Can Current Explainability Help Provide References in Clinical Notes to Support Humans Annotate Medical Codes?

Byung-Hak Kim[†,1] **Zhongfen Deng**[2] **Philip S. Yu**[2] **Varun Ganapathi**[1]

[†]project lead

[1]AKASA, Inc. [2]University of Illinois Chicago

Abstract

The medical codes prediction problem from clinical notes has received substantial interest in the NLP community, and several recent studies have shown the state-of-the-art (SOTA) code prediction results of full-fledged deep learning-based methods. However, most previous SOTA works based on deep learning are still in early stages in terms of providing textual references and explanations of the predicted codes, despite the fact that this level of explainability of the prediction outcomes is critical to gaining trust from professional medical coders. This raises the important question of how well current explainability methods apply to advanced neural network models such as transformers to predict correct codes and present references in clinical notes that support code prediction. First, we present an e**x**plainable **R**ead, **A**ttend, and **C**ode (xRAC) framework and assess two approaches, attention score-based xRAC-ATTN and model-agnostic knowledge-distillation-based xRAC-KD, through simplified but thorough human-grounded evaluations with SOTA transformer-based model, RAC. We find that the supporting evidence text highlighted by xRAC-ATTN is of higher quality than xRAC-KD whereas xRAC-KD has potential advantages in production deployment scenarios. More importantly, we show for the first time that, given the current state of explainability methodologies, using the SOTA medical codes prediction system still requires the expertise and competencies of professional coders, even though its prediction accuracy is superior to that of human coders. This, we believe, is a very meaningful step toward developing explainable and accurate machine learning systems for fully autonomous medical code prediction from clinical notes.

1 Introduction

Within current medical systems, the prediction of medical codes from clinical notes is a practical and essential need for every healthcare delivery organization (Dev, 2021). A human coder or health care provider scans medical documentation in electronic health records, identifying important information and annotating codes for that specific treatment or service. With a diverse range of medical services and providers (primary care clinics, specialty clinics, emergency departments, mother-baby units, outpatient and inpatient units, etc.), the complexity of human coders' tasks grows, while productivity standards fall as charts take longer to review. Thus, even partial automation of the annotation workflow will save significant time and effort that human coders currently spend. The biggest challenge, however, is directly identifying appropriate medical codes from thousands of high-dimensional codes from unstructured free-text clinical notes (Dong et al., 2022).

Lately, advanced deep learning-based methods for predicting medical codes based on clinical notes (Kim and Ganapathi, 2021; Sun et al., 2021; Liu et al., 2021; Yuan et al., 2022) have achieved state-of-the-art prediction performance and even reached parity with human coders' performance (Kim and Ganapathi, 2021). However, most current works on medical code prediction based on deep learning models do not provide the end-user with references from the clinical notes to explain why the predicted codes were presented/chosen. There have been some related works that provide the rationales or text highlights from clinical notes to explain why the predictions were made to support humans clinical decision making (Taylor et al., 2021; Cao et al., 2020; Mullenbach et al., 2018; Wood-Doughty et al., 2022). However, to the best of our knowledge, there is still a gap in studies that have thoroughly analyzed explainability to extract supporting text for code prediction, especially made by state-of-the-art (SOTA) transformer-based models such as the RAC model (Kim and Ganapathi, 2021).

Two examples are the attention score-based ap-

Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI), pages 26 - 34

proach first introduced in Mullenbach et al. (2018) and the model-independent knowlege-distillation based method recently initiated in Wood-Doughty et al. (2022). The first approach utilizes the per-label attention mechanism to select key sentences for prediction decisions; however, if the model does not have the per-label attention layer, it cannot generate text snippets, and even worse, applying this method to transformer-based architecture and deploying it to production comes with a range of compute and memory challenges (Vaswani et al., 2017). In the second knowledge-distillation approach, a large neural network is distilled into linear student models in a post-hoc manner without sacrificing much accuracy of the teacher model while retaining many advantages of linear models including explainability and smaller model size, which is beneficial for deployment.

This paper makes two primary contributions:

- First, we present a general and explainable xRAC framework that generates evidentiary text snippets for a predicted code, which are oriented towards the needs of a deployment scenario of the RAC model. Then, to better assess the explainability of our xRAC framework, human-grounded evaluations is conducted with two groups of internal annotators, one group with and one group without medical coding expertise. We find that the proposed xRAC framework can benefit professional coders but not lay annotators who lack relevant expertise and competencies.

- Second, we propose code-prior matching and text-prior matching losses to augment the original binary cross-entropy (BCE) loss used to train the RAC model. Because trained models with BCE loss typically tend to focus more on the frequent medical codes and their associated clinical notes portions of the dataset, these new losses are to help distribute the gradient update evenly across all of the codes and note tokens, regardless of code's frequency and token's relevance to codes, so as to improve the xRAC model's prediction as a whole.

2 xRAC Framework

2.1 xRAC-ATTN

The original RAC architecture is built on the code-title guided attention module that considerably improves the per-label attention mechanism first introduced in Mullenbach et al. (2018). This enhanced attention module is to address the extreme sparsity of the large code output space with so called code-title embedding. Because code titles (or descriptions) contain important semantic information and meaning of the codes, the RAC model obtains its embedding from its textual description as shown in the Table 2 examples. Specifically, the code description is fed into an embedding layer, which is then followed by a CNN and Global Max Pooling layer to learn the code embedding.

Therefore, the first xRAC-ATTN directly leverages the attention scores learned in the RAC model to generate the evidence text for each code i. In particular, the attention scores $\mathbf{w}_i^{\text{ATTN}} = (w_{i,1}^{\text{ATTN}}, ..., w_{i,n_x}^{\text{ATTN}})$ on the input tokens for code-i is computed as follows:

$$\mathbf{w}_i^{\text{ATTN}} = \text{Softmax}\left(\frac{\mathbf{e_i}\mathbf{U}_\mathbf{x}^\mathbf{T}}{\sqrt{d}}\right), \tag{1}$$

where $\mathbf{e_i} \in \mathbb{R}^{1\times d}$ is one row of $\mathbf{E_t} \in \mathbb{R}^{n_y \times d}$ which is the code embeddings from the code descriptions, $\mathbf{U_x} \in \mathbb{R}^{n_x \times d}$ is the text representation outputted by the Reader, d is the dimension of code embedding, n_x is the number of tokens in the input document, and n_y is the number of codes in the dataset.

2.2 xRAC-KD

The application of the original idea of knowledge distillation (Hinton et al., 2014) requires specific adjustments to the problem setting of medical codes prediction. Knowledge distillation is typically used to train a compact neural network from a large or ensemble of neural network models. Unlike those standard approaches, xRAC-KD transfers the large RAC-based "teacher" model into a set of reliable and explainable "student" linear models by distilling the predictions made by the large teacher model.

Assume that we have a trained "teacher" neural network $f_{\text{teacher}}(\mathbf{x}_t)$ and training data.[1] xRAC-KD approximates $f_{\text{teacher}}(\mathbf{x}_t)$ with a collection of student linear models $f_{\text{student}}(\mathbf{x}_s) = (f_{s,0}(\mathbf{x}_s), ..., f_{s,n_y}(\mathbf{x}_s))$ defined as

$$f_{s,i}(\mathbf{x}_s) = \mathbf{w}_i^{\text{KD}}\mathbf{x}_s, \tag{2}$$

[1]Note that because there is flexibility in using different representations for the same clinical note, we use different notations $\mathbf{x}_t$ and $\mathbf{x}_s$ to denote a tokenized clinical note. We use Word2Vec for $\mathbf{x}_t$ and bag of words for $\mathbf{x}_s$.

where $\mathbf{w}_i^{\text{KD}} = (w_{i,1}^{\text{KD}}, ..., w_{i,n_x}^{\text{KD}})$. $f_{\text{teacher}}(\mathbf{x}_t)$ produces predicted probability vector $\hat{\mathbf{y}}_t = (\hat{y}_{t,1}, ..., \hat{y}_{t,n_y})$. First xRAC-KD converts $\hat{\mathbf{y}}_t$ to $\mathbf{q}_t = (q_{t,1}, ..., q_{t,n_y})$ which is defined as

$$q_{t,i} = T\text{logit}(\hat{y}_{t,i}) = T \log\left(\frac{\hat{y}_{t,i}}{1-\hat{y}_{t,i}}\right), \quad (3)$$

where a temperature parameter T is to adjust the logit values and set it to 1 for convenience.

Then, as a distillation loss to train the student models $f_{\text{student}}(\mathbf{x}_s)$, xRAC-KD uses the L1 regularized regression loss between $\mathbf{q}_t$ and the student's predicted output vectors $\mathbf{q}_s$ written as follows

$$||\mathbf{q}_t - \mathbf{q}_s||_2 + \lambda||\mathbf{w}_i^{\text{KD}}||_1, \quad (4)$$

with λ parameter. xRAC-KD does not use any additional loss term with respect to the training data's hard labels (either 0 or 1). Once the distilled student models $f_{\text{student}}(\mathbf{x}_s)$ are ready, xRAC-KD finally transforms the output vector $\mathbf{q}_s$ back to the prediction vector $\hat{\mathbf{y}}_s = (\hat{y}_{s,0}, ..., \hat{y}_{s,n_y})$ easily as follows:

$$\hat{y}_{s,i} = \text{expit}\left(\frac{q_{s,i}}{T}\right) = \frac{1}{1+\exp(-q_{s,i}/T)}. \quad (5)$$

The logit and expit transforms defined in Eq. (3) and (5) pairs that are inverse to each other are a fundamental improvement over the initial method presented in Wood-Doughty et al. (2022). Previously, the distilled models showed consistently low precision scores and it was hypothesized for the independence of the distilled linear models. However, by comparing the first and last rows of Table 1, it turns out that this new pair has resulted in a clear outperformance across the board over the logistic regression baseline unlike the initial approach.

2.3 Supporting Text Extraction

Lastly, the evidence text of the xRAC-ATTN and xRAC-KD models is constructed by first locating the *n*-gram with the highest average weight score for each code i calculated as

$$\arg\max_j \sum_{n-gram} w_{i,j}, \quad (6)$$

then m tokens on either side of the n gram are included to obtain the final subsequence of evidence with length of $n + 2m$. We set n to 4 and m to 5.

2.4 xRAC with Augmented Losses

The RAC model utilizes a transformer encoder and an attention-based architecture to attain SOTA performance. It also makes use of code descriptions to obtain code embeddings. Although the code embeddings obtained from the code description capture the semantic meaning of each code, due to the natural characteristics of medical coding, most of the codes appear just a few times compared to other common codes associated with common diseases.

Similarly, not all tokens in a given piece of text can be learned sufficiently and equally during the training process; therefore, frequent code embedding (as well as token embeddings) will receive more updates than infrequent codes (and tokens). In other words, trained models with BCE loss tend to focus more on the frequent codes and their associated clinical notes portions in the dataset; therefore, we propose code-prior matching and text-prior matching losses to supplement the BCE loss to encourage the models better handle imbalance issues and improve the model's overall prediction.

Code Prior Matching (CPM): To alleviate the issue of frequent codes receiving more updates than infrequent codes during training, CPM is applied to the second to the last output of the Coder, $\mathbf{V_x} \in \mathbb{R}^{n_y \times d}$ defined as

$$\mathbf{V_x} = \text{Softmax}\left(\frac{\mathbf{E_t}\mathbf{U_x^T}}{\sqrt{d}}\right)\mathbf{U_x}. \quad (7)$$

The CPM can help the model learn evenly across all codes, regardless of frequency, by imposing constraints on the learned $\mathbf{V_x}$. This prior matching module is implemented by a discriminator D_{cpm}, which shares the same structure as D_{lpm} in Deng et al. (2021) and introduces a regularization loss for each code as

$$\begin{aligned} l_c^i = -(&\mathbb{E}_{\mathbf{c_p}\sim\mathbb{Q}}[\log D_{\text{cpm}}(\mathbf{c_p})] + \\ &\mathbb{E}_{\mathbf{v_i}\sim\mathbb{P}}[\log(1 - D_{\text{cpm}}(\mathbf{v_i}))]), \end{aligned} \quad (8)$$

where $\mathbf{v_i} \in \mathbb{R}^{1\times d}$ is one row of $\mathbf{V_x}$ which is the vector for one code in the dataset, $\mathbb{P}$ is the code embedding distribution learned by the model, $\mathbf{c_p}$ is a prior vector of the same size as $\mathbf{v_i}$ for the given code generated by a uniform distribution $\mathbb{Q}$ in the interval of $[0, 1)$, and l_c^i is the prior matching loss for code-i.[2] We take the average of l_c^i losses

[2]We chose a compact uniform distribution on $[0, 1)$ as the

from all codes to obtain the final CPM loss L_{C} as follows:

$$L_{\mathrm{C}} = \frac{1}{n_y} \sum_{i=1}^{n_y} l_c^i. \tag{9}$$

Text Prior Matching (TPM): In the RAC model, not all tokens in a particular clinic text note can be learned equally, as the Reader focuses more on tokens related to frequent codes in the data set. To help the model's gradient be updated equally for all tokens in the input, a TPM loss is applied on $\mathbf{U_x}$ output of the Reader. The TPM is also implemented by a discriminator D_{tpm} similar to D_{cpm}, where it introduces another prior matching loss L_{T} shown as

$$\begin{aligned} l_t^i = -(&\mathbb{E}_{\mathbf{t_p}\sim\mathbb{Q}}[\log D_{tpm}(\mathbf{t_p})] + \\ &\mathbb{E}_{\mathbf{u_i}\sim\mathbb{P}}[\log(1 - D_{tpm}(\mathbf{u_i}))]), \end{aligned} \tag{10}$$

$$L_{\mathrm{T}} = \frac{1}{n_x} \sum_{i=1}^{n_x} l_t^i, \tag{11}$$

where $\mathbf{u_i}$ is one row of U_x that is the embedding of a token in the input document, $\mathbb{P}$ is the distribution of text embedding learned by the model, $\mathbf{t_p}$ is the prior embedding vector for the given token in the input document also generated by a uniform distribution $\mathbb{Q}$ in the interval of $[0, 1)$, and l_t^i is the TPM for a token in the input; we then use the average loss for all tokens in the input document as the final TPM loss L_{T}, similar to Eq. (9). This loss can make the model evenly learn the embeddings for all tokens in the input, which will be fed to the Coder for code prediction.

Overall Training Loss: Finally, the total augmented loss is written as

$$L_{\text{total}} = L_{\text{BCE}} + \alpha * L_{\mathrm{C}} + \beta * L_{\mathrm{T}}, \tag{12}$$

where α and β are parameters to balance L_{C} and L_{T} respectively. The updated RAC model trained with L_{total} instead of BCE loss, is first used for xRAC-ATTN and its performance is shown in the third row of Table 1. Although we used the original RAC model as a teacher model to distill from in xRAC-KD, this updated RAC model can also be used. Comparing the second (RAC model trained with BCE loss) and third rows (updated RAC model trained with L_{total}) in Table 1 shows modest improvements in both standard and hierarchical micro F1 scores, indicating that prior matching modules modestly help to address the imbalanced issues.

prior, which worked better in practice than other priors, such as Gaussian, unit ball, or unit sphere as shown in previous works (Deng et al., 2021; Hjelm et al., 2019).

3 Experimental Results

3.1 MIMIC-III Dataset

The MIMIC-III Dataset (MIMIC v1.4 Johnson et al. (2016)) is a freely accessible medical database that contains de-identified medical data from over 40,000 patients who visited the Beth Israel Deaconess Medical Center between 2001 and 2012.[3] We extract the discharge summaries and the corresponding medical codes, for this study. For a direct comparison with previous works, we use the same data processing, and data split described in (Mullenbach et al., 2018). This processing results in 47,724 samples for training, 1,632 and 3,373 samples for validation and testing, respectively, with an average number of 16 codes assigned to each discharge summary. More dataset statistics, can be found in Table 2 of (Mullenbach et al., 2018).

3.2 Training Details

The xRAC models follow the same training details as the RAC model, which can be found in the original RAC paper (Kim and Ganapathi, 2021). The xRAC-ATTN model is also trained with the same hyperparameters as the RAC model.[4] The xRAC-ATTN model's extra hyperparameters include α and β in Eq. (12), with values of 0.5 and 0.8 respectively. The temperature for the xRAC-KD model is set to 1, λ to 1e-3, and the maximum iteration for the training is set to 800.[5]

3.3 xRAC Model Performance

In addition to the same standard flat metrics used in previous RAC model evaluations, recently introduced hierarchical metrics (e.g. CoPHE (Falis et al., 2021), set-based metrics (Kosmopoulos et al., 2015)) are used. These two metrics take the hierarchical structure of the ICD codes tree into consideration for evaluating codes prediction. The CoPHE

[3] One reason for using the MIMIC-III dataset for this study is that it has been used as standard benchmark in previous studies (Kim and Ganapathi, 2021; Mullenbach et al., 2018), allowing meaningful head-to-head comparisons with our work. We believe that the proposed xRAC model is not limited to the MIMIC-III dataset and will also work well with a MIMIC-IV dataset, but MIMIC-IV-Note is currently not available to the public.

[4] The maximum sequence length is 4096, and there are four stacks of attention layers with single attention head. The code and text embedding dimensions are 300 and the batch size is 16.

[5] For the choices of hyper-parameters, we fine-tuned the model by running a linear search of these hyper-parameters to find the best value at which the model's performance peaks.

Table 1: Medical codes prediction results (in %) by ML systems on the MIMIC-III-full-label testing set as described in Kim and Ganapathi (2021). The bold value shows the best (and highest) value for each column metric. The logistic regression results are taken from Mullenbach et al. (2018), and the RAC results come from Kim and Ganapathi (2021). All numbers are the results of a single run with fixed random seeds, as practiced in the previous literature (Kim and Ganapathi, 2021; Mullenbach et al., 2018) for apples-to-apples comparisons. Note that our baseline is the most recent SOTA model RAC, and our xRAC-ATTN outperforms RAC in most metrics.

Model	**AUC**		**Standard F1**		**Precision@n**			**Hierarchical F1**	
	Macro	Micro	Macro	Micro	5	8	15	CoPHE	Set-Based
Logistic Regression	56.1	93.7	1.1	27.2		54.2	41.1		
RAC	**94.8**	**99.2**	**12.7**	58.6	**82.9**	75.4	**60.1**	62.7	64.0
xRAC-ATTN (ours)	**94.8**	99.1	12.6	**58.8**	**82.9**	**75.6**	**60.1**	**62.9**	**64.3**
xRAC-KD (ours)	93.6	98.7	7.4	46.0	69.4	61.6	48.6	51.8	54.5

metric further utilizes depth-based hierarchical representation and the count of codes at different ancestral levels of the tree to evaluate model's prediction, providing more meaningful evaluation in this context.

The results of the xRAC-ATTN and the xRAC-KD are shown in the last two rows of Table 1 respectively. First, when compared to the prior RAC model trained with BCE loss, the xRAC-ATTN model improves both standard and hierarchical micro F1 scores, as noted by comparing the second and third rows of Table 1, suggesting that the prior matching modules modestly help and effectively improve the SOTA scores. Second, while the xRAC-KD student model (shown in the last row) performs slightly worse than that of the RAC-based teacher model (shown in the second row), it still significantly outperforms the logistic regression baseline (shown in the first row, which was trained from scratch and has the same level of model complexity) across the board, which was not the case in Wood-Doughty et al. (2022).

3.4 Human-Grounded Evaluation

Human Evaluation Design: Human-grounded evaluation is important for evaluating the explainability. Because medical code annotation involves domain knowledge specific to medical coding, human evaluation is challenging; thus, we conducted a human evaluation with two groups of internal annotators. **Group A** had two annotators without medical coding experience and **Group B** had six certified professional coders. While both groups followed the same annotation instructions and guidelines, Group A was supervised by one manager and Group B was supervised by two managers with professional coder management experience to ensure annotation consistency (i.e., inter-annotator agreement) within each group. Group A worked full-time for two weeks to finish all the annotation, while Group B worked part-time for three weeks. Because the two groups of annotators involved in the human evaluation process are well aware that the task involves the medical notes of anonymized patients, the study does not require IRB approval and does not raise any ethical concerns.

Annotation Task Design: We select the overlap of codes predicted between the xRAC-ATTN and xRAC-KD models on the MIMIC-III-full-label testing set and combine the code descriptions and the corresponding textual explanations generated by each model together in a question sheet [6]. We then provide the sheet to Groups A and B for evaluation. Specifically, the question sheet contains six columns which are Question ID, Code and Description, Explanation Text Snippet, Highly Informative, Informative, and Irrelevant (see Table 2 for sample questions). Each code has two different text snippets extracted by two models, respectively. The annotators need to assign one of the three choices, which are highly informative, informative, and irrelevant to every explanation text snippet extracted to support the appearance of the predicted code.

Highly informative is defined as if the text snippet provides an accurate explanation for the pre-

[6]The MIMIC-III dataset's entire test set is used for human evaluation. Specifically, both the xRAC-ATTN and xRAC-KD models take clinical note from each example in the test set as input and predict multiple codes associated with this note. Because each model can predict differently for each example in the test set, we select all the test examples from the two models that are predicted with the same codes to compare their explainability. As a result, there are a total of 3,813 test examples predicted with the same codes by the xRAC-ATTN and xRAC-KD models.

Table 2: Two example questions provided for human evaluation: The codes in these two questions are the same, “521.00, Dental caries, unspecified", however, the two explanation text snippets classified as A) and B) are extracted by two different models, xRAC-ATTN and xRAC-KD. The information about the models is hidden from human annotators, and the order of text snippets for the same code is permuted to prevent the annotators from guessing the models based on the order. Note that **HI**, **I**, and **IR** stand for Highly Informative, Informative, and Irrelevant, respectively.

Question ID	**Code and Description**	**Explanation Text Snippet**	**HI**	**I**	**IR**
1	521.00, Dental caries, unspecified	A) surgical or invasive procedure left **and right heart catheterization** coronary angiogram multiple dental extractions			
1	521.00, Dental caries, unspecified	B) balloon s p dental extractions **s p exploratory laparotomy** and cholecystectomy fungal sepsis discharge			

Table 3: The overall informativeness of xRAC-ATTN and xRAC-KD retrieved explanatory text snippets. The left half represents the outcome of Group A's annotation, while the right half represents the outcome of Group B's evaluation. **HI**, **I**, and **IR** stand for Highly Informative, Informative, and Irrelevant, respectively. Percent denotes the ratio of informative text snippets (HI and I) to the total extracted snippets, which is 3,813 (in %).

Model	**Group A (Lay Annotators)**				**Group B (Professional Coders)**			
	HI	I	IR	Percent	HI	I	IR	Percent
xRAC-ATTN	1652	1389	772	79.75	1283	1094	1436	62.34
xRAC-KD	865	1318	1630	57.25	145	212	3456	9.36

Table 4: The evaluation agreements on Highly Informative and Informative text snippets between Groups A and B as measured by Jaccard Similarity (in %). Note that we evaluated the annotation consistency between two groups as described in Section 3.4, and the annotation consistency (or correctness) of lay annotators (Group A) is lower than 40% even provided with the same textual references as for professional coders (Group B).

Model	**Jaccard Similarty**	
	HI	I
xRAC-ATTN	39.2	18.5
xRAC-KD	7.0	5.0

dicted code. Otherwise, it is informative as long as the annotators believe that the text snippet adequately explains the presence of the given code, is related to the code's description, or has a close meaning to the code's description. Because the medical note contains domain knowledge, it is difficult for annotators to assign a finer-grained scale to the textual evidence when deciding between highly informative, informative, and irrelevant.

The final question sheet has a total of 3,813 codes predicted with different supporting text snippets. Unlike all previous studies, which typically collect less than 100 samples from clinicians (e.g., Mullenbach et al. (2018)), the task design of our study is quite unique, as is the volume of questions to our knowledge.

Human Evaluation Results: The results of human evaluation for the explainability of xRAC framework are shown in Tables 3 and 4. Table 3 shows the overall result of the informativeness of the text snippets extracted by xRAC-ATTN and xRAC-KD. The percentage column in Table 3 represents the percentage of explanations annotated as highly informative or informative, excluding irrelevant explanations. Thus, the irrelevant explanations generated by our model are about 20-40% as shown in Table 3.

One can see that there is a much larger gap in xRAC-KD between Group A and Group B than between xRAC-ATTN. Each group of annotators adhered to use the same standard to evaluate the textual explanation and was monitored by managers with professional coder management experience to ensure that there was no annotation variation among annotators in the same group. However, the large deviation between the two groups (Groups A and B) is understandable due to the domain knowledge gap between professional coders and lay annotators. Because of their limited medical knowledge and understanding, lay annotators tend to assign more highly informative and informative to the extracted textual explanation. Whereas, professional coders are much stricter on the informativeness of textual explanations.

In other words, this implies that xRAC-ATTN is a more viable choice than xRAC-KD to extract a text snippet from clinical notes to support code prediction. However, Table 4 shows that the consistency score measured by Jaccard Similarity between two groups is lower than 40% even with xRAC-ATTN. This suggests that the automated extraction system must continue to rely on professional coders' feedback and domain experience, and that text snippets alone are insufficient to replace them. In other words, there is still room to improve explainability for a lay person without expertise to appropriately code.

4 Conclusion

In this paper, a xRAC framework is presented to obtain supporting evidence text from clinical notes that justify the predicted medical codes from medical code prediction systems. We have demonstrated that the proposed xRAC framework may help even complex transformer-based models (e.g., RAC model) to attain high accuracy with a decent level of explainability (which is of high value for deployment scenarios) through quantitative experimental studies and qualitative human-grounded evaluations. It was also shown for the first time that, given the current state of explainability methodologies, using the proposed explainable yet accurate medical codes prediction system still requires professional coders' expertise and competencies.

Limitations

The current human-grounded evaluation studies only a simplified scenario: the impact of clinical-text-based explanations provided alongside predictions on explainability as judged by humans with and without professional coding backgrounds. This exercise sheds light on a key element that is necessary for these AI coding-based models to be useful in real-world deployment scenarios, but does not definitively ascertain that these coding predictions provided alongside explanations of the prediction would enable a transition to AI-driven coding autonomously. First, we have not studied how to incorporate the proposed xRAC framework into a human-in-the-loop situation with human coder feedback, which may be a very common scenario of deployment in practice. Second, we have not compared a full AI-driven coding model with humans-in-the-loop to a human-only process, in terms of speed, manpower needed, and accuracy. Limitations of the prediction model may become relevant in these situations, as human coders must occasionally combine disparate pieces of information together (Dong et al., 2022). Third, while the MIMIC-III dataset provides a useful benchmark for evaluating approaches, it is not representative of the wide range of clinical notes, so it would be

beneficial to expand to other data sets with a wider range of codes.

Ethics Statement

First and foremost, an automated and explainable machine learning system for medical code prediction aims to streamline the medical coding workflow, reduce the backlog of human coders by increasing productivity, and assist human coders quickly navigating complex and extended charts while reducing coding errors (Crawford, 2013). Second, an automated and explainable system is designed to lessen the administrative burden on providers, allowing them to focus on providing care rather than mastering the complexities of coding. Furthermore, better automated and explainable software can improve clinical documentation, enhance the overall picture of its quality, and eventually redirect lost healthcare dollars to more meaningful purposes (Shrank et al., 2019).

Acknowledgements

The authors would like to thank everyone who helped with the human-grounded evaluation, especially Juliann Chaparro, Tracy Salyers, Peg Leland, Jasmine Porter, Jenn Arlas, Amber Taylor, Stetson Bauman, Colin Wilde, Beth Cable, Dana Hunter, and Amy Raymond. Angela Kilby and Jiaming Zeng's suggestions to improve the manuscripts are also gratefully acknowledged.

References

Pengfei Cao, Chenwei Yan, Xiangling Fu, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020. Clinical-coder: Assigning interpretable ICD-10 codes to Chinese clinical notes. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, volume System Demonstrations, page 294–301, Association for Computational Linguistics.

Mark Crawford. 2013. Truth about computer-assisted coding: A consultant, HIM professional, and vendor weigh in on the real CAC impact. *Journal of American Health Information Management Association*, 84(7):24–27.

Zhongfen Deng, Hao Peng, Dongxiao He, Jianxin Li, and Philip Yu. 2021. HTCInfoMax: A global model for hierarchical text classification via information maximization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3259–3265, Online. Association for Computational Linguistics.

Sachin Dev. 2021. The healthcare delivery organization cios' guide to computer-assisted coding. *Gartner Research*.

Hang Dong, Matúš Falis, William Whiteley, Beatrice Alex, Shaoxiong Ji, Jiaoyan Chen, and Honghan Wu. 2022. Automated clinical coding: what, why, and where we are? *npj Digital Medicine*, 5(159).

Matúš Falis, Hang Dong, Alexandra Birch, and Beatrice Alex. 2021. 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*, pages 907–912, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2014. Distilling the knowledge in a neural network. In *NeurIPS 2014 Deep Learning and Representation Learning Workshop*.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of European Conference on Computer Vision 2016 (ECCV 2016)*.

Byung-Hak Kim and Varun Ganapathi. 2021. Read, Attend, and Code: Pushing the limits of medical codes prediction from clinical notes by machines. In *Proceedings of the 6th Machine Learning for Healthcare Conference (MLHC 2021)*, volume 149 of *Proceedings of Machine Learning Research*, pages 196–208. PMLR.

Aris Kosmopoulos, Ioannis Partalas, Eric Gaussier, Georgios Paliouras, and Ion Androutsopoulos. 2015. Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery*, 29(3):820–865.

Yang Liu, Hua Cheng, Russell Klopfer, Matthew R. Gormley, and Thomas Schaaf. 2021. Effective convolutional attention network for multi-label clinical document classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, pages 5941–5953, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, page 1101–1111, New Orleans, Louisiana. Association for Computational Linguistics.

William H. Shrank, Teresa L. Rogstad, and Natasha Parekh. 2019. Waste in the US health care system: Estimated costs and potential for savings. *JAMA Network*, 322(15):1501–1509.

Wei Sun, Shaoxiong Ji, Erik Cambria, and Pekka Marttinen. 2021. Multitask balanced and recalibrated network for medical code prediction. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2021)*.

Niall Taylor, Lei Sha, Dan W Joyce, Thomas Lukasiewicz, Alejo Nevado-Holgado, and Andrey Kormilitzin. 2021. Rationale production to support clinical decision-making. In *Machine Learning for Health (ML4H 2021)*, volume Extended Abstract.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*.

Zach Wood-Doughty, Isabel Cachola, and Mark Dredze. 2022. Model distillation for faithful explanations of medical code predictions. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, page 412–425, Dublin, Ireland. Association for Computational Linguistics.

Zheng Yuan, Chuanqi Tan, and Songfang Huang. 2022. Code Synonyms Do Matter: Multiple synonyms matching network for automatic icd coding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*.