CENT: Context Engineering Framework for Normalization of Clinical Trial Procedures

Sanya B. Taneja[1], Ziqing Ji[2], Hans Verstraete[1], Ali Samadani[1]

[1]Johnson and Johnson Innovative Medicine, [2]University of Washington

Correspondence: asamadan@its.jnj.com

Abstract

Clinical Concept Normalization is essential for clinical research applications involving trial protocols, such as patient-trial matching. Existing approaches focus heavily on specific domains and need large, annotated datasets. To address these challenges, we propose CENT, a context engineering framework that combines semantic matching for candidate retrieval and Large Language Model (LLM) prompting for disambiguation. CENT achieves superior performance on clinical procedures normalization in both binary and hierarchical metrics compared to semantic matching or LLM-only approaches, without requiring fine-tuning. Advanced prompt strategies, including Chain-of-Thought and Tree-of-Thoughts, achieve high performance at practical cost. CENT is scalable and adaptable for normalization across diverse clinical vocabularies in real-world clinical applications.

1 Introduction

Clinical Concept Normalization (CCN), often referred to as entity linking, standardizes terms in clinical texts, such as clinical trial protocols, to standard concepts in clinical coding systems (e.g., ICD, SNOMED-CT, LOINC). In practice, normalization for clinical research applications is often performed as an ad hoc step, either as a manual effort or more commonly with string matching before downstream tasks, such as patient-trial matching (Wong et al., 2025). This limits both scalability and reliability in real-world deployments, where concept variations and ambiguous terminology are common (Wong et al., 2023).

For clinical trial protocols, CCN ensures reliable information retrieval and enables comparative analysis across trials in terms of eligibility criteria, disease evaluation steps or schedule of activities, predictive modeling using data from historic trials, and patient trial matching (Danhauser et al., 2025; Lopez et al., 2025).

Despite advances in semantic matching with pretrained and fine-tuned BERT-based models and, more recently, transformer models, biomedical entity linking remains challenging without domain-specific adaptation. Prompting general-purpose Large Language Models (LLMs) to assign codes to diagnosis and procedure descriptions has shown subpar performance and hallucinated codes (Soroush et al., 2024). Moreover, relying solely on LLMs for entity linking requires fine-tuning for each clinical coding system, which is expensive (Berkowitz et al., 2025).

We introduce **C**ontext **E**ngineered **N**ormalization for Clinical **T**rials (CENT) framework that leverages semantic matching to retrieve highly relevant information, which is then used in structured prompts to perform CCN. In CENT, semantic matching efficiently generates high-recall candidate matches, while LLM prompting improves disambiguation among closely related candidates without the need for extensive annotated data or vocabulary-specific fine-tuning.

Objective: To develop and evaluate a context engineering framework, CENT, for CCN in trial protocols, focusing on laboratory procedures, and benchmarking performance using binary and hierarchical metrics on a publicly available dataset.

Contributions:

- A context engineering framework, CENT, that combines semantic matching for context retrieval with LLM prompting for CCN.
- Evaluation of CENT using various prompt strategies on a publicly available clinical procedures dataset, assessed with both binary and hierarchical metrics.
- Comparative evaluation of CENT against embedding models and LLM-only prompting for CCN.

Proceedings of the 25th Workshop on Biomedical Language Processing (BioNLP 2026), pages 729–741

1.1 Related Work

Current state-of-the-art CCN systems and benchmark datasets primarily focus on normalizing diseases, symptoms, and chemicals, while clinical procedures and laboratory measurements are underrepresented. This results in incomplete normalization of clinical trial protocols and is a significant barrier for real-world automated protocol analysis and benchmarking. While CCN tools such as MedLinker, SciSpacy, and cTAKES support entity linking for procedures (Kartchner et al., 2023), these systems typically rely on string matching, leading to suboptimal performance.

Semantic matching models, such as SapBERT (Liu et al., 2021) that is fine-tuned with Unified Medical Language System (UMLS) concepts, have demonstrated strong similarity-based matching for biomedical entities (Kartchner et al., 2023). Semantic matching approaches with domain-specific embeddings achieve high recall and are effective at candidate generation, however, they struggle in candidate disambiguation as the size of the target vocabulary increases (Kartchner et al., 2023). Moreover, these systems are primarily optimized for diseases, drugs, and chemicals, as most benchmark datasets (e.g., NCBI Disease, BC5CDR) focus on these domains (Ferré et al., 2020; Yang et al., 2023).

Recent work has explored the use of LLMs and generative AI approaches for CCN. Berkowitz et al. (2025) developed RAGNorm, a system that combines semantic matching with LLM prompting to normalize biomedical texts to SNOMED-CT codes. RAGNorm uses general-purpose models for both semantic matching and prompting, and while it outperformed the baseline semantic matching methods, the authors highlight challenges with generalizability and a decline in performance with increased target vocabulary size. Soroush et al. (2024) found that prompt-based LLMs frequently hallucinate invalid or non-specific codes when relying on internal knowledge of clinical vocabularies, and that prompt engineering alone could not consistently produce correct codes for all concept descriptions. Other studies have noted the high computational cost and limited improvements with LLM-only pipelines compared to semantic matching and domain-specific embeddings in the biomedical normalization task (Dobbins, 2024; Wang et al., 2024). Jahan et al. (2024) observed that baseline models such as BioBART still outperform LLMs in recall for chemical and disease normalization, especially with small datasets and large vocabularies.

Combining the strengths of semantic matching and LLM prompting, however, appears to be a promising approach for CCN in specialized domains (Borchert et al., 2024; Berkowitz et al., 2025). Strategic prompt design with curated context beyond simple prompts can help overcome the limitations of LLM-only normalization approaches. In this work, we present CENT, a framework that integrates 1) domain-specific semantic matching to retrieve relevant context, followed by 2) a structured prompt design that embeds the retrieved context, and 3) explicit evaluation rubric to guide the LLM in performing CCN.

Evaluating CCN systems remains challenging due to variability across datasets, biases in scoring metrics, the lack of clear separation between entity recognition and subsequent normalization tasks, and the predominant focus of benchmark datasets on diseases and drugs (Kartchner et al., 2023; Ferré et al., 2020). Furthermore, prior evaluations rely primarily on binary metrics such as accuracy, precision, and recall, overlooking the hierarchical nature of clinical coding systems, which warrants more nuanced assessments that account for relationships among concepts (Ferré and Langlais, 2023). To address these gaps, in this study, we adapt hierarchical evaluation metrics to provide a more comprehensive assessment of CCN performance for normalization of clinical procedures.

2 Datasets

In this work, we evaluate our proposed CCN framework on the LOINC to CPT mappings of laboratory procedures available through UMLS (National Library of Medicine, 2006). LOINC, created by the Regenstrief Institute, is the universal standard for laboratory tests and clinical observations (Regenstrief Institute, 2025). The CPT vocabulary, created and maintained by the American Medical Association, is used for reporting procedural activities and services for billing and reimbursement (American Medical Association, 2025). Both vocabularies are commonly used for coding of activities related to clinical trials.

The UMLS LOINC–CPT mappings dataset contains 2,019 LOINC concepts linked to CPT codes; in some cases, multiple LOINC concepts map to a single CPT code. We excluded mappings to 21

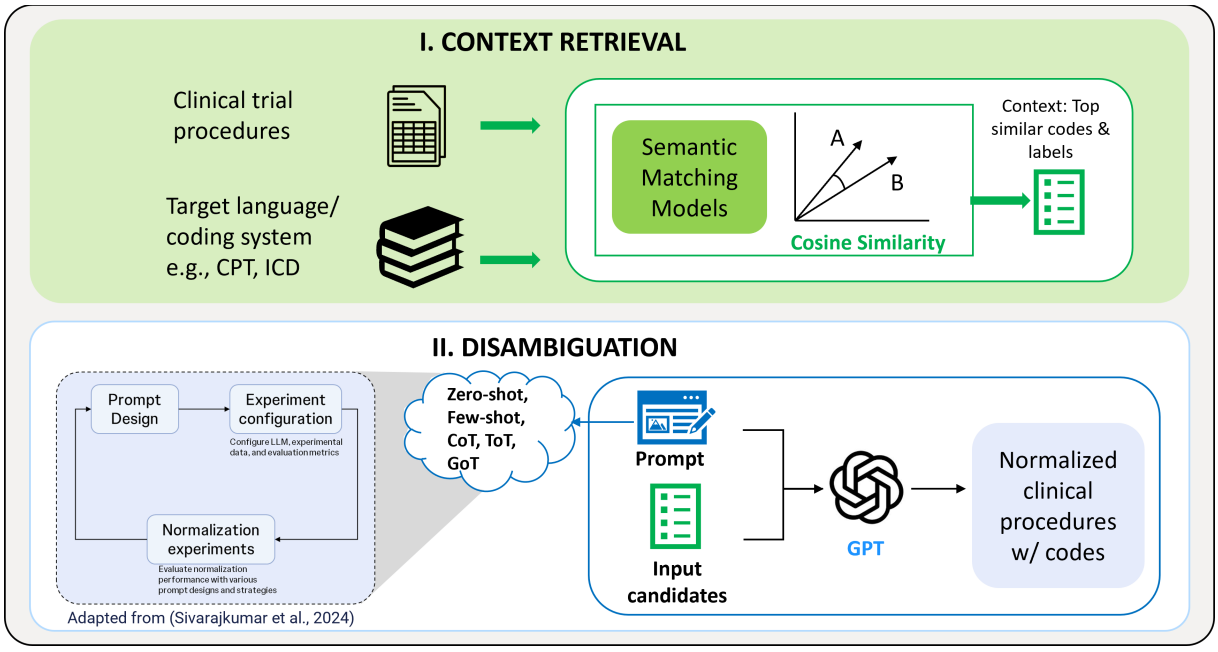

Figure 1: Overview of the CENT framework (I) Context retrieval with semantic matching, and (II) context engineering and prompt designs for disambiguation with GPT models. Iterative prompt design is based on prior work (Sivarajkumar et al., 2024) with zero-shot, few-shot, Chain-of-Thought (CoT), Tree-of-Thoughts (ToT), and Graph-of-Thought (GoT) prompt strategies.

deprecated CPT codes (44 mappings removed). We also excluded three non-specific allergen-related CPT codes—86001, 86003, and 86005—removing 314, 1,017, and 95 mappings, respectively, as these codes are unrelated to trial procedures and outside the scope of normalization. After these exclusions, 593 LOINC-CPT mappings remained. LOINC and CPT concept labels were further processed to remove punctuation and tokenized (average LOINC tokens = 7.6, average CPT tokens = 11.3).

We used CPT vocabulary (v2015) codes and labels as the target concept data for evaluation, as this was the last publicly available version of CPT. These CPT concepts were filtered to include the measurement domain only, resulting in 1,584 CPT target codes and labels.

2.1 Protocol-derived Datasets

We additionally applied CENT to two internal datasets of clinical assessments extracted from schedule of activities tables in clinical trial protocols. Dataset 1 includes 669 unique laboratory, diagnostic, survey, and billing procedures partially mapped to CPT codes, and Dataset 2 includes 139 laboratory/biomarker procedures from 20 trials mapped to a custom laboratory vocabulary (N=248 concepts). We applied CENT to predict target concepts for these datasets. For these datasets, we report quantitative metrics and expert review outcomes for the experiments with these protocol-derived datasets.

3 Methodology

We propose the CENT framework for CCN that systematically combines semantic matching with LLM prompting. CENT is designed to retrieve highly relevant context as top-ranked candidate codes and labels, and deliver it to the LLM in the form of structured, sectioned prompts (Figure 1).

3.1 Context Retrieval with Semantic Matching

In the first stage, CENT uses semantic matching with two fine-tuned domain-specific BERT-based models, SapBERT and BioBERT-mnli, to retrieve candidate concepts from the target vocabulary. SapBERT is a word-based model fine-tuned with UMLS concepts for medical entity linking and generates candidates based on closeness of concept embeddings in semantic space (Liu et al., 2021). BioBERT-mnli is a sentence transformer model fine-tuned with natural language inference and text similarity datasets that has been used to effectively classify sentences from randomized controlled trials (Deka et al., 2022). For each input term (e.g., LOINC concept), the models embed the concept

label and the resulting embedding is used to compute cosine similarity scores with embeddings of the target concepts (e.g., CPT concepts). The top 5 concepts with the highest scores from each model are added to a set and the scores are normalized. The concepts are then sorted by similarity scores and de-duplicated. The top unique candidates (up to 10) are used as the curated context formatted as code + label. Note that similarity scores are not passed to the LLM. By deliberately constraining the search space, this step serves as context engineering, shaping and refining the information available for downstream reasoning and matching.

Implementation: For SapBERT, we tokenize the texts with truncation enabled, embed in batches, with padding of all inputs to maximum tokens within the batch. For BioBERT-mnli, we use sentence-transformers with mean pooling, truncation, and dynamic batch padding. Both models generate 768-dimensional embeddings. We compute cosine similarity between input and target embeddings and retrieve the top k candidates per model. To further characterize the retrieval stage, we report recall@k on the LOINC-CPT dataset, where a query is counted as retrieved if ground truth CPT appears in the top-k candidate set.

3.2 Structured Prompts for Disambiguation

In the second stage, the query code and its label along side with the curated contextual information from the previous step are embedded in a structured prompt for the LLM. Prompts are organized into clearly defined sections, including 1) description of the CCN task, 2) explicit rubric to guide the LLM in performing CCN, 3) instructions to score LLM thoughts, 4) examples of the CCN to be done along with reasoning for the selected matches, and 5) the input query and the list of candidate matched codes with their labels (curated context). The LLM is instructed to select the single best candidate or return "no match" if appropriate.

For the CCN task in this paper, we designed zero-shot and few-shot prompt strategies using the DSPy library (Khattab et al., 2023) and adapted the open-source Graph-of-Thoughts repository (Besta et al., 2024) to implement the Chain-of-Thought (CoT) (Wei et al., 2022), Tree-of-Thoughts (ToT) (Yao et al., 2023), and Graph-of-Thoughts (GoT) strategies. Broadly, the operations involved are (1) prompt the LLM (*generate_prompt*) to map input concept to corresponding target concept based on the context of candidate codes and labels, (2) parse and score the LLM response (*score_prompt*), (3) improve the final response based on the context (*tot_improve_prompt*) or aggregate responses (*aggregate_prompt*), and (4) keep the best scoring response. The scores are calculated by the LLM on a scale of 0-5 using the *score_prompt*. For details of prompts used in the task, see Appendix A. Each prompt strategy was tested and improved iteratively. The prompt design and iterative evaluation was based on prior research on task-specific prompt design and improved performance of heuristic and ensemble prompts in complex clinical information extraction and disambiguation tasks (Sivarajkumar et al., 2024).

Implementation: ToT is defined with branch factor 10 and depth 3 while GoT is defined with branch factors 10 and 20. We used the GPT4.1 (OpenAI, 2025a) and GPT4o-mini (OpenAI, 2024) models to assess the impact of the model size on performance and cost with temperature equal to 1.0 and maximum output tokens equal to 10000.

For the comparative evaluation with direct LLM prompting without engineering context, we provided the input term along with 1,584 CPT target codes and labels to the GPT4o-mini model. The total prompt and context was approximately 65,000 tokens (context window of model is 100,000 tokens). For comparative context-retrieval models, we utilized the OpenAI API for text-embedding-ada-002 model (OpenAI, 2022). We accessed GPT models via an Azure OpenAI endpoint with access-controlled API keys. CENT is designed to be model-agnostic and can be used with private endpoints and locally deployed LLMs, if required.

4 Evaluation

We evaluated CENT with binary metrics, hierarchical metrics, and total cost (in US dollars) for CCN on the LOINC-CPT dataset. Binary metrics included precision@5, recall@5, and F1 score. Cost was measured by the input and output tokens in the prompts based on the OpenAI API pricing for the models (OpenAI, 2025b). We report the total cost of the LOINC-CPT normalization task with 593 mappings. The performance evaluation also included comparison against: 1) semantic matching only using SapBERT, 2) semantic matching only using OpenAI text-embedding-ada-002 model, 3) context retrieval using OpenAI text-embedding-ada-002 model followed by LLM prompting for disambiguation, and 4) direct LLM prompting without

engineered context.

4.1 Hierarchical Metrics

Hierarchical metrics take into account hierarchical characteristics of predicted codes in CCN, including concept position within the ontology, ancestral and descendant information, and relationship with the correct match. Unlike binary metrics that only determine if a predicted code is an exact match, hierarchical metrics incorporate partial correctness, capture ontological distance, and reward higher semantic similarity for close mismatches. We included two hierarchical metrics in our evaluation, Wu-Palmer similarity and Resnik similarity.

Wu-Palmer Similarity. Wu-Palmer Similarity (Wu and Palmer, 1994) is a structural similarity measure that quantifies the closeness of two concepts within an ontology, based on their depth (distance from the root concept) and the depth of the lowest common ancestor (LCA). For concepts a and b:

$$\text{Wu–Palmer}(a,b) = \frac{2 \times \text{depth}(\text{LCA}(a,b))}{\text{depth}(a) + \text{depth}(b)} \quad (1)$$

The score ranges from 0 to 1, with higher values indicating greater semantic overlap.

Resnik Similarity. Resnik similarity (Resnik, 1995) is calculated as the information content (IC) of the LCA of two nodes in the hierarchy to measure the closeness of two nodes by how informative their LCA is.

$$\text{Resnik}(a,b) = IC(\text{LCA}(a,b)) \quad (2)$$

IC is calculated using the proposed method by Seco et al. (2004) to use the ontology's structure:

$$ic(c) = 1 - \frac{\log(\text{descendants}(c) + 1)}{\log(\text{total concepts})} \quad (3)$$

Resnik similarity is higher for concepts with LCA lower in the hierarchy.

The UMLS API (National Library of Medicine, 2025) was used to extract ontological relationships, including ancestors, descendants, and depth for CPT codes. Ancestors and descendants were identified recursively through the UMLS API endpoints *parent* and *children* to preserve the hierarchical order. For cases with "no match", hierarchical metrics values were set to 0.

5 Results and Discussion

Table 1 presents the evaluation results of CENT on the LOINC-CPT dataset. CENT consistently outperformed the comparative methods, including semantic matching alone and direct LLM prompting-across both binary and hierarchical metrics. The highest F1-score was achieved by CENT with CoT prompting (0.82), while the highest Wu-Palmer similarity (0.86) and Resnik similarity (0.87) were observed with ToT prompting. When compared to embedding-only approaches, CENT achieved a marked improvement in precision (highest precision 0.86 with CENT vs 0.15 with SapBERT-only) and Wu-Palmer similarity (0.77-0.86 with CENT vs 0.53 with SapBERT-only), indicating that normalized codes by CENT are not only more accurate but, if not an exact match, are closer to the correct matches in the ontology. Direct LLM prompting performed poorly across all metrics with prohibitive costs, confirming prior findings that this approach is infeasible for CCN (Soroush et al., 2024).

A comparison of embedding models within CENT (SapBERT + BioBERT-mnli) and a general-purpose OpenAI model (text-embedding-ada-002) further underscores the value of domain-specific embeddings for CCN. Using zero-shot prompting, CENT with SapBERT + BioBERT-mnli achieved an F1 score of 0.79, compared to 0.64 for the OpenAI text-embedding-ada-002 model followed by LLM zero-shot prompting. These findings corroborate prior reports that domain-specific models outperform general-purpose models on clinical data classification tasks (Gao et al., 2025).

Across prompt strategies, CENT with CoT achieved high performance (F1-score 0.82 with GPT4.1 and 0.80 with GPT4o-mini), while ToT and GoT were stronger in hierarchical metrics (Wu-Palmer similarity 0.86 and 0.85, respectively). GoT, despite adding operational complexity and cost, did not offer improved performance except for the lowest percentage of "no match" predictions (0%), followed by ToT (0.3%). The "no match" percentage indicates the failure to identify any matches. In contrast, CENT with zero-shot and few-shot prompts resulted in the highest percentage of "no match" predictions (14.1% and 12.8%, respectively). Error analysis has shown that, when allowed, CENT over-predicts codes (i.e., predicted code set includes more than one candidate) rather than "no match".

Cost is a critical factor for real-world and re-

Approach	Prompt Strategy	Precision @5	Recall @5	F1	Wu-Palmer	Resnik	% "no match"
CENT GPT4.1	Zero-shot	**0.87**	0.74	0.80	0.77	0.77	14.8
CENT GPT4.1	Few-shot	0.86	0.76	0.81	0.79	0.79	10.8
CENT GPT4.1	CoT	0.86	0.79	**0.82**	0.84	0.84	5.1
CENT GPT4.1	ToT	0.81	**0.81**	0.81	**0.86**	**0.87**	0.3
CENT GPT4.1	GoT	0.79	0.78	0.78	0.85	0.85	0.7
CENT GPT4o-mini	CoT	0.82	0.78	0.80	0.84	0.85	7.2
CENT GPT4o-mini	ToT	0.79	0.79	0.79	0.85	0.86	0.5
CENT GPT4o-mini	GoT	0.78	0.78	0.78	0.85	0.86	**0**
Embedding only (SapBERT)	-	0.15	0.74	0.25	0.53	0.56	-
Embedding only (ada-002)	-	0.15	0.76	0.25	0.55	0.62	-
ada-002 + GPT4.1	Zero-shot	0.64	0.64	0.64	0.81	0.82	0.3
LLM only (GPT4o-mini)	Few-shot	0.05	0.17	0.08	0.50	0.58	3.1

Table 1: **Top.** Evaluation of CENT with binary and hierarchical metrics for LOINC-CPT mappings (N=593). "no match" indicates cases where the model did not return any normalized code. **Bottom.** Evaluation of comparative approaches: (i) SapBERT only for CCN, (ii) OpenAI ada-002 for context retrieval and GPT4.1 for disambiguation, (iii) OpenAI ada-002 only, and (iv) LLM-only with GPT4o-mini for CCN.

Approach	Prompt Strategy	F1-score	Cost (USD)
CENT*	Zero-shot	0.79	0.36
CENT*	Few-shot	0.80	0.37
CENT*	CoT	**0.82**	12.89
CENT*	ToT	0.81	37.57
CENT*	GoT	0.78	59.89
CENT$^+$	CoT	0.80	0.56
CENT$^+$	ToT	0.79	2.66
CENT$^+$	GoT	0.78	3.27
LLM-only$^+$	Few-shot	0.08	102.5

Table 2: Cost (in USD) of API calls and F1-score for the prompt strategies used to evaluate CENT with LOINC-CPT dataset (N=593). "*" denotes GPT4.1 used for disambiguation and "$^+$" denotes GPT4o-mini model.

search applications. Table 2 presents the cost (in USD) of the API calls used for CENT with different prompt strategies in the LOINC-CPT normalization. While ToT and GoT prompting have higher costs ($37.57 and $59.89, respectively) compared to other strategies, our experiments showed that comparable performance is achievable with the smaller GPT4o-mini model using these advanced prompt strategies, but with significantly lower cost (e.g., cost for CENT with GPT4o-mini & ToT was $2.66 vs $37.57 for CENT with GPT4.1 & ToT).

Table 3 presents the results of the retrieval ablation analysis in the LOINC-CPT mapping experiment. Union of candidates from both models (SapBERT and BioBERT-mnli) achieves the highest recall (recall@10 improves to 0.85), improving the downstream CCN task driven by the retrieved context (union of candidates). Moreover, increasing the number of per-model candidates (k) from 5 to 10 produces no appreciable gains, indicating that CENT remains effective even with a small number of candidates.

To the best of our knowledge, this work represents the first implementation of advanced prompt strategies (ToT, GoT) for the CCN task. Compared to general tasks evaluated by Besta et al. (2024) where GoT outperformed other strategies, it is evident that model and prompt strategy selection for domain-specific tasks such as CCN cannot be gen-

Retrieval	Recall @1	Recall @5	Recall @10
SapBERT only	0.51	0.74	0.80
BioBERT-mnli only	0.59	0.79	0.82
Union (SapBERT ∪ BioBERT-mnli)	–	0.83	0.85

End-to-end (CENT+CoT)	F1	% "no match"
Union candidates, $k = 5$ per encoder	0.82	5.1
Union candidates, $k = 10$ per encoder	0.83	5.3

Table 3: Retrieval ablation and end-to-end sensitivity for LOINC-CPT mappings. Recall@k measures whether the ground truth CPT code is present in the top-k retrieved candidates.

eralized out of the box. The combined use of binary and hierarchical metrics provides a more complete evaluation: binary metrics alone may over- or underestimate model performance, but hierarchical metrics can indicate whether the predicted codes are proximally closer to the ground truth code in the ontology. For instance, CENT with ToT prompting showed lower F1-scores than CoT but higher Wu-Palmer and Resnik scores, indicating that even when predictions are incorrect, they are semantically closer to the ground truth.

Internal Validation: We applied CENT to the protocol-derived procedures, which are noisier and less standardized than LOINC labels. Examples of procedures included laboratory panels (chemistry, hematology), abbreviated texts (PSA, UPEP), imaging procedures (MRI Abdomen), biomarker assays (Serum biomarkers). For the 139 laboratory/biomarker procedures normalized to custom vocabulary concepts (Dataset 2), CENT (few-shot) achieved precision@5=0.77, recall@5=0.65, and F1=0.71. This dataset included instances with multiple target concepts and instances with no valid matches; "no match" is counted correct only when the ground truth is empty. For the dataset with 669 procedures (Dataset 1), we ran CENT to predict CPT codes, and 3 clinician-experts manually reviewed a random sample of 59 laboratory procedures. Each of 3 experts reviewed 20 samples with partial overlap. The outcomes were 28/59 (47.5%) exact matches to expert annotation, 17/59 (28.8%) partial matches (sibling/parent relationships), and 14/59 (23.7%) incorrect CENT match when compared to expert annotations.

Together with the benchmark dataset, these results suggest CENT can operate on noisy and often non-standard clinical texts common in real-world data (e.g., clinical protocols). Notably, the optimal choice of models and prompt strategy should be guided not only by performance metrics but also by specific use case requirements, the target vocabulary, acceptable operational costs, and the prevalence of "no match" scenarios in the real-world data. CENT enables high-recall candidate generation and robust candidate disambiguation for CCN without model fine-tuning or annotated training datasets. CENT does not require model training and allows for substitution of prompt formats, LLMs, and reference clinical vocabulary system as needed for domain adaptation.

6 Conclusion

We present a context engineering framework, CENT, for CCN that combines semantic matching with domain-specific models for context engineering and LLM prompting for disambiguation. CENT consistently outperformed other methods - including semantic matching alone and LLM-only - on laboratory procedures with both binary and hierarchical metrics without any fine-tuning or annotated training data. CENT enables scalable, cost-effective normalization to support real-world clinical research applications. Ongoing work involves applying CENT to use cases in clinical trial optimization and benchmarking, and exploring further prompt strategies to refine normalization.

Limitations

This study has several limitations. First, the evaluation mainly focused on a single dataset with laboratory procedures, along with two internal datasets for validation. The framework will need to be evaluated on larger, diverse datasets with more entity types and clinical vocabularies. Second, prompt design is currently a manual process in CENT that must be tailored for other CCN tasks. Finally, our prompts were limited to GPT models and cost calculations are based on current API pricing. The performance may not reflect future changes in models and re-evaluation of CENT is warranted with newer LLMs.

References

American Medical Association. 2025. Current procedural terminology (cpt®).

Jacob S. Berkowitz, Apoorva Srinivasan, Jose Miguel Acitores Cortina, Yasaman Fatapour, and Nicholas P Tatonetti. 2025. Biomedical text normalization through generative modeling. *Journal of Biomedical Informatics*, 167:104850.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690.

Florian Borchert, Ignacio Llorca, and Matthieu-P Schapranow. 2024. Improving biomedical entity linking for complex entity mentions with LLM-based text simplification. *Database: The Journal of Biological Databases and Curation*, 2024:baae067.

Katharina Danhauser, Yingding Wang, Christoph Klein, Uta Tacke, Larissa Mantoan, Laura Aurica Ritter, Florian Heinen, Chiara Nobile, and Moritz Tacke. 2025. Using large language models to extract information from pediatric clinical reports. *PLOS Digital Health*, 4(7):e0000919.

Pritam Deka, Anna Jurek-Loughrey, and Deepak P. 2022. Evidence extraction to validate medical claims in fake news detection. In *Health Information Science: 11th International Conference, HIS 2022, Virtual Event, October 28–30, 2022, Proceedings*, page 3–15, Berlin, Heidelberg. Springer-Verlag.

Nicholas J. Dobbins. 2024. Generalizable and Scalable Multistage Biomedical Concept Normalization Leveraging Large Language Models. *arXiv preprint*. ArXiv:2405.15122 [cs].

Arnaud Ferré, Louise Deléger, Robert Bossy, Pierre Zweigenbaum, and Claire Nédellec. 2020. C-Norm: a neural approach to few-shot entity normalization. *BMC Bioinformatics*, 21(S23):579.

Arnaud Ferré and Philippe Langlais. 2023. An analysis of entity normalization evaluation biases in specialized domains. *BMC Bioinformatics*, 24(1):227.

Yuhe Gao, Runxue Bao, Yuelyu Ji, Yiming Sun, Chenxi Song, Jeffrey P Ferraro, and Ye Ye. 2025. Transfer learning with clinical concept embeddings from large language. *AMIA Summits on Translational Science Proceedings*, 2025:167.

Israt Jahan, Md Tahmid Rahman Laskar, Chun Peng, and Jimmy Xiangji Huang. 2024. A comprehensive evaluation of large Language models on benchmark biomedical text processing tasks. *Computers in Biology and Medicine*, 171:108189.

David Kartchner, Jennifer Deng, Shubham Lohiya, Tejasri Kopparthi, Prasanth Bathala, Daniel Domingo-Fernández, and Cassie S. Mitchell. 2023. A Comprehensive Evaluation of Biomedical Entity Linking Models. *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, 2023:14462–14478.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, and 1 others. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021. Self-Alignment Pretraining for Biomedical Entity Representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4228–4238, Online. Association for Computational Linguistics.

Ivan Lopez, Akshay Swaminathan, Karthik Vedula, Sanjana Narayanan, Fateme Nateghi Haredasht, Stephen P. Ma, April S. Liang, Steven Tate, Manoj Maddali, Robert Joseph Gallo, Nigam H. Shah, and Jonathan H. Chen. 2025. Clinical entity augmented retrieval for clinical information extraction. *npj Digital Medicine*, 8(1):45.

National Library of Medicine. 2006. Loinc to cpt mapping. `https://www.nlm.nih.gov/research/umls/mapping_projects/loinc_to_cpt_map.html`. Accessed: 2024-02-03.

National Library of Medicine. 2025. Umls rest api documentation. `https://documentation.uts.nlm.nih.gov/rest/home.html`. Accessed: 2026-02-03.

OpenAI. 2022. text-embedding-ada-002.

OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence.

OpenAI. 2025a. GPT-4.1.

OpenAI. 2025b. OpenAI API Pricing.

Regenstrief Institute. 2025. Loinc: A universal standard for identifying laboratory observations.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*.

Nuno Seco, Tony Veale, and Jer Hayes. 2004. An intrinsic information content metric for semantic similarity in wordnet. In *Proceedings of the 16th European Conference on Artificial Intelligence*, ECAI'04, page 1089–1090, NLD. IOS Press.

Sonish Sivarajkumar, Mark Kelley, Alyssa Samolyk-Mazzanti, Shyam Visweswaran, and Yanshan Wang. 2024. An empirical evaluation of prompting strategies for large language models in zero-shot clinical natural language processing: algorithm development and validation study. *JMIR Medical Informatics*, 12:e55318.

Ali Soroush, Benjamin S. Glicksberg, Eyal Zimlichman, Yiftach Barash, Robert Freeman, Alexander W. Charney, Girish N. Nadkarni, and Eyal Klang. 2024. Large Language Models Are Poor Medical Coders — Benchmarking of Medical Code Querying. *NEJM AI*. Publisher: Massachusetts Medical Society.

Andy Wang, Cong Liu, Jingye Yang, and Chunhua Weng. 2024. Fine-tuning large language models for rare disease concept normalization. *Journal of the American Medical Informatics Association*, 31(9):2076–2083.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Cliff Wong, Sam Preston, Qianchu Liu, Zelalem Gero, Jaspreet Bagga, Sheng Zhang, Shrey Jain, Theodore Zhao, Yu Gu, Yanbo Xu, Sid Kiblawi, Srinivasan Yegnasubramanian, Taxiarchis Botsis, Marvin Borja, Luis M. Ahumada, Joseph C. Murray, Guo Hui Gan, Roshanthi Weerasinghe, Kristina Young, and 6 others. 2025. Universal Abstraction: Harnessing Frontier Models to Structure Real-World Data at Scale. *arXiv preprint*. ArXiv:2502.00943 [cs].

Cliff Wong, Sheng Zhang, Yu Gu, Christine Moung, Jacob Abel, Naoto Usuyama, Roshanthi Weerasinghe, Brian Piening, Tristan Naumann, Carlo Bifulco, and Hoifung Poon. 2023. Scaling Clinical Trial Matching Using Large Language Models: A Case Study in Oncology. In *Proceedings of the 8th Machine Learning for Healthcare Conference*, pages 846–862. PMLR. ISSN: 2640-3498.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico, USA. Association for Computational Linguistics.

Siyu Yang, Peiliang Zhang, Chao Che, and Zhaoqian Zhong. 2023. B-LBConA: a medical entity disambiguation model based on Bio-LinkBERT and context-aware mechanism. *BMC Bioinformatics*, 24(1):97.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *arXiv preprint*. ArXiv:2305.10601 [cs].

A Appendix: Prompts

We present the prompts for the LOINC-CPT normalization task. **zero_shot** and **few_shot** prompts included simple instructions for the task along with the input LOINC concept and candidate concepts from context retrieval. For Chain-of-Thought (CoT), Tree-of-Thoughts (ToT), and Graph-of-Thought (GoT) prompting, we extended the graph-of-thoughts framework (Besta et al., 2024) to create a LOINC-CPT normalization task. Graph-of-thoughts describes prompt operations as 'transformations of thoughts' that ultimately give the final response from the language model. The **generate_prompt** generates new thoughts from a single thought with a branching factor to define the number of thoughts generated. The **tot_improve_prompt** enhances a given thought using information provided in another thought (could be generated previously). The **aggregate_prompt** aggregates thoughts into new ones in GoT. Thus, for the LOINC-CPT normalization task, CoT was modeled with the `generate` operation, ToT with `generate` and `improve` operations, and GoT with `generate` and `aggregate` operations. All three prompt strategies used the **score_prompt** for scoring the LLM responses. Examples for the prompts were generated from UMLS, however, these can be alternately created by experts or by varying the ground truth data. Other configurations of the operations can be implemented depending on the task and desired complexity of operations.

zero-shot prompt:

<Instruction>

You are an expert clinical assistant. You are provided with a LOINC measurement concept delimited by <> and a list of candidate CPT concepts for matching, delimited by []. These candidates are generated using a BERT model, which assesses the similarity between the embeddings of the LOINC query and all CPT concepts. The first candidate in the ranked list is the most similar CPT concept to the LOINC query based on BERT similarity. For the given LOINC concept and candidate list, your task is to -

1. Assess the candidate CPT concepts based on your knowledge about laboratory terms and clinical vocabularies. Create a new candidates list by reordering and/or removing concepts. If the original list is correct, do not create a new list. If there are no good candidate concepts for the LOINC concept, create an empty list.
2. Then using the new list of candidates, select the best match for the given LOINC concept. There is only one best CPT match in the candidate list for the given LOINC concept. A good match in the candidate list of CPT concepts is when the candidate exactly matches the LOINC concept or is related to the LOINC concept.
3. Return the matched code in a list selected only from the candidates provided. If there is no good match in the list of candidates, return ['no match']. Return the output as list only and no other text.

</Instruction>

Term: <term>
Candidates: [candidates]

---

few-shot prompt:

<Instruction>

You are an expert clinical assistant. You are provided with a LOINC measurement concept delimited by <> and a list of candidate CPT concepts for matching, delimited by []. These candidates are generated using a BERT model, which assesses the similarity between the embeddings of the LOINC query and all CPT concepts. The first candidate in the ranked list is the most similar CPT concept to the LOINC query based on BERT similarity. For the given LOINC concept and candidate list, your task is to -

1. Assess the candidate CPT concepts based on your knowledge about laboratory terms and clinical vocabularies. Create a new candidates list by reordering and/or removing concepts. If the original list is correct, do not create a new list. If there are no good candidate concepts for the LOINC concept, create an empty list.
2. Then using the new list of candidates, select the best match for the given LOINC concept. There is only one best CPT match in the candidate list for the given LOINC concept. A good match in the candidate list of CPT concepts is when the candidate exactly matches the LOINC concept or is related to the LOINC concept.
3. Return the matched code in a list selected only from the candidates provided. If there is no good match in the list of candidates, return ['no match']. Return the output as list only and no other text.

</Instruction>

<Examples>

Term: 22360-2:HTLV I IgG Ab [Presence] in Serum
Candidates: [("86687","Antibody HTLV-I"), ("86296","HEPATITIS A ANTIBODY HAAB IGG AND IGM"), ("86709","Hepatitis A antibody HAAb IgM antibody"), ("82784","Gammaglobulin immunoglobulin IgA IgD IgG IgM each"), ("86701","Antibody HIV-1"), ("81381","HLA Class I typing high resolution ie alleles or allele groups one allele or allele group eg B*5701P each"),

("86001","Allergen specific IgG quantitative or semi-quantitative each allergen")]
Answer: ["86687"]

Term: 14416-2:Calcium [Moles/volume] in Pleural fluid
Candidates: [("82330","Calcium ionized"), ("82310","Calcium total"), ("82331","Calcium after calcium infusion test"), ("82340","Calcium urine quantitative timed specimen"), ("82360","Calculus quantitative analysis chemical"), ("80410","Calcitonin stimulation panel eg calcium pentagastrin This panel must include the following Calcitonin 82308 x 3")]
Answer: ["82310"]
</Examples>

Term: <term>
Candidates: [candidates]

---

generate_prompt for CoT, ToT, GoT:

<Instruction>
You are an AI medical coding assistant designed to help healthcare providers accurately map LOINC codes to their corresponding or nearest CPT codes. You will be provided with a LOINC laboratory code and name in the format query_lab_code:query_lab_name, referred to as the LOINC query, hereafter. Occasionally, you may also receive a ranked list of candidate CPT code matches, with each match formatted as a list of tuples; for example: []. These candidates are generated using a BERT model, which assesses the similarity between the embeddings of the LOINC query and all CPT concepts. The first tuple in the ranked list is the most similar CPT concept to the LOINC query based on BERT similarity.

Your goal is to identify the most appropriate CPT match(es) for the LOINC query. A suitable CPT match should:

1. Exactly correspond to the LOINC concept, or
2. Be closely related (e.g., same lab family, sub-test of the LOINC query, or a broader lab panel that includes the LOINC query).

Your output **must** follow this exact format and be enclosed between <Match> and </Match> tags: <Match>[("CPT_CODE","CPT_LABEL"), ("CPT_CODE","CPT_LABEL")]</Match>

Example of correct output: <Match>[("86256","Fluorescent noninfectious agent antibody titer each antibody"), ("82330","Calcium ionized")]</Match>
</Instruction>

<Approach>
1. Assess Candidate List Availability: If a candidate list is provided, review each candidate, then reorder, add, or remove concepts as needed to refine the list. If no candidate list is provided (i.e., the list is empty, "None", or ""): Generate a list of potential CPT matches for the LOINC query.
2. Evaluate Candidates: Use the refined candidate list to select the best CPT match(es) for the LOINC query. There may be multiple suitable matches or none at all.
3. Return Results: Output a list of the best matches. If no matches are found, return ['no match'].
</Approach>

<Examples>
LOINC query: 22360-2:HTLV I IgG Ab [Presence] in Serum
Candidate matches: [("86687","Antibody HTLV-I"), ("86296","HEPATITIS A ANTIBODY HAAB IGG AND IGM"), ("86709","Hepatitis A antibody HAAb IgM antibody"), ("82784","Gammaglobulin immunoglobulin IgA IgD IgG IgM each"), ("86701","Antibody HIV-1"), ("81381","HLA Class I typing high resolution ie alleles or allele groups one allele or allele group eg B*5701P each"), ("86001","Allergen specific IgG quantitative or semi-quantitative each allergen")]
<Match>[("86687","Antibody HTLV-I")]</Match>

LOINC query: 14416-2:Calcium [Moles/volume] in Pleural fluid
Candidate matches: [("82330","Calcium ionized"), ("82310","Calcium total"), ("82331","Calcium after calcium infusion test"), ("82340","Calcium urine quantitative timed specimen"), ("82360","Calculus quantitative analysis chemical"), ("80410","Calcitonin stimulation panel eg calcium pentagastrin This panel must include the following Calcitonin 82308 x 3")]
<Match>[("82310","Calcium total")]</Match>
</Examples>

<Input>
LOINC query: {query_lab_code}:{query_lab_name}
Candidate matches: {candidates}
</Input>

---

tot_improve_prompt for ToT:

<Instruction>
The following list represents the list of CPT matches for the LOINC query query_lab_code:query_lab_name:cpt_matches. These CPT matches might include wrong matches or might be an empty list shown as []. Your task is to find the correct CPT matches. Make sure if there are identified CPT matches, they are truly related to the LOINC query by ensuring that these matches are either exactly correspond to the LOINC query, or are closely related. Your output must follow this exact format and be enclosed between <Match> and </Match> tags:

<Match>
["CPT_CODE": "<CPT_CODE>", "CPT_LABEL": "<CPT_LABEL>","CPT_CODE": "<CPT_CODE>", "CPT_LABEL": "<CPT_LABEL>"]
</Match>

Replace <CPT_CODE> with the actual CPT code match and <CPT_LABEL> with its corresponding label. Only return valid matches inside the tags — do not include extra text.
Example of correct output: <Match>[("86256","Fluorescent noninfectious agent antibody titer each antibody"), ("82330","Calcium ionized")]</Match>
</Instruction>

<Approach>
To fix the incorrect list of CPT matches:

1. If there are CPT matches:
   (a) For each pair of CPT match, check which one is more related to the LOINC query and retain it and drop the other.
   (b) Iterate over the pairs of CPT matches, until you are left with a single match.
   (c) If the remaining CPT match is the exact same concept as LOINC query, sibling concept, parent concept, keep it, otherwise, remove it from the list.
   (d) Return the resulting match if any
2. If there are no CPT matches, indicate as an empty list

You are an AI medical coding assistant designed to help healthcare providers accurately map LOINC codes to their corresponding or nearest CPT codes. Use your knowledge to find the best match for query_lab_code:query_lab_name.
</Approach>

<Examples>
LOINC query: 13051-8:Protein S [Units/volume] in Platelet poor plasma
Matches: [("85306","Protein S [Units/volume] in Platelet poor plasma"), ("85305","Clotting inhibitors or anticoagulants protein S total")]
Reason: 85306 is the exact match, while 85305 is a sibling match. <Match>[("85306","Protein S

[Units/volume] in Platelet poor plasma")]</Match>

LOINC query: 14252-1:Smooth muscle Ab [Presence] in Serum
Matches: [("86015","Actin smooth muscle antibody ASMA each"), ("86256","Fluorescent noninfectious agent antibody titer each antibody"),(86381,"Mitochondrial antibody eg M2 each")]
Reason: The actual match is 86256 which is not in the list of matches. The closest match 86255 is a sibling match, while 86015 and 86381 are cousin matches (more distant than the sibling match).
<Match>[("86256","Fluorescent noninfectious agent antibody titer, each antibody")]</Match>
</Examples>

LOINC query: {query_lab_code}:{query_lab_name}

---

score_prompt for CoT, ToT, GoT:

<Instruction>
The following is the list of CPT matches for LOINC query query_lab_code:query_lab_name: current. CPT matches are formatted as a list of tuples, where each tuple in the list represents a match and has only two elements, where the first element is a CPT code and the second element is its corresponding CPT label. Score each match in terms of how closely it represents the LOINC query. A score of 5 implies that the match is the exact match, a parent match gets a score of 4, a sibling match gets a score of 3, and grandparent or grand child matches get a score of 2, cousins get a score of 1, and everything else get a score of 0. You may provide reasoning for your scoring, but you should put the final scores in a list between the tags <Score> and </Score>. The order of final score list corresponds to the order of CPT matches i.e., the first and second scores correspond to the first and second CPT matches, respectively.
</Instruction>

<Example>
LOINC query: 13051-8:Protein S [Units/volume] in Platelet poor plasma
Matches: [("85306","Protein S [Units/volume] in Platelet poor plasma"), ("85305","Clotting inhibitors or anticoagulants protein S total")]
Reason: 85306 is the exact match, while 85305 is a sibling match.
<Score>[5,3]</Score>
</Example>
LOINC query: {query_lab_code}:{query_lab_name}

---

aggregate_prompt for GoT:

<Instruction>
The following list of lists presents the potential CPT matches for query_lab_code:query_lab_name: current. Combine and merge them into one list to only retain top 5 most relevant ones, with no repeated matches. Your output must follow this exact format and be enclosed between <Match> and </Match> tags:
<Match>[("<CPT_CODE>","<CPT_LABEL>"), ("<CPT_CODE>","<CPT_LABEL>")]</Match>
</Instruction>