recall approach, we relied on the LLM's ability to retrieve information from its instruction prompt.(23) For each term in our datasets, we included the entire list of possible SNOMED CT terms and their corresponding IDs (N=106 for the oncology-specific evaluation and N=750 for the broader dataset). This dictionary served as a reference for GPT-4 to accurately recall and provide the SNOMED CT IDs when prompted with the generated synonym.

Semantic Embedding

Semantic search is a method that leverages the semantic similarity between phrases to improve the accuracy of text normalization.(24) This approach involves using text embedding models to project both the medical terms and the standardized vocabulary into a high-dimensional space where semantically similar terms are closer together. We evaluated embedding models from MTEB Leaderboard(25) and used within the Sentence Transformers Python module. The models under consideration were "text-embedding-ada-002", "all-MiniLM-L6-v2",(26) "bge-large-en-v1.5",(27) "ember-v1", "GIST-large-Embedding-v0",(28) "sf_model_e5", and "SFR-Embedding-Mistral".(29) We compared performance of these models against our SNOMED CT reference set (N=750) using the metrics described above. We selected the model that provided the highest accuracy in mapping synonyms to the correct SNOMED CT codes, which we then used for the normalization process in the subsequent phases of our study (i.e. here and for RAG).

Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG)(15) combines semantic search and generative language models to perform text normalization. We used a two-step process. First, we used a retrieval system built on GIST-large-Embedding-v0 to fetch the most relevant (based on cosine similarity) SNOMED CT entries based on the input medical term. The top retrieved entries were then passed to GPT-4, which generated the final normalized term by synthesizing the information from the retrieved entries and its pre-existing knowledge. The number of entries retrieved is determined by taking the square root of the total number of potential terms, a number that was adjusted (Supplemental Figure 1) to have a manageable yet comprehensive set of results for the language model to work with.

RESULTS

Selecting an Embedding Model

For each embedding model, we calculated the accuracy through the percentage of synonyms correctly matched to the SNOMED CT codes. The only closed-source model evaluated, "text-embedding-ada-002," has a dimensionality of 1536 and achieved an accuracy of 74.82%. The smallest model, "all-MiniLM-L6-v2," uses 384 dimensions and reached 64.73%. Out of the “medium” 1024 dimensional models evaluated, "bge-large-en-v1.5" had an accuracy of 73.59%, "ember-v1" scored 73.34%, "GIST-large-Embedding-v0" produced 76.23%, and "sf_model_e5" achieved 76.08%. The largest model assessed, "SFR-Embedding-Mistral" with a dimensionality of 4096, yielded an accuracy of 81.63%. We chose the "GIST-large-Embedding-v0" for its balance of size and performance for use in the subsequent phases of our study (Figure 2).

Tuning the Number of Terms Returned

The Retrieval-Augmented Generation (RAG) method required optimization of the number of terms retrieved to inform the language model. We experimented with varying retrieval sizes to find a balance between excluding the true answer and passing a cumbersome amount of information to GPT-4. We identified an optimal number that equated to the square root of the total number of potential terms (Supplemental Figure 1).

SNOMED CT Dataset Results

The TF-IDF-based String Matching gave an accuracy of 67.0% on the domain specific set of 106 terms, and 46.6% on the broader set of 750 terms. The Zero-Shot Recall

method achieved an accuracy of 4.6% and 0.9%. The Prompt Recall method recorded accuracies of 86.4% and 40.8%. The Semantic Search approach yielded accuracies of 86.0% and 76.2%. The RAG method attained an accuracy of 89.8% for the domain-specific dataset and 80.0% for the broader dataset. The results and computational-complexity metrics of these text normalization methods are presented in Table 1 and Figure 3.

Table 1: Model Performance and Utility

	Domain Specific Terms (N=106)			**Cross-Domain Terms (N=750)**		
Approach	**# API Calls**	**Average Tokens per call**	**Accuracy**	**# API Calls**	**Average tokens per call**	**Accuracy**
TF-IDF based String Matching	0	N/A	67.0%	0	N/A	46.6%
Zero-Shot Recall	106	36	4.6%	750	36	0.9%
Prompt Recall	106	1606	86.4%	750	11,529	40.8%
Semantic Search	0	N/A	86.0%	0	N/A	76.2%
RAG	106	189	89.8%	750	455	80.0%

Figure 3: Boxplots of Model Performance. Accuracy of four different normalization methods—Zero-Shot Recall, Prompt Recall, Semantic Search, and RAG—across: (A) 106 Domain Specific SNOMED CTs and (B) 750 Randomly Sampled SNOMED CTs.

TAC2017 Dataset Results

The TAC2017 Adverse Drug Reaction dataset served as a validation set for our normalization methods. Applying the RAG approach to this dataset resulted in a Micro-Precision of 89.6%, Micro-Recall of 86.5%, and a Micro-F1 Score of 88.0%. The Macro-Precision was 89.1%, with a Macro-Recall of 86.1% and a Macro-F1 Score of 87.6%. Our Semantic Search method resulted in a Micro-F1 of 80.83% and a Macro-F1 of 80.55%. Since this approach was unable to classify terminology as unmapped, precision and recall do not apply here. These metrics are tabulated in Table 2, which also includes the performance of other competitors in the TAC2017 (20) challenge, offering a benchmark for the RAG method's performance.

Table 2: Comparison of our RAG and Semantic Search Framework with TAC2017 Task 4 Results

System	Micro-Precision	Micro-Recall	Micro-F1	Macro-Precision	Macro-Recall	Macro-F1
RAG	**89.56**	86.52	**88.01**	**89.07**	86.12	**87.55**
UTH CCB	84.17	**89.84**	86.91	83.02	**89.06**	85.33
UTH CCB	85	87.75	86.35	84.04	86.67	84.79
UTH CCB	82.42	90.78	86.4	80.83	89.9	84.53

Semantic Search*	N/A	N/A	80.83	N/A	N/A	80.55
CONDL	88.81	77.16	82.58	88.2	75.76	80.5
PRNA SUNY	86.14	74.89	80.12	85.32	72.76	77.97
PRNA SUNY	81.55	78.24	79.86	79.8	76.03	77.25
PRNA SUNY	83.6	74.14	78.59	82.22	71.44	75.87
CONDL	74.56	80.96	77.63	73.06	79.92	75.55
CONDL	74.56	80.96	77.63	73.06	79.92	75.55
MC UC3M	73.4	80.25	76.67	72.1	80.38	75.29
MC UC3M	73.4	80.25	76.67	72.1	80.38	75.29
CHOP	71.78	50.14	59.04	70.12	49.84	57.27

* Since the semantic search approach lacks an unmapped option, false negatives are not possible. Consequently, precision and recall are always equal to the F1 score.

DISCUSSION

Our study demonstrates the potential of LLMs to improve the accuracy of medical text normalization, which is an advancement over traditional string-matching or simple machine learning approaches that often struggle with the nuances of medical language.

The Zero-Shot method, with fast response times and low computational overhead due to OpenAI's processing, offers a widely accessible and easy-to-set-up option. However, consistent with previous findings,(12) its reliance on specific assumptions about GPT-4's knowledge of SNOMED CT and its ability to recall details from its training dataset severely limits its accuracy. The Prompt Recall method benefits from having a wealth of knowledge at its disposal, ensuring that the model has all the necessary information to select the optimal term. However, this method is hampered by its high token requirements, which can quickly become costly and result in longer compute times, and the performance suffers significantly when given a larger possible set of terms to normalize. This latter limitation, especially, would limit its practical use in real-world scenarios. In contrast, the Semantic Search approach stands out for its speed and cost-effectiveness, leveraging precomputed embeddings to bypass the need for real-time response generation. While its simpler architecture may overlook complex contextual signals, potentially limiting its effectiveness, it strikes a strong balance between efficiency and performance that make it the most desirable option in scenarios where access to a generative LLM is not viable.

Of the approaches, Retrieval-Augmented Generation (RAG) demonstrated the strongest performance in our study. This proficiency can be largely attributed to its dual-phase process, which combines the precision of semantic retrieval with the contextual fluidity of generative language models. RAG's initial retrieval phase acts as a targeted filter, narrowing down the vast expanse of medical terminology to the most pertinent SNOMED CT entries. This pre-processing step is important, as it primes the model with a distilled set of data that is highly relevant to the input query, thereby increasing the likelihood of accurate normalization. The generative phase of RAG then capitalizes on this curated subset, building upon the retrieved information to produce a normalized output that is not only precise but also contextually enriched. This is particularly helpful when disambiguating terms that have multiple SNOMED CT codes depending on the context.

Methods that rely on prompting often include a large and comprehensive amount of information within the prompt itself, which can be prohibitively expensive in terms of token usage and computational power.(30) In contrast, RAG's retrieval phase efficiently identifies and retrieves only the most relevant entries, greatly reducing the number of tokens processed (Table 1). This selective retrieval not only saves on computational resources but also streamlines the normalization process, making it more agile and cost-effective.

While more cost effective than other approaches, it is still important to note that requiring both retrieval and generation steps necessitates computational power. The

demands translate into potential costs and longer processing times than a simple string-matching approach, which could be a barrier to real-time application in clinical settings.

Ultimately, the best approach for text normalization depends on circumstances such as budget, computational resources, and the nature of the target terminology. For instance, well-known concepts like ICD-10 codes may be more suitable for zero-shot prompting since they may be better represented in the LLM's training data.

The results of our study pave the way for several future research directions. One such direction is the integration of RAG-normalization into clinical systems. For example, within Electronic Health Records (EHRs), this could enhance the normalization process by enabling the identification of adverse drug reactions (ADRs) in clinical notes—an area often overlooked by the structured data elements of EHRs. Additionally, there is an opportunity to fine-tune the RAG approach for specific medical subdomains, such as genetics or rare diseases, where the terminology is particularly specialized. The incorporation of a user feedback loop could additionally improve our pipeline, where clinicians can provide input to correct and refine the model's outputs. This could lead to continuous improvement in the model's performance. Lastly, exploring ways to scale the RAG approach in a cost-effective manner, perhaps by utilizing more efficient language models or distributed computing resources, would make this technology more accessible and practical for widespread use.

ACKNOWLEDGEMENTS

NPT is supported by NIH NIGMS R35GM131905

REFERENCES

1. Kong HJ. Managing unstructured big data in healthcare system. Vol. 25, Healthcare Informatics Research. Korean Society of Medical Informatics; 2019. p. 1–2.
2. Leaman R, Khare R, Lu Z. Challenges in clinical natural language processing for automated disorder normalization. J Biomed Inform. 2015 Oct 1;57:28–37.
3. Kreimeyer K, Foster M, Pandey A, Arya N, Halford G, Jones SF, et al. Natural language processing systems for capturing and standardizing unstructured clinical information: A systematic review. Vol. 73, Journal of Biomedical Informatics. Academic Press Inc.; 2017. p. 14–29.
4. Ahmad Aliero A, Sulaimon Adebayo B, Olanrewaju Aliyu H, Gogo Tafida A, Umar Kangiwa B, Muhammad Dankolo N. Systematic Review on Text Normalization Techniques and its Approach to Non-Standard Words [Internet]. Vol. 185, International Journal of Computer Applications. 2023. Available from: https://www.researchgate.net/publication/374160515
5. Dankolo N, Ahmad Aliero A, Sulaimon Adebayo B, Olanrewaju Aliyu H, Gogo Tafida A, Umar Kangiwa B, et al. Systematic Review on Text Normalization Techniques and its Approach to Non-Standard Words [Internet]. Vol. 185, International Journal of Computer Applications. 2023. Available from: https://www.researchgate.net/publication/374160515
6. Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space [Internet]. 2013. Available from: http://ronan.collobert.com/senna/
7. Scholarworks@uno S, Handler A. An empirical study of semantic similarity in WordNet and An empirical study of semantic similarity in WordNet and Word2Vec Word2Vec An empirical study of semantic similarity in WordNet and Word2Vec [Internet]. 2014. Available from: https://scholarworks.uno.edu/td/1922
8. Steck H, Ekanadham C, Kallus N. Is Cosine-Similarity of Embeddings Really About Similarity? 2024 Mar 8; Available from: http://arxiv.org/abs/2403.05440
9. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention Is All You Need. 2017 Jun 12; Available from: http://arxiv.org/abs/1706.03762
10. Clark P, Tafjord O, Richardson K. Transformers as Soft Reasoners over Language. 2020 Feb 13; Available from: http://arxiv.org/abs/2002.05867
11. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language Models are Unsupervised Multitask Learners [Internet]. 2018. Available from: https://github.com/codelucas/newspaper
12. Wang A, Liu C, Yang J, Weng C. Fine-tuning large language models for rare disease concept normalization. Journal of the American Medical Informatics Association. 2024 Jun 3;
13. Shin J, Tang C, Mohati T, Nayebi M, Wang S, Hemmati H. Prompt Engineering or Fine Tuning: An Empirical Assessment of Large Language Models in Automated

Software Engineering Tasks. 2023 Oct 10; Available from: http://arxiv.org/abs/2310.10508
14. Henrickson L, Meroño-Peñuela A. Prompting meaning: a hermeneutic approach to optimising prompt engineering with ChatGPT. AI Soc. 2023;
15. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. 2020 May 22; Available from: http://arxiv.org/abs/2005.11401
16. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks [Internet]. 2020. Available from: https://github.com/huggingface/transformers/blob/master/
17. Gao Y, Xiong Y, Gao X, Jia K, Pan J, Bi Y, et al. Retrieval-Augmented Generation for Large Language Models: A Survey. 2023 Dec 18; Available from: http://arxiv.org/abs/2312.10997
18. Xiong G, Jin Q, Lu Z, Zhang A. Benchmarking Retrieval-Augmented Generation for Medicine. 2024 Feb 20; Available from: http://arxiv.org/abs/2402.13178
19. Vuokko R, Vakkuri A, Palojoki S. Systematized Nomenclature of Medicine–Clinical Terminology (SNOMED CT) Clinical Use Cases in the Context of Electronic Health Record Systems: Systematic Literature Review. Vol. 11, JMIR Medical Informatics. JMIR Publications Inc.; 2023.
20. Roberts K, Demner-Fushman D, Tonning JM. Overview of the TAC 2017 Adverse Reaction Extraction from Drug Labels Track [Internet]. Available from: https://dailymed.nlm.nih.gov/
21. Salton G, Buckley C. Term Weighting Approaches in Automatic Text Retrieval. Inf Process Manag. 1988;513–23.
22. Kojima T, Gu SS, Reid M, Matsuo Y, Iwasawa Y. Large Language Models are Zero-Shot Reasoners. 2022 May 24; Available from: http://arxiv.org/abs/2205.11916
23. Kuratov Y, Bulatov A, Anokhin P, Sorokin D, Sorokin A, Burtsev M. In Search of Needles in a 11M Haystack: Recurrent Memory Finds What LLMs Miss. 2024 Feb 16; Available from: http://arxiv.org/abs/2402.10790
24. Wang S, Koopman R. Semantic embedding for information retrieval [Internet]. Available from: http://www.worldcat.org/
25. Muennighoff N, Tazi N, Magne L, Reimers N. MTEB: Massive Text Embedding Benchmark. 2022 Oct 13; Available from: http://arxiv.org/abs/2210.07316
26. Wang W, Wei F, Dong L, Bao H, Yang N, Zhou M. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. 2020 Feb 25; Available from: http://arxiv.org/abs/2002.10957
27. Xiao S, Liu Z, Zhang P, Muennighoff N. C-Pack: Packaged Resources To Advance General Chinese Embedding. 2023 Sep 14; Available from: http://arxiv.org/abs/2309.07597
28. Solatorio A V. GISTEmbed: Guided In-sample Selection of Training Negatives for Text Embedding Fine-tuning. 2024 Feb 26; Available from: http://arxiv.org/abs/2402.16829
29. Meng R, Liu Y, Rayhan Joty S, Xiong C, Zhou Y, Yavuz S. SFR-Embedding-Mistral Enhance Text Retrieval with Transfer Learning. Salesforce AI Research Blog. 2024;

30. Jiang N, Luo C, Lakshman V, Dattatreya Y, Xue Y. Massive Text Normalization via an Efficient Randomized Algorithm. In: WWW 2022 - Proceedings of the ACM Web Conference 2022. Association for Computing Machinery, Inc; 2022. p. 2946–56.

FIGURE LEGENDS

Figure 1 caption: Methodology Flowchart. Step-by-step approach for four different normalization methods— (A) Zero-Shot Recall: Utilizes a single prompt to elicit the correct term from the model without any prior examples or fine-tuning. (B) Prompt Recall: Feeds the model a comprehensive list of terms, prompting it to select the most appropriate one based on the input context. (C) Semantic Search: Matches input terms with the closest semantic equivalents using a precomputed vector space of embeddings. (D) RAG: First retrieves most semantically relevant terms and then uses the generative decoder to choose the best matching term. An embedding space visualization illustrates the differences between Semantic Search (E) and RAG (F).

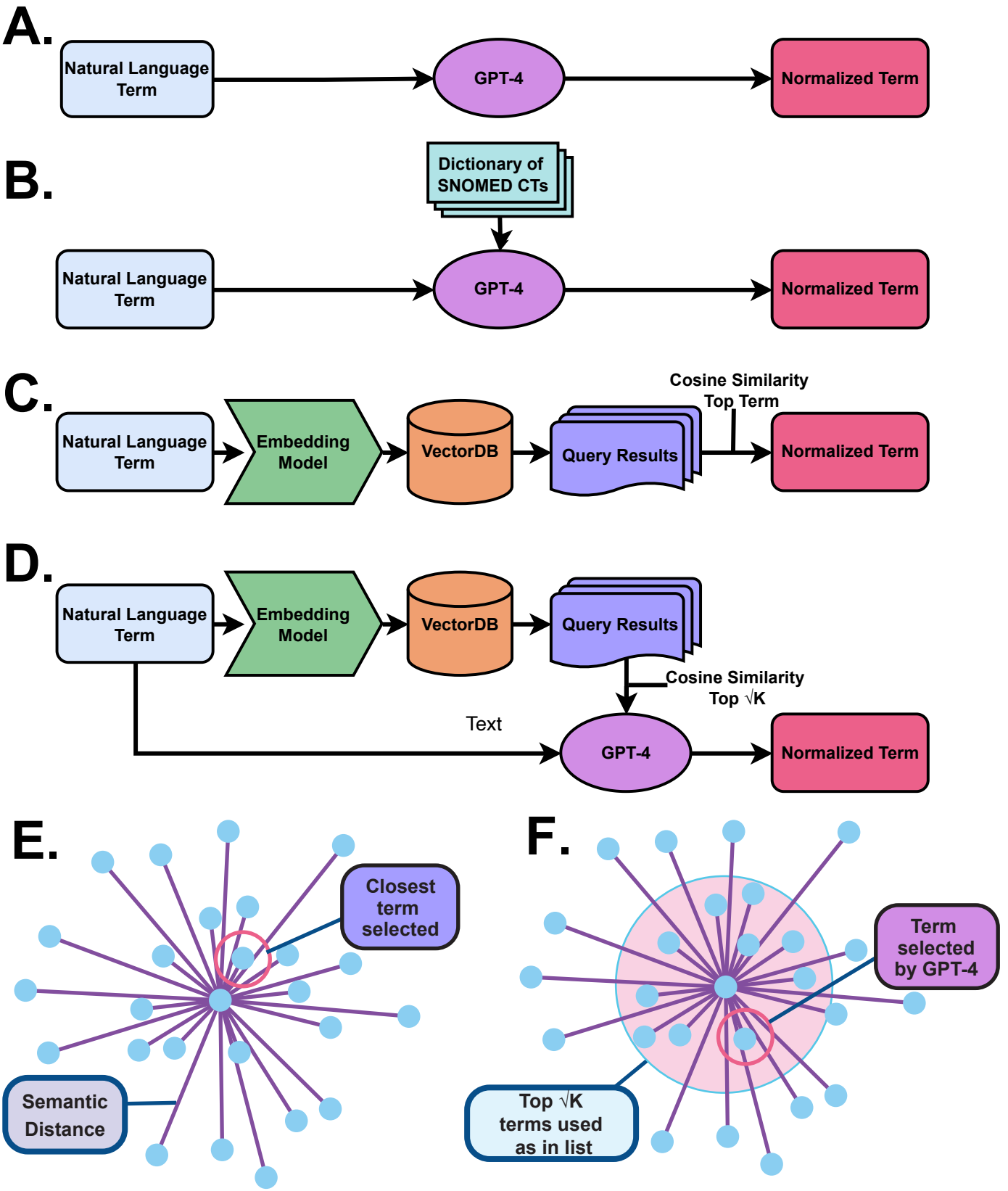
Alt text: Colorful flowchart of LLM approaches and networks of semantic embedding distances for relevant approaches.

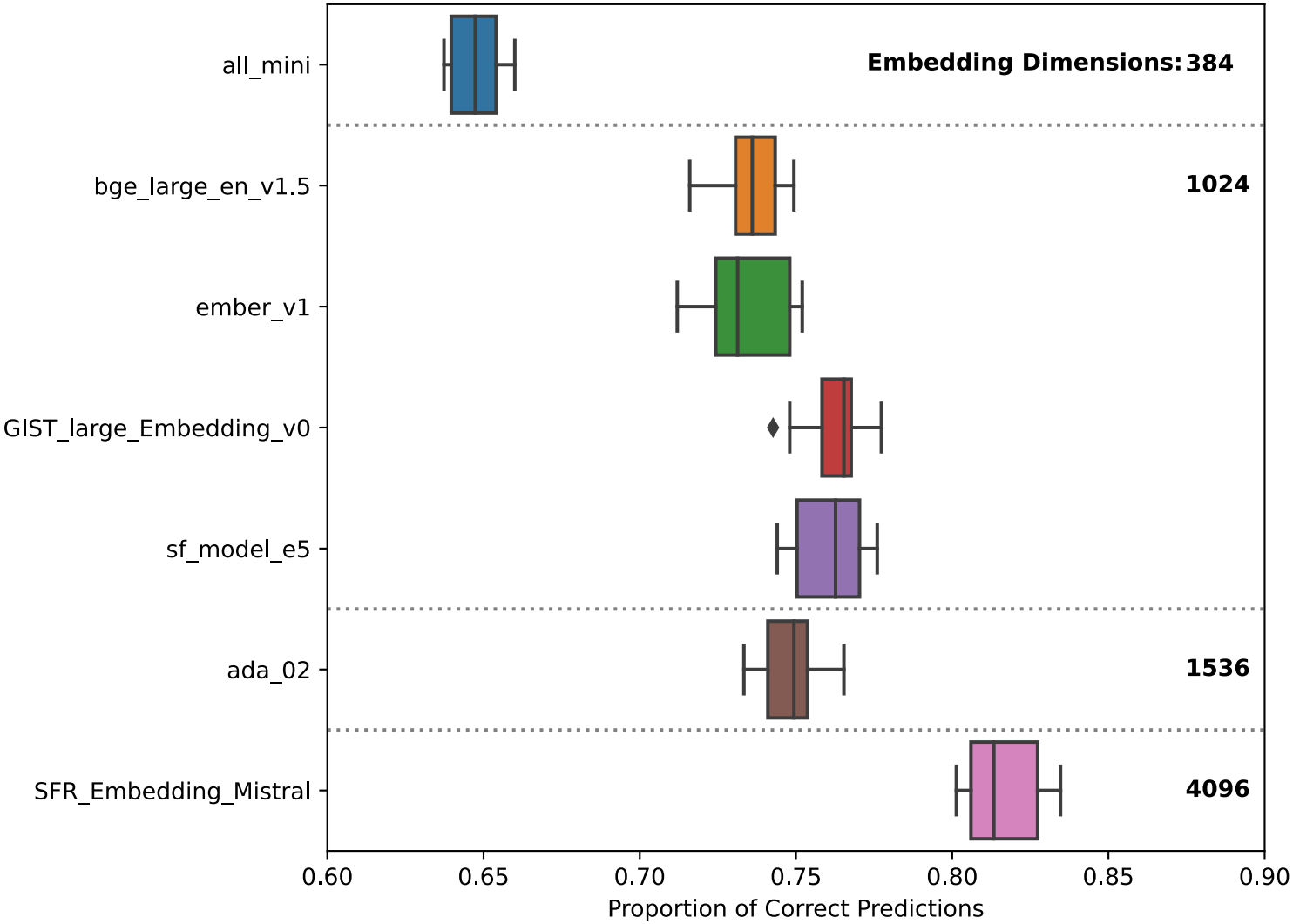
Figure 2 caption: Embedding Evaluation. Comparison of Semantic Search text normalization accuracy across each assessed embedding model. The models are grouped by the number of embedding dimensions used.

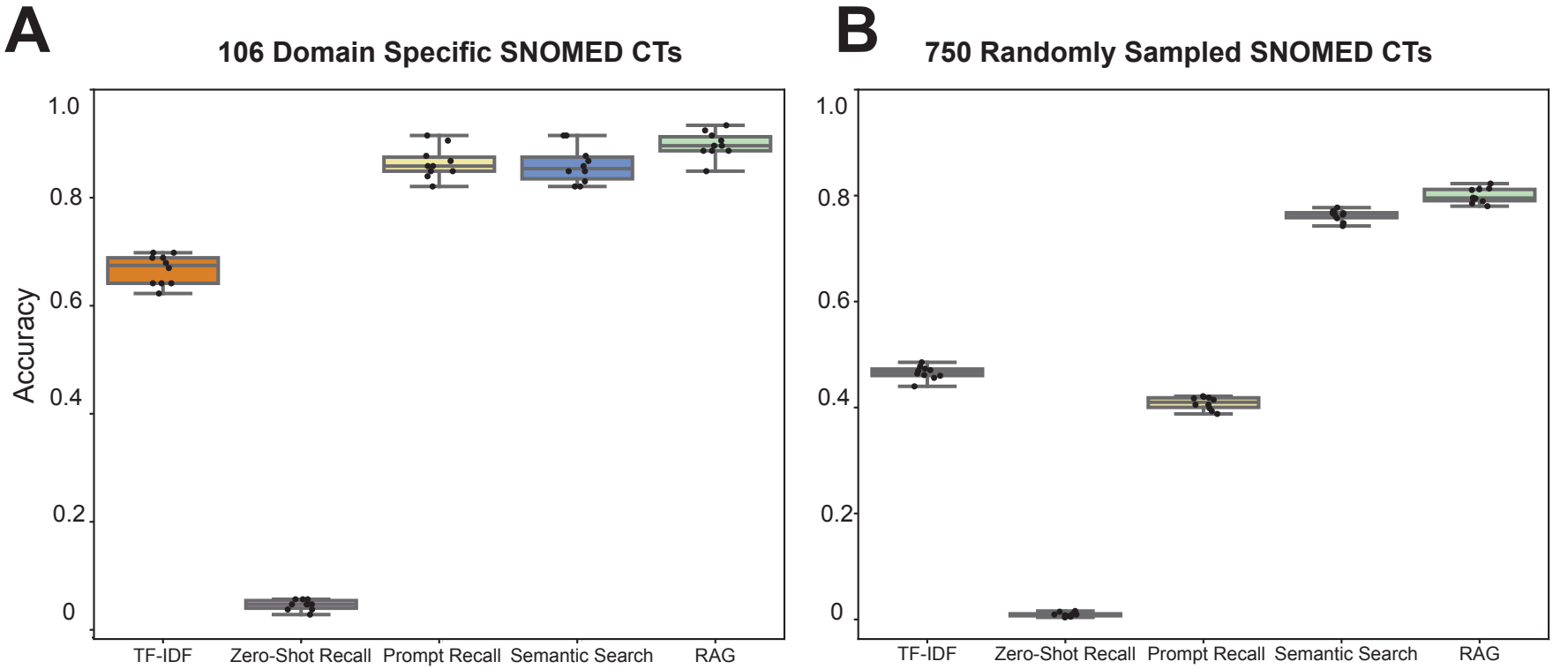
Alt text: Boxplot comparison of embedding models by dimensions, with GIST-large-Embedding-v0 performing the strongest out of medium size (1024 dimensions) models.

Figure 3 caption: Boxplots of Approach Performance. Accuracy of four different normalization methods—Zero-Shot Recall, Prompt Recall, Semantic Search, and RAG—across: (A) 106 Domain Specific SNOMED CTs and (B) 750 Randomly Sampled SNOMED CTs.

Alt text: Boxplot comparison of different approaches. The Semantic Search and RAG models perform most strongly, with RAG achieving the highest accuracy.


A.
Natural Language Term
GPT-4
Normalized Term
B.
Dictionary of SNOMED CTs
Natural Language Term
GPT-4
Normalized Term
C.
Natural Language Term
Embedding Model
VectorDB
Query Results
Cosine Similarity Top Term
Normalized Term
D.
Natural Language Term
Embedding Model
VectorDB
Query Results
Cosine Similarity Top √K
Text
GPT-4
Normalized Term
E.
Closest term selected
Semantic Distance
F.
Term selected by GPT-4
Top √K terms used as in list


Embedding Dimensions:
384
1024
1536
4096
all_mini
bge_large_en_v1.5
ember_v1
GIST_large_Embedding_v0
sf_model_e5
ada_02
SFR_Embedding_Mistral
0.60
0.65
0.70
0.75
0.80
0.85
0.90
Proportion of Correct Predictions


A
106 Domain Specific SNOMED CTs
B
750 Randomly Sampled SNOMED CTs
Accuracy
1.0
0.8
0.6
0.4
0.2
0
TF-IDF
Zero-Shot Recall
Prompt Recall
Semantic Search
RAG