Journal of the American Medical Informatics Association, 2024, 1–10
https://doi.org/10.1093/jamia/ocae132
Research and Applications

Research and Applications

Can GPT-3.5 generate and code discharge summaries?

Matúš Falis, MScR[1,*], Aryo Pradipta Gema, MScR[1], Hang Dong, PhD[2], Luke Daines, PhD[3], Siddharth Basetti, MBSS[4], Michael Holder, MMedSci[5], Rose S. Penfold, BMBCh[6,7], Alexandra Birch, PhD[1], Beatrice Alex, PhD[8,9]

[1]School of Informatics, The University of Edinburgh, Edinburgh EH8 9AB, United Kingdom, [2]Department of Computer Science, University of Exeter, Exeter EX4 4QF, United Kingdom, [3]Centre for Medical Informatics, Usher Institute, University of Edinburgh, Edinburgh EH16 4UX, United Kingdom, [4]Department of Research, Development and Innovation, National Health Service Highland, Inverness IV2 3JH, United Kingdom, [5]Centre for Population Health Sciences, Usher Institute, The University of Edinburgh, Edinburgh EH16 4UX, United Kingdom, [6]Ageing and Health, Usher Institute, The University of Edinburgh, Edinburgh EH16 4UX, United Kingdom, [7]Advanced Care Research Centre, The University of Edinburgh, Edinburgh EH16 4UX, United Kingdom, [8]Edinburgh Futures Institute, The University of Edinburgh, Edinburgh EH3 9EF, United Kingdom, [9]School of Literatures, Languages and Cultures, The University of Edinburgh, Edinburgh EH8 9LH, United Kingdom

*Corresponding author: Matúš Falis, MScR, School of Informatics, The University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, United Kingdom (mfalis2@ed.ac.uk)

Abstract

Objectives: The aim of this study was to investigate GPT-3.5 in generating and coding medical documents with International Classification of Diseases (ICD)-10 codes for data augmentation on low-resource labels.

Materials and Methods: Employing GPT-3.5 we generated and coded 9606 discharge summaries based on lists of ICD-10 code descriptions of patients with infrequent (or generation) codes within the MIMIC-IV dataset. Combined with the baseline training set, this formed an augmented training set. Neural coding models were trained on baseline and augmented data and evaluated on an MIMIC-IV test set. We report micro- and macro-F1 scores on the full codeset, generation codes, and their families. Weak Hierarchical Confusion Matrices determined within-family and outside-of-family coding errors in the latter codesets. The coding performance of GPT-3.5 was evaluated on prompt-guided self-generated data and real MIMIC-IV data. Clinicians evaluated the clinical acceptability of the generated documents.

Results: Data augmentation results in slightly lower overall model performance but improves performance for the generation candidate codes and their families, including 1 absent from the baseline training data. Augmented models display lower out-of-family error rates. GPT-3.5 identifies ICD-10 codes by their prompted descriptions but underperforms on real data. Evaluators highlight the correctness of generated concepts while suffering in variety, supporting information, and narrative.

Discussion and Conclusion: While GPT-3.5 alone given our prompt setting is unsuitable for ICD-10 coding, it supports data augmentation for training neural models. Augmentation positively affects generation code families but mainly benefits codes with existing examples. Augmentation reduces out-of-family errors. Documents generated by GPT-3.5 state prompted concepts correctly but lack variety, and authenticity in narratives.

Key words: ICD coding; data augmentation; large language model; clinical text generation; evaluation by clinicians.

Background and significance

Large-scale multilabelled text classification (LMTC) tasks in Natural Language Processing (NLP) associate input documents with a set of output labels from a large label space, often hierarchically with a big-head long-tail distribution and data sparsity issues. Medical document coding is the task of assigning structured codes from a medical ontology—for example, the International Classification of Diseases (ICD) (https://www.who.int/standards/classifications/classification-of-diseases)—to clinical documents, a task performed by specially trained hospital staff. Coding consumes human resources that could be allocated to patient care. To ease this burden, research in machine learning and NLP cast medical document coding as an LMTC task.[1]

In automatic ICD coding, discharge summaries serve as input, yielding codes from a specified ICD version (eg, ICD-10-CM) (https://www.cdc.gov/nchs/icd/icd-10-cm.htm). ICD coding faces distribution challenges mirroring other LMTC tasks. Few common conditions (eg, hypertension), contrast with many underrepresented or absent in corpora, such as MIMIC-IV.[2] Moreover, limited real-world data availability, often restricted for privacy reasons, compounds these challenges. However, modern deep learning ICD-coding approaches (eg, CAML,[3] HLAN,[4] RAC[5]) are data-driven, and adversely affected by data sparsity unless explicitly designed to handle label under-representation. Techniques such as auxiliary information,[6–9] or data augmentation and synthesis[10–12] attempt to mitigate these issues. ICD-coding models with pretrained encoders at best match the current state-of-the-art—usually involving domain-specific versions of BERT.[13] Large language models (LLMs) such as GPT-3 and its newer variants[14] (eg, GPT-3.5) or Large Language Model Meta AI (LLaMA)[15] have recently displayed state-of-the-art performance on several tasks with emerging

Received: January 2, 2024; **Revised:** May 12, 2024; **Editorial Decision:** May 17, 2024; **Accepted:** May 22, 2024

capabilities.[16] In the medical domain, notably Med-PaLM 2[17] matching human performance in multiple-choice medical schools' exams. While using models such as GPT-3.5 is problematic with real discharge summaries due to privacy issues, these models have the potential to aid in generating synthetic discharge summaries for training local models.

Recently, LLMs, notably GPT-3.5, have become a new standard for advanced NLP tasks, especially ones reliant on understanding natural language. These models retain and apply background knowledge observed during training, yet can also produce fluent but inaccurate information (known as hallucination).[18] Writing and coding discharge summaries require extensive background knowledge making it of interest to explore an LLM's capability of reading, coding, and generating discharge summaries. An LLM capable of handling medical text could address the data sparsity issue by synthesizing new data. This study aims to investigate the viability of GPT-3.5-generated medical documents for data augmentation in training local neural models and their credibility in clinical settings. We investigate GPT-3.5 given its performance in natural language understanding tasks requiring background knowledge, within an ethical experimental setting for clinical note data (by disabling content monitoring by the service provider). Furthermore, we explore GPT-3.5's ability to code real discharge summaries and self-generated text.

Recent studies have prompted discussions on GPT's utility in medicine, including applications in medical chatbots,[19] or radiology.[20] Yeung et al[21] compared the ChatGPT with a clinical GPT model[22] on generating patient vignettes. While research exists in generating data in low-resource settings in similar domains (eg, law[23]) to the best of our knowledge, GPT's performance in generating discharge summaries based on input conditions and its ability to perform ICD coding has not yet been reported.

Objective

This study aims to assess GPT-3.5's efficacy in the context of automated ICD-10 coding and investigate its viability as:

- A data generator for ICD-10 coding, enhancing the training of local neural models by including GPT-3.5-generated discharge summaries, especially for rare labels;
- An automated ICD-coding classifier, either using synthetic text with explicit code descriptions or real data as prompts; and
- A discharge summary generator focusing on producing clinically accurate and plausible synthetic data from expert perspectives.

Materials and methods

We queried GPT-3.5 (gpt-3.5-turbo) (https://platform.openai.com/docs/models/gpt-3-5), from hereon referred to as *G*, through the OpenAI Python API (https://platform.openai.com/docs/api-reference?lang=python) to generate patient discharge summaries based on specific conditions and procedures represented by ICD-10-CM and ICD-10-PCS code descriptions from gold standard labels associated with MIMIC-IV discharge summaries (from the table hosp/d_icd_diagnoses.csv.gz). These label combinations were chosen to closely follow real scenarios to correspond to correlations between real labels. Note that sharing MIMIC data via any online API is prohibited (https://physionet.org/news/post/415; our method was consulted with and approved by PhysioNet, as we merely use the descriptions of attached codes, which are not considered part of the dataset).

While various dataset splits have been proposed since the release of coded discharge summaries in MIMIC-IV, we specifically chose not to follow the one proposed by Edin et al[24] due to its exclusion of the long tail, which contrasts with our aim of addressing this aspect through generation techniques. Instead, we adopted the dataset split proposed by Nguyen et al[25] (Table 1) that preserved the long tail, aligning better with our focus. Nonetheless, we utilized the implementation of common ICD-coding models produced by Edin et al[24] for our analysis.

Label selection

Candidate source documents for generation were selected from MIMIC-IV based on label populations in our selected split. We identified codes common across training, validation, and test sets, choosing those appearing up to five times in the training set, resulting in 195 unique codes (compared to 15 353 unique few-shot codes in training).

For these 195 codes, we produced a list of codes belonging to their families (identified by the head of the code). Families with at least 1 relatively frequent code (population >100 in training) and at least 1 code exclusive to the test set (zero-shot) were retained leaving 16 families. The constraint of having at least 1 frequent code was included to explore the scenario where the model may predict a label due to its dominance in the population, and the increase in population of other labels potentially leading to more confusion.

From these 16 families, we randomly selected 10 to generate from (E10, G43, H35, H81, S00, S02, S06, T82, T84, T85). Of these, 114 codes had a population lower than 100 and are henceforth termed *generation codes* (a list of codes is available in Supplementary Material 1).

Preparation of samples for generation

As there exist correlations among labels (eg, different complications of type 1 diabetes, cancer correlating with the presence of chemotherapy), rather than creating random combinations of ICD codes we opted to work our way back from existing real scenarios.

In Nguyen's MIMIC-IV training set, we found documents with the 98 relevant few-shot codes (the 16 zero-shot were by-definition absent). Some documents contained multiple relevant codes. We have collected documents for each of the

Table 1. MIMIC-IV Nguyen dataset split. Zero-Shot corresponds to labels absent from the training set

	Train[a]	Dev[b]	Test[c]
# Documents	110 442	4017	7851
# Labels (total)	1 784 304	65 516	124 518
# Labels (unique)	25 230	6738	9159
# Labels (unique zero-shot)	N/A[d]	291	587
# Labels (unique few-shot)	15 300	919	1646

Few-shot corresponds to labels appearing at least once but no more than 5 times in the training set.
[a] Training set.
[b] Development set.
[c] Test set.
[d] Not Applicable (zero-shot implies absence in the training set).

relevant codes and cloned them to bring their population up to 100. To increase variety, we have randomly dropped up to 5 of the assigned nonrelevant labels within clones to create the new set of labels for generation (referred to as the silver standard).

We identified documents containing the siblings of the 16 zero-shot labels. A silver standard set was created for each of these documents substituting the sibling code with the zero-shot code (similar to the zero-shot approach in[10]). If multiple siblings were present, a random one was replaced with the zero-shot code. This resulted in 9606 input sets of labels—6779 unique and 2827 duplicated.

Generation

Natural language generation is the task of producing natural language text based on a set of input data. We used the model "gpt-3.5-turbo-0613" given its wide recognition in the field of LLMs, relative cost-effectiveness, and time efficiency (compared to gpt-4). We utilized a temperature (parameter in the 0-1 range controlling randomness) of 0 to produce deterministic outputs. We set the temperature to 0.1 for duplicates, allowing output variation.

Within the prompt (see Supplementary Material 2), we specified to write a discharge summary for a patient with a list of standard descriptions of their conditions and procedures based on our silver standard. We added further specifications:

- Length of up to 4000 words (following the maximum cut-off point in previous work[24]) The overall input and output token restriction of GPT-3.5 is 4096 (inclusive of the prompt);
- Inclusion of social and family history;
- Anonymization was required for personal and location data (due to uncertainty of the anonymity of the data used in training of GPT-3.5), maintaining numeric information when relevant despite potential removal in preprocessing;
- Explicit ICD code mentions within the main text were to be avoided to prevent model association or potential errors;
- Clear numeric values were preferred rather than ranges, especially for time-related codes;
- Providing a specific concept for codes involving the umbrella term "other" encompassing a range of conditions;
- Omission of the keyword "unspecified" present in standard descriptions opting for a more natural means of expression; and
- Coding of the discharge summary was to be positioned at the end in a regular pattern (codes in square brackets) for model coding assessment.

An example of the generation process can be seen in Figure 1.

The generated documents were processed to find the discharge diagnoses and extract the assigned codes with a regular expression. In total, we have generated 9606 synthetic training documents. We removed all mentions of ICD-10 codes and preprocessed the documents the same as the baseline data. Then, we merged these documents with the *baseline* training set (110 442 MIMIC-IV documents), forming the *augmented* training set (120 048 documents). The baseline and augmented settings used the same validation and test sets with 4017 and 7851 real documents, respectively.

Analysis between the generated text, and MIMIC-IV ICD-10-coded discharge summaries (the entire dataset, and the subset used for source label sets used in generation) showed differences in word count (per label and overall—Figures 2A and 2B) between synthetic and real data while retaining similar distributions of label counts (Figure 2). The synthetic discharge summaries tend to be shorter which could be the result of GPT-3.5's 4096 token limit. The synthetic discharge summaries were accepted for publication by PhysioNet will be made available at https://doi.org/10.13026/bnc2-1a81.

Local neural models

Most recent LMTC neural architectures are encoder-decoder models whose encoder processes the input text to generate a latent representation.

Architectures using a non-BERT-like encoder (eg, in CAML,[3] LAAT,[26] or Multi-Res CNN[27]) utilize noncontextual (eg, Word2Vec[28]) word embeddings, while BERT[29]-like encoders (eg, in PLM-ICD[30]) enable contextual token representation. The decoder determines a probability for each label based on the latent representation. A probability threshold determines positive predictions.

Evaluation

We have conducted 4 evaluation rounds:

- Local neural model evaluation: Assessing CAML, LAAT, and Multi-Res CNN models' performance on Nguyen's test set. Models were trained either solely on Nguyen's training set or enhanced with data generated by *G* (the augmented training set).
- GPT's coding on real data: Evaluation of *G*'s coding ability on MIMIC-IV using Nguyen's test set;
- GPT's coding on GPT-generated data: Evaluation of *G*'s coding ability on generated documents (with provided code descriptions in the prompt)
- Acceptability of generated data in clinical practice: Reviewing *G*-generated discharge summaries by clinical professionals to gauge their suitability in clinical settings.

Local neural model evaluation

We used the codebase and training procedure of Edin et al[24] to train and evaluate CAML, LAAT, and Multi-Res CNN on the ICD-coding task using Nguyen's split. Each model is trained with 20 epochs on the training set. Throughout the training, we evaluated the model on the validation set, selecting the model with the highest mean average precision from each run as the final model. Subsequently, this final model underwent evaluation on the test set. Additionally, we explored PLM-ICD, considered a state-of-the-art model for this task. However, its performance on the baseline and augmented data was notably lower than previously reported. This model has been reported to be unstable during training, which depends on random seeds. We decided against tuning the random seed, and only report the performance for CAML, LAAT, and Multi-Res CNN.

The models' test performance was evaluated using standard information retrieval metrics commonly used in LMTC tasks—micro- and macro-averaged Precision, Recall, and F1 scores.

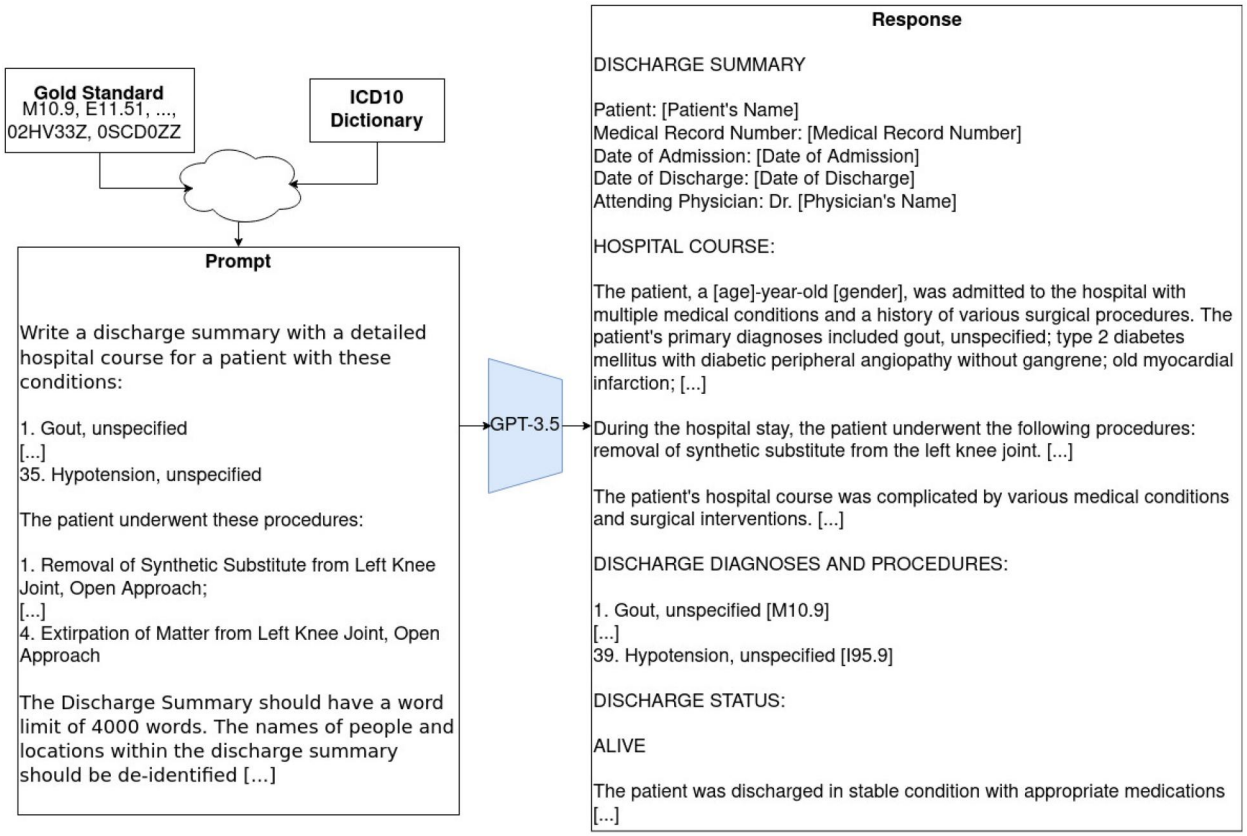

Figure 1. An example generation of a synthetic discharge via GPT-3.5.

Micro-averaging assigns equal weight to each prediction, favoring high-population classes (eg, hypertension). Macro-averaging, in contrast, computes the performance for each unique label and averages across the label space, giving each label's average result equal weight regardless of their population. This highlights poor performance in less common classes. Our primary evaluation metrics common with the majority of previous work are micro-F1 and macro-F1 scores. Metrics are further explained in Supplementary Material 3.

GPT's coding on real clinical notes

We used the Azure AI Services API (https://azure.microsoft.com/en-gb/products/ai-services) to test GPT's ability to assign diagnosis codes based on real clinical notes. The API returns a free text response which we further processed to retrieve the predicted ICD-10 codes. In postprocessing, we verify the response format. For correctly structured arrays of JSON objects we simply extract the predictions. For incorrectly structured outputs, we employ a regular expression pattern to extract all diagnoses and ICD code pairs. The result is a list of predicted diagnoses and corresponding ICD-10 codes for each clinical note. Figure 3 illustrates the API call workflow.

For reproducibility, we specified the model version and the API version as "gpt-3.5-turbo-0613" and "2023-03-15-preview," respectively. All parameters were set to zero for deterministic responses from G, including temperature, top P, frequency penalty, and presence penalty. The system prompt directed G to act as a clinician assigning ICD-10 diagnosis codes to clinical notes, specifying the expected output format as JSON objects with keys "diagnosis" and "icd_code." Refer to Figure 3 for all hyperparameter details. Our code implementation can be found in our Github repository (https://github.com/EdinburghClinicalNLP/chatgpt_icd_coding).

We opted out of human review of the data for 2 reasons. First, the terms of the data use agreement of MIMIC-IV (https://physionet.org/news/post/415) did not grant us the authority to permit a third party to process the data for abuse detection. Second, we assessed the likelihood of harmful misuse to be low given the sensitive nature of the clinical notes.

In our evaluation of GPT's performance, we have also employed hierarchical evaluation techniques—set-based hierarchical evaluation[31] and Count-Preserving Hierarchical Evaluation (CoPHE).[32] These metrics award partial credit to mispredicted labels by extending prediction and gold standard sets with their ancestor labels. While set-based evaluation ancestor labels track only the presence of descendants, in CoPHE, ancestor labels link to the count of descendants, penalizing over- and under-predictions within code families.

Comparing set-based and CoPHE results helps to evaluate the model's tendency to over-/under-predict. A lower CoPHE score indicates this phenomenon. See Supplementary Material 4 for details on calculating hierarchical scores.

We utilize macro-averaged metrics from weak hierarchical confusion matrices[10] to summarize in-family versus out-of-family (OOF) prediction errors. These metrics are chosen to explore how expanding the population of codes within code families to a minimum of 100 instances impacts within-family performance. Within-family errors involve false positives that align with false negatives within the same family in the gold standard. On the other hand, an OOF error for a false negative in the gold standard lacks a false positive

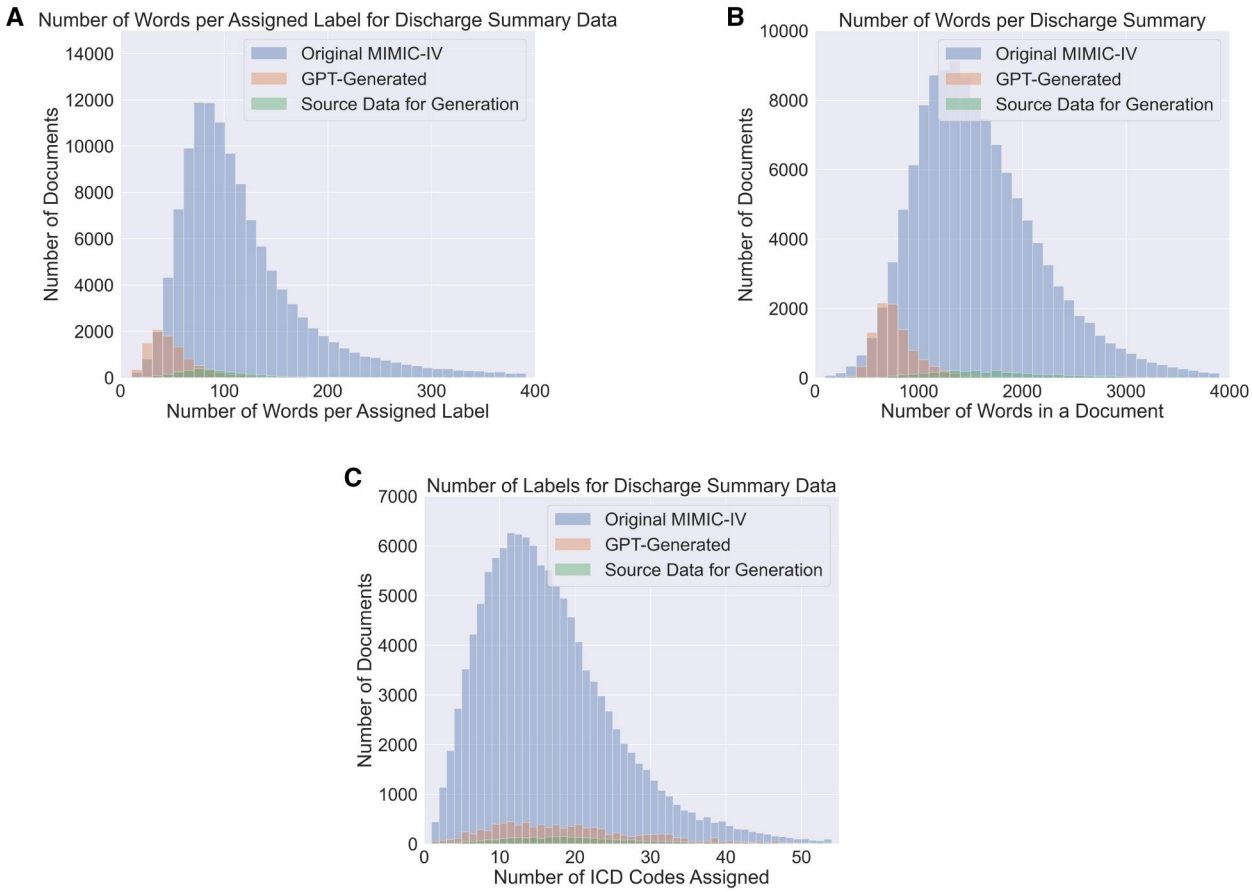

Figure 2. A comparison between the discharge summary data in MIMIC-IV, seed MIMIC discharge summaries for generation (the source data), and the generated discharge summaries. Subfigures 2A and 2B focus on the number of words in documents, indicating that the GPT-generated data generally contains fewer words overall and per assigned label compared to the real data from MIMIC-IV. Subfigure 2C demonstrates that, although there's a variance in document size, the distribution of the number of labels per document remains relatively similar across the datasets.

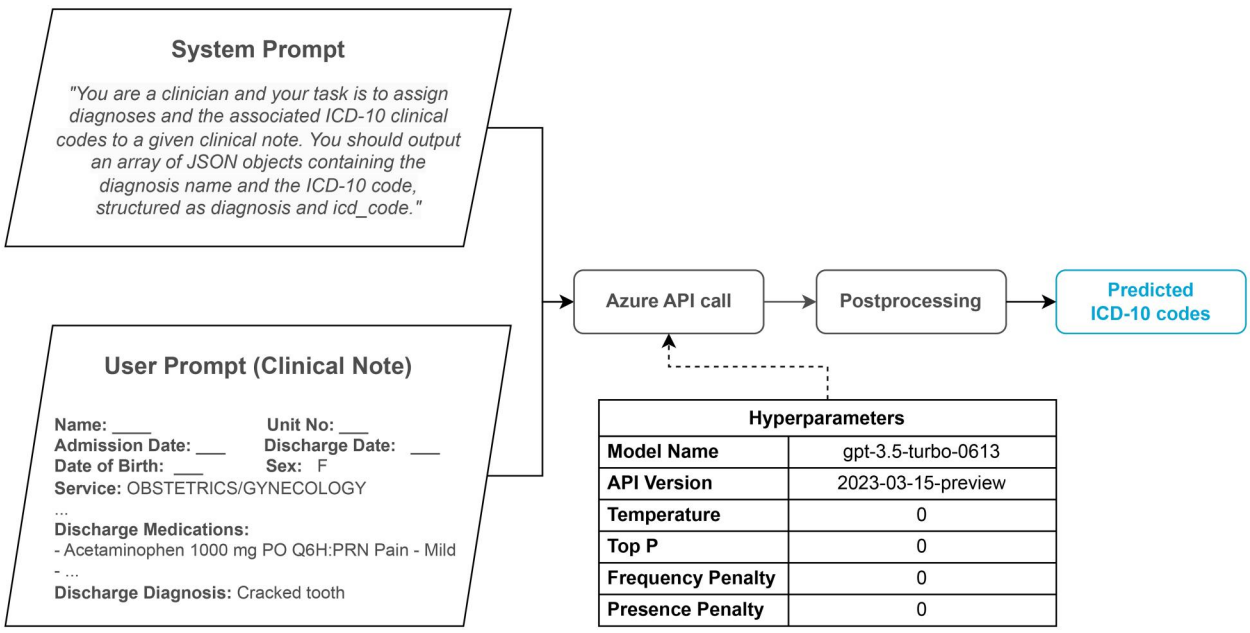

Figure 3. The workflow of the GPT-3.5 prediction. We used Azure AI Services API to query GPT-3.5 and we employed a postprocessing step to extract the predicted diagnoses and ICD-10 codes for each clinical note.

within the prediction set from the same family to match. Our primary goal in generating synthetic data is to reduce OOF errors, enhancing true positive predictions or ensuring mispredictions occur within family.

GPT-3.5's coding on synthetic data

The prompt asked *G* to code the conditions and procedures mentioned in the document it generated. This experiment tested *G*'s ability to assign ICD-10 codes to concepts presented in their standard descriptions. Alongside this, the prompt required creating a patient's social and family history, which might have led to the model introducing new conditions like substance abuse and potentially coding them, despite not being part of the initial prompt.

Acceptability of generated data in clinical practice

Four clinical professionals (coauthors S.B., L.D., M.H., and R.S.P.) assessed the quality of the generated data. As the data was generated based on labels associated with MIMIC-IV discharge summaries, this evaluation included both synthetic discharge summaries generated by *G* and discharge summaries from MIMIC-IV. The clinicians were presented with 20 discharge summaries—10 synthetic and 10 real (based on whose adjusted gold standard the synthetic ones were generated).

Each discharge summary was assessed for:

- Correctness—accuracy in describing patient conditions and procedures;
- Informativeness—clarity and sense in supporting information (eg, test results, medication suggestions);
- Authenticity (patient)—whether such a patient could exist;
- Authenticity (clinical scenario)—whether the hospital course was plausible as reported; and
- Acceptability—suitability of the document for clinical use.

Additionally, they separately evaluated correctness and informativeness for both nonlow-resource and low-resource labels to gauge *G*'s ability to generate low-resource data. Scores from 1 (failure to perform) to 5 (perfect performance) were assigned to each metric, with accompanying comments justifying the score. An example evaluation by a clinician can be seen in Figure 4.

Results

Local neural model evaluation

We assessed performance across 3 code subsets: the entire codeset in MIMIC-IV (*overall*), a restricted generation set ($f_{(gen)}$) containing only the 114 low-population candidate generation labels, and a set of codes from the code families present in $f_{(gen)}$ (f). Results are shown in Table 2. Baseline results for CAML and LAAT align closely with prior findings by Nguyen et al[25] for *overall* metrics. While Nguyen et al[25] did not report on Multi-Res CNN, its performance trends were similar to CAML and LAAT in comparison to Edin et al[24] on nonfiltered codesets. LAAT excels in micro-F1, Multi-Res CNN leads in macro-F1, and CAML generally lags behind. Baseline models outperform augmented ones in *overall* micro-F1, a common observation when enhancing

Synthetic Discharge Summary	**Non-Low-Resource Labels**
DISCHARGE SUMMARY Patient: [Patient's Name] Medical Record Number: [Medical Record Number] Date of Admission: [Date of Admission] Date of Discharge: [Date of Discharge] Attending Physician: Dr. [Physician's Name] HOSPITAL COURSE: The patient, a [age]-year-old [gender], was admitted to the hospital with multiple medical conditions and a history of various surgical procedures. The patient's primary diagnoses included gout, unspecified; type 2 diabetes mellitus with diabetic peripheral angiopathy without gangrene; old myocardial infarction; [...] mechanical loosening of internal left knee prosthetic joint, initial encounter; [...] During the hospital stay, the patient underwent the following procedures: removal of synthetic substitute from the left knee joint, open approach; [...] The patient's hospital course was complicated by various medical conditions and surgical interventions. The patient received appropriate medical management, [...] The patient's surgical procedures, including the removal of synthetic substitute from the left knee joint and insertion of a spacer, [...] DISCHARGE DIAGNOSES AND PROCEDURES: 1. Gout, unspecified [...] 39. Hypotension, unspecified DISCHARGE STATUS: ALIVE The patient was discharged in stable condition with appropriate medications [...]	M10.9: Gout, unspecified; E11.51: Type 2 diabetes mellitus with diabetic peripheral angiopathy without gangrene;[...] 0SCD0ZZ: Extirpation of Matter from Left Knee Joint, Open Approach **Low-Resource Labels** T84.033A: Mechanical loosening of internal left knee prosthetic joint, initial encounter **Correctness** Correctness Score - Non-Low-Resource (1-5): *5* Correctness Score - Low-Resource (1-5): *5* Correctness Comment: *All codes are present* **Informativeness** Informativeness Score - Non-Low-Resource (1-5): *3* Informativeness Score - Low-Resource (1-5): *2* Informativeness Comment: *Within the discharge summary text, the code descriptions are listed in a long paragraph and provided with minimal context or narrative. The description of the low resource code is limited to naming the code only.* **Authenticity** Authenticity Score - Patient (1-5): *1* Authenticity Score - Scenario (1-5): *2* Authenticity Comment:*This would be a very complex discharge summary to write given the very large number of codes. Unfortunately the summary has little or no attempt at providing a narrative or context for why the given conditions occurred. There is minimal effort at describing the treatment, or joining up the conditions, or providing a sense of what is most important in this clinical case.* **Acceptability** Acceptability Score (1-5): *1* Acceptability Comment: *Too vague, generic phrases, no context or narrative.*

Figure 4. An example evaluation of a synthetic discharge summary by a clinical expert.

Table 2. A comparison between local neural network models (MRCNN stands for Multi-Res CNN) trained on baseline (*base*) and augmented (*aug*) training sets is evaluated using micro- and macro-averaged *F*1 scores (*mi* and *ma* respectively) on 3 codesets—*overall* on all codes present in MIMIC-IV; f comprising all codes within the families we chose for generation; and $f_{(gen)}$ corresponding to candidate codes used in generation with a population of at most 100 in the training set.

	F1 ↑						WHCM error ↓			
Experiment	$mi_{overall}$	$ma_{overall}$	mi_f	ma_f	$mi_{f(gen)}$	$ma_{f(gen)}$	OOF_f	IF_f	$OOF_{f(gen)}$	$IF_{f(gen)}$
$CAML_{base}$	**53.65**	3.87	**38.43**	3.03	17.41	6.64	66.53	25.05	83.81	9.83
$CAML_{aug}$	53.54	**3.90**	38.41	**3.78**	**20.68**	**11.86**	**65.98**	**23.77**	**79.79**	**9.17**
$LAAT_{base}$	**57.29**	**6.18**	**43.59**	4.96	**26.79**	14.48	58.57	**28.35**	74.03	12.20
$LAAT_{aug}$	57.18	6.09	43.36	**5.38**	25.70	**14.98**	**55.93**	29.78	**73.65**	**11.98**
$MRCNN_{base}$	**55.66**	6.40	40.16	5.04	26.80	13.92	52.72	32.41	**69.68**	15.24
$MRCNN_{aug}$	54.69	**6.46**	**42.69**	**5.85**	**30.39**	**17.68**	**49.65**	**32.36**	70.41	**10.22**

The highest score in each metric for each model pair (baseline versus augmented) is highlighted in bold. Weak hierarchical confusion matrix (WHCM) error rates are produced for codesets f and $f_{(gen)}$. Performance on the common test set is reported using the macro-averaged proportion of errors that were out-of-family (OOF) and within family (IF). The best (lowest) error rate for each error type for each model pair (baseline versus augmented) is presented in bold.

Table 3. Results of GPT-3.5's coding ability on real and self-generated data.

	Leaf-only			Set-based			CoPHE		
Evaluation set	P	R	F1	P	R	F1	P	R	F1
GPT-3.5 real	9.46	33.51	14.76	10.59	44.87	17.13	10.30	44.33	16.72
GPT-3.5 synthetic	59.06	40.72	48.20	66.46	41.32	50.96	67.20	41.55	51.35
Best baseline (LAAT) real	60.42	54.46	57.29	61.28	54.50	57.68	60.84	54.33	57.39

lower-resource label performance. Nonetheless, macro-F1 scores improved for 2 out of 3 models within the *overall* codeset and for all models on f and $f_{(gen)}$. Multi-Res CNN and CAML macro-F1 scores display sizable relative improvement (26% and 78%, respectively) in $f_{(gen)}$.

Augmented models performed on par with or outperformed baseline models in micro-F1 scores for f and $f_{(gen)}$. Augmented Multi-Res CNN outperforms its baseline in micro-F1 for both f and $f_{(gen)}$, indicating benefits for the code family from augmenting less-populous members. Augmented LAAT shows improvement in macro-F1 in both f and $f_{(gen)}$ but lags in micro-F1. LAAT's performance may have been biased toward high-population classes and the augmentation's boosting of low-frequency classes (misrepresenting their frequency) may have introduced confusion. Apart from having a recurrent encoder (Bi-LSTM), the LAAT model employed in this experiment is about twice the size of the Multi-Res CNN (21.9M versus 11.9M parameters). This added model complexity may have enabled better performance on already frequent labels, but increased the need for more examples of lower-resource labels.

Comparing within-family and out-of-family errors (Table 2), augmented models generally exhibit fewer out-of-family errors on f and $f_{(gen)}$. An exception to this is augmented Multi-Res CNN in $f_{(gen)}$, whose slight increase in OOF came with a sizable reduction in within-family error.

Unlike baseline models, augmented ones occasionally predicted codes absent from the original data, although incorrectly, except for 1 correctly predicted code (S02.63XA) by a Multi-Res CNN model trained on augmented data. While consistent enhancement in zero-shot code performance was not achieved through augmentation, the potential for improvement is evident.

GPT's coding ability on real and synthetic data

We examined *G*'s ability to code real MIMIC-IV documents and generate coded documents with explicit code descriptions in the prompt. The results (Table 3) show that the performance on prompt-guided self-generated (synthetic) data resembles that of local models on the MIMIC-IV test set, not surpassing it. Hierarchical metrics show higher precision, recall, and consequently, F1-score in CoPHE compared to set-based hierarchical evaluation indicating errors coming from within-family misprediction, rather than incorrectly estimating the number of expected labels.

However, the performance on the MIMIC-IV test set is notably low, especially in precision. The improvement in the precision from leaf-only results to hierarchical is minimal. This implies that incorrect predictions were more likely to be out-of-family. Moreover, results on CoPHE are lower than on the set-based hierarchical evaluation indicating a tendency of the model to over-/under-predict within the scope of the family—an issue previously reported in local ICD-coding models[32] and present in the reported hierarchical results for baseline LAAT.

These results demonstrate that *G* can identify ICD-10 codes in self-generated mentions based on provided descriptions if presented within the prompts. Its performance when tasked with standard ICD coding without explicitly identified concepts or nonstandard surface forms of the concepts significantly deteriorates.

Acceptability of generated data in clinical practice

We calculated the inter-evaluator agreement for the 7 metrics using Fleiss' kappa (κ).[33] As κ is designed for categorical variables and does not fully capture ordinal scores, also produced the mean scores for each metric (μ). The results are presented in Table 4.

The evaluators' agreement was poor ($\kappa < 0$) in examples from MIMIC-IV for Correctness and Informativeness of nonlow-resource codes, and the acceptability of the discharge summaries. For the other metrics, a $\kappa > 0$ was reached but never exceeded 0.4 (lower than moderate agreement). All

Table 4. Evaluator agreement (κ) and mean scores (μ) for samples from MIMIC-IV (real), versus GPT-generated (synthetic) data

Metrics	κ_{real}	μ_{real}	$\kappa_{synthetic}$	$\mu_{synthetic}$
Correctness—nonlow resource	−0.386	4.175	−0.163	4.375
Correctness—low resource	0.043	4.350	0.206	4.525
Informativeness—nonlow resource	−0.155	4.550	−0.220	2.775
Informativeness—low resource	0.241	4.675	−0.277	3.000
Authenticity—patient	0.340	4.750	−0.078	3.150
Authenticity—scenario	0.373	4.775	−0.333	2.250
Acceptability	−0.056	4.550	0.035	2.225

mean scores are higher than 4. Hence, while the clinicians disagreed on the exact scores, they rated real discharge summaries positively. The disagreement may be due to clinicians being UK-based with significant differences in reporting style within the United Kingdom and the United States (where MIMIC-IV is from).

For GPT-generated summaries, slight agreement was seen in acceptability, and fair agreement in the correctness of low-resource labels. All other metrics had poor agreement. Both correctness metrics scored above 4, with low-resource correctness surpassing 4.5—an encouraging outcome for our primary goal of generating low-resource code data. Mean informativeness in the low-resource scenario and authenticity of scores were at least 3. Once again, performance on the low-resource codes exceeded nonlow-resource codes. Other metrics had μ scores above 2. Informativeness and authenticity for nonlow-resource codes had a poor agreement ($\kappa < 0$), while acceptability had some agreement with the lowest mean score of 2.225.

While *G* generally produces correct notes, the clinical evaluators have identified several challenges in the generation of natural-looking clinical notes:

GPT-3.5 tends to do verbatim reproductions of the prompted diagnoses list

G tends to copy all concepts mentioned in the prompt when generating a clinical note. While instruction following is a desirable behavior, excluding noncrucial details is essential when generating a natural-looking clinical note. Real clinical notes often omit irrelevant and less critical findings for brevity, particularly if the information is inferrable from surrounding contexts such as medications and treatments. For instance, *G* unnecessarily noted a normal BMI.

GPT-3.5 may phrase diagnoses in an unnatural manner

G tends to use an overly technical and unnatural style when specifying diagnoses. For instance, *G* mentioned "anaemia, which was unspecified," in the generated clinical note as it was prompted with "D64.9: Anemia, unspecified." *G* also occasionally introduces vague phrases (eg, "geriatric team provided supportive care, including behavioural interventions and medication management") without further detail. This contrasts with the more streamlined language of real clinical notes.

GPT-3.5 lacks details when introducing supporting information

G tends to introduce crucial supporting information without sufficient details. For instance, *G* mentioned "Following a traumatic event" without further specification of the mentioned traumatic event, which is unacceptable in the clinical setting. This omission limits the overall informativeness of the patient's medical context, potentially hindering the notes' usability for a comprehensive view.

GPT-3.5 may introduce spurious supporting information

G sometimes introduces improbable but possible details. For instance, *G* overemphasized the significance of a patient's anxiety disorder regarding an episode of syncope and a subsequent facial fracture, which the clinicians consider unlikely.

GPT-3.5 failed to present diagnoses as interconnected events

G does not effectively present diagnoses as interconnected, resulting in fragmented notes that lack coherence. The clinicians described *G*-generated clinical notes as collections of unrelated facts. For example, *G* presented complications of Type 1 diabetes mellitus separately without illustrating their relation. The lack of coherence between diagnoses may impede the plausibility of the clinical note and undermine the overall acceptability and usefulness of synthetic notes.

GPT-3.5 failed to prioritize and emphasize critical diagnoses

G struggles to prioritize diagnoses based on clinical significance, which undermines the authenticity of the portrayed scenario. For example, *G* often places critical conditions on the same level as minor issues, such as impacted ear wax, cataracts, and conjunctival hemorrhage. Hence, we concluded that *G* struggles to effectively convey the relative clinical significance of certain diagnoses.

Discussion, conclusion, and future studies

In this work, we have investigated the capability of GPT-3.5's potential in augmenting ICD-10 coding for local neural models in low-resource scenarios. While overall performance dipped with synthetic data augmentation, filtered codeset evaluation showed improvements, especially in advanced models like LAAT and Multi-Res CNN. Error analysis indicated augmented models made fewer out-of-family predictions, with some shift to within-family errors (closer to the correct answer). Augmentation showed promise in improving the prediction of generated codes and their siblings. Zero-shot labels did not consistently benefit from the augmentation, emphasizing the need for real data in augmentation success. However, a zero-shot code learned from the synthetic data was predicted correctly. The potential of LLM-generated discharge summaries should further be explored with different (eg, local or specialized) LLMs, prompt engineering, and further supplementing the prompt with external knowledge (eg, from ontologies).

In guided synthetic settings with ICD-10 descriptions, GPT-3.5 showed partial code identification ability displaying lesser over-/under-prediction tendencies than previously reported local models. It, however, struggled in the realistic scenario without in-prompt aid, performing below locally-trained models. Hence, the explored setup of producing a synthetic document based solely on the associated ICD codes is unsuitable for deployment in a clinical setting.

Clinician-evaluated synthetic discharge summaries showed correctness in individual codes, yet lacked naturalness and coherence compared to real data, resulting in lower informativeness, authenticity, and acceptability scores. Synthetic

summaries failed to represent holistic patient narratives or prioritize critical diagnoses.

One potential solution to generating synthetic discharge summaries involves restructuring the prompt to order diagnoses chronologically, providing their corresponding timestamps. This could guide LLMs in creating synthetic notes mirroring the chronological progression of a patient's medical journey, enhancing coherence and prioritization.

Another promising solution is to retrieve real clinical notes as in-context learning examples to help guide the generation process[34] to aid LLMs in generating more realistic and coherent content. As this study focuses on evaluating LLMs' existing capability, we opted to evaluate it in a zero-shot framework. Future work may explore this idea's potential for generating more realistic-looking clinical notes.

Limitations

In this study, while the annotation experts are involved as coauthors, we ensured that they were independent from the development of the algorithms that involved the synthetic data. While the evaluation utilized few clinical experts ($n = 4$), they provided sufficient expertise in evaluating the notes. The study was blinded with respect to the real/synthetic status of documents, but according to the experts, the synthetic data differed from real enough to be distinguishable.

Author contributions

Matúš Falis proposed exploring data augmentation for ICD coding with GPT-3.5 and developed the main ideas of the publication with Aryo Pradipta Gema, Hang Dong, Alexandra Birch, and Beatrice Alex. Matúš Falis, Aryo Pradipta Gema, Hang Dong, Alexandra Birch, and Beatrice Alex contributed to the experimental design. Matúš Falis designed and performed the document synthesis and the evaluation of GPT's ICD-10 coding performance on GPT-generated data. Aryo Pradipta Gema evaluated GPT's ICD-10 coding performance on MIMIC-IV data. Matúš Falis, Aryo Pradipta Gema, Hang Dong, and Luke Daines designed the evaluation of discharge summaries by clinical staff. Luke Daines, Siddharth Basetti, Michael Holder, and Rose S. Penfold performed the clinical expert evaluation of the data, and Matúš Falis and Aryo Pradipta Gema analyzed its results. Matúš Falis and Aryo Pradipta Gema wrote the manuscript. All authors participated in editing the manuscript. Beatrice Alex and Alexandra Birch supervised the project and provided feedback and input on the experiments and analyses.

Supplementary material

Supplementary material is available at *Journal of the American Medical Informatics Association* online.

Funding

This work is supported by the United Kingdom Research and Innovation (grant EP/S02431X/1), UKRI Centre for Doctoral Training in Biomedical AI at the University of Edinburgh, School of Informatics. H.D. was supported by the Engineering and Physical Sciences Research Council (EPSRC, grant EP/V050869/1), Concur: Knowledge Base Construction and Curation. R.S.P. is a fellow on the Multimorbidity Doctoral Training Programme for Health Professionals, which is supported by the Wellcome Trust (grant 223499/Z/21/Z). B.A. was funded by Legal and General PLC as part of the Advanced Care Research Centre and by the National Institute for Health Research (NIHR) for the Artificial Intelligence and Multimorbidity: Clustering in Individuals, Space and Clinical Context (AIM-CISC, grant NIHR202639). The funders had no role in conduct of the study, interpretation or the decision to submit for publication. The views expressed are those of the authors and not necessarily those of the funders..

Conflicts of interest

We have identified no competing interests.

Data availability

The synthetic discharge summary data generated as part of this study will be shared on reasonable request to the corresponding author upon presenting a certificate of completion of the CITI Data or Specimens Only Research course from the Collaborative Institutional Training Initiative program (https://physionet.org/about/citi-course/). The data has been accepted for publication and will be made available via PhysioNet (https://doi.org/10.13026/bnc2-1a81).

References

1. Dong H, Falis M, Whiteley W, et al. Automated clinical coding: what, why, and where we are? *NPJ Digit Med*. 2022;5(1):159.
2. Johnson AE, Bulgarelli L, Shen L, et al. MIMIC-IV, a freely accessible electronic health record dataset. *Sci Data*. 2023;10(1):31.
3. Mullenbach J, Wiegreffe S, Duke J, et al. Explainable prediction of medical codes from clinical text. In: *Proceedings of NAACL-HLT*; 2018:1101-1111.
4. Dong H, Suárez-Paniagua V, Whiteley Explainable automated coding of clinical notes using hierarchical label-wise attention networks and label embedding initialisation. *J Biomed Inform*. 2021;116:103728.
5. Kim BH, Ganapathi V. Read, attend, and code: pushing the limits of medical codes prediction from clinical notes by machines. In: *Machine Learning for Healthcare Conference*. PMLR; 2021:196-208.
6. Rios A, Kavuluru R. Few-shot and zero-shot multi-label learning for structured label spaces. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*. Vol. 2018. NIH Public Access; 2018:31-32.
7. Song C, Zhang S, Sadoughi N, et al. Generalized zero-shot text classification for icd coding. In: *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*; 2021:4018-4024.
8. Ren W, Zeng R, Wu T, et al. 2022. Hicu: leveraging hierarchy for curriculum learning in automated icd coding, arXiv, arXiv:2208.02301, preprint: not peer reviewed.
9. Wang S, Tang D, Zhang L, et al. Hienet: bidirectional hierarchy framework for automated icd coding. In: *International Conference on Database Systems for Advanced Applications*. Springer; 2022:523-539.
10. Falis M, Dong H, Birch A, et al. Horses to zebras: ontology-guided data augmentation and synthesis for icd-9 coding. In: *Proceedings of the 21st Workshop on Biomedical Language Processing*. Association for Computational Linguistics; 2022.
11. Kim D, Yoo H, Kim S. 2022. An automatic icd coding network using partition-based label attention, arXiv, arXiv:2211.08429, preprint: not peer reviewed.

12. Barros J, Rojas M, Dunstan J, et al. Divide and conquer: An extreme multi-label classification approach for coding diseases and procedures in Spanish. In: *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*; 2022:138-147.
13. Afkanpour A, Adeel S, Bassani H, et al. 2022. Bert for long documents: a case study of automated icd coding, arXiv, arXiv:2211.02519, preprint: not peer reviewed.
14. Ouyang L, Wu J, Jiang X, et al., et al. Training language models to follow instructions with human feedback. In: Koyejo S, Mohamed S, Agarwal A, eds. *Advances in Neural Information Processing Systems*. Vol 35. Curran Associates, Inc.; 2022:27730-27744. https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53-be364a73914f58805a001731-Paper-Conference.pdf
15. Touvron H, Lavril T, Izacard G, et al. 2023. Llama: open and efficient foundation language models, arXiv, arXiv:2302.13971, preprint: not peer reviewed.
16. Zhao WX, Zhou K, Li J, et al. 2023. A survey of large language models, arXiv, arXiv:2303.18223, preprint: not peer reviewed.
17. Singhal K, Azizi S, Tu T, et al. 2022. Large language models encode clinical knowledge, arXiv, arXiv:2212, preprint: not peer reviewed.
18. Ji Z, Lee N, Frieske R, et al. Survey of hallucination in natural language generation. *ACM Comput Surv*. 2023;55(12):1-38. https://doi.org/10.1145/3571730
19. Lee P, Bubeck S, Petro J. Benefits, limits, and risks of gpt-4 as an ai chatbot for medicine. *N Engl J Med*. 2023;388(13):1233-1239.
20. Lecler A, Duron L, Soyer P. Revolutionizing radiology with gpt-based models: Current applications, future possibilities and limitations of chatgpt. *Diagn Interv Imaging*. 2023;104(6):269-274.
21. Yeung JA, Kraljevic Z, Luintel A, et al. 2023. Ai chatbots not yet ready for clinical use, medRxiv:2023-03.
22. Kraljevic Z, Bean D, Shek A, et al. 2022. Foresight-deep generative modelling of patient timelines using electronic health records, *CoRR*.
23. Ghosh S, Evuru CK, Kumar S, et al. 2023. Dale: generative data augmentation for low-resource legal nlp, arXiv, arXiv:2310.15799, preprint: not peer reviewed.
24. Edin J, Junge A, Havtorn JD, et al. 2023. Automated medical coding on mimic-iii and mimic-iv: a critical review and replicability study, arXiv, arXiv:2304.10909, preprint: not peer reviewed.
25. Nguyen TT, Schlegel V, Kashyap A, et al. 2023. Mimic-iv-icd: a new benchmark for extreme multilabel classification, arXiv, arXiv:2304.13998, preprint: not peer reviewed.
26. Vu T, Nguyen DQ, Nguyen A. 2020. A label attention model for icd coding from clinical text, arXiv, arXiv:2007.06351, preprint: not peer reviewed.
27. Li F, Yu H. Icd coding from clinical text using multi-filter residual convolutional neural network. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol 34; 2020:8180-8187.
28. Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. *Adv Neural Inf Process Syst*. 2013;26(2):3111-3119.
29. Devlin J, Chang MW, Lee K, et al. 2018. Bert: pre-training of deep bidirectional transformers for language understanding, arXiv, arXiv:1810.04805, preprint: not peer reviewed.
30. Huang CW, Tsai SC, Chen YN. 2022. Plm-icd: automatic icd coding with pretrained language models, arXiv, arXiv:2207.05289, preprint: not peer reviewed.
31. Kosmopoulos A, Partalas I, Gaussier E, et al. Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Min Knowl Disc*. 2015;29(3):820-865.
32. Falis M, Dong H, Birch A, et al. Cophe: a count-preserving hierarchical evaluation metric in large-scale multi-label text classification. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*; 2021:907-912.
33. Fleiss JL. Measuring nominal scale agreement among many raters. *Psychol Bull*. 1971;76(5):378-382.
34. Lewis PSH, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Larochelle H, Ranzato M, Hadsell R, et al., eds. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual*; 2020. https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Journal of the American Medical Informatics Association, 2024, 00, 1–10
https://doi.org/10.1093/jamia/ocae132
Research and Applications