Do Large Language Models **Struggle** With Medical Coding?

Resource Paper: **Large Language Models Are Poor Medical Coders — Benchmarking of Medical Code Querying**

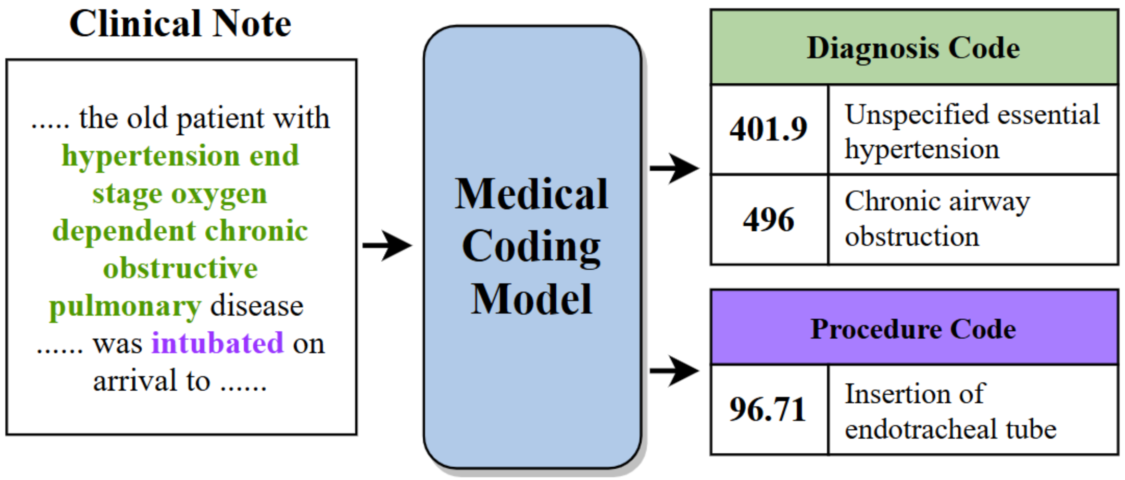
Ali Soroush, M.D., M.S. Benjamin S. Glicksberg, Ph.D. Eyal Zimlichman, M.D., M.Sc. Yiftach Barash, M.D., M.Sc. Robert Freeman, R.N., M.S.N., N.E.-B.C. Alexander W. Charney, M.D., Ph.D. Girish N. Nadkarni, M.D., M.P.H. Eyal Klang, M.D.

Romen Samuel Rodis Wabina, MSc
PhD candidate, Data Science for Healthcare and Clinical Informatics
Data Scientist, Department of Clinical Epidemiology and Biostatistics

Quick Overview

- Properly coded medical information is vital for decision-making, health surveillance, research, and reimbursement[1-2].

- Various coding systems:
  - International Classification of Diseases (ICD)
    - Clinical Modification (CM)
  - Current Procedural Terminology (CPT)



- Automated medical code assignment uses rule-based, machine learning (ML), or deep learning (DL)
  - See Shaoxiong et al (2024) for a unified review[3]

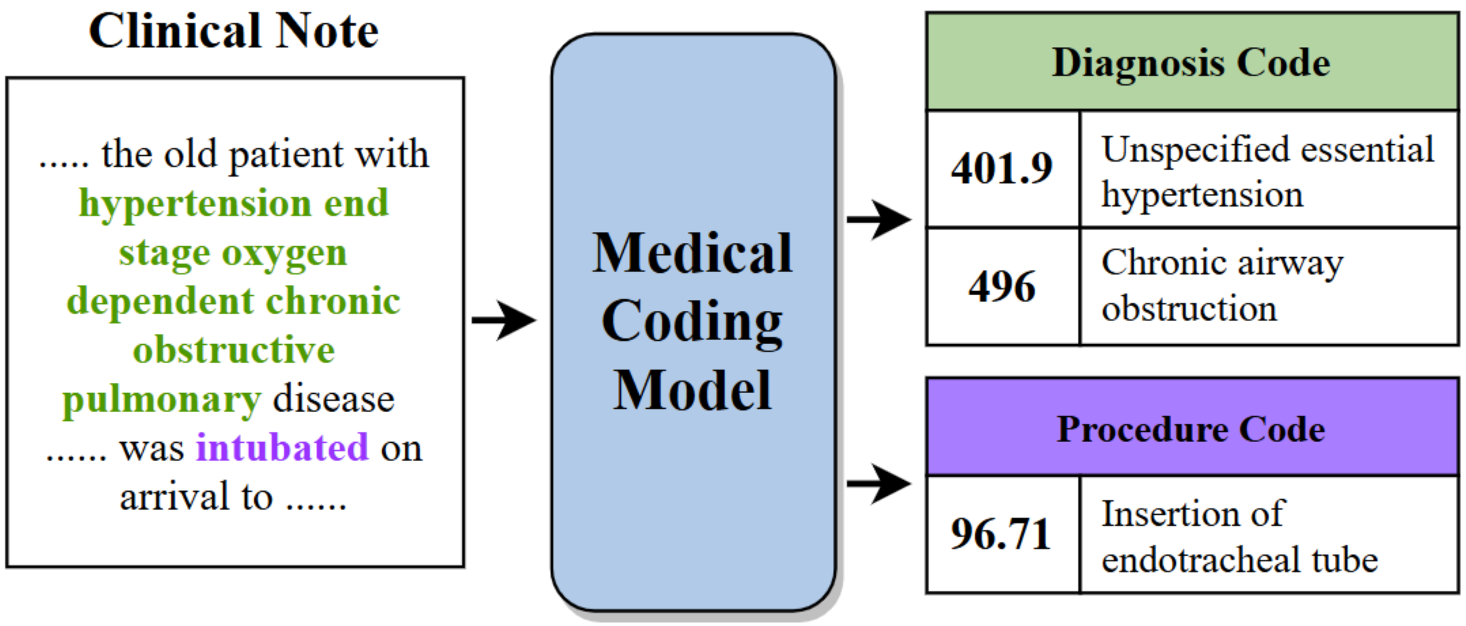
- Large language models (LLMs) have shown remarkable text processing and reasoning capabilities
  - Recent studies show that LLMs extract fewer correct medical codes[3].
    - LLMs are highly error-prone when mapping clinical codes[3-4].

[1] Park, J. K., Kim, K. S., Lee, T. Y... & Kim, C. B. (2000). The accuracy of ICD codes for cerebrovascular diseases in medical insurance claims. *Journal of Preventive Medicine and Public Health*, *33*(1), 76-82.
[2] Burks, K., Shields, J., Evans, J., Plumley, J., Gerlach, J., & Flesher, S. (2022). A systematic review of outpatient billing practices. *SAGE Open Medicine*, *10*, 20503121221099021.
[3] Ji, S., Li, X., Sun, W., Dong, H., Taalas, A., Zhang, Y., ... & Marttinen, P. (2024). A unified review of deep learning for automated medical coding. *ACM Computing Surveys*, *56*(12), 1-41.
[4] Simmons, A., Takkavatakarn, K., McDougal, M., Dilcher, B., Pincavitch, J., Meadows, L., ... & Sakhuja, A. (2024). Benchmarking Large Language Models for Extraction of International Classification of Diseases Codes from Clinical Documentation. *medRxiv*, 2024-04.

Objectives

- **Can popular LLMs (GPT-3.5, GPT-4, Gemini Pro, Llama2-70b Chat) reliably query medical billing codes from clinical text?**

- To quantify and benchmark the performance of GPT-3.5 Turbo, GPT-4, Gemini Pro, and Llama2-70b Chat in querying medical codes from clinical data.
  - Evaluate how well these models generate correct ICD-9-CM, ICD-10-CM, and CPT codes based on **exact match accuracy**.

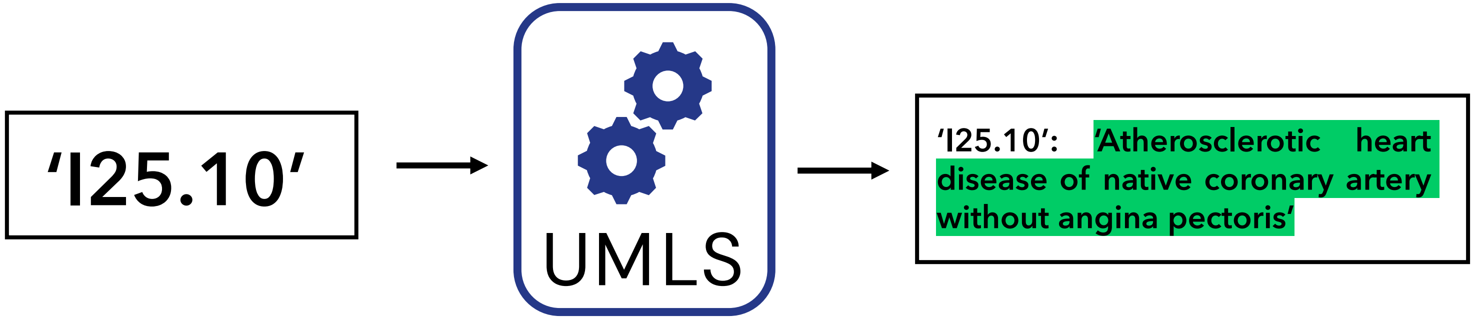


Code Datasets

- Extracted IC9-CM, ICD-10-CM, and CPT billing codes from Mount Sinai Data Warehouse[1]
  - Used a REST API to interface with the data warehouse, enabling efficient extraction of data

- Code Datasets
  - **ICD-9-CM**, **ICD-10-CM**, and **CPT**

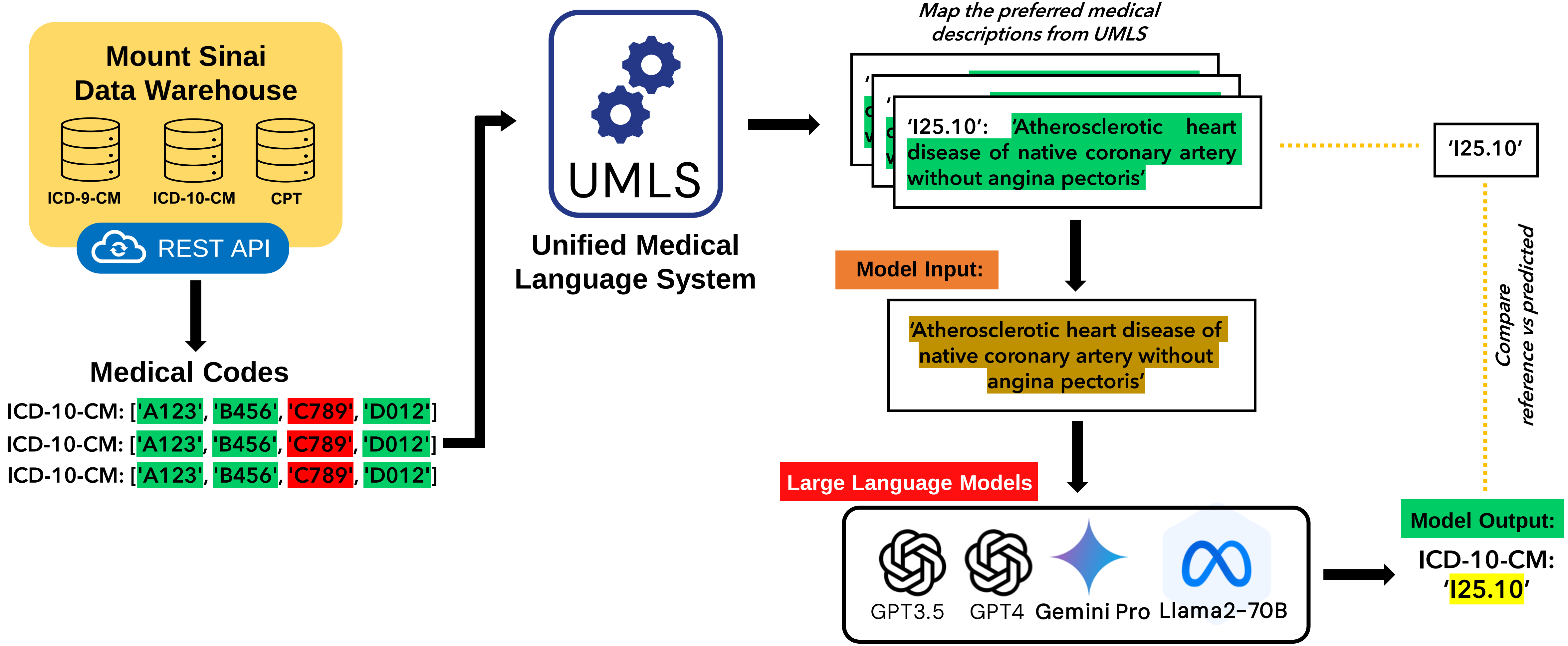
- The extracted data (medical codes) were mapped and standardized using the Unified Medical Language System (UMLS) Metathesaurus
  - UMLS – a comprehensive biomedical terminology resource
  - Used UMLS to obtain the preferred description for each code.



Unified Medical Language System

[1] Icahn School of Medicine at Mount Sinai. (2023). *Mount Sinai Data Warehouse (MSDW).* Research Roadmap. Retrieved February 19, 2025, from https://researchroadmap.mssm.edu/reference/systems/msdw/

Methodological Framework



LLM Code Generation

- Once the codes are harmonized through the UMLS, clinical descriptions are fed into the LLMs.

- Four LLMs were utilized
  - GPT3.5 Turbo (March 2023, June 2023, & November 2023 versions)
  - GPT4 (March 2023, June 2023, & November 2023 versions)
  - Gemini Pro
  - Llama2-70b Chat

- Primary task: **Generate the medical code when given the preferred code description**

- LangChain was used to standardize LLM API calls
- Tested temperatures of 0.2, 0.4, 0.6, 0.8, and 1.0
  - No meaningful difference in overall accuracy
  - Selected 0.2 as final temperature
- No LLMs were truncated
  - All models were set to 50 maximum output tokens

Prompt Development

- Selected a random sample of 100 codes from each coding system (ICD9, ICD10, and CPT).
- Tested various wordings and structures to reliably generate valid codes.
    - Iteratively refined the prompt until it consistently produced correctly formatted outputs without errors.
- The iterative process was qualitative
    - No specific number of development rounds mentioned.

'What is the most correct <code system> billing code for this description: <description>.

Only generate a single, VALID <code system> billing code. Do not explain. ALWAYS respond in the following format:

Code: <code system>: <sample code>'.

Performance Metrics

1. **Exact match**: compares each generated code with its corresponding reference code
   - Determine the percentage of pairs that are identical

     Reference: ['A123', 'B456', 'C789', 'D012', 'E345']

     Predicted: ['A123', 'B456', 'C788', 'D012', 'E346']

     Exact Match = 3/5 = 60%

2. **METEOR**: comparison through exact matches, stemming, synonyms, and word order

     Reference: ['diabetes mellitus without complication']

     Predicted: ['diabetes without complications']

   - Both text share the words 'diabetes' and 'without' (2 matches)
   - Predicted add 's' at the end of complications vs no 's' has the same root meaning
   - Range: 0.0 – 1.0 (higher, better)

3. **BERTScore**: computes cosine similarity using contextual embeddings rather than token matches
   - Captures contextual meaning and semantic relationship
   - Range: 0.0 – 1.0 (higher, better)

[1] **METEOR**: Metric for Evaluation of Translation with Explicit Ordering; [2] **BERT**: Bidirectional Encoder Representations from Transformers

Performance Evaluation – Exact Match

- GPT-4 had the highest exact match rates and Llama2-70b Chat scored the lowest
  - Both GPT-3.5 and GPT-4 demonstrated improved exact match performance with each successive model (March to November).

Model	ICD-9-CM	ICD-10-CM	CPT
GPT-3.5 Turbo (Mar)	26.6% (25.6–27.6%)	17.1% (16.5–17.7%)	28.4% (27.0–29.9%)
GPT-3.5 Turbo (June)	26.7% (25.7–27.7%)	17.8% (17.2–18.4%)	26.2% (24.7–27.6%)
GPT-3.5 Turbo (Nov)	28.9% (27.9–29.9%)	18.2% (17.6–18.8%)	31.9% (30.4–33.4%)
GPT-4 (Mar)	42.3% (41.2–43.4%)	27.5% (26.8–28.1%)	44.0% (42.4–45.6%)
GPT-4 (June)	44.1% (43.0–45.2%)	28.4% (27.7–29.1%)	42.6% (41.0–44.2%
GPT-4 (Nov)	45.9% (44.8–47.0%)	33.9% (33.2–34.6%)	49.8% (48.2–51.5%)
Gemini Pro	10.7% (10.0–11.4%)	4.8% (4.5–5.1%)	11.4% (10.3–12.4%)
Llama2-70b Chat	1.2% (1.0–1.5%)	1.5% (1.4–1.7%)	2.6% (2.1–3.1%)

- At the code system level, ICD-9-CM and CPT codes had more exact matches than ICD-10-CM, except for Llama2-70b Chat, which had the lowest match rate with ICD-9-CM.

Performance Evaluation – METEOR & BERTScore

- Using LLM-generated codes, they retrieved its medical description from UMLS
  - Compare the medical descriptions of LLM-generated codes and original code using METEOR and BERTScores

Model	METEOR			BERTScore		
	ICD-9-CM	ICD-10-CM	CPT	ICD-9-CM	ICD-10-CM	CPT
GPT-3.5 Turbo (Mar)	0.415 (0.406–0.424)	0.399 (0.393–0.407)	0.461 (0.448–0.474)	0.857 (0.855–0.860)	0.866 (0.864–0.868)	0.868 (0.864–0.871)
GPT-3.5 Turbo (June)	0.414 (0.405–0.422)	0.405 (0.398–0.412)	0.433 (0.421–0.446)	0.856 (0.854–0.859)	0.870 (0.868–0.871)	0.859 (0.855–0.863)
GPT-3.5 Turbo (Nov)	0.437 (0.428–0.445)	0.400 (0.394–0.406)	0.495 (0.485–0.507)	0.863 (0.861–0.866)	0.866 (0.864–0.868)	0.878 (0.874–0.882)
GPT-4 (Mar)	0.564 (0.555–0.573)	0.510 (0.504–0.516)	0.596 (0.583–0.609)	0.899 (0.896–0.901)	0.899 (0.897–0.900)	0.904 (0.901–0.908)
GPT-4 (June)	0.579 (0.569–0.588)	0.522 (0.516–0.528)	0.586 (0.573–0.599)	0.903 (0.901–0.906)	0.902 (0.901–0.904)	0.901 (0.897–0.904)
GPT-4 (Nov)	0.593 (0.585–0.602)	0.581 (0.575–0.587)	0.655 (0.642–0.667)	0.907 (0.904–0.909)	0.918 (0.917–0.920)	0.921 (0.918–0.925)
Gemini Pro	0.245 (0.240–0.250)	0.250 (0.245–0.254)	0.295 (0.284–0.306)	0.812 (0.809–0.814)	0.824 (0.822–0.826)	0.816 (0.813–0.820)
Llama2-70b Chat	0.100 (0.094–0.106)	0.129 (0.125–0.132)	0.182 (0.172–0.192)	0.749 (0.747–0.751)	0.774 (0.773–0.776)	0.770 (0.766–0.773)

- GPT-4 (Nov) achieved a METEOR score of 0.593 and a BERTScore of 0.907, indicating a very close match between the generated code description and the original.
- Gemini Pro and Llama2-70b Chat demonstrated substantially lower textual similarity scores—with METEOR scores roughly around 0.245 and 0.100, and BERTScores approximately 0.812 and 0.743 respectively

Code Generation: **Error Analysis**

Error Analysis Metrics:

- **Incorrect** code: LLM-generated code does not match the correct reference code.

  Reference: [I82.409'] (acute embolism and thrombosis)

  Predicted: ['I82.49']

- **Valid** code: LLM-generated code exists in the UMLS Metathesaurus, regardless of whether it exactly matches the reference code.

  Predicted: ['I82.409'] (valid as long as it appears in UMLS)

- **Fabricated** code: LLM-generated code that **does not exist in the UMLS** at all.

- **Code generation frequency**: Average number of codes the model outputs per prompt.
  - LLM outputs three codes, e.g., I82.409, I82.410, I82.411, indicating **over-generation**.

- **Matched length**: Average number of codes the model outputs per prompt.
  - LLM outputs three codes

- **Matched digits:** how many digits align with the correct code

Error Analysis: **ICD-9-CM**

- GPT-4 outperformed other models with the lowest incorrect code rate (53.9%) and the highest valid code percentage (97.1%).
  - Llama2-70b Chat performed poorly, with almost all generated codes being incorrect (98.8%), only 54.1% valid codes, and the highest fabricated code rate (45.9%).

Metric	GPT-3.5	**GPT-4**	Gemini Pro	**Llama2-70b Chat**
Incorrect codes, n (% of total)	5467 (71.0%)	4149 (53.9%)	6869 (89.2%)	7601 (98.8%)
Valid code, % (95% CI)	96.1% (95.6%–96.6%)	97.1% (96.6%–97.5%)	88.9% (88.1%–89.6%)	54.1% (53.0%–55.2%)
Fabricated code, % (95% CI)	3.9% (3.4%–4.4%)	2.9% (2.4%–3.5%)	11.1% (10.4%–11.8%)	45.9% (44.8%–47.0%)
Code frequency, mean (95% CI)	4.9 (4.7–5.0)	3.0 (3.0–3.1)	6.5 (6.3–6.6)	17.5 (16.9–18.1)
Matched length, % (95% CI)	71.8% (70.6%–73.0%)	73.9% (72.5%–75.2%)	62.7% (61.5%–63.8%)	58.1% (57.0%–59.2%)
Matched digits, % (95% CI)	56.3% (55.6%–57.0%)	63.3% (62.6%–64.0%)	53.2% (52.6%–53.8%)	30.8% (30.2%–31.4%)

- GPT-4's low code generation frequency (mean 3.0) implies it generates a focused output, while Llama2-70b Chat's high frequency (mean 17.5) suggests over-generation that could be associated with hallucination.

- GPT-4 has better structural alignment, which reduces fabricated codes
  - Higher matched length and digits percentages for GPT-4 (73.9% and 63.3%, respectively)

Error Analysis: **ICD-10-CM**

- All models, except GPT-4, struggle with ICD-10-CM code generation
  - GPT-4 is comparatively more reliable and less prone to hallucination, whereas the others tend to generate more errors and extraneous output.
  - GPT-3.5 showed a high incorrect codes (81.7%), which has high difference to its GPT-4 counterpart (65.8%).

Metric	GPT-3.5	**GPT-4**	Gemini Pro	**Llama2-70b Chat**
Incorrect codes, n (% of total)	13,025 (81.7%)	10,492 (65.8%)	15,170 (95.1%)	15,693 (98.4%)
Valid code, % (95% CI)	82.7% (82.0%–83.3%)	81.5% (80.7%–82.2%)	62.6% (61.8%–63.4%)	69.7% (69.0%–70.4%)
Fabricated code, % (95% CI)	17.3% (16.7%–18.0%)	18.5% (17.8%–19.2%)	37.4% (36.6%–38.2%)	30.3% (29.6%–31.0%)
Code frequency, mean (95% CI)	93.6 (88.6–98.7)	3.7 (3.7–3.8)	46.2 (44.0–48.4)	63.1 (60.4–65.9)
Matched length, % (95% CI)	57.4% (56.6%–58.3%)	64.7% (63.8%–65.7%)	58.9% (58.1%–59.7%)	31.3% (30.6%–32.1%)
Matched digits, % (95% CI)	57.0% (56.6%–57.4%)	67.6% (67.2%–68.0%)	51.6% (51.3%–52.0%)	37.5% (37.1%–37.8%)

- Gemini Pro & Llama2-70b exhibit high incorrect codes (95.1% and 98.4%, respectively)
  - Fabricated codes (37.4% for Gemini Pro) and (30.3% for Llama2-70b Chat)
  - Gemini Pro & Llama2-70b have significant challenges in accurately capturing ICD-10-CM codes.

Error Analysis: **CPT**

- GPT-4 outperformed other models with the lowest incorrect code rate (53.9%) and the highest valid code percentage (97.1%).
  - Llama2-70b Chat performed poorly, with almost all generated codes being incorrect (98.8%), only 54.1% valid codes, and the highest fabricated code rate (45.9%).

Metric	GPT-3.5	**GPT-4**	Gemini Pro	**Llama2-70b Chat**
Incorrect codes, n (% of total)	2502 (68.1%)	1843 (50.2%)	3225 (88.6%)	3579 (97.4%)
Valid code, % (95% CI)	94.0% (93.0%–94.9%)	93.9% (92.8%–95.0%)	84.1% (82.8%–85.3%)	54.8% (53.1%–56.4%)
Fabricated code, % (95% CI)	6.0% (5.1%–7.0%)	6.1% (5.0%–7.2%)	15.9% (14.7%–17.2%)	45.2% (43.6%–46.9%)
Code frequency, mean (95% CI)	8.4 (7.5–9.3)	2.6 (2.5–2.7)	15.3 (14.0–16.7)	60.3 (56.3–64.4)
Matched length, % (95% CI)	99.7% (99.4%–99.9%)	98.5% (98.0%–99.1%)	98.7% (98.3%–99.1%)	98.8% (98.4%–99.1%)
Matched digits, % (95% CI)	59.5% (58.7%–60.4%)	63.3% (62.3%–64.2%)	53.7% (53.0%–54.5%)	40.8% (40.1%–41.6%)

- All models generate outputs of the correct overall length (matched lengths = around 98–99%)
  - Percentage of matched digits is considerably lower for Llama2-70b Chat (40.8%)—inaccurate coding.
- Gemini Pro and Llama-2-70 Chat (base forms) are not yet reliable medical coders
  - High rates of incorrect and fabricated codes
  - Excessive output frequencies
  - Not yet reliable medical coders without further fine-tuning or tool integration.

Discussion: **Are LLMs poor medical coders?**

- None of the evaluated models achieved a high **exact match** rate overall, with even the best model (GPT-4) reaching only about 46% for ICD-9-CM, 34% for ICD-10-CM, and 50% for CPT codes.
  - GPT-4 performed the best in terms of exact match rates and multiple measures of conceptual similarity.
  - GPT-4 had the lowest rate of fabricated codes (ICD-9-CM, ICD-10-CM, and CPT)

- LLM-generated CPT and ICD-9-CM codes are more accurate than ICD-10-CM codes.
  - ICD-10-CM codes are longer, alphanumeric, and more granular.
  - LLMs frequently produced overgeneralized or entirely incorrect codes
    - Unable to fully comprehend the detailed alphanumeric structure of medical billing codes.

- Error patterns (e.g., missing digits, extra characters, or fabricated codes) suggest LLMs do not have complete internal representation of medical coding rules.
  - Base LLMs struggle with matching alphanumeric codes (e.g., ICD-10-CM) to their descriptions.
    - LLMs can correctly generate the initial three digits but fail to accurately extend the code
    - LLMs does not fully internalize the precise formatting rules.

Discussion: **Are LLMs poor medical coders?**

- Tokenization Challenges
  - LLMs break into subword units, which can obscure precise structure of medical codes
  - Loss of critical information regarding exact character order and composition

ICD-10-CM code: ['E11.9'] diabetes mellitus without complications

Tokenization: ['E11', '.', '9']

- Sensitivity of medical codes
  - Medical codes require strict adherence to specific formatting
    - Minor deviations can result to incorrect or fabricated codes
  - The sensitivity to exact characters is neglected by LLMs trained on general language
- LLM Training Limitations
  - Models are trained on natural language, resulting in overgeneralized and frequently imprecise
- Potential solutions
  - Fine-tuning, RAG frameworks

Discussion: **How about previous studies?**

- Previous studies have also reported that general-purpose (base) LLMs are suboptimal for medical coding tasks.
  - LLMs hallucinate medical codes, generating imprecise or fabricated outputs[1,4]
    - LLMs only rely on statistical patterns rather than a true understanding of the strict coding rules, leading to significant inaccuracies.
  - Fine-tuned models (Spark NLP) achieved 76% exact match compared to GPT-4 (58%) and GPT-3.5 (40%)[5]

- Aside from fine-tuning, LLMs can improve its medical coding performance by:
  - RAG frameworks[2]
  - Hierarchical-aware uncertainty estimation[3]

[1] Simmons, A., Takkavatakarn, K., McDougal, M., Dilcher, B., Pincavitch, J., Meadows, L., ... & Sakhuja, A. (2024). Benchmarking Large Language Models for Extraction of International Classification of Diseases Codes from Clinical Documentation. *medRxiv*, 2024-04.
[2] Kwan, K. (2024). Large language models are good medical coders, if provided with tools. *arXiv preprint arXiv:2407.12849*.
[3] Maatouk, O. (2025). Leveraging LLMs for ICD Coding and Uncertainty Estimation: Can the model's awareness of the hierarchical structureof ICD-10 codes impact its prediction performance?.
[4] Addimando, S. A. From Words to Codes: Large Language Models for ICD-9 Extraction in Clinical Documents.
[5] Kocaman, V. (2023, April 20). *Comparing Spark NLP for Healthcare and ChatGPT in Extracting ICD10-CM Codes from Clinical Notes*. John Snow Labs. https://www.johnsnowlabs.com/comparing-spark-nlp-for-healthcare-and-chatgpt-in-extracting-icd10-cm-codes-from-clinical-notes/

Conclusion

- Base LLMs alone are poorly suited for medical code mapping tasks.
  - While models can approximate its meaning, LLMs display unacceptable lack of precision and high rate for falsifying codes.

- Higher performance was observed with more frequently occurring, shorter codes and simpler descriptions

- This study have found out that current base LLMs struggle with simple code queries
  - Enhancements through fine-tuning, integration with specialized tools, or retrieval-augmented generation could be essential for adapting LLMs to reliably perform medical code querying tasks.

Questions?

Coding Structures

- **ICD-9-CM**
  - Format: 3-5 numeric structures, with a possible decimal point after the first three digits
  - Range: codes are 001-999.99 (disease classification)
    - V-codes (V01-V91) and E-codes (E000-E999) for supplementary information

- **ICD-10-CM**
  - Format: Alphanumeric (3-7 characters)
  - 1st character: always a letter (A-Z); disease category
  - 2nd-3rd characters: numbers (0-9); body system and disease classification
  - 4th-7th characters: Alphanumeric and provide additional specificity
    - 4th digit: condition (e.g., severity, cause)
    - 5th digit: anatomical site
    - 6th digit: severity or type of encounter
    - 7th digit: extension

- **CPT**
  - 5 numeric digits, sometimes followed by modifiers

Study Limitations

- The study did not integrate additional strategies to improve LLM performance:
    - Advanced prompt engineering
    - Tools and frameworks
    - Retrieval augmented generation
    - Model fine-tuning

Performance Metrics

1. **METEOR**: comparison through exact matches, stemming, synonyms, and word order

Reference: ['diabetes mellitus without complication']
Predicted: ['diabetes without complications']

- Both text share the words 'diabetes' and 'without' (2 matches)
- Predicted add 's' at the end of complications vs no 's' has the same root meaning

2. **BERTScore**: computes cosine similarity using contextual embeddings rather than token matches
   - Captures contextual meaning and semantic relationship

Predicted	Reference	Cosine
diabetes	diabetes	**1.00**
without	without	**1.00**
complications	complication	**0.98**
(missing)	mellitus	**0.40**

$$\text{Precision} = \frac{1.00 + 1.00 + 0.98}{3} = 0.99$$

$$\text{Recall} = \frac{1.00 + 1.00 + 0.98 + 0.40}{4} = 0.85$$

$$\text{BERTScore(F1)} = 2 * \frac{(0.99 \times 0.85)}{(0.99 + 0.85)} \approx 0.91$$

[1] **METEOR**: Metric for Evaluation of Translation with Explicit Ordering; [2] **BERT**: Bidirectional Encoder Representations from Transformers