MedCodER: A Generative AI Assistant for Medical Coding

Krishanu Das Baksi[1]*, Elijah Soba[2]*, John J. Higgins[2], Ravi Saini[1], Jaden Wood[2], Jane Cook[2], Jack Scott[2], Nirmala Pudota[1], Tim Weninger[3], Edward Bowen[2], Sanmitra Bhattacharya[2],

[1]Deloitte & Touche Assurance & Enterprise Risk Services India Private Limited, [2]Deloitte & Touche LLP, [3]University of Notre Dame

Abstract

Medical coding standardizes clinical data but is both time-consuming and error-prone. Traditional Natural Language Processing (NLP) methods struggle with automating coding due to the large label space, lengthy text inputs, and the absence of supporting evidence annotations that justify code selection. Recent advancements in Generative Artificial Intelligence (AI) offer promising solutions to these challenges. In this work, we introduce MedCodER, an *emerging* Generative AI framework for automatic medical coding that leverages extraction, retrieval, and re-ranking techniques as core components. MedCodER achieves a micro-F1 score of 0.62 on International Classification of Diseases (ICD) code prediction, significantly outperforming state-of-the-art methods. Additionally, we present a new dataset containing medical records annotated with disease diagnoses, ICD codes, and supporting evidence texts (https://doi.org/10.5281/zenodo.13308316). Ablation tests confirm that MedCodER's performance depends on the integration of each of its aforementioned components, as performance declines when these components are evaluated in isolation.

1 Introduction

The International Classification of Diseases (ICD)[1], developed by the World Health Organization (WHO)[2], is a globally recognized standard for recording, reporting, and monitoring diseases. In the United States, the use of ICD codes is mandated by the U.S. Department of Health and Human Services (HHS) for entities covered by the Health Insurance Portability and Accountability Act for insurance purposes.

ICD codes have undergone various revisions over time to reflect advancements in medical science[3]. The 10th revision, known as ICD-10-CM (referred to as ICD-10 hereafter) in the U.S, is the standard for modern clinical coding and comprises over 70,000 distinct codes. These codes follow a specific alphanumeric structure (Hirsch et al., 2016) and are organized into a hierarchical ontology based on the medical concepts they represent. ICD-10 differs significantly from previous versions, making translation between versions challenging.

Accurate ICD coding is essential for medical billing, health resource allocation, and medical research (Campbell and Giadresco, 2020). This task is performed by specialized professionals known as medical or clinical coders, who use a combination of manual techniques and semi-automated tools to process large volumes of medical records. Their primary responsibility is to accurately assign ICD-10 codes to medical records based on documented diagnoses and procedures. The coding process is often time-consuming and costly, and the difficulty depends on the complexity of the patient records and the level of detail in the documentation. Errors in ICD coding can have significant financial and legal implications for patients, healthcare providers, and insurers. Despite the critical importance of accurate coding, few reliable solutions exist to supplement or automate this process.

Automation of ICD coding is an active research area within the NLP community. While various approaches have been proposed, recent methods typically frame this task as a multi-label classification problem: given the raw text of a medical record, the goal is to predict each of the relevant ICD codes (Yan et al., 2022). Although the objective is straightforward, several challenges make automatic ICD coding difficult. These include the

*These authors contributed equally to this work.

[1]https://www.cms.gov/medicare/coding-billing/icd-10-codes

[2]https://www.who.int/standards/classifications/classification-of-diseases

[3]https://www.cdc.gov/nchs/hus/sources-definitions/icd.htm

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Industry Track), pages 449–459
April 30, 2025 ©2025 Association for Computational Linguistics

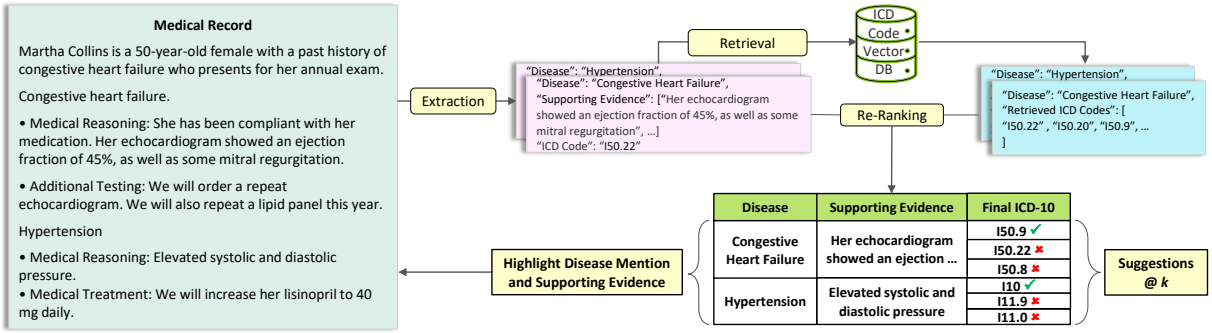

Figure 1: A schematic diagram of the MedCodER framework illustrates three primary components: extraction of disease diagnoses, supporting evidence and an initial list of ICD-10 codes, retrieval of candidate ICD-10 codes for the extracted diagnosis using a vector database, and re-ranking these combined codes to produce a final list of k ICD-10 codes. Extracted disease mentions and supporting evidence are mapped back to the medical record for in-context highlighting, aiding medical coders in the coding process.

extremely large label space, the diversity and lack of standardization in medical record data, and the severely imbalanced distribution of labels. State of the art NLP techniques still fall short of fully automating the process, and these methods often lack interpretability.

Large Language Models (LLMs) have shown remarkable capabilities in text generation and reasoning, particularly in zero-shot scenarios. However, early efforts to apply LLMs for automatic ICD coding have produced unsatisfactory results (Boyle et al., 2023; Soroush et al., 2024). In the present work, we hypothesize that augmenting the intrinsic (parametric) knowledge of LLMs with complementary techniques, such as retrieval (Lewis et al., 2020) and re-ranking (Sun et al., 2023), can significantly improve their accuracy in this domain.

Furthermore, evaluation and benchmarking for automatic ICD coding tools, particularly those based on Generative AI, are challenged by restrictive licensing terms and lack of expert annotations. Medical records contain sensitive data that discourage the use of third party API providers such as OpenAI or Anthropic. In addition, a majority of datasets in this space only contain ICD-10 labels and not the text that justifies it. In practice, the justification of an ICD-10 code is just as important as its classification.

To address the challenges associated with applying Generative AI approaches to ICD coding and the lack of third-party-friendly ICD coding datasets, this paper makes the following contributions:

1. We introduce an open-source dataset designed for evaluating ICD coding methodologies, including those based on Generative AI. This dataset includes not only ICD-10 codes but also extracted diagnoses and supporting evidence texts, which facilitate the development and assessment of interpretable ICD coding methods.

2. We describe the **Med**ical **Cod**ing using **E**xtraction, **R**etrieval, and re-ranking (**MedCodER**) framework, an accurate and interpretable *emerging* approach to ICD coding that leverages LLMs along with retrieval and re-ranking techniques. MedCodER first extracts disease diagnoses, supporting evidence, and an initial list of ICD-10 codes from medical records. It then retrieves candidate ICD-10 codes using semantic search and re-ranks the combined codes from previous steps to produce the final ICD-10 code predictions.

3. We evaluate the performance of the MedCodER framework compared to state-of-the-art (SOTA) methods using our dataset.

2 Related Research

2.1 Automatic ICD Coding

Automated ICD coding is a challenging NLP problem, approached through rule-based (Kang et al., 2013; Farkas and Szarvas, 2008), traditional machine learning (Scheurwegs et al., 2016, 2017), and deep learning methods (Ji et al., 2024). Recent methods often treat it as a multi-label classification task, utilizing architectures like convolutional (Mullenbach et al., 2018; Cao et al., 2020), recurrent (Yu et al., 2019; Guo et al., 2020), graph neural networks (Wang et al., 2020), and transformers (Huang et al., 2022). Although generative AI and

LLMs have been explored for ICD coding (Boyle et al., 2023; Soroush et al., 2024), results have been mixed.

An analysis by Edin et al. 2023 compared SOTA ICD coding models on MIMIC datasets and found that PLM-ICD (Huang et al., 2022) excelled on MIMIC IV, but common ICD coding challenges persisted, with *more than half* of ICD-10 codes misclassified. This suggests the potential of zero-shot models like LLMs for more reliable solutions.

LLM-based ICD coding research has yielded mixed outcomes. One study achieved only a 34% match rate using a dataset from Mount Sinai (Soroush et al., 2024), while an LLM-guided tree search method achieved competitive results (Boyle et al., 2023), though it lacked transparency in code selection and was resource-intensive.

2.2 Disease Extraction

Disease extraction, a key component of both traditional medical coding and the MedCodER framework, involves identifying disease entities from medical records and is a form of Named Entity Recognition (NER) in biomedical NLP (Durango et al., 2023). While often overlooked in ICD coding methods, disease NER is crucial for accurate retrieval and re-ranking of ICD codes.

Domain-specific models like BioBERT (Lee et al., 2019), pre-trained on biomedical literature, achieve high F1 scores (86-89%) on benchmark datasets but are more effective with data similar to their training sets. Recent advancements such as Universal Named Entity Recognition (UniNER), Generalist Model for Named Entity Recognition (GLiNER), and NuExtract all have shown competitve zero-shot performance on traditional NER by training or fine-tuning Large Language models.

Unlike general NER, which may identify a broad range of disease mentions, ICD-10 extraction focuses on diagnosing diseases relevant for coding, reducing noise and minimizing errors in billing and documentation. Our approach targets precise disease extraction aligned with ICD-10 codes.

2.3 Retrieval and Re-ranking

While traditional NLP methods often frame automatic ICD coding as a multi-label classification task, it can also be approached as a retrieval and re-ranking problem. In this perspective, the goal is to retrieve the most relevant ICD codes for a given medical record and then re-rank them into a prioritized list. This approach addresses the challenge of dealing with large label spaces by filtering out irrelevant codes, resulting in a more manageable set of candidates.

Prior work has explored the retrieval and re-ranking paradigm using pre-trained ICD coding models (Tsai et al., 2021). In this approach, the top k most probable codes are selected from the pre-trained model and re-ranked based on label correlation. However, its effectiveness is limited by the retriever's ability to produce relevant codes within the top k. Embedding models have also been utilized to retrieve relevant codes for a given medical record (Niu et al., 2023). While promising, this approach is limited by the challenges of long input texts and lacks a clear rationale for ICD-10 code selections. In contrast, the MedCodER framework addresses these limitations by extracting disease-related text segments to enhance the retrieval of relevant ICD-10 codes.

3 MedCodER Framework

Here we introduce the MedCodER framework, which is illustrated in Fig. 1. MedCodER is an interpretable and explainable ICD coding framework comprised three components: (1) extraction, (2) retrieval, and (3) re-ranking. In this section, we describe each component and its relevance to ICD-10 coding.

3.1 Step 1: Disease Diagnoses, Supporting Evidence & ICD-10 Code Extraction

MedCodER begins by employing an LLM to extract disease diagnoses, supporting evidence, and ICD-10 codes from medical records. Disease diagnoses refer to clinical terms for a patient's condition, while supporting evidence includes related details such as test results and medications. We prompt the LLM to output these entities in JSON format (see Appendix A).

Drawing inspiration from Chain-of-Thought (CoT) prompting (Wei et al., 2022), we asked the LLM to first reason about relevant text from the medical record before generating ICD-10 codes, mimicking the workflow of medical coders (Appendix A). The extracted diagnoses are used in the retrieval step, while the supporting text and generated ICD-10 codes are used in the re-ranking step. To mitigate against hallucinations in the LLM output, we match the extracted text to the medical record text using fuzzy matching and BM25 similarity scores.

3.2 Step 2: ICD-10 Retrieval Augmentation

Following the LLM text extraction, we generate a candidate set of ICD-10 codes through semantic search between extracted diagnoses and the descriptions of valid ICD-10 codes. This approach mitigates the large label space issue by reducing the number of potential codes to a more manageable set.

For the semantic search, we compiled textual descriptions of valid codes from the ICD-10 ontology and equivalent descriptions from the Unified Medical Language System (UMLS) Metathesaurus[4], providing accurate handling of medical synonyms. We then embedded these descriptions and tagged each code with metadata related to the ontology, such as chapter, block, and category (Boyle et al., 2023). During inference, disease diagnoses are embedded, and the top k most similar ICD-10 codes based on cosine distance are retrieved for each diagnosis. This results in a ranked list of ICD-10 codes directly mapped to specific diagnoses, enhancing interpretability.

3.3 Step 3: Code-to-Record Re-ranking

In the final step, the retrieved codes from the Step 2 and those generated by the LLM are re-ranked to produce the final list of predicted ICD-10 codes. This re-ranking is performed using an LLM, but only the extracted diagnoses and supporting evidence are considered, allowing the LLM to prioritize based on relevant information. We follow the RankGPT framework (Sun et al., 2023), with modifications specific to ICD-10 coding.

4 Experimental Methodology

4.1 Dataset

Because current ICD coding benchmark datasets, like MIMIC III and IV, have restrictions on use with off-the-shelf, externally-hosted LLMs, and because they lack annotations of supporting evidence text, they cannot be used in typical Generative AI solutions. To address this, we created a new dataset that extends the Ambient Clinical Intelligence Benchmark (ACI-BENCH) dataset (Yim et al., 2023). ACI-BENCH is a synthetic dataset containing 207 transcribed conversations that simulate doctor-patient interactions. These notes were reviewed and revised, as necessary, by medical domain experts to ensure their accuracy and realism, closely mimicking real-world clinical notes.

We extended the ACI-BENCH dataset by manually annotating each clinical note with ICD-10 codes, disease diagnoses, and supporting evidence texts. This task was performed with the assistance of an expert medical coder, who has over 20 years of experience and holds certifications such as the American Health Information Management Association (AHIMA) Certified Coding Specialist (CCS) and the American Academy of Professional Coders (AAPC) Certified Professional Coder (CPC). Of the 207 clinical notes, three were deemed unworthy of coding. The remaining notes were coded in two batches: the first batch included 184 notes, 360 ICD-10 codes with diagnoses, and 737 supporting evidence texts, and is used to evaluate the results of various MedCodER components. The second batch, consisting of 20 notes, is intended for use as a hold out set.

4.2 Methodology

We evaluate the performance of MedCodER's components using the extended ACI-BENCH dataset and comparing them with SOTA approaches. Because most automatic ICD coding baselines produce a single ICD-10 code per diagnosis, we compare our k@1 results against these. We also demonstrate performance trade-offs with increasing values of k. For non-LLM baselines, we use publicly available pre-trained weights, and for LLM-based experiments, we use top-performing models[5], such as GPT-4o, Claude 3.5 Sonnet and Llama 405B (MedCodER with GPT-4o is simply referred to as MedCodER henceforth; results of ICD-10 coding with Claude and Llama models are shown in the Appendix B).

4.3 Metrics

We report results with micro precision and micro recall for each sub-task. Consistent with current evaluation approaches for NER and ICD coding, we focus on micro metrics because, in extremely large label spaces, it is crucial to treat each instance equally rather than each class. This approach emphasizes the performance of our framework per document rather than per ICD-10 code.

To evaluate disease diagnoses extraction, we use set-based, exact-match metrics. Our metric choice is motivated by the retrieval subtask. Because vector search is location-independent, we disregard

[4]https://www.nlm.nih.gov/research/umls/index.html

[5]As per the HELM Lite leaderboard https://crfm.stanford.edu/helm/lite/latest/#/leaderboard

Model	Recall	Precision	F1
BioBERT	0.44	0.07	0.12
UniNER	0.67	0.11	0.19
GLiNER	0.78	0.15	0.25
NuExtract v1.5	**0.85**	0.79	0.82
MedCodER	**0.85**	**0.81**	**0.83**

Table 1: Disease diagnoses extraction results.

Model	Recall	Precision	F1
PLM-ICD	0.57	0.31	0.40
Simple Prompt	0.52	0.32	0.40
LLM Tree-Search	0.38	0.10	0.16
MedCodER@1	**0.68**	**0.57**	**0.62**

Table 2: ICD-10 coding results for MedCodER compared to SOTA baselines.

text positions when computing extraction performance. Additionally, we treat exact matches case insensitively, differing from traditional NER evaluations.

5 Results

In this section, we present the results of both the baselines and the MedCodER framework.

5.1 Disease Diagnoses and Supporting Evidence Extraction

The results of disease diagnoses extraction are shown in Table 1. We find that MedCodER's disease diagnoses extraction for ICD-10 coding outperforms most other NER specialized models, validating our hypothesis that prompting for specific ICD-10 diagnoses is better for this task. Although NuExtract was able to approximate the performance of GPT-4o in disease extraction, its performance significantly declined when prompted for both disease and supporting evidence. Because disease extraction directly determines the ICD-10 codes produced, these results also represent an upper bound on ICD-10 coding performance.

Because this dataset is the first to include supporting evidence for ICD-10 codes and their associated diagnoses, we lacked a baseline for comparison. In our experiments with various prompting approaches, partial match recall ranged from 0.75 to 0.82, and precision ranged from 0.24 to 0.30 (detailed results are omitted due to space constraints). The low precision indicates that the model extracts some non-relevant evidence, potentially introducing errors in the re-ranking process where supporting evidence texts are used. Despite the low precision, our full framework results in Table 2 suggest that the extracted supporting evidence aids re-ranking. This task is more nuanced and challenging than disease extraction, highlighting the need for performance improvements in future work.

5.2 ICD-10 Coding

Table 2 presents MedCodER results when filtering for only the top ranked ICD-10 code per diagnosis. For baselines, we used the pre-trained weights of PLM-ICD on MIMIC IV from Edin et al. (2023) and a 50-call limit for the LLM Tree-Search. These methods represent the SOTA deep learning (Edin et al., 2023) and generative AI based solutions (Boyle et al., 2023) for automatic ICD-10 coding. MedCodER outperforms these baselines, significantly enhancing ICD-10 coding performance while remaining interpretable. The LLM Tree-Search method performed lower than expected, which we attribute to the call limit and error propagation mentioned in their work.

We observe that GPT-4o outperforms both Claude 3.5 Sonnet and Llama 405B (Appendix B), which can be attributed to its enhanced extraction and re-ranking capabilities.

5.3 Ablation Results

To evaluate the efficacy of retrieval and re-ranking on ICD coding performance, we conducted an ablation study. The results are shown in Fig. 2. The variations of MedCodER used in the study are:

- MedCodER-`Prompt`: Uses only the ICD-10 codes from MedCodER prompt. This value does not change with the number of retrieved documents k.
- MedCodER-`Retrieve`: Uses only the retrieved ICD-10 codes, without re-ranking.
- MedCodER-`Prompt+Retrieve`: Uses both prompted and retrieved ICD-10 codes, without re-ranking.
- MedCodER: The entire framework with each constituent component, *i.e.*, prompted and retrieved ICD-10 codes after re-ranking.

We observe that re-ranking the combined set of prompted and retrieved ICD-10 codes outperforms

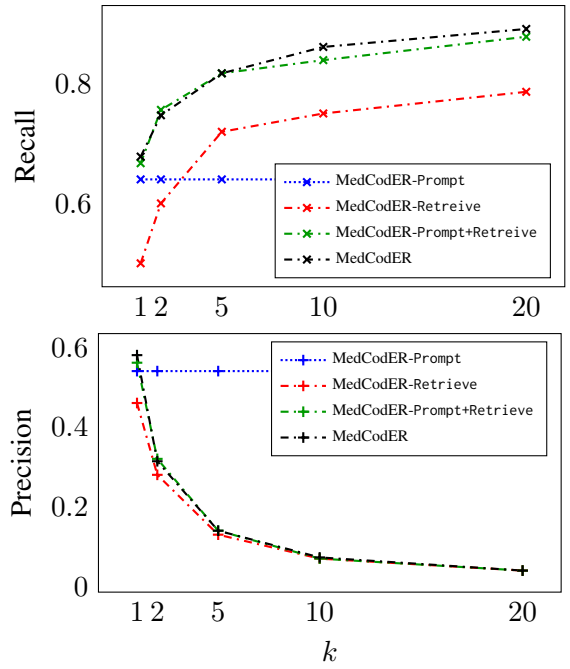

Figure 2: Recall and Precision @k for variations of MedCodER framework

using either method alone. Recall increases monotonically with addition of retrieval, meaning our search produces semantically relevant hits. As expected, the precision decays as we produce more output codes. Contrary to prior work (Soroush et al., 2024), our results with MedCodER-Prompt show that LLMs can perform well on ICD-10 prediction with careful prompt engineering. We attribute this to prompt design, where the LLM is prompted to first generate the diagnoses and supporting evidence texts before it is prompted to generate the ICD-10 codes, akin to CoT prompting (Wei et al., 2022).

5.4 Error Analysis

We conducted an error analysis to highlight MedCodER's limitations and suggest future research directions.

Table 3 presents failure cases for each component of our framework (k=1). We show cases where the extracted disease diagnosis matched the ground truth to highlight errors in prompting and retrieval approaches for ICD-10 coding. We observed that that even when the codes are incorrect, they are often very close semantically. Additionally, MedCodER can overcome prompting and retrieval shortcomings due to its re-ranking capability.

Medical Record Snippet and Ground Truth Diagnosis	**Ground Truth ICD-10 and Description**
Regarding her depression, the patient feels that it is well managed on Effexor	F32.A: Depression, unspecified

Model	**Prediction**	**?**
MedCodER-Prompt	F32.9	✗
MedCodER-Retrieve	F33.9	✗
MedCodER	F32.A	✓

Edema and ecchymosis surrounding the knee. Positive pain to palpation. Assessment: Right Knee Contusion	S80.01XA: Contusion of right knee, initial encounter

Model	**Prediction**	**?**
MedCodER-Prompt	S80.01XA	✓
MedCodER-Retrieve	S80.01	✗
MedCodER	S80.01XA	✓

Today I discussed conservative options for left shoulder impingement with the patient	M75.42: Impingement syndrome of left shoulder

Model	**Prediction**	**?**
MedCodER-Prompt	M75.40	✗
MedCodER-Retrieve	M75.42	✓
MedCodER	M75.42	✓

His examination is consistent with rather severe post-traumatic stenosing tenosynovitis of the right index finger.	M65.321: Trigger finger, right index finger

Model	**Prediction**	**?**
MedCodER-Prompt	M22.40	✗
MedCodER-Retrieve	M17.2	✗
MedCodER	M22.2X1	✗

Table 3: Error analysis of each variation of the MedCodER framework with associated disease diagnosis

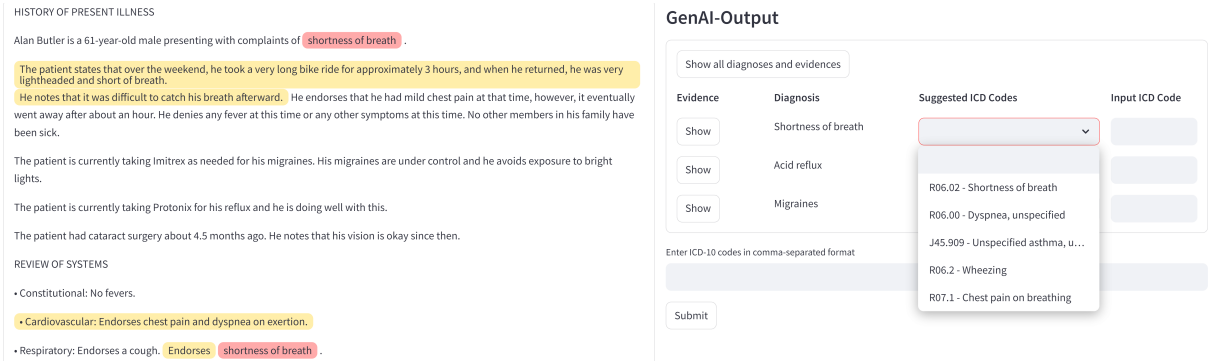

Figure 3: A representation of MedCodER in action. On the left, the medical record is annotated with the disease diagnosis for shortness of breath and its supporting evidence texts. On the right, the corresponding top 5 ICD-10 code suggestions are shown. Other diagnoses and supporting evidence texts can be toggled to show or hide using the 'Show' buttons next to them.

6 Discussion

Unlike fully automated ICD coding solutions, MedCodER is an AI-assisted coding tool to enhance medical coding workflows. To illustrate this, we designed a preliminary but functional user interface (Figure 3) which is current being beta-tested by our coders prior to production integration with an enterprise medical coding tool. For each predicted diagnosis, a button in the UI is available to highlight the corresponding text spans containing disease mentions and supporting evidence texts. Additionally, a dropdown menu displays MedCodER's top five most relevant ICD-10 codes per diagnosis. Coders can review and select a code from the dropdown or input a different code.

In future work, we intend to investigate biomedical domain-specific LLMs, as MedCodER depends on the LLM's understanding of diseases, supporting evidence, and ICD-10 codes. Our framework's flexibility in replacing individual components allows us to integrate the latest SOTA models as the generative AI landscape evolves. For example, Appendix C demonstrates the results of MedCodER utilizing MedCPT (Jin et al., 2023), a domain-specific embedding model trained on PubMed articles, as the backend embedder for retrieval, instead of the OpenAI text-embedding-ada-002 model used in our current work.

Although the dataset discussed in this paper is in text format, real-world medical records often come in other formats, such as scanned or digital PDFs. These formats require additional preprocessing to handle any handwritten sections, tables, and other poorly-formatted data. Furthermore, the fixed context length of LLMs may require extra pre-processing steps for longer records. We hypothesize that performance should remain relatively consistent for larger records, provided they are divided into smaller consecutive chunks and processed sequentially.

7 Conclusions

In conclusion, we present MedCodER–an innovative, interpretable framework that surpasses current SOTA methods in automated ICD coding. By integrating extraction, retrieval, and re-ranking techniques with LLMs, MedCodER achieves a synergy that no single component can match alone. Our analyses confirm that this holistic approach not only boosts coding accuracy but also maintains transparency in code selection. Additionally, our error analysis has pinpointed key areas for future improvement, paving the way for more robust and efficient solutions. Finally, our preliminary integration of MedCodER as an AI-based assistant for medical coders demonstrates its potential to enhance both efficiency and accuracy in clinical settings, promising significant practical benefits.

References

Joseph S. Boyle, Antanas Kascenas, Pat Lok, Maria Liakata, and Alison Q. O'Neil. 2023. Automated clinical coding using off-the-shelf large language models. *Preprint*, arXiv:2310.06552.

Sharon Campbell and Katrina Giadresco. 2020. Computer-assisted clinical coding. *Health Information Management Journal*, 49(1):5–18.

Pengfei Cao, Chenwei Yan, Xiangling Fu, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng

Chong. 2020. Clinical-coder: Assigning interpretable ICD-10 codes to Chinese clinical notes. In *ACL*, pages 294–301.

María C. Durango, Ever A. Torres-Silva, and Andrés Orozco-Duque. 2023. Named entity recognition in electronic health records: A methodological review. *Healthcare Informatics Research*, 29(4):286–300.

Joakim Edin, Alexander Junge, Jakob D. Havtorn, Lasse Borgholt, Maria Maistro, Tuukka Ruotsalo, and Lars Maaløe. 2023. Automated medical coding on mimic-iii and mimic-iv: A critical review and replicability study. In *SIGIR*. ACM.

Richárd Farkas and György Szarvas. 2008. Automatic construction of rule-based icd-9-cm coding systems. In *BMC bioinformatics*, volume 9, pages 1–9. Springer.

Donglin Guo, Guihua Duan, Ying Yu, Yaohang Li, Fang-Xiang Wu, and Min Li. 2020. A disease inference method based on symptom extraction and bidirectional long short term memory networks. *Methods*, 173:75–82.

JA Hirsch, G Nicola, G McGinty, RW Liu, RM Barr, MD Chittle, and L Manchikanti. 2016. Icd-10: history and context. *American Journal of Neuroradiology*, 37(4):596–599.

Chao-Wei Huang, Shang-Chi Tsai, and Yun-Nung Chen. 2022. PLM-ICD: Automatic ICD coding with pretrained language models. In *Proceedings of the 4th Clinical Natural Language Processing Workshop*, pages 10–20, Seattle, WA. ACL.

Shaoxiong Ji, Xiaobo Li, Wei Sun, Hang Dong, Ara Taalas, Yijia Zhang, Honghan Wu, Esa Pitkänen, and Pekka Marttinen. 2024. A unified review of deep learning for automated medical coding. *ACM Computing Surveys*.

Qiao Jin, Won Kim, Qingyu Chen, Donald C Comeau, Lana Yeganova, W John Wilbur, and Zhiyong Lu. 2023. Medcpt: Contrastive pre-trained transformers with large-scale pubmed search logs for zero-shot biomedical information retrieval. *Bioinformatics*, 39(11).

Ning Kang, Bharat Singh, Zubair Afzal, Erik M van Mulligen, and Jan A Kors. 2013. Using rule-based natural language processing to improve disease normalization in biomedical text. *Journal of the American Medical Informatics Association*, 20(5):876–881.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeurIPS*, 33:9459–9474.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *NAACL-HLT*, pages 1101–1111, New Orleans, Louisiana. ACL.

Kunying Niu, Yifan Wu, Yaohang Li, and Min Li. 2023. Retrieve and rerank for automated icd coding via contrastive learning. *Journal of Biomedical Informatics*, 143:104396.

Elyne Scheurwegs, Boris Cule, Kim Luyckx, Léon Luyten, and Walter Daelemans. 2017. Selecting relevant features from the electronic health record for clinical code prediction. *Journal of biomedical informatics*, 74:92–103.

Elyne Scheurwegs, Kim Luyckx, Léon Luyten, Walter Daelemans, and Tim Van den Bulcke. 2016. Data integration of structured and unstructured sources for assigning clinical codes to patient stays. *Journal of the American Medical Informatics Association*, 23(e1):e11–e19.

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*, 1(5):AIdbp2300040.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *EMNLP*, pages 14918–14937, Singapore. ACL.

Shang-Chi Tsai, Chao-Wei Huang, and Yun-Nung Chen. 2021. Modeling diagnostic label correlation for automatic icd coding. *Preprint*, arXiv:2106.12800.

Wenlin Wang, Hongteng Xu, Zhe Gan, Bai Li, Guoyin Wang, Liqun Chen, Qian Yang, Wenqi Wang, and Lawrence Carin. 2020. Graph-driven generative models for heterogeneous multi-task learning. *AAAI*, 34(01):979–988.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 35:24824–24837.

Chenwei Yan, Xiangling Fu, Xien Liu, Yuanqiu Zhang, Yue Gao, Ji Wu, and Qiang Li. 2022. A survey of automated international classification of diseases coding: development, challenges, and applications. *Intelligent Medicine*, 2(03):161–173.

Wen-wai Yim, Yujuan Fu, Asma Ben Abacha, Neal Snider, Thomas Lin, and Meliha Yetisgen. 2023. Aci-bench: a novel ambient clinical intelligence dataset for benchmarking automatic visit note generation. *Nature Scientific Data*.

Ying Yu, Min Li, Liangliang Liu, Zhihui Fei, Fang-Xiang Wu, and Jianxin Wang. 2019. Automatic icd code assignment of chinese clinical notes based on multilayer attention birnn. *Journal of biomedical informatics*, 91:103114.

Appendix A Prompts

Simple Prompt

You are an expert clinical coder. Given a medical record, your task is to output all relevant ICD-10 codes that are relevant to the text. Output the ICD10 codes as a comma separated list.

Medical Record:
{medical_note}

ICD10 codes:

MedCodER Prompt

You are an expert clinical coder. Your task is to identify all the disease diagnoses present in the given Medical Note.
Medical Note:
{medical_note}

The output must be a valid JSON list, where each element of the list must contain the following:
1. Disease: The disease mentioned in the Medical Note.
2. Supporting Evidence: The list of sentences from the Medical Note which contain information related to diagnosis, assessment, medical reasoning, treatment plans, medications, referrals for the Disease. Do not include sentences about the medical history of the patient.
3. ICD-10-CM Code: The ICD-10 code for the Disease.

Here is an example output:
[
{
"Disease": "<disease diagnosis 1>",
"Supporting Evidence": [<list of sentences which which contain any kind of information related to diagnosis, assessment, medical reasoning, treatment plans, medications, referrals for disease diagnosis 1>],
"ICD-10-CM Code": <ICD-10-CM Code for diagnosis 1>
},
{
"Disease": "<disease diagnosis 2>",
"Supporting Evidence": [<list of sentences which which contain any kind of information related to diagnosis, assessment, medical reasoning, treatment plans, medications, referrals for disease diagnosis 2>],
"ICD-10-CM Code": <ICD-10-CM Code for diagnosis 2>
},
]

Output only the JSON and nothing else.
Output:

Table 4: Baseline simple prompt and the MedCodER prompt

Appendix B MedCodER with various SOTA LLMs

Model	Recall	Precision	F1
Llama 405B	0.56	0.37	0.45
Claude 3.5 Sonnet	**0.68**	0.24	0.35
GPT-4o	**0.68**	**0.57**	**0.62**

Table 5: ICD-10 coding results @1 for MedCodER with various SOTA LLMs

Appendix C MedCodER with MedCPT embeddings

Model	Recall	Precision	F1
Llama 405B	0.54	0.36	0.43
Claude 3.5 Sonnet	0.52	0.36	0.42
GPT-4o	**0.68**	**0.39**	**0.49**

Table 6: ICD-10 coding results @1 for MedCodER with various LLMs using MedCPT embeddings for retrieval