Data Quality in Clinical Coding: A Critical Analysis and Preliminary Study

Supriya Khadka
Centre for Computational Sciences and Mathematical Modelling, Coventry University
Coventry, United Kingdom
khadkas25@uni.coventry.ac.uk

Xiaorui Jiang*
Health Informatics Research Group, Information School, University of Sheffield
Sheffield, United Kingdom
xiaorui.jiang@sheffield.ac.uk

Vasile Palade
Centre for Computational Sciences and Mathematical Modelling, Coventry University
Coventry, United Kingdom
vasile.palade@coventry.ac.uk

ABSTRACT

Clinical coding is a vital yet complex component of healthcare practice. While automated coding systems have advanced significantly, they still rely on imperfect training data, which affects the quality of their predictions. A key issue contributing to this problem, often overlooked in current research, is the presence of errors and undercoding in widely used clinical coding datasets. In this work, we uncover substantial undercoding and annotation errors in commonly used datasets and present the first empirical study on their impact on the performance of automated clinical coding algorithms. We develop a three-stage pipeline combining a large language model (LLM)-based coding evidence extractor, a multiclass classifier trained on silver-labeled evidence from the MIMIC-IV [14] dataset, and a verification step using LLMs to assess the validity of each code. This approach reveals that approximately 80% of clinical notes in the MDACE [3] dataset and 86% of notes in CodiEsp [25] are likely to be undercoded or contain errors. Furthermore, correcting the errors leads to a relative improvement of 4% in precision and 7% in recall for the current state-of-the-art clinical coding model, PLM-ICD [9]. These findings make it clear that not only the algorithm, but also dataset integrity, plays a critical role in automated clinical coding.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**;
• **Applied computing** → **Health informatics**.

KEYWORDS

Automated Clinical Coding, Evidence Extraction, LLM-based Verification, Undercoding, EHR Errors

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ACM'25, Month DD–DD, 2025, Place
© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Supriya Khadka, Xiaorui Jiang, and Vasile Palade. 2018. Data Quality in Clinical Coding: A Critical Analysis and Preliminary Study. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (ACM'25).* ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Clinical Coding is the process of assigning systematic codes to patients' medical records, which includes diagnoses, treatments, lab tests, and medications. These codes are crucial for billing, insurance claims, policymaking, and healthcare analysis. Currently, clinical coding is manually performed by trained professionals, making the process costly, time-consuming, and error-prone [36] [16]. As a result, researchers have been working on developing automated medical coding systems for a long time. Automated medical coding involves performing medical coding using various forms of automation. The primary focus of automated clinical coding has been on International Classification of Diseases (ICD) codes, likely because these codes are used for diagnosing diseases, making them more universal and fostering uniformity within the research community. Despite several decades of research, this problem remains challenging due to the inherent complexity of clinical codes.

Clinical codes have a large label space and an imbalanced distribution. With the transition from ICD-9 to ICD-10 in 1992, the number of codes increased significantly, from 3,882 to approximately 70,000 [32]. This increase in granularity, combined with the skewed frequency distribution of codes in available datasets, presents substantial challenges. Clinical texts themselves add complexity due to their long, unstructured nature, frequent use of ambiguous abbreviations, and domain-specific vocabulary [16]. Each clinical document must be assigned multiple ICD codes reflecting all the diagnoses it contains. As a result, much of the current research frames automated coding as a multi-label classification task [3, 4, 9, 10, 13, 26, 33, 35, 36]. While the best-performing models are indeed multi-label classifiers [11], they are heavily reliant on widely used electronic health record (EHR) datasets such as MIMIC-III [15], MIMIC-IV [14] and CodiEsp [25]. Consequently, any quality issues in these datasets, such as misannotations or missing codes, are likely to be learned and propagated by the models, ultimately affecting prediction quality.

Our work focuses on identifying two types of issues in these datasets: **undercoding** and **annotation errors**. Given that datasets differ across institutions, both in terms of clinical text style and annotation strategy [17], we observe corresponding variations in

the types of errors they contain. These differences also affect the performance of state-of-the-art (SoTA) deep learning models. In this study, we investigate such issues in two widely used datasets: MDACE [3] and CodiEsp [25].

We design a three-stage pipeline for code prediction: an LLM-based evidence extractor, a multiclass classifier which is trained on the silver evidences obtained from MIMIC-IV [14] dataset, and an LLM-based verifier to produce the final set of codes. These codes are then analyzed for undercoding and annotation errors using two complementary approaches: an ensemble-based setup and a manual review process as shown in Figure 1. We observe noticeable performance improvements when re-evaluating existing best-performing deep learning models on the corrected datasets. Our major contributions are:

- Identification and quantification of undercoding and annotation errors in two widely used clinical coding datasets.
- Demonstration of performance improvements in existing models when evaluated on corrected datasets.
- Proposal of a promising, interpretable clinical coding pipeline that synergises large and small language models to provide valuable diagnostic insights without relying on large annotated datasets.

2 CODE PREDICTION PIPELINE

We found that the most effective way to identify undercoding and annotation errors was to build a code prediction pipeline that offers transparent explanations at each step, clarifying why a code is included or omitted. The pipeline starts with preprocessing clinical texts, followed by mapping extracted evidence spans to ICD-10 codes using a multiclass classifier. Finally, an LLM-based verifier reviews the full text to filter out any inappropriate codes.

2.1 Clinical Text Preprocessing

We used a variety of preprocessing strategies depending on the nature of the clinical text. The MDACE dataset, based on MIMIC-III [15], contains longer, messier, but more structurally consistent notes. As prior studies have noted, only a few sections of clinical texts are information-rich [38]. In this preprocessing phase, we extracted the most informative sections from the MDACE dataset: *Chief Complaint, History of Present Illness, Past Medical History, Family History, Social History,* and *Discharge Diagnosis.*

In several instances, a single sentence may be responsible for multiple codes, especially when different conditions are listed together. To isolate distinct clinical concepts, we split these sentences into smaller phrases using punctuation, such as commas (,), colons (:), and semicolons (;), as well as conjunctions like "and". These phrases were then further preprocessed to remove unnecessary punctuation and de-identified content. Each word from the phrases is then passed through a lookup table to expand abbreviations.

Upon evaluating the recall of extracted evidence against MDACE annotations, we found that relevant information was also derived from additional sections, such as *Medications, Brief Hospital Course,* and descriptions of the patient's *critical care status.* Restricting preprocessing to only the major sections proved insufficient. Therefore, we extended the coverage to include these additional sections. However, many of the extracted phrases contained non-clinical or irrelevant details. To refine the content, we passed all phrases through an evidence extraction module, which is designed to retain only clinically meaningful and codeable information (Section 2.2).

In contrast, the CodiEsp dataset is structurally simpler. Each line in a clinical text in CodiEsp typically corresponds to a single sentence, and these sentences are often densely packed with clinical information. As a result, the dataset required less preprocessing compared to MDACE. Nevertheless, we still applied phrase-splitting using punctuation- and conjunction-based rules. Given the smaller number of phrases and the sentence-level structure, we chose to omit the evidence extraction step for the experiments on CodiEsp. However, a notable issue with CodiEsp is that the original clinical texts are in Spanish and the qualities of the English translations provided in the CodiEsp dataset were very poor, often omitting critical clinical details present in the original Spanish texts. In our analysis, we re-translated them using Google Translate.

2.2 Evidence Extraction

The main purpose of the evidence extraction module is to include only the phrases that are clinically relevant and potentially codable. Inspired by [12] and [37], evidence extraction was accomplished using LLMs. Figure 2 shows the prompt used for evidence extraction. The prompt was initially designed in a simple way to include *any medical condition, symptom, diagnosis, finding, or infection resulting from a procedure* (see the "Include sentences if it contain" section in Figure 2). However, during tuning the prompt, we observed that since the LLM primarily focused on medical-condition-related aspects, it sometimes extracted content that was clinical in nature but not relevant for coding, such as lab results, negative diagnoses, or general narrative text. At the same time, it occasionally missed important coded details, such as causes of the condition, the patient's social history, or critical care information. Therefore, we iteratively enriched the prompt by introducing some clinically meaningful "exclusion rules" (see the "Exclude sentences if it contains" section in Figure 2). Initial experiments also showed that few-shot approach worked best for evidence extraction, particularly in guiding what to include and exclude. Ultimately, the prompt in Figure 2 proved to yield the best performance. While the figure displays only one example, our actual prompt used 2–3 examples for each clause.

2.3 Code Prediction

Codes are then predicted based on the extracted evidence phrases. To accomplish this, a multiclass classifier capable of assigning a code to a given evidence phrase is required. Due to the fact that a suitable dataset for training such a classifier does not exist, we constructed a custom dataset consisting of evidence phrases and their corresponding codes. Specifically, we built a silver-labeled evidence dataset from the MIMIC-IV corpus [14], where each entry contains a phrase and its associated ICD-10 code.

2.3.1 Silver Evidence Dataset Creation. The dataset creation process for training our multiclass classifier is illustrated in part (B) of Figure 1. We first preprocess the clinical texts from MIMIC-IV using the same procedure described in Section 2.1. In parallel, we collect the "gold-standard" human-annotated ICD codes and their corresponding descriptions for each clinical note. To identify relevant evidence phrases, we use the MiniLM model from

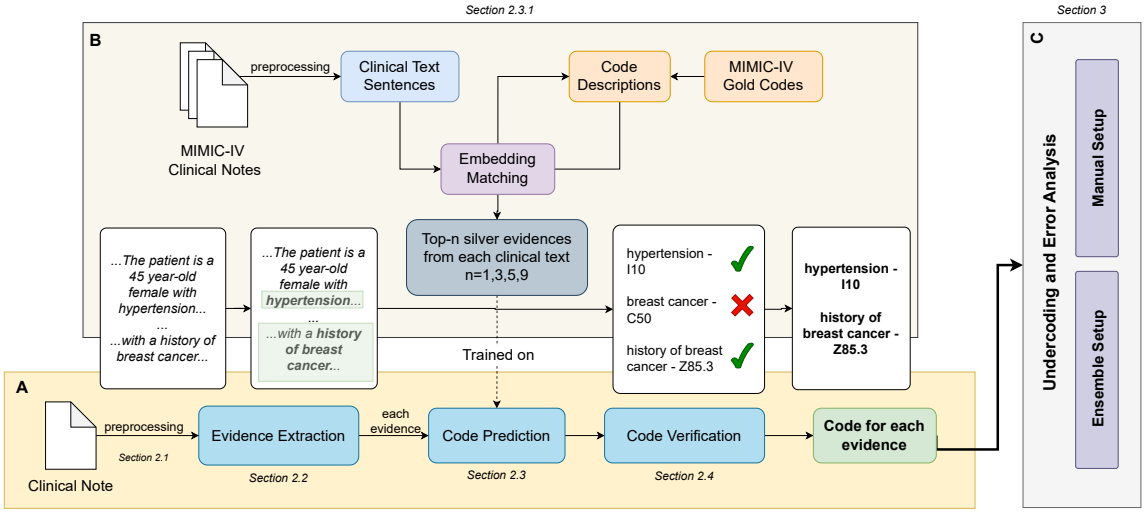

Figure 1: Methodology Diagram: Part (A) of the diagram describes the initial pipeline for code prediction. Part (B) is the silver evidence creation process, which is used to train the code predictor. Part (C) shows the undercoding and error analysis process which is done through ensemble analysis and manual analysis.

Prompt 1: Evidence Extraction Prompt

System Prompt

Identify **all candidate sentences** that contain symptoms, problems, causes, diagnosis, or medical conditions from the given list. Treat **each candidate sentence** as a distinct and separate entity.

Include sentences if it contains
- Any medical condition, symptom, diagnosis, finding, or infection resulting from a procedure.
- Explicit mention of a diagnosis, symptom, or medical condition along with a medication. *Example: "patient given clonazepam 5mg insomnia"*
- Causes/contributing factors of diagnosis. *Example: "fell from stairs",*
- Dependence on any external medical device. *Example: wheelchair, pacemaker*
- Critical patient care information. *Example: 'do not resuscitate'*

Exclude sentences if it contains
- Laboratory results, vitals, chemical values, or test results without any associated diagnosis or condition. *Example: "glucose 337 urea n 8 creat 0.7 sodium 137"*
- Medications administered during treatment unless explicitly tied to a diagnosis or condition. *Example: "omeprazole 20 mg po daily"*
- General narrative text without clear medical relevance. *Example: "patient is feeling better today.",*
- Sentences that you do not understand or that are purely numerical. *Example: "h h 11.3 34.7"*

Strict Instructions:
- Do not **modify, combine, or alter** the candidate sentences. Extract them **exactly as they appear** in the text.
- The response should be a **bulleted list** of **unique and relevant candidate sentences** without any explanation or commentary.
- **List as many relevant candidate sentences as possible.**

User Prompt

Candidate Evidences: {candidate_sentences}

Figure 2: Prompt for Evidence Extraction

Sentence-BERT [28] to compute embeddings and match each code description to sentences within the corresponding clinical text. For every code description, we retrieve the top 1, top 3, top 5, and top 9 sentence matches. The top match for each code in each clinical note is then selected as a silver-labeled evidence instance for our dataset. We did not use our LLM-based evidence extraction method for silver evidence creation due to the scale of this phase. Given that MIMIC-IV contains over 100,000 long clinical texts, applying LLMs would have been computationally and financially impractical.

We needed to assess whether the extracted evidence was meaningful enough for training a multiclass classifier. As a verification step, we trained PLM-ICD on top-n evidence sets derived from MIMIC-IV and observed a relative improvement of up to 20% over the baseline. A detailed breakdown is provided in Appendix A.

We also evaluated this method against human-annotated evidence spans in the MDACE [3] dataset. For each code in MDACE, we applied our silver-evidence extraction method and manually assessed the resulting matches. We found an 85% match rate between the extracted and annotated evidence. Most mismatches occurred because our method identified a piece of text that better aligned with the code description than the original annotated evidence.

For example, code I130, which corresponds to *Hypertensive heart and chronic kidney disease with heart failure and stage 1 through stage 4 chronic kidney disease, or unspecified chronic kidney disease.* MDACE annotators had labeled *Hypertension* as the supporting evidence. However, our method ranked *chronic systolic heart failure* and *chronic renal insufficiency baseline creat 1.5* as better matches, placing *Hypertension* as the third-best match. The combination of *Hypertension* and *chronic systolic heart failure* arguably provides the most accurate clinical context for this code. But, since our method extracts and evaluates individual phrases in isolation, it cannot capture such combinations. So, we conservatively classified this outcome as an inaccurate match.

These two verification methods confirm the validity of the obtained silver evidences and demonstrate that it is suitable for training a multiclass classifier.

2.3.2 Training Code Classifier. The multiclass classifier is trained on the extracted silver evidence, with each ICD code treated as a distinct class and its corresponding silver evidence serving as training examples. Prior to training, we perform some preprocessing. Since certain evidence phrases appear under multiple codes, we resolve these conflicts by selecting the code with the best-matching description based on embedding similarity. For example, while *chronic systolic heart failure* may appear as evidence for *I130* (as shown in the example above), it is more accurately aligned with *I5022 - Chronic systolic (congestive) heart failure*. In such cases, we use MiniLM-based embedding matching to assign the phrase to the most appropriate code.

We fine-tune the ClinicalBERT model [1] to train our multiclass classifier. Codes with fewer than five silver evidence examples are excluded from training to ensure meaningful representation. In total, we obtained silver evidence for 6,311 unique codes. This results in the exclusion of a large number of infrequent codes. However, many of these had only a single instance in the training set and none in the test set, making it impossible to evaluate performance reliably. Including such codes led to unstable training and misleading metrics, as the model either overfit to rare examples or failed to learn any meaningful pattern. By focusing on sufficiently represented codes, we ensure better generalization and more robust classifier performance.

2.4 Code Verification

The codes predicted by the classifier based on the extracted evidence are then passed through an LLM-based verifier. This verification step, inspired by a few recent attempts [21], aims to ensure that any code which should not have been assigned is filtered out. For example, it helps eliminate false positives arising from negative statements that may have been incorrectly split (e.g., *"no fever, headache or nausea"* being parsed as *"no fever"* and *"headache or nausea"*) or from symptoms that are already covered by a diagnosis.

We provide the original clinical text along with the predicted codes as input to the verifier. We also experimented with including the extracted evidence in the input, but this made the input sequence too long and resulted in worse performance compared to using just the clinical text and predicted codes. Multiple prompt variations and input formats were tested during this phase. The simplest prompt, asking the model *to extract supporting evidence if any existed for a given code*, achieved the highest recall but suffered from low precision. In contrast, a more refined prompt that instructed the model to remove redundant codes or those already implied by other diagnoses yielded slightly lower recall but significantly higher precision. The final version of the prompt used for verification is shown in Figure 3.

3 UNDERCODING AND ERROR DISCOVERY

We define **undercoding** as any code that should have been annotated in the clinical text but was not. We use the term **error** to refer to any other form of annotation issue observed in the datasets. We further classify errors into six specific categories, which are detailed in the results section in Table 1.

Prompt 2: Code Verification Prompt

System Prompt

You are a medical coding assistant. **Follow the output format strictly; any deviation will be rejected.** You are given **a clinical text** and a list of **predicted clinical codes**, each formatted as:

code, description

For **each clinical code,** decide

- **Retain** the code if it is clinically valid and meaningful in the context of the given clinical text.
- **Discard** the code if it is:
  – A routine symptom or expected sign of a broader condition
  – Not documented
  – Redundant or implied by the main diagnosis without requiring separate coding

Always extract supporting evidence from the text. If none, write: Evidence: unavailable

Strict Output Instructions

For **each input code**, your output must follow exactly one of the following formats:

- **If discarded**
  Discarded - [reason]
- **If retained**
  Retained
  Original code kept - [justification]
  Evidence: [quote from clinical note or "unavailable"

User Prompt

Possible Codes: [codes, descriptions]
Clinical Text: [clinical text]

Figure 3: Prompt for Code Verification

The codes produced by our prediction pipeline are evaluated using two independent setups to identify undercoding and annotation errors: an ensemble-based setup and a manual evaluation setup. In the *ensemble setup* (detailed in Section 3.1), we aggregate the outputs from various combinations of LLMs acting as extractors and verifiers. Probable errors are flagged using a strict two-thirds majority rule, as described in Section 3.1. In the *manual evaluation setup* (detailed in Section 3.2), we use the responses from the three best-performing LLMs on a curated sample. Human annotators then assess whether a predicted code represents an instance of undercoding, and whether codes from the original dataset are likely annotation errors. The details of this annotation process are provided in Section 3.2.

3.1 Ensemble Evaluation Setup

Let $\mathcal{E} = \{E_1, E_2, \ldots, E_m\}$ denote the set of LLM-based evidence extractors, and $\mathcal{V} = \{V_1, V_2, \ldots, V_n\}$ the set of LLM-based verifiers. Each extractor-verifier pair $M_{ij} = E_i V_j$, where $1 \leq i \leq m$ and $1 \leq j \leq n$, defines a unique model combination. Let $\mathcal{M} = \{M_{ij}\}$ represent the complete set of extractor-verifier models, with cardinality $|\mathcal{M}| = m \times n = N$.

Let $\mathcal{C} = \{t_1, t_2, \ldots, t_T\}$ be the set of clinical texts. For each clinical text $t \in \mathcal{C}$, we denote:

- $K_t \subseteq \mathcal{Y}$: the gold-standard set of ICD-10 codes (i.e., assigned by professional coders), where $\mathcal{Y}$ is the universe of possible codes.

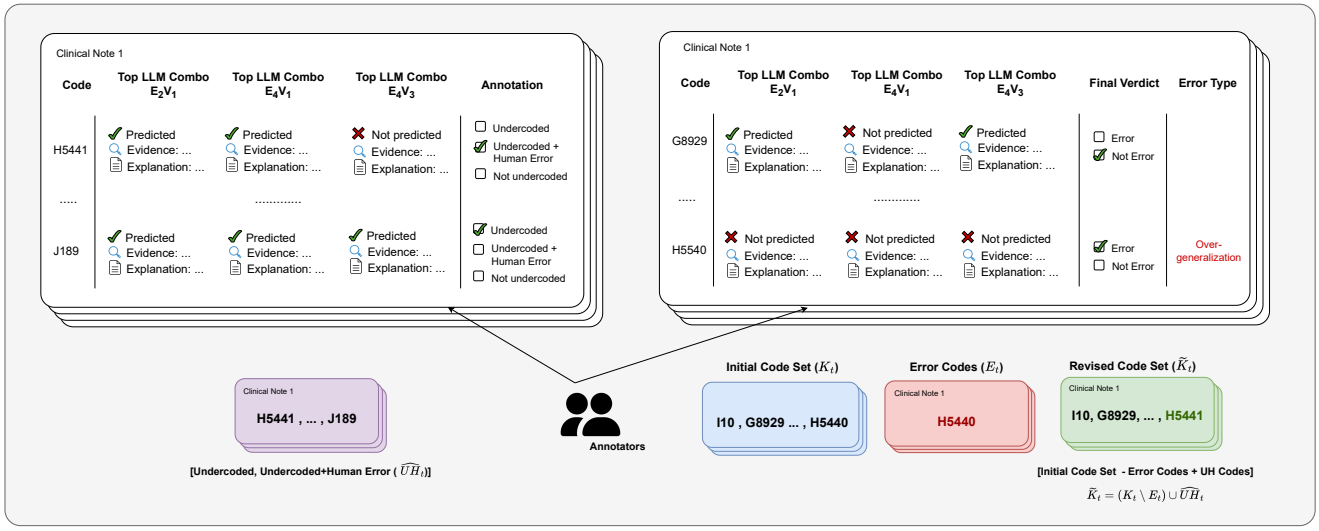

Figure 4: Manual Setup of Undercoding and Error Analysis

- $P_{ij}(t) \subseteq \mathcal{Y}$: the set of ICD-10 codes predicted by model $M_{ij} \in \mathcal{M}$ for text t.

We define the vote count for a code $y \in \mathcal{Y}$ as the number of model combinations that predict y for a given text t:

$$\text{votes}_t(y) = \sum_{M_{ij} \in \mathcal{M}} \mathbf{1}\left[y \in P_{ij}(t)\right] \tag{1}$$

A code is included in the ensemble prediction set $\hat{K}_t$ if it is predicted by at least two-thirds majority of the models:

$$\hat{K}_t = \left\{ y \in \mathcal{Y} \;\middle|\; \text{votes}_t(y) > \frac{2N}{3} \right\} \tag{2}$$

Based on this, we define the following evaluation sets:

- **True Positives**: $TP_t = \hat{K}_t \cap K_t$
- **Undercoded Codes**: $U_t = \hat{K}_t \setminus K_t$

We identify the initial set of **probable error codes** as those present in the gold-standard annotation but missing from the ensemble prediction: $Er_t = K_t \setminus \hat{K}_t$. It is refined through two filters:

- **Classifier Misses**: Codes that could not be predicted by the multiclass classifier from the extracted evidence.
- **Rare Codes**: Codes that appear fewer than five times in MIMIC-IV and were excluded during classifier training.

By removing these two categories from Er_t, we obtain the final set of probable error codes for manual evaluation and further analysis (Section 3.2), which are the codes likely to have been mistakenly assigned by human annotators, rather than missed by our predictor.

3.2 Manual Evaluation Setup

We evaluate all model combinations in $\mathcal{M}$ on the MDACE dataset and select the three best-performing combinations for manual evaluation. While we acknowledge that this introduces some circularity, using the same dataset to both identify flaws and select the top-performing LLMs, we needed a practical basis for narrowing down the combinations, as manually annotating outputs from all models was not feasible. We also recognize a potential source of bias in this process: by limiting manual evaluation to only three LLM combinations, we may overlook codes predicted by other models. To mitigate this, we include the ensemble-based method in parallel, allowing the two approaches to complement each other.

Each note is reviewed (re-annotated) based on the evidence provided by the LLMs, along with the original clinical text and the reasoning behind why certain LLMs accepted or rejected specific codes to aid reviewers' decision-making. If any of the top-performing LLMs predict a code, it is included for annotation.

For undercoding analysis, each predicted code is assigned to one of the following three annotation categories:

- **Undercoded**: The code was not present in the original annotation but should have been, based on the available evidence.
- **Undercoded + Human Error**: The code is undercoded and also serves as a correct replacement for one of the mistakenly assigned codes in the original annotation.
- **Not Undercoded**: The evidence is insufficient to justify the addition of the predicted code.

For error analysis, each original code is labeled as either:

- **Error**: The code is incorrect based on the evidence and should not have been included. If a code is marked as an error, it is further categorized into one of six specific error types: *Overgeneralization, Clinical Detail Mismatch, Contradictory Codes, History as Active, False Positive* and *No Evidence* (detailed in Section 5.3).
- **Not Error**: The evidence suffices to support the assignment of the original code.

The errors identified through this process are also corrected, as detailed in Section 3.3 and illustrated in Figure 4.

3.3 Correcting Errors

As described in Section 3.2, we performed two kinds of annotation for each clinical text t,

- Error analysis identified a subset of codes $E_t \subseteq K_t$ that were incorrectly assigned.
- Undercoding analysis yielded a set of missing codes $\widehat{UH}_t \subseteq \mathcal{Y} \setminus K_t$ that serve as correct replacements for the erroneous codes in E_t (Undercoding + Human Error).

The revised gold-standard code set for each clinical text t, where identified error codes are replaced with their appropriate undercoded alternatives, is defined as:

$$\widetilde{K}_t = (K_t \setminus E_t) \cup \widehat{UH}_t \tag{3}$$

Here, we consider only those undercoded codes that serve as correct replacements for identified errors. This ensures that when models are evaluated against the revised sets $\widetilde{K}_t$, their true performance is more accurately reflected—often revealing improvements that were previously underreported.

4 EXPERIMENTAL SETUP

LLMs Used: We experimented with five LLMs for evidence extraction: **GPT-4.1-mini**, **GPT-4.1**, **Deepseek-R1**, **Gemini-1.5-pro**, and **Gemini-2.0-flash**. For verification, we used the latter four models, as we found that GPT-4.1-mini performed poorly in predicting ICD-10 codes. This resulted in a total of 20 extractor-verifier combinations (5 extractors × 4 verifiers). As recommended by the *PhysioNet* guidelines, we used *Azure* to access GPT and Deepseek models, and *Google Cloud Platform* to access the Gemini models.

Datasets: We focus on ICD-10 diagnosis codes in our experiments and use two datasets: **MDACE** and **CodiEsp**. Both datasets were chosen for analyses because they provide human-annotated evidence spans for assigned ICD-10 codes.

MDACE is a curated subset of the MIMIC-III dataset, containing human-annotated evidence for each code in selected clinical texts. It includes 302 "Discharge Summary" notes from in-patient records. After excluding addendums and the validation set reserved for baseline comparisons, we include 239 clinical texts in our experiments. A larger-scale experiment involving multiple LLMs was not feasible due to associated computational and financial costs. We chose MDACE over a random subset of MIMIC-III or MIMIC-IV because it includes annotated evidence, allowing us to perform verification in the first phase using the same dataset. Our second dataset is CodiEsp. We conduct our experiments using its test set, which contains 250 clinical texts.

Evaluation Setup: For the ensemble setup, we use all 239 clinical texts from MDACE and 250 from the CodiEsp test set. For the manual setup, we randomly select 100 clinical notes from the MDACE dataset and use all 250 notes from the CodiEsp test set for human evaluation. Due to the length and complexity of MDACE notes, we limited the manual review to 100 texts, whereas the shorter length of CodiEsp notes allowed us to include the entire test set.

Baseline Models: We use four baseline models: **CAML**[26], **MultiResCNN**[19], **LAAT**[33] and **PLM-ICD**[9]. For MDACE, the baselines were trained on the MIMIC-IV dataset, given the similarity in text structure between MIMIC-III and MIMIC-IV. Initially, we also tested these MIMIC-IV-trained models on the CodiEsp dataset. However, they showed very poor performance due to significant differences in text structure and possibly a domain mismatch.

To address this, we created a hybrid training set by combining parts of the CodiEsp training set with clinical notes from MIMIC-IV that predominantly contain codes found in the CodiEsp dataset. This approach helped minimize the introduction of additional code complexity. While this setup also introduces some circularity, it allows us to study how the erroneous nature of CodiEsp annotations affects model performance.

5 RESULTS

5.1 Code Prediction

For evidence extraction, we use recall to measure how many MDACE-annotated evidences are captured by the LLM. Precision is not considered, as LLMs often extract relevant but unannotated text, leading to unfairly low precision scores. We obtained recall of more than **95%** for all the extractors, with Deepseek-R1 giving the best result. The extracted evidence is then passed to the code classifier, followed by the code verifier. The classifier achieved an accuracy of **82.47%** when evaluated on MDACE-labeled evidence. Interestingly, the codes with the lowest classification accuracy were also among the most error-prone codes identified in Section 5.3, so the real accuracy of the classifier is expected to be much higher, justifying its suitability as code predictor. Finally, the predicted codes are verified using an LLM-based verifier. The complete set of predicted codes from all extractor-verifier combinations for both MDACE and CodiEsp is provided in Appendix B.

We selected the best-performing model combinations based on F1 scores, as our evaluation considers both undercoding and annotation errors. GPT-4.1 consistently achieved the highest F1 scores as a verifier across combinations. However, to avoid bias from relying solely on one model, we selected two top-performing combinations with GPT-4.1 as verifier and one best-performing combination where GPT was not used as the verifier. We obtained the highest recall when Deepseek-R1 served as the verifier, while precision was highest when Gemini-2.0-flash was used as the extractor. Our $(Extractor, Verifier)$ combinations are: (GPT-4.1, GPT-4.1), (Gemini-2.0-Flash, GPT-4.1) and (Gemini-2.0-Flash, Gemini-1.5-Pro).

5.2 Undercoding Findings

Using the ensemble method, we found the undercoding rate to be **71.63%**. We also observed that the most frequently occurring codes tend to be both undercoded and error-prone. In particular, Chapter Z codes (Chapter Z titled "Factors Influencing Health Status and Contact with Health Services") were among the most commonly undercoded . This is partly due to their high frequency in the dataset and partly because Chapter Z codes represent factors influencing health status and encounters with health services. Since they are not direct diagnoses, they are often overlooked during coding.

In the 100 MDACE clinical notes manually evaluated, the (potential) undercoding rate was as high as **76.34%**, closely aligning with the ensemble-based estimate. To enable a fair comparison, we filtered the ensemble predictions to include only the same set of notes used in the manual evaluation. Both methods revealed similar trends in the most frequently undercoded codes. As shown in

Table 1: Error Categories, Examples, and Proportions

Category	Definition	Example	MDACE(%)	CodiEsp(%)
Overgeneralization	Specific details missed; broader code used	*Right eye blindness* coded as *blindness, one eye, unspecified*	**39.8**	**37.9**
Detail Mismatch	Code misses key modifiers	*Coronary artery disease(CAD) with chest pain* coded as *CAD without chest pain*	**35.9**	7.5
Contradictory	Conflicting codes used together	*Infectious* and *noninfectious gastroenteritis* coded for the same text	11.7	15.1
History as Active	Past conditions coded as current.	*History of breast cancer* coded as active.	6.8	2.5
False Positive	Code assigned despite negative results	*Urinary tract infection* coded despite negative findings	3.9	**32.4**
No Evidence	No clinical basis in text for the given code	*Hyperkalemia* code with only *hyperlipidemia* as evidence	1.9	4.6

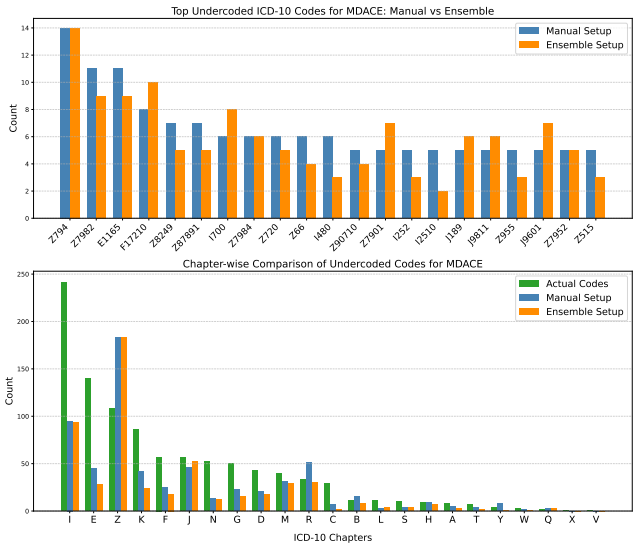

Figure 5: MDACE Undercoding: Top: ensemble and manual methods identify similar undercoded codes. Bottom: undercoding is more common in high-frequency chapters.

Figure 5, there is a clear relationship between the actual frequency of a code and its likelihood of being undercoded, in both manual and ensemble analyses. This shows that ensemble analysis can be more or less reliable in detecting undercoding.

On the other hand, undercoding in CodiEsp was minimal, with an average rate of only **5.3%**, possibly because of the fact that the texts in CodiEsp (short death reports) are often densely coded. The limited undercoding observed was mostly concentrated in Z-codes and followed a similar frequency-driven pattern as seen in MDACE. Given its low incidence, we did not conduct manual annotation of undercoded codes for the CodiEsp dataset.

5.3 Error Detection Findings

The MDACE dataset contained 1,004 codes across 100 clinical notes. When we compared the predicted codes against the original annotations, we identified 103 errors, resulting in an overall error rate of **10.3%**. Among these, our pipeline successfully generated 56 correct replacements, which were subsequently incorporated into the revised gold-standard dataset. The identified errors were further categorized into six distinct types, as detailed in Table 1.

In CodiEsp, the error rate was even higher than in MDACE. Using ensemble setup, we found an overall error rate of 22.26%. Among the 250 manually reviewed clinical notes, the original annotations included a total of 2,841 codes. Comparison with the predicted codes revealed 670 error cases, yielding an error rate of **23.58%**.

We observed that MDACE predominantly exhibited two types of errors: overgeneralization and clinical detail mismatch. These cases involved the assignment of incorrect codes where more specific or clinically accurate alternatives existed, often identifiable during undercoding analysis. Instances of false positives or codes lacking supporting evidence were rare in MDACE, likely due to the presence of human-annotated evidence spans.

CodiEsp also displayed two primary types of errors: **overgeneralization** and **false positives**. Unlike MDACE, in CodiEsp we identified multiple clinical notes in which diagnoses explicitly ruled out (i.e., documented as negative findings) were nonetheless coded, violating official ICD-10 coding guidelines. This represents a clear case of **overcoding**.

Notably, because both datasets included human-annotated evidence spans, instances of "no evidence" errors were infrequent. While the original English version of the CodiEsp dataset contained a higher number of such errors, our translated version significantly reduced their occurrence.

Finally, we note that the error codes in both MDACE and CodiEsp followed the same distribution pattern as the undercoded codes, as illustrated in Figures 6 and 7. Manual and ensemble analyses produced comparable results, and the most frequently occurring codes and ICD chapters were also the most error-prone.

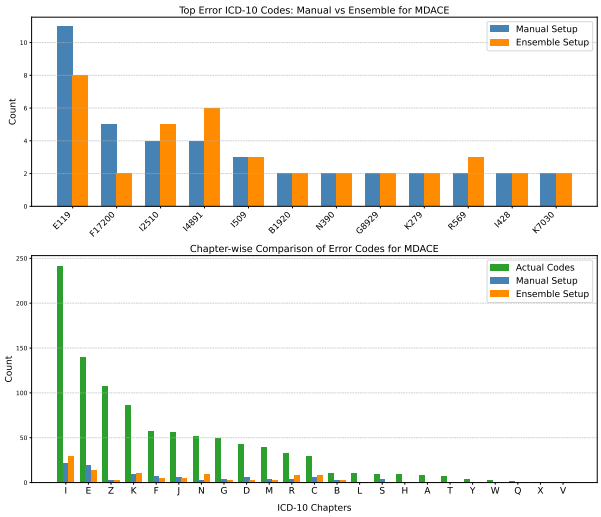

Figure 6: MDACE Error: Top: ensemble and manual methods identify similar error codes. Bottom: error is more common in high-frequency chapters.

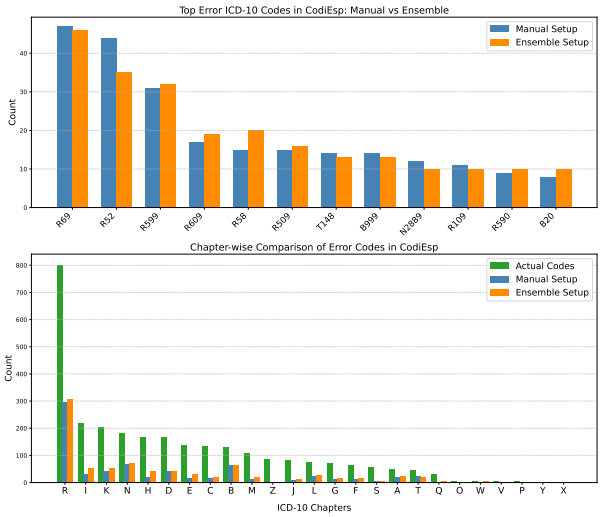

Figure 7: CodiEsp Error: Top: ensemble and manual methods identify similar error codes. Bottom: error is more common in high-frequency chapters.

Table 2: Impact of Correcting Errors on MDACE. MDACE+ indicates results using the corrected version of the MDACE

		Micro			Macro		
Model	**Prompt**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
CAML	MDACE	0.332	0.491	0.396	0.163	0.205	0.166
CAML	MDACE+	**0.337**	**0.521**	**0.409**	**0.172**	**0.221**	**0.178**
MultiresCNN	MDACE	0.689	0.354	0.468	0.192	0.191	0.183
MultiresCNN	MDACE+	**0.690**	**0.372**	**0.483**	**0.198**	**0.197**	**0.188**
LAAT	MDACE	0.464	0.413	0.437	0.229	0.235	0.219
LAAT	MDACE+	**0.473**	**0.441**	**0.457**	**0.248**	**0.258**	**0.240**
PLM-ICD	MDACE	0.635	0.603	0.618	0.341	0.353	0.335
PLM-ICD	MDACE+	**0.639**	**0.639**	**0.639**	**0.347**	**0.368**	**0.346**

Table 3: Impact of Correcting Errors on CodiEsp. CodiEsp+ indicates results using the corrected version of the CodiEsp

		Micro			Macro		
Model	**Prompt**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
CAML	CodiEsp	0.425	0.387	0.405	0.225	0.218	0.207
CAML	CodiEsp+	0.365	**0.409**	0.385	0.215	**0.224**	**0.209**
MultiresCNN	CodiEsp	0.347	0.233	0.278	0.120	0.123	0.111
MultiresCNN	CodiEsp+	0.308	**0.266**	**0.285**	0.118	**0.133**	**0.115**
LAAT	CodiEsp	0.383	0.493	0.431	0.216	0.254	0.218
LAAT	CodiEsp+	0.316	**0.504**	0.389	0.204	**0.257**	0.212
PLM-ICD	CodiEsp	0.535	0.400	0.457	0.251	0.239	0.227
PLM-ICD	CodiEsp+	0.479	**0.449**	**0.463**	0.240	**0.248**	**0.229**

5.4 Impact of Correcting Errors

We first run the baseline models on the original set of annotated gold codes, and then re-evaluate them on the revised gold codes based on Equation 3. The impact differs across the two datasets. For MDACE, we observe a clear improvement across all evaluation metrics when tested on the corrected dataset, as shown in Table 2. Since the model was trained on MIMIC-IV, it generalized well to MDACE. For CodiEsp, the improvement is less pronounced. This is partly because the training data combined both MIMIC-IV and CodiEsp, which introduced inconsistency. Removing error codes improved recall across all models, but precision dropped, reflecting noise introduced by incorrect labels in the training set (Table 3).

6 RELATED WORK

Early automated medical coding systems were based on rule-based techniques. More recently, the task has predominantly been framed as a multi-label classification problem, as each clinical document may be assigned multiple ICD codes. Alternative formulations have included multi-class classification, information retrieval, and sequence-to-sequence modeling. Initial classification efforts employed traditional machine learning methods [17, 24, 27]. Deep learning models, particularly those based on LSTMs [10, 36] and convolutional architectures [19, 22], soon became the dominant approach, often augmented with attention mechanisms [3, 4, 13, 26, 33, 35]. Among recent models, PLM-ICD [9] has achieved state-of-the-art results on both MIMIC-III and MIMIC-IV datasets, as shown in a comprehensive reproducibility study [5].

6.1 Evidence-based Clinical Coding

Beyond classification accuracy, a key concern is explainability. Several studies have focused on making clinical coding more interpretable by identifying the evidence behind each code. Mullenbach et al. [26] introduced a CNN-based model with attention to highlight relevant document segments. The CLEF eHealth 2020 CodiEsp track [25] extended this idea to Spanish clinical reports, introducing an evaluation track specifically for evidence-based coding. Duque et al.[4] proposed a keyphrase-based model for interpretable ICD-10

coding in Spanish. Recently, Lopez et al. [23] leveraged transformer-based architectures to improve explainability, while Cheng et al. [3] created the MDACE dataset with annotated evidence spans. Yang et al. [37] proposed LLM-Codex, a two-stage method that uses an LLM to generate evidence and an LSTM to verify it. Edin et al. [6] demonstrated that unsupervised methods can also achieve explainability levels comparable to supervised models.

6.2 LLMs in Clinical Coding

Several studies have tested LLMs for clinical coding tasks. Boyle et al. [2] implemented a tree-search approach over ICD code hierarchies using LLMs. Others have used LLMs to generate silver- and bronze-quality training data [20], paraphrase discharge summaries [7], and extract code evidence [37]. Recent benchmarking work [18, 31, 34] has assessed the performance of LLMs on standard datasets. While initial results show some promises, especially on rare codes, these studies agree that LLMs are not yet reliable enough for direct deployment in clinical coding systems.

6.3 Undercoding and Error Analysis

Few studies provide in-depth error analyses of clinical coding datasets, often with a narrow focus on undercoding. Searle et al. [30] identified widespread undercoding in MIMIC-III, particularly among frequently used codes. Remmer et al. [29] reported similar findings in Swedish discharge summaries. Jin et al. [13] developed methods to predict ICD codes from partially labeled data, addressing undercoding indirectly. Gan et al. [8] critiqued existing evaluation metrics for their misalignment with clinical realities, while Edin et al. [6] highlighted that longer notes do not affect model performance, contrary to previous assumptions. Despite these efforts, most prior work does not focus on error discovery or deep analysis of dataset flaws.

7 CONCLUDING REMARKS

The quality of clinical coding datasets is an important but often overlooked topic of consideration in research, just as much as the models and algorithms themselves. When machine learning algorithms are trained on subpar or flawed datasets, the results inevitably reflect those limitations. In our work, we have uncovered various kinds of flaws in commonly used datasets, including undercoding, erroneous coding, and overcoding, to name a few. Because of these errors, the true potential of the algorithms cannot be realized, and they are unlikely to be usable in real-life clinical settings.

We also advocate for evidence-based clinical coding through our work, as it holds great value in uncovering such errors. Our proposed transparent pipeline enables coders to clearly trace which evidence leads to which code, and whether that code is valid or not. This approach allowed us to pinpoint the causes of errors and address them systematically, as demonstrated by our experiments on MDACE and CodiEsp datasets. This method also shows promise for adaptation into practical code recommendation systems.

Our results further validate the use of LLM-based ensemble methods for code verification and detecting undercoding. Across all experimental setups, we observed strong alignment between ensemble outcomes and manual evaluations, confirming the value of ensemble approaches in supporting reliable, scalable coding verification, which could also serve as a powerful tool for clinical documentation auditing.

Beyond specific error identification, our findings underscore a deeper challenge: the lack of transparency and traceability in current benchmark datasets. To overcome this, we recommend that future clinical coding datasets include evidence-linked annotations, not only to facilitate verification but also to better prepare models for deployment in real-world clinical environments.

ACKNOWLEDGMENTS

The corresponding author of the paper is partially supported by the Royal Society of the UK (IESR1231175).

A SILVER EVIDENCES VALIDATION

The F1 scores of PLM-ICD trained on the top 1, 3, 5, and 9 evidence phrases extracted from the silver-labeled MIMIC-IV dataset are presented in Table **??**. The scores show a relative improvement of up to **20%** compared to the baseline. The results partially implied that (i) the our approach was able to extract high quality silver evidence and (ii) accurately extracting evidence is the key for the success of automated medical coding.

B CODE PREDICTION PIPELINE RESULTS

The evidence extraction phase for MDACE achieved a high recall of over 95% across all extractors, as shown in Table 6. Due to the shorter length of CodiEsp documents, evidence extraction was not applied to that dataset. The results of code verification for MDACE using the combined extractor-verifier setup and for CodiEsp using only verifiers are presented in Tables 7 and 4, respectively.

C GENERATIVE AI USE DISCLOSURE

The authors claim that generative AI has not been used in undertaking the research or writing the paper.

REFERENCES

[1] Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. *arXiv preprint arXiv:1904.03323* (2019).

[2] 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. *arXiv preprint arXiv:2310.06552* (2023).

[3] Hua Cheng, Rana Jafari, April Russell, Russell Klopfer, Edmond Lu, Benjamin Striner, and Matthew R Gormley. 2023. MDACE: MIMIC documents annotated with code evidence. *arXiv preprint arXiv:2307.03859* (2023).

[4] Jingcheng Du, Qingyu Chen, Yifan Peng, Yang Xiang, Cui Tao, and Zhiyong Lu. 2019. ML-Net: multi-label classification of biomedical texts with deep neural networks. *Journal of the American Medical Informatics Association* 26, 11 (2019), 1279–1285.

[5] 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 *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, Taipei, Taiwan. https://doi.org/10.1145/3539618.3591918

[6] Joakim Edin, Maria Maistro, Lars Maaløe, Lasse Borgholt, Jakob Drachmann Havtorn, and Tuukka Ruotsalo. 2024. An Unsupervised Approach to Achieve Supervised-Level Explainability in Healthcare Records. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 4869–4890. https://doi.org/10.18653/v1/2024.emnlp-main.280

[7] Matúš Falis, Aryo Pradipta Gema, Hang Dong, Luke Daines, Siddharth Basetti, Michael Holder, Rose S Penfold, Alexandra Birch, and Beatrice Alex. 2024. Can

Table 4: Performance of different verifier–extractor combinations on MDACE. Bold values indicate the combinations selected for manual annotation. Underlined values indicate the highest precision, recall, and micro F1 score.

Verifier →	GPT-4.1			Gemini-1.5-Pro			Gemini-2.0-Flash			Deepseek-R1		
Extractor ↓	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
GPT-4.1-mini	0.359	0.623	0.456	0.360	0.539	0.432	0.320	0.591	0.415	0.313	0.639	0.421
GPT-4.1	0.370	0.616	**0.462**	0.369	0.566	0.447	0.338	0.597	0.432	0.332	0.645	0.438
Gemini-1.5-Pro	0.361	0.626	0.458	0.350	0.570	0.434	0.331	0.600	0.427	0.329	0.645	0.435
Gemini-2.0-Flash	0.385	0.623	**0.476**	0.380	0.570	**0.456**	0.350	0.587	0.439	0.350	0.642	0.453
Deepseek-R1	0.355	0.633	0.455	0.349	0.566	0.432	0.326	0.615	0.426	0.313	0.647	0.422

Table 5: F1-score comparison for Top-N silver evidences using PLM-ICD on MIMIC-IV. The underlined value is the baseline (trained and tested on MIMIC-IV); bold values indicate improvements over the baseline.

Train ↓ Test →	Top 1	Top 3	Top 5	Top 9	MIMIC-IV
Top	**0.688**	**0.604**	0.543	0.497	0.427
Top 3	**0.611**	**0.725**	**0.697**	**0.637**	0.485
Top 5	0.591	**0.680**	**0.731**	**0.696**	0.513
Top 9	0.575	**0.616**	**0.699**	**0.723**	0.531
MIMIC-IV	0.225	0.534	0.541	0.532	**0.594***

Table 6: Evidence Extraction Recall on MDACE

gpt-4.1-mini	gpt-4.1	gem-1.5-pro	gem-2.0-flash	DS-R1
0.958	0.959	0.962	0.959	**0.966**

Table 7: Verifier results on CodiEsp. All results were used for manual annotation. Underlined values indicate the highest precision, recall, and micro F1 score.

Model	Precision	Recall	F1 Score
GPT-4.1	0.753	0.384	0.509
Gemini-1.5-Pro	0.664	0.311	0.423
Gemini-2.0-Flash	0.606	0.482	0.537
Deepseek-R1	0.579	0.380	0.459

GPT-3.5 generate and code discharge summaries? *Journal of the American Medical Informatics Association* 31, 10 (2024), 2284–2293.

[8] Yidong Gan, Maciej Rybinski, Ben Hachey, and Jonathan K Kummerfeld. 2024. Aligning AI Research with the Needs of Clinical Coding Workflows: Eight Recommendations Based on US Data Analysis and Critical Review. *arXiv preprint arXiv:2412.18043* (2024).

[9] 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*, Tristan Naumann, Steven Bethard, Kirk Roberts, and Anna Rumshisky (Eds.). Association for Computational Linguistics, Seattle, WA, 10–20. https://doi.org/10.18653/v1/2022.clinicalnlp-1.2

[10] Jinmiao Huang, Cesar Osorio, and Luke Wicent Sy. 2019. An empirical evaluation of deep learning for ICD-9 code assignment using MIMIC-III clinical notes. *Computer methods and programs in biomedicine* 177 (2019), 141–153.

[11] 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 Comput. Survey* 56, 12, Article 306 (Oct. 2024), 41 pages. https://doi.org/10.1145/3664615

[12] Xiaorui Jiang, Kulsoom Khan, Sumithra Thinakara Vasantha, and Sajjad Haider. 2025. Evidence Extraction for Automated Medical Coding: Preliminary Evaluation. In *Proceedings of the 2024 8th International Conference on Natural Language Processing and Information Retrieval (NLPIR'24)*. Association for Computing Machinery, New York, NY, USA, 18–23. https://doi.org/10.1145/3711542.3711580

[13] Yucheng Jin, Yun Xiong, Dan Shi, Yifei Lin, Lifang He, Yao Zhang, Joseph M Plasek, Li Zhou, David W Bates, and Chunlei Tang. 2023. Learning from undercoded clinical records for automated International Classification of Diseases (ICD) coding. *Journal of the American Medical Informatics Association* 30, 3 (2023), 438–446.

[14] Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. 2023. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific data* 10, 1 (2023), 1.

[15] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3, 1 (2016), 1–9.

[16] Rajvir Kaur, Jeewani Anupama Ginige, and Oliver Obst. 2023. AI-based ICD coding and classification approaches using discharge summaries: A systematic literature review. *Expert Systems with Applications* 213 (2023), 118997.

[17] Ramakanth Kavuluru, Anthony Rios, and Yuan Lu. 2015. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artificial intelligence in medicine* 65, 2 (2015), 155–166.

[18] Simon A Lee and Timothy Lindsey. 2024. Do Large Language Models understand Medical Codes? *arXiv preprint arXiv:2403.10822* (2024).

[19] Fei Li and Hong Yu. 2020. ICD coding from clinical text using multi-filter residual convolutional neural network. In *proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 8180–8187.

[20] Rumeng Li, Xun Wang, and Hong Yu. 2023. Two Directions for Clinical Data Generation with Large Language Models: Data-to-Label and Label-to-Data. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 7129–7143. https://doi.org/10.18653/v1/2023.findings-emnlp.474

[21] Rumeng Li, Xun Wang, and Hong Yu. 2024. Exploring LLM Multi-Agents for ICD Coding. arXiv:2406.15363 [cs.CL] https://arxiv.org/abs/2406.15363

[22] Chin Lin, Chia-Jung Hsu, Yu-Sheng Lou, Shih-Jen Yeh, Chia-Cheng Lee, Sui-Lung Su, and Hsiang-Cheng Chen. 2017. Artificial intelligence learning semantics via external resources for classifying diagnosis codes in discharge notes. *Journal of medical Internet research* 19, 11 (2017), e380.

[23] Guillermo López-García, José M Jerez, Nuria Ribelles, Emilio Alba, and Francisco J Veredas. 2023. Explainable clinical coding with in-domain adapted transformers. *Journal of Biomedical Informatics* 139 (2023), 104323.

[24] Ben J Marafino, Jason M Davies, Naomi S Bardach, Mitzi L Dean, and R Adams Dudley. 2014. N-gram support vector machines for scalable procedure and diagnosis classification, with applications to clinical free text data from the intensive care unit. *Journal of the American Medical Informatics Association* 21, 5 (2014), 871–875.

[25] Antonio Miranda-Escalada, Aitor Gonzalez-Agirre, Jordi Armengol-Estapé, and Martin Krallinger. 2020. Overview of Automatic Clinical Coding: Annotations, Guidelines, and Solutions for non-English Clinical Cases at CodiEsp Track of CLEF eHealth 2020. *CLEF (Working Notes)* 2020 (2020).

[26] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. *arXiv preprint arXiv:1802.05695* (2018).

[27] Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. 2014. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association* 21, 2 (2014), 231–237.

[28] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[29] Sonja Remmer, Anastasios Lamproudis, and Hercules Dalianis. 2021. Multi-label diagnosis classification of Swedish discharge summaries–ICD-10 code assignment using KB-BERT. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. 1158–1166.

[30] Thomas Searle, Zina Ibrahim, and Richard JB Dobson. 2020. Experimental evaluation and development of a silver-standard for the MIMIC-III clinical coding dataset. *arXiv preprint arXiv:2006.07332* (2020).

[31] 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 (2024), AIdbp2300040. https://doi.org/10.1056/AIdbp2300040 arXiv:https://ai.nejm.org/doi/pdf/10.1056/AIdbp2300040

[32] Michael Subotin and Anthony Davis. 2014. A System for Predicting ICD-10-PCS Codes from Electronic Health Records. In *Proceedings of BioNLP 2014*, Kevin Cohen, Dina Demner-Fushman, Sophia Ananiadou, and Jun-ichi Tsujii (Eds.). Association for Computational Linguistics, Baltimore, Maryland, 59–67. https://doi.org/10.3115/v1/W14-3409

[33] Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. 2020. A label attention model for ICD coding from clinical text. *arXiv preprint arXiv:2007.06351* (2020).

[34] Yichen Wang, Yuting Huang, Induja R Nimma, Songhan Pang, Maoyin Pang, Tao Cui, and Vivek Kumbhari. 2024. Validation of GPT-4 for clinical event classification: A comparative analysis with ICD codes and human reviewers. *Journal of Gastroenterology and Hepatology* (2024).

[35] Yuzhou Wu, Zhigang Chen, Xin Yao, Xuechen Chen, Zeren Zhou, and Jinkai Xue. 2022. JAN: Joint attention networks for automatic ICD coding. *IEEE journal of biomedical and health informatics* 26, 10 (2022), 5235–5246.

[36] Pengtao Xie and Eric Xing. 2018. A Neural Architecture for Automated ICD Coding. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Iryna Gurevych and Yusuke Miyao (Eds.). Association for Computational Linguistics, Melbourne, Australia, 1066–1076. https://doi.org/10.18653/v1/P18-1098

[37] Zhichao Yang, Sanjit Singh Batra, Joel Stremmel, and Eran Halperin. 2023. Surpassing GPT-4 medical coding with a two-stage approach. *arXiv preprint arXiv:2311.13735* (2023).

[38] Tong Zhou, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Kun Niu, Weifeng Chong, and Shengping Liu. 2021. Automatic ICD coding via interactive shared representation networks with self-distillation mechanism. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 5948–5957.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009