Master thesis

Degree of Master in Information Technology - Data Science, AI and Machine Learning (120 credits)

Leveraging LLMs for ICD Coding and Uncertainty Estimation

Can the models awareness of the hierarchical structure of ICD-10 codes impact its prediction performance?

Computer Science and Engineering, 30 credits

Halmstad 2025-01-31
Ola Maatouk

HALMSTAD UNIVERSITY

ABSTRACT

Manual ICD coding of clinical notes with International Classification of Diseases (ICD) codes is a time-consuming and error-prone task, yet critical for effective healthcare management, billing, and research. This thesis investigates the use of Large Language Models (LLMs) to automate the prediction and assignment of ICD codes from clinical notes, aiming to enhance both the efficiency and accuracy of the process while reducing the reliance on human coders.

We developed a novel approach to leverage the hierarchical structure of ICD codes, which reflects the nested and interrelated nature of diagnoses, to improve the predictive capabilities of LLMs. By incorporating this hierarchical awareness, the model is better equipped to capture the dependencies among medical codes, thereby minimizing errors associated with irrelevant or less likely disease classifications.

Additionally, we implemented uncertainty estimation techniques during the inference phase to assess the confidence of the model's predictions. By quantifying prediction uncertainty, we can identify cases where the model may be less certain, providing valuable insights into areas where human oversight might still be necessary.

Our results show that integrating hierarchical structures reduces the likelihood of predicting irrelevant diseases that are distant from the actual disease in the hierarchical tree, highlighting the potential of this approach to improve ICD code assignment.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervisors, Amira Soliman, Awais Ashfaq, and Prayag Tiwari, for their outstanding guidance throughout this thesis project. Their valuable insights, thoughtful feedback, and continuous support were essential in guiding this work and achieving a successful outcome.

I am also grateful to my examiners, Jens Lundstrom and Slawomir Nowaczyk, who engaged in discussions with me throughout this work. Their input and suggestions were crucial for refining my ideas and moving the work forward. This thesis greatly benefited from their feedback and support.

I am thankful to have had such dedicated and supportive individuals alongside me on this journey.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

LLM Large Language Model

NLP Natural Language Processing

ICD International Classification of Diseases

ICD-9 International Classification of Diseases, Ninth Revision

ICD-10 International Classification of Diseases, Tenth Revision

MCD Monte Carlo Dropout

MIMIC-IV Medical Information Mart for Intensive Care IV

TD Total Distance

IIC Irrelevant ICD Chapter

IIB Irrelevant ICD Block

IIT Irrelevant ICD Third-Level

1 INTRODUCTION

Clinical notes are free-text medical records generated by healthcare specialists during patient visits. These notes are a rich source of information and contain valuable data, including symptoms, diagnoses, laboratory tests, procedures and medications [19].

A widely used approach for ICD coding involves the use of International Classification of Diseases (ICD) [52], a standardized coding format supported by the World Health Organization (WHO) [52, 11]. ICD coding is a task that involves assigning proper ICD codes to a clinical note, which result to classify and translate diagnoses and health issues into a standardized coding format [52]. ICD coding for the clinical note is an important aspect of healthcare as it enables efficient and standardized information retrieval and storage, enhances decision-making processes, supports billing processes to ensure accurate payments, and contributes to the overall quality of patient care [11, 52, 16].

Currently, ICD coding is performed manually by trained coders[22]. This manual process poses challenges, including high costs and proneness to errors, with an estimated annual expenses in the US, approaching $25 billion [50]. This is primarily due to the need for individuals with medical expertise and specialized ICD coding skills, making coder training challenging and resource-intensive task. Moreover, accurately assigning proper codes to input clinical notes is also challenging, even for professional coders, as a single document may require multiple ICD codes. The task's complexity is further compounded by the vast number of codes in the ICD taxonomy. For example, ICD version 9 (ICD-9) contains 18,000 codes, and ICD version 10 (ICD-10) contains 142,000 [11]. Additionally, manual ICD coding is time-consuming, taking an average of 34 minutes to assign codes to each patient [52].

As the volume of medical data increases, researchers are leveraging various Natural Language Processing (NLP) techniques to automate and semi-automate ICD coding aiming to enhance efficiency and accuracy while reducing human effort and coding errors [52]. Traditional approaches, such as rule-based systems and classical machine learning models, focus on extracting relevant information from clinical texts and mapping it to appropriate ICD codes [28]. However, these methods often struggle to handle the complexity and variability of clinical language, limiting their effectiveness in achieving full automation [44].

Recent advancements in Artificial Intelligence (AI), particularly in LLMs, have introduced new opportunities for automating ICD coding [17]. These AI-driven models utilize deep learning to better understand context and semantics, enabling more accurate and nuanced predictions [16]. By integrating these models into automatic coding systems, the performance and reliability of ICD coding can be improved [32].

While full automation in ICD coding aims to eliminate human intervention, it may face challenges in accurately capturing the nuanced and complex nature of clinical data. These challenges can lead to legal and ethical risks, as incorrect coding could result in improper treatments, billing errors, or compromised patient care. Therefore, maintaining human oversight is crucial to ensure responsible and ethical use of automated systems. Healthcare professionals must leverage their expertise to validate and adjust automated decisions, thereby ensuring patient safety [39]. Semi-automatic coding offers a balanced approach by combining machine capabilities with human expertise, allowing for greater precision and the integration of critical contextual insights. This approach recognizes the need for human involvement in managing the complexities of medical coding, resulting in more accurate ICD code assignments [45].

1.1 PROBLEM FORMULATION

This work explores the use of LLMs for predicting ICD codes from clinical notes. The selection of LLMs is driven by their remarkable capability in NLP [21]. A key focus of our work involves investigating how the hierarchical structure of ICD codes can be integrated into the LLM generative model. ICD codes follow a tree-like structure, with broader categories at higher levels and more specific diseases at lower levels [8].

In ICD-10, there are 22 chapters, and its hierarchical coding system includes chapters, blocks, three-character categories, and four-character subcategories. The coding format can extend up to seven characters. The first three characters represent the main disease category, while the fourth through seventh characters, when present, provide additional detail to specify a particular diagnosis or clinical information within that disease category. For example, the ICD-10 code hierarchy for hand and finger injuries (chapter S) includes a structured system that ranges from broad injury categories to specific finger injuries. This hierarchical coding system is organized as follows:

- Chapter level: Broad disease categories (Ex. S00-S99: Injuries to different body regions).
- Block level: Blocks within categories represent related conditions, offering increased detail (Ex. S60-S69: Injuries to the wrist, hand, and fingers).
- Three character codes: Further specificity, narrowing down to specific conditions or disease groups (Ex. S63: Dislocation, sprain, and strain of wrist and hand).
- Four character codes: For certain diseases, a fourth level provides additional detail within the three-character code (Ex. S63.1: Dislocation of the thumb).

The primary goal of this work is to explore how utilizing this hierarchical structure can reduce prediction errors in healthcare coding. Specifically, we aim to leverage this hierarchy to infer codes that are reasonably close to the true labels, ensuring that errors are not completely off-target or dubious in nature. In addition, we will assess the model's prediction reliability during the inference phase by measuring uncertainty, which will provide insights into cases where human oversight may still be necessary.

The main objectives of this project are:

- Hierarchical structure utilization: Investigating the utilization of the hierarchical structure in ICD codes to reduce the predictions of irrelevant disease situated at considerable distances from the actual disease in the hierarchical tree.
- Uncertainty estimation: Exploring the use of uncertainty estimation during the inference phase to understand prediction reliability and identify cases needing human oversight.

1.2 RESEARCH QUESTIONS

Our research questions are:

1. How can the hierarchical information of ICD codes be incorporated into model training?
2. How does the model awareness of the hierarchical structure of ICD codes affect model predictions performance?
3. How can model uncertainty be estimated to improve our understanding of the model's prediction reliability?

2 RELATED WORK

The automation of ICD coding has been a significant area of research in the intersection of NLP and healthcare. Numerous studies have explored various methodologies to enhance the accuracy and efficiency of ICD code prediction from clinical texts. This literature review provides an overview of the key advancements in this domain.

2.1 PREVIOUS APPROACHES FOR ICD CODING

The exploration of NLP, along with various machine learning and deep learning methods, has been a focal point for addressing the challenge of assigning ICD codes to clinical notes and aiding human coders in accurate and efficient code assignment [22]. The challenge of ICD coding was addressed as a text classification task [38, 51, 41, 16, 35]. This task involves a large set of labels, making it a complex multi-label text classification problem [26, 11].

2.1.1 *Traditional machine learning approach*

Traditional Machine Learning (ML) methods like Logistic Regression (LR), random forests, K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) have played a significant role in ICD code assignment [22].

In [28], a hierarchical SVM model was implemented to investigate codes related to cancer in death certificates. In [1], LR was used to address the task of predicting ICD-10-CM (Clinical Modification) codes from electronic health records, where ICD-10-CM is a version of the ICD-10 coding system used for diagnosis coding. The study focuses on building a scalable machine learning system that addresses challenges related to imbalanced data. Two system architectures were implemented, Convolutional Neural Networks (CNN) and LR model. CNN proved effective for high-frequency codes, while LR model demonstrated superior performance for low-frequency codes.

However, a significant challenge in utilizing traditional ML is the need for extensive medical expertise during the feature selection stage.

Despite dedicated research efforts, there was considerable potential to enhance the effectiveness and accuracy of these methods by deep learning approaches [22].

2.1.2 *Deep Learning Approaches*

Deep learning methods have proven to be more effective than traditional ML techniques, especially in feature engineering. Models such as CNN and Recurrent Neural Networks (RNN) utilize hierarchical structures to automatically extract significant features from raw data through multiple layers of nonlinear transformations. Unlike traditional methods that rely on manually crafted features, deep learning algorithms can directly learn complex patterns from raw input data [55]. This capability enhances the accuracy of tasks like text and document classification [25, 36].

The utilization of these deep neural networks has been a focal point of interest for many researchers aiming to enhance the effectiveness and accuracy of ICD code assignment [52]. The use of RNNs and their variants such as Long Short-Term Memory (LSTM) is prevalent in capturing sequential features in textual data [48]. In [3], a hierarchical LSTM was applied to transfer knowledge, intending to avoid predicting incorrect subordinate ICD codes for a given parent ICD code. In [32], CNN is used to introduce the MultiResCNN model, a convolutional neural network with multiple filters and a residual layer designed for the assignment of ICD-9-CM codes to discharge summaries, where ICD-9-CM (Clinical Modification) refers to the version of the ICD-9 coding system used for diagnosis coding. The MultiResCNN utilizes a multi-filter convolutional layer to capture diverse text patterns of different lengths, along with a residual convolutional layer to enhance the receptive field. [9] utilized dilated convolution as a mechanism to capture underlying semantic features. This technique enables each kernel to cover a wide range of information by extending the receptive field, enhancing the model's ability to perceive clinical content for accurate code assignment.

2.1.3 *Large Language Models (LLMs)*

In recent NLP research, LLMs using the transformer architecture have become widely recognized, showing excellent performance in various natural language understanding tasks [16, 35]. These models undergo pretraining on extensive text datasets, aiming for different language modelling goals. After pretraining, they are fine-tuned for specific tasks, enabling them to excel in problems like classification [16] and text generation [54].

Different transformer architectures used in LLMs such as encoder-decoder, causal decoder, and prefix decoder. The main differences in these architectures are related to the used transformer blocks and the attention mechanism. The attention mechanism prioritizes significant input tokens by assigning them greater weights, thus directing the model's focus towards them while disregarding irrelevant tokens. In transformers, attention computation involves generating query, key, and value mappings for input sequences. The attention score is derived from the multiplication of query and key, which is subsequently utilized to determine the weighting of values. different attention approaches were used in LLMs such as Self-Attention, Cross-Attention, and Sparse-Attention [43].

Various LLMs, such as BioBERT [30], BioGPT [37] and ClinicalBERT [17], have been adapted for biomedical tasks. These models are pretrained on text specific to medical domain to enhance performance in downstream tasks, reporting enhanced performance in biomedical tasks such as text mining, named entity recognition, and question answering [30].

Despite the success of LLMs in various tasks, applying them to large-scale multi-label classification, particularly in automatic ICD coding, remains a challenge. One of the challenges , beside the main challange of large set of labels, is the long input texts issue. Clinical notes length often exceed the maximum sequence length set by LLMs, leading to decreased performance upon truncation.

In [16], PLM-ICD model, that achieved state-of-the-art results, was introduced. The challenge of long clinical notes in ICD coding was addressed by efficiently handling them through text segmentation. This approach enables the extraction of contextualized representations from each segmented chunk, which are then concatenated to form a comprehensive document representation. The model employs a label-wise attention mechanism to transform the document representation into a label-specific form, enhancing its focus on relevant information for each ICD code. The final step involves predicting ICD code probabilities based on the transformed document representation.

In another study [35], The authors propose XR-LAT model, which builds upon the baseline PLM-ICD architecture. This method uses a waterfall training strategy, utilizing four sub-models with the same architecture as the baseline PLM-ICD model. It employs a four-level hierarchical code tree based on the ICD-9-CM structure (chapter, block, category, code) to provide a structured representation of labels. Knowledge transfer is facilitated through a bootstrapping method, where trained model weights from higher levels initialize the weights of lower levels, enhancing the learning process.

In another research [54], the proposed model GPsoap is pretrained to generate free-text Assessment and Plan (AP) sections from subject and object sections containing symptoms and lab results. Then, generative fine-tuning is applied to convert the task into a sequence-to-sequence problem, generating unique code descriptions. The generated code descriptions are then used to predict the corresponding ICD codes.

A recent review study [11] conducted an extensive comparative analysis of deep learning and LLM-based approaches for ICD coding. Contrary to common assumptions, the study found that reducing the length of clinical notes from 4,000 to 2,500 tokens had only a minimal impact on performance. Additionally, it introduced a new partitioning scheme for the MIMIC-III [20] and MIMIC-IV [19] datasets, where codes with fewer than ten occurrences were eliminated, the size of the test set was doubled, and multi-label stratified sampling was employed for document selection. This splitting ensured that patients were present in either the training or testing set, but not both, thereby refining data splits and standardizing evaluation benchmarks. The study reported the results of six state-of-the-art models, including PLM-ICD, LAAT, MultiResCNN, CAML, Bi-GRU, and CNN, on the new proposed splits of MIMIC-III and MIMIC-IV datasets, establishing a benchmark for future studies.

2.2 UTILIZING ICD CODE HIERARCHY FOR DISTANCE CALCULATION

The hierarchical structure of ICD codes has been a subject of interest in healthcare informatics. Previous studies have highlighted the potential benefits of incorporating this hierarchical structure into the coding process [35] [8] [49].

Some studies have investigated methods to calculate the distances between ICD codes in this hierarchical structure and using this distance to address specific issues. In [18], the distance between sets of hierarchical taxonomic clinical concepts is used to measure patient similarity.

In [15], the hierarchical distance $d(x, y)$ between two icd codes, x and y , is calculated as the minimum number of edges $p_{min}(x, y)$ normalized by the sum of their depths in the hierarchy $(l(x) + l(y))$. This measure accounts for both the hierarchy depth and the direct

connection between the two codes.
The formula is:

$$d(x,y) = \frac{p_{min}(x,y)}{l(x) + l(y)} \tag{1}$$

A special case of this formula is when the Least Common Subsumer (LCS) of two nodes x and y is the root node, then the minimum number of edges $p_{min}(x,y)$ is equal to the sum of their depths $(l(x) + l(y))$. In this scenario, the hierarchical distance $d(x,y)$ becomes 1, indicating the nodes are maximally distant.

In [13], the authors present a distance metric based on the hierarchical structure of the ICD tree to measure the distance between two ICD codes. The study is limited to the first three digits of the ICD-10 codes. The distance between two codes of the same chapter is calculated by taking the absolute difference between their numeric parts and multiplying by 0.01. For any two codes from different chapters, the distance is 1. For example, the distance between K20 and K29 is equal to (29 - 20) * 0.01 = 0.09, and the distance between K20 and I10 is equal to 1. The distance range between two ICD codes is 0 to 1, with 1 indicating maximum dissimilarity.

2.3 ADDRESSING NON-DIFFERENTIABLE LOSS FUNCTIONS IN MODEL TRAINING

Loss functions are central to neural network training, providing a quantitative measure of how well the model's predictions match the target outputs. For backpropagation, a widely used algorithm for training neural networks, to function effectively, the loss function must be differentiable [47]. Differentiability allows gradients to be computed and propagated backward through the computational graph, enabling the model to update its parameters in the direction that reduces error [47]. Non-differentiable loss functions, however, interrupt this process, preventing gradient-based optimization by breaking the computational graph [6]. This fundamental requirement is emphasized in discussions of backpropagation and gradient-based learning methods [5] [47] .

Non-differentiable loss functions present significant challenges for gradient-based optimization methods, such as backpropagation. To address this, researchers have developed several strategies to approximate these loss functions with differentiable alternatives, allowing effective model training. In [10], the authors discuss gradient approximation techniques that estimate gradients for non-differentiable functions, allowing optimization algorithms to proceed even when the exact gradient cannot be computed. Various other methods have been explored to approximate non-differentiable loss functions directly. For instance, in [40], the authors address the non-differentiability of the Contour Dice Coefficient (CDC) by training a secondary neural network to approximate it. This differentiable approximation is combined with Dice loss to improve segmentation results in medical imaging tasks. Another study, [12], demonstrates the use of differentiable proxies, where a neural network proxy replaces non-differentiable ranking metrics like Spearman correlation, enabling their use in training. Surrogate loss models are particularly valuable when dealing with loss functions that involve complex, non-smooth behavior or discrete choices. These models approximate the non-differentiable loss with a differentiable function, enabling the use of gradient-based optimization methods like backpropagation [56].

2.4 UNCERTAINTY

Given the critical and sensitive nature of ICD coding, it is essential to minimize the potential for errors. To address this challenge, research has explored uncertainty estimation techniques, which aim to enhance the interpretability and reliability of model predictions [2].

In machine learning, uncertainty is typically categorized into two types: aleatoric and epistemic. Aleatoric uncertainty, or data uncertainty, arises from inherent randomness and noise within the data itself. In contrast, epistemic uncertainty stems from limitations in the model or the insufficiency of the data. Unlike aleatoric uncertainty, epistemic uncertainty can be reduced by gathering more data or improving the model [4].

Various techniques have been developed to quantify epistemic uncertainty, including Bayesian neural networks, ensemble methods, and Monte Carlo Dropout (MCD) [4]. Each of these methods offers distinct advantages and insights into model confidence.

Among these approaches, MCD has emerged as a outstanding method for quantifying uncertainty in machine learning models [14, 23]. The foundational work by Gal and Ghahramani introduced MCD as a technique for approximating Bayesian inference in neural networks. Their research demonstrated that applying dropout not only during training but also during inference can effectively capture and represent model uncertainty through multiple stochastic forward passes [14].

The application of MCD was further extended to multi-label classification tasks. In [29], the researchers highlighted the importance of uncertainty estimates in improving the reliability of model predictions, particularly in scenarios involving multiple labels where traditional methods may fall short.

In our work, we focus on epistemic uncertainty, utilizing the MCD technique in inference phase to provide decision-makers with clear insights into the model's confidence in its ICD code predictions.

3 METHODOLOGY

In this section, we describe the methodological approach employed to develop and fine-tune a T5-based LLM (Text-To-Text Transfer Transformer) [46], integrated with the hierarchical structure of ICD-10 codes.

The methodology includes dataset description, data processing steps, the model architecture, the incorporation of the ICD-10 code hierarchy into the training process, model fine-tuning, new defined evaluation metrics and uncertainty estimation.

3.1 LLMS USED IN THIS WORK

In this work, experiments were conducted using three encoder-decoder-based generative models: T5-base (Text-to-Text Transfer Transformer), FLAN-T5-base (Fine-Tuned Language Net T5), and BART (Bidirectional and Auto-Regressive Transformer). All three models share a common encoder-decoder transformer architecture, which is widely employed for sequence-to-sequence tasks.

This architecture comprises an encoder that processes input sequences into contextual representations using self-attention mechanisms and a decoder that generates output sequences token by token. The decoder leverages both self-attention and cross-attention mechanisms to align the generated outputs with input features. Figure 1 provides a general illustration of the encoder-decoder structure, emphasizing its key components and flow of information.

Below is a detailed description of each model and its architectural features:

- T5-base: employs a unified text-to-text framework, where all NLP tasks are reformulated as text generation problems. It uses an encoder-decoder architecture, with the encoder processing input text into hidden representations and the decoder generating output sequences token by token. T5-base is pretrained on a large corpus using a denoising autoencoding objective, where random spans of input text are masked, and the model learns to reconstruct the missing spans. This versatile framework enables T5-base to perform well across a wide range of NLP tasks [46].

  The model comprises 12 layers, 768 hidden units, and 12 attention heads. T5's architecture is based on the original transformer, with the following modifications [46]:

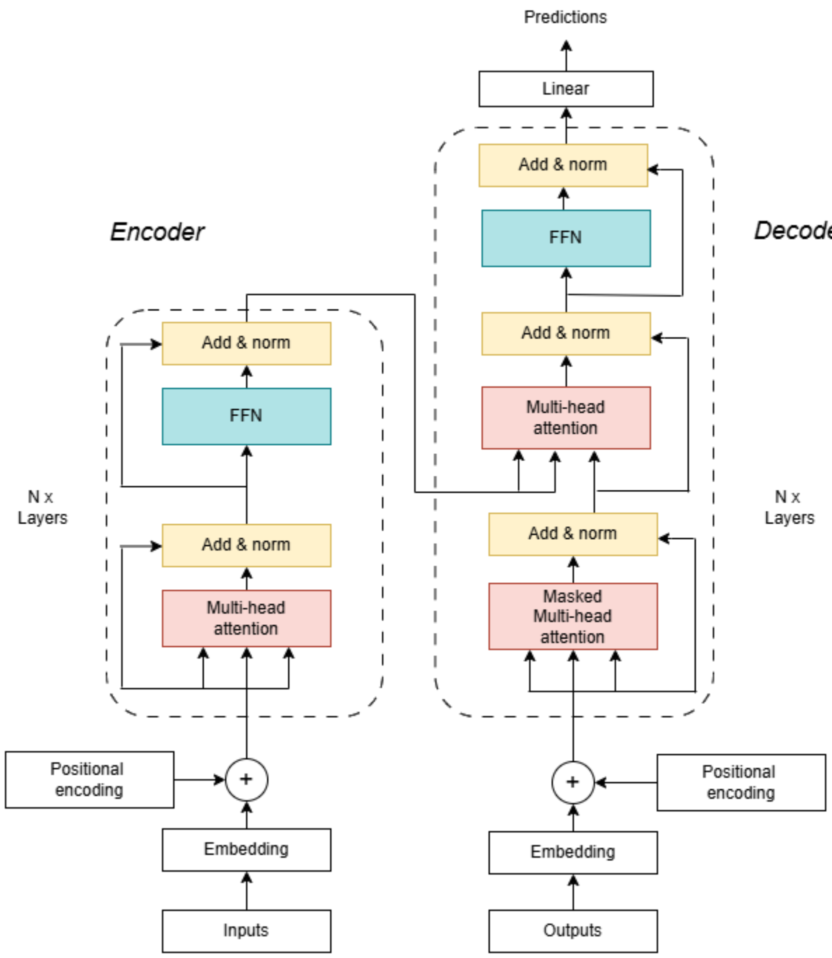


Figure 1: Encoder-decoder transformer architecture, illustrating the flow of information. Left part is the encoder, right part is the decoder. The encoder processes the input sequence into contextual representations using self-attention, while the decoder generates the output sequence token by token, leveraging both self-attention and cross-attention to align outputs with input features.

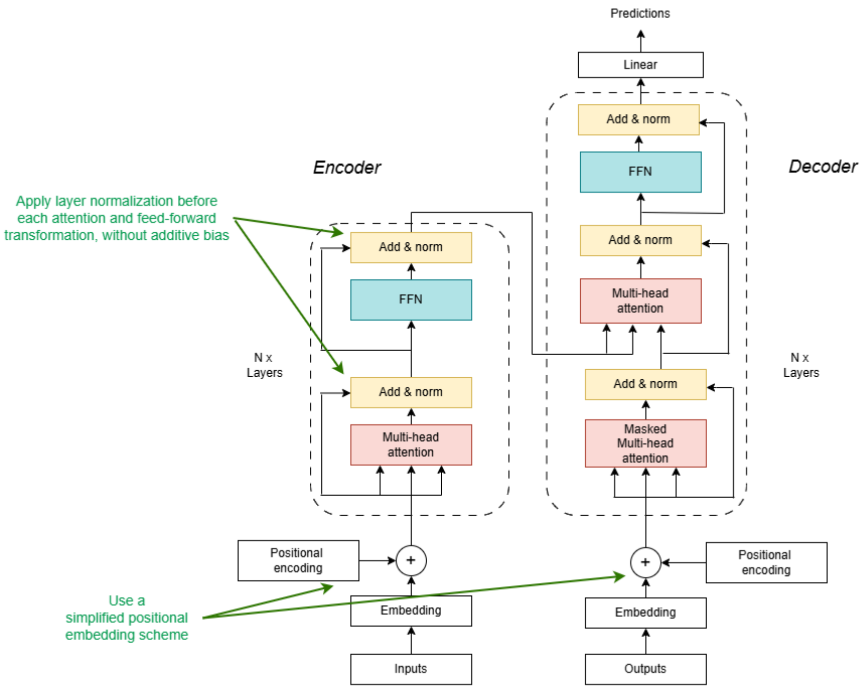


Figure 2: Modifications introduced by T5 to the encoder-decoder transformer architecture

– Layer normalization is applied immediately before each attention and feed-forward transformation, outside the residual path.

– No additive bias is used in layer normalization, relying solely on scaling.

– A simple position embedding scheme adds a scalar to the logits used for attention weight computation.

– Dropout is applied extensively throughout the network, including attention weights, feed-forward layers, and skip connections.

In Figure 2 we illustrate these modifications.

- FLAN-T5-base: builds upon the T5 architecture by incorporating instruction-tuning during pretraining. This process involves training the model on various tasks phrased as natural language instructions, such as "Summarize this paragraph" or "Translate this text," to improve its adaptability to zero-shot and few-shot learning scenarios [53]. FLAN-T5-base shares the same encoder-decoder architecture as T5-base, with 12 layers, 768 hidden units, and 12 attention heads, but its instruction-tuned pretraining makes it more effective in handling complex, instruction-driven tasks.

- BART: is another encoder-decoder model designed for text generation and comprehension tasks. Unlike the original transformer which uses a unidirectional encoder, BART's encoder is bidirectional, allowing it to understand context from both directions of the input text. The decoder operates in an autoregressive manner, generating tokens sequentially. BART is pretrained using a denoising autoencoding objective, where corrupted input sequences (e.g., with missing or shuffled tokens) are reconstructed. BART comprises 12 layers, 1024 hidden units, and 16 attention heads [31].

Each of these models leverages self-attention and cross-attention mechanisms to capture dependencies between input and output tokens. However, their pretraining objectives and fine-tuning strategies differ, influencing their adaptability to specific tasks.

3.2 DATASET

MIMIC-IV dataset is utilized in this work, which is a large, openly available dataset released on January 6th, 2023 and used in many research studies in the healthcare domain [19]. MIMIC-IV includes data from patients admitted to the emergency department or ICU between 2008 and 2019, annotated with either ICD-9 or ICD-10 codes [19].

3.3 DATA PROCESSING

Data preprocessing is a crucial step in preparing the clinical texts for model fine-tuning. In our work, data processing includes three main aspects: Clinical notes processing, ICD-10 codes processing, and data splitting.

3.3.1 *Clinical notes processing*

We preprocess the clinical notes in the MIMIC-IV dataset by converting all text to lowercase and removing words consisting of non-alphabetic characters. This preprocessing approach, originally employed in [42] for clinical notes on the MIMIC-III dataset, has since been adopted in many later studies [16, 24].

Each clinical note is prefixed with "Assign icd codes:" as explained in section 3.5. Following this, the text is tokenized using the pretrained T5 tokenizer, and the length of clinical notes is truncated to a maximum of 2,500 tokens. This token limit was chosen based on findings from a recent review study [11].

3.3.2 *ICD-10 codes processing*

For ICD-10 code processing, we started with 16,156 unique ICD-10 codes from the MIMIC-IV dataset. We filtered out codes with a frequency less than or equal to 10, which removed 10,599 codes. After filtering, 5,557 codes remained. For these codes, we ensured that all codes follow the standard format, which begins with a letter followed by numbers. This step removes any codes containing letters in positions other than the first character. After filtering, each code is truncated to a length of four characters: one letter and three digits. This length was chosen after experimentation to manage memory usage and computational requirements, as detailed in section 3.6.2.

These truncated codes are then added as special tokens to the pre-trained T5 tokenizer, as further explained in section 3.6.2. This processing resulted in a refined dataset containing 2,830 ICD-10 codes.

3.3.3 *Data splitting*

To ensure that each subset of the data (training, validation, and test sets) reflects the diversity of clinical notes and their associated labels, we employed multi-label stratified sampling for data splitting. We increased the test set size to provide more robust evaluation of model performance. To avoid data leakage, all records of each patient were assigned to either the training or testing set, but not both. This approach aligns with the steps mentioned in [11]. The final split resulted in a training set of 89,077 samples, a validation set of 13,375 samples, and a test set of 19,785 samples.

3.4 T5-BASE MODEL FOR ICD CODING

T5-Base is a versatile generative language model. It employs an encoder-decoder architecture, allowing it to treat every text-based problem as a text-to-text task. This capability enables T5-Base to adapt to a broad range of NLP tasks by framing them as text generation problems [46].

T5-Base is pre-trained on a diverse set of text-based tasks using a denoising autoencoder objective, where it learns to predict missing parts of a text sequence based on the context provided by the remaining text [46]. This pre-training provides a useful foundation for the ICD coding task, as the model's ability to understand complex text structures can be leveraged to generate ICD codes from medical narratives.

3.5 USING TASK PREFIXES IN T5 FOR ICD CODING

T5 model utilizes a text-to-text framework that distinguishes different NLP tasks using specific prefixes [46]. This approach allows the model to handle various tasks by transforming them into text generation problems, making it versatile with a single architecture [33]. While other LLMs also use prompt engineering to guide their responses [7], the prefix-based approach is particularly integral to T5's design [34].

For our work on ICD coding, we used the prefix "Assign ICD codes:" with each clinical note to clearly communicate the task to the model. This adaptation during fine-tuning allows the model to learn to interpret the clinical input and generate the corresponding ICD-10 code as output. We experimented with different prefixes and found that "Assign ICD codes:" was the most effective in guiding the model to perform the task accurately.

3.6 INCORPORATING THE ICD-10 CODE HIERARCHY INTO THE TRAINING PROCESS

By leveraging the relationships between broader disease categories and specific diagnoses, we aim to guide the model to generate more relevant predictions. In this section, we propose a novel loss function that complement cross-entropy loss with a distance-based loss as in Eq. 7. The distance-based loss penalizes the model based on how far the predicted ICD code is from the ground truth in the ICD-10 hierarchy, encouraging the model to prioritize predictions that are reasonably close to the true labels.

3.6.1 *Define a Distance-based Loss Function*

The key idea of this work is to integrate the ICD-10 code hierarchy into the training process. Our approach aims to discourage the model from predicting codes that are completely irrelevant to the clinical note by assigning a higher error penalty for such predictions.

For example, if we have a clinical note with two ground truth codes: E11.9 (Diabetes without complications) and I20.0 (Cardiovascular disease), and let us assume that the model predicts two codes: E11.2 (a diabetes code related to kidney complications) and J18.9 (Respiratory infection). The evaluation of these predictions would be handled as follows: With only cross-entropy loss, the model seeks to minimize the difference between the predicted probability distribution and the true distribution of the correct codes. In this scenario, the first predicted code (E11.2) is related to diabetes, but it incorrectly indicates kidney complications instead of diabetes without complications (E11.9). While this prediction is incorrect, it's still somewhat relevant to the broader context of diabetes. On the other hand, the second predicted code (J18.9) corresponds to a respiratory infection, which is entirely unrelated to the ground truth codes for diabetes or cardiovascular disease. This prediction is not only incorrect but also clinically irrelevant to the patient's condition, such mispredictions may impact diagnosis and treatment.

Using cross-entropy loss alone may penalize both misclassifications equally, without considering the clinical relevance of each prediction. However, by incorporating the hierarchical structure of ICD-10 codes into the training process, we can help differentiate between misclassifications that are clinically relevant and those that are not.

To address this, we propose a new loss function that complements cross-entropy loss with a distance-based loss. This combined loss function penalizes the model based on both the accuracy of the wrongly predicted ICD codes and the distance between the predicted and actual codes within the ICD-10 hierarchy.

Our distance-based loss function is designed to calculate the vertical penalty, which applies when there are discrepancies between different hierarchical levels (e.g., chapter, block, third level). These discrepancies are important in the clinical context because higher-level codes, such as chapters or blocks, correspond to broader diagnostic categories. A prediction that is distant from the true higher level is considered more clinically irrelevant, and thus penalized more heavily to encourage the model to focus on clinically relevant codes.

All horizontal differences at the same hierarchical level do not impact the distance metric (penalty). For instance, if the ground truth code is 'I21.1' the penalty is the same for predictions such as 'I21.0' and 'I21.9' as they belong to the same hierarchical level.

Our proposed distance-based loss function calculates the overall distance between incorrectly predicted ICD codes and the ground truth codes. The distance loss is computed as the sum of these distances, which are measured based on how much each predicted code far from the set of ground truth codes.

The proposed distance-based loss function is defined as:

$$\text{Distance Loss} = \sum_{i=1}^{M} \min\left(\text{distance}(\hat{c}^{i}, C)\right) \tag{2}$$

where M is the number of predicted ICD codes, $\hat{c}^i$ is the i-th predicted ICD code, and C is the set of ground truth codes.

The distance metric is computed based on the hierarchical structure of ICD codes, where penalties are assigned according to the first differing character between two codes. This metric is defined as:

$$\text{distance}(C_1, C_2) = \begin{cases} \lambda^{-(i+1)} & \text{if } i = \min\{k : c_1(k) \neq c_2(k)\}, \\ 0 & \text{if } c_1(k) = c_2(k) \quad \forall k \in \{1, 2, \dots, n\}. \end{cases} \tag{3}$$

Here, C_1 and C_2 represent the ICD codes being compared, n is the total number of hierarchical levels, and $c_k(C)$ extracts the character at level k from code C. The parameter λ is a scaling factor that determines the magnitude of the penalty at each hierarchical level.

This formulation ensures that the penalty is determined by the first differing character, with higher-level differences (e.g., at the chapter or block level) incurring larger penalties due to the decay factor $\lambda^{-(i+1)}$. In contrast, if all characters at every hierarchical level match, the distance is 0. Additionally, all horizontal differences at the same hierarchical level do not impact the distance metric, as the binary penalty only considers whether characters match or not, regardless of their values.

For example, consider two ICD codes $C_1 = 'A12.3'$ and $C_2 = 'B12.3'$, with a scaling factor $\lambda = 10$. The codes differ at the first hierarchical level because the first characters ('A' and 'B') are not the same. Thus, the penalty is computed as:

$$\text{distance}(C_1, C_2) = \lambda^{-(1+1)} = \lambda^{-2} = 10^{-2} = 0.01.$$

Now consider another pair $C_3 = 'A12.3'$ and $C_4 = 'A23.3'$. Here, the codes match at the first level ('A') but differ at the second level ('1' and '2'). The penalty is:

$$\text{distance}(C_3, C_4) = \lambda^{-(2+1)} = \lambda^{-3} = 10^{-3} = 0.001.$$

Finally, if two codes $C_5 = 'A12.3'$ and $C_6 = 'A12.3'$ are identical, the distance is:

$$\text{distance}(C_5, C_6) = 0.$$

One of the major challenges in this approach was ensuring that the distance loss is differentiable, a crucial requirement for maintaining smooth gradient flow during model training [27]. The proposed distance loss function in Eq.2 is not differentiable because the minimum function used to calculate the distance loss is non-differentiable. Additionally, the distance metric (penalty) proposed in Eq.3, which is a binary penalty, is also non-differentiable.

To address this, we replace the traditional minimum function with a smooth minimum function, which provides a differentiable approximation. The smooth minimum function is defined as:

$$\text{smooth\_min}(x, \beta) = -\frac{1}{\beta} \log \left(\sum_i \exp(-\beta x_i) \right) \tag{4}$$

where x represents the set of distance metrics between the predicted ICD code $\hat{c}_i$ and each of the ground truth codes in C, and β is a parameter that controls the smoothness of the approximation.

Therefore, we apply the smooth minimum function to these distances, as shown in the following equation:

$$\text{Distance Loss} = \sum_{i=1}^{M} \text{smooth\_min}(\text{distance}(\hat{c}_i, C)) \tag{5}$$

where M is the number of predicted ICD codes, N is the number of ground truth codes, $\hat{c}_i$ is the i-th predicted ICD code, and C represents the set of ground truth codes.

For the distance metric, we employ a differentiable approximation that compares the corresponding digits of two ICD codes. The distance metric between two ICD codes is calculated by comparing their numeric parts digit by digit (in section 3.6.2, we explain how the first letters of ICD-10 codes are converted to digits).

The distance metric is defined as:

$$\text{distance}(C_1, C_2) = \sum_{k=1}^{L} \left(\lambda^{-(k+1)} \left(\frac{1}{1 + \exp\left(-|c_1(k) - c_2(k)|\right)} - 0.5 \right) \right) \tag{6}$$

where L is the length of the longer ICD code, $c_1(k)$ and $c_2(k)$ represent the k-th digits of ICD codes C_1 and C_2, λ is the scaling factor that decays with each subsequent digit.

To ensure the loss function remains a differentiable and smooth approximation of the originally proposed distance-based loss, this formulation introduces unavoidable horizontal differences. While it provides an upper bound for the proposed loss (as shown in Table 1), optimizing this formulation inherently optimizes the original loss as well. The penalty for horizontal differences (i.e., differences between corresponding digits at the same hierarchical level) is designed to be significantly lower than the penalty for vertical differences (i.e., differences across different hierarchical levels). By leveraging the decay factor λ, the formulation ensures that differences at higher hierarchical levels (e.g., top-level digits) incur a much larger penalty, aligning with the hierarchical structure of the ICD codes. This smooth approximation allows penalties to increase continuously with the magnitude of the difference while maintaining differentiability for all comparisons.

For example, when calculating the distance (reflecting the penalty) between the ground truth code "Z990" and the codes "R990", "M990", "Z190", "Z490", "Z910", and "Z940", with $\lambda = 10$, we obtain the following results in tabel 1

Table 1: Distance penalties for predicted ICD codes compared to the ground truth code "Z990", using both the original non-differentiable and the differentiable approximated distance metrics.

Code	**Level**	**Appx. distance Eq.6**	**Original distance Eq.3**
R990	Chapter	0.0231	0.01
M990	Chapter	0.0231	0.01
Z190	Block	0.0050	0.001
Z490	Block	0.0049	0.001
Z910	Third digit	0.0005	0.0001
Z940	Third digit	0.0005	0.0001

From the results of the distances (Appx. penalty) in the table, we can observe the following:

- Smooth penalty differences at the same level: For codes differing at the same level (e.g., "R990" vs. "M990", differing in chapter level), the penalties are very similar, both being 0.0231. Similarly, for differences in the block level (e.g., "Z190" and "Z490"), the penalties are also very close, with values of 0.0050 and 0.0049, respectively.

- Significant penalty differences across levels: When comparing penalties for differences across different levels, there is a noticeable difference. For instance, the penalties for differences at the chapter level (0.0231 for "R990" and "M990") are significantly higher than those at the third digit level (0.0005 for "Z910" and

"Z940"). This indicates that discrepancies at higher hierarchical levels result in greater penalties.

Thus, penalties increase smoothly for differences at the same level but show considerable variation when comparing differences across different levels. This approach aligns with our goal of encouraging the model to focus on accurately predicting the broader, clinically significant categories (higher-level characters) before refining its focus on the more specific details (lower-level characters).

The final Loss function combines the Cross-Entropy Loss and Distance Loss:

$$\text{Loss} = \text{Cross-Entropy Loss} + \alpha \cdot \text{Distance Loss} \tag{7}$$

Where α is a hyperparameter that balances the contribution of the distance loss relative to the cross entropy loss and it requires fine-tuning.

By incorporating this hierarchical distance-based penalty into our loss function, we aim to enhance the model's ability to generate clinically relevant ICD codes and reduce the prediction of irrelevant codes.

3.6.2 *Overcoming the Challenge of Decoding Model Output*

Fine-tuning T5 for ICD coding typically does not require decoding the model's output. However, since we are using a distance-based loss function that requires comparing the predicted ICD-10 codes with the ground truth codes, decoding the output became necessary. This additional step of decoding disrupted the computation graph and interrupted the gradient flow during model training.

To address this issue, we propose a method for encoding ICD-10 codes in a way that represents the code itself. We convert the first letter of each ICD-10 code into a predefined numeric value while leaving the numerical part of the code unchanged. Then, we add these converted codes to the tokenizer's vocabulary as special tokens. Each code was assigned a unique index reflecting the code itself.

Increasing the tokenizer's vocabulary size results in larger embedding layers, which, in turn, increases memory consumption and computational demands. As a result, we limited the ICD-10 codes to a length of four characters. The original size of our T5 tokenizer was 32,100, and after adding the converted codes as special tokens, the vocabulary size increased to 58,998. This was the maximum size we could use given the available resources.

This encoding approach allows us to tokenize ICD-10 codes with indexes that represent the codes, thereby eliminating the need for decoding during model training. For example, the codes "B129" and "B172" are encoded as "34129" and "34172", respectively, where "34" corresponds to the numeric mapping for 'B', and "129" and "172" remain unchanged. By using these indexes, we can compare predicted and ground truth codes directly and measure the distance between them without the need for decoding.

The suggested direct tokenization facilitates the comparison of predicted and ground truth codes by their indexes, thus avoiding the decoding step. Preserving the computation graph with this approach ensures a smooth gradient flow during model fine-tuning.

3.7 FINE-TUNING T5 FOR ICD CODING

To adapt T5 model for ICD coding, we fine-tune it on the MIMIC dataset, which contains clinical notes annotated with ICD-10 codes, as described in section 3.2. To evaluate the impact of our proposed loss function, we fine-tune both the original T5-Base model and our modified model, referred to as T5-HI (T5 with Hierarchical Information). This allows us to compare the performance of both models on the ICD coding task. Fine-tuning optimizes the models' weights, enhancing their ability to predict ICD codes based on clinical notes and providing a clear evaluation of how the new loss function improves performance.

3.8 EVALUATION METRICS

In this section, we introduce the evaluation metrics used to assess the performance of our ICD code prediction model. We use several metrics include F1 micro score, TD (Total Distance) and evaluations at three hierarchical levels of ICD codes: the chapter level, the block level, and the third level, corresponding to the first, second, and third characters of the ICD code, respectively. These metrics capture the model's accuracy across broad and specific categories, providing a comprehensive view of its performance.

3.8.1 *Total Distance (TD)*

TD represents the cumulative measure of the distance between predicted ICD codes and their corresponding ground truth codes. This metric provides an overall assessment of the alignment between predicted and actual ICD codes. A lower TD indicates that the predictions are closer to the true ICD codes, reflecting better performance. By evaluating TD, we gain insights into the overall accuracy of the model in predicting ICD codes, taking into account the closeness of predictions to the true values.

3.8.2 *Irrelevant ICD Chapter (IIC)*

The chapter level, represented by the first character of the ICD code, categorizes diseases into broad groups. IIC measures how often the predicted codes belongs to an incorrect disease category, indicating incorrect chapter-level predictions. IIC occurs when the first character (chapter) of the predicted ICD code does not match the chapter of the ground truth code. This metric helps evaluate how frequently the model misclassifies conditions into entirely incorrect categories.

3.8.3 *Irrelevant ICD Block (IIB)*

The block level, indicated by the second character of the ICD code, provides a more specific classification within each chapter. Irrelevant ICD Block (IIB) measures the number of instances where the predicted block does not match the ground truth block, even though the chapter is correctly predicted. This metric evaluates the model's precision in recognizing specific subcategories within the broader chapter.

3.8.4 *Irrelevant ICD Third-Level (IIT)*

The third level, represented by the third character of the ICD code, provides a detailed classification within a block. Irrelevant ICD Third-Level (IIT) captures cases where the predicted third-level code is incorrect, even though both the chapter and block are predicted correctly. This metric evaluates the model's ability to make specific predictions within a specific category.

3.9 MODEL UNCERTAINTY ESTIMATION

We employed the Monte Carlo Dropout (MCD) method to assess the uncertainty in our model's predictions. This method involves generating multiple predictions for the same input and analyzing the variability in the predictions to quantify uncertainty [14].

This method introduces stochasticity during inference by keeping dropout layers active even in evaluation mode. By performing multiple forward passes with dropout enabled, the model generates diverse probability distributions, allowing us to quantify uncertainty.

For each label, uncertainty was calculated based on the variability of the predicted probabilities across multiple forward passes. Specifically, we computed the mean probability and the variance for each label as follows:

Mean Probability (μ_l):

$$\mu_l = \frac{1}{N} \sum_{i=1}^{N} p_l^{(i)} \tag{8}$$

where $p_l^{(i)}$ is the predicted probability for label l in the i-th forward pass, and N is the total number of passes.

Uncertainty = Variance (σ_l^2):

$$\sigma_l^2 = \tfrac{1}{N} \textstyle\sum_{i=1}^{N} (p_l^{(i)} - \mu_l)^2 \tag{9}$$

Suppose we have four possible labels, and the model is run twice with MCD. The probabilities predicted for each label across the two runs are:

- Run 1: [0.1, 0.6, 0.8, 0.2]
- Run 2: [0.3, 0.9, 0.6, 0.1]

For label 1:

$$\mu_1 = \frac{0.1 + 0.3}{2} = 0.2, \quad \sigma_1^2 = \frac{(0.1 - 0.2)^2 + (0.3 - 0.2)^2}{2} = 0.02$$

For label 2:

$$\mu_2 = \frac{0.6 + 0.9}{2} = 0.75, \quad \sigma_2^2 = \frac{(0.6 - 0.75)^2 + (0.9 - 0.75)^2}{2} = 0.045$$

Higher variance indicates greater uncertainty, as seen for label 2, compared to label 1.

The variance reflects the spread of probabilities, indicating the model's confidence in its predictions. A higher variance suggests greater uncertainty, while a lower variance implies higher confidence. For overall uncertainty, variances across all possible labels can be aggregated.

It is important to note that the ability of MCD to provide meaningful uncertainty estimates relies significantly on the model's generalization capability. If the model is biased or overconfident, it may assign low uncertainty values to incorrect predictions, as illustrated in the above example, where incorrect predictions for label 2 still showed relatively low uncertainty. This limitation underscores the need for robust model training and careful evaluation of the uncertainty estimates, particularly for underrepresented labels or complex input samples.

By incorporating this probabilistic approach, MCD enables a more nuanced understanding of the model's confidence. For instance, predictions with low variance (e.g., label 1 in the example above) can generally be trusted more, while predictions with high variance (e.g., label 2) may require human oversight.

The reliability of MCD for estimating uncertainty depends significantly on the model's generalization ability. While MCD introduces stochasticity during inference to quantify epistemic uncertainty, its effectiveness is influenced by how well the model represents the underlying data distribution. A biased or overconfident model may produce low uncertainty values for incorrect predictions, leading to a false sense of confidence in those outputs. This limitation highlights the need for robust model training to reduce biases and improve generalization, ensuring that uncertainty estimates accurately reflect the model's confidence. In scenarios where the model overfits to frequent labels or fails to generalize well, the uncertainty estimates may not reliably identify errors, emphasizing the importance of critical evaluation when interpreting these values in practice.

3.10 EXPERIMENTAL SETTING

To implement and evaluate our custom loss function, we designed a custom training loop for all experiments instead of using the Hugging Face Trainer. This custom training loop was also used for experiments that involve only cross-entropy loss to ensure consistency across all experiments. The learning rate for T5-base and FLAN-T5-base was set to 0.0001, while for models using LoRA with T5-base and BART the learning rate set to 0.001 . A weight decay of 0.01 was applied consistently across all models. These hyperparameter values were determined after conducting some experiments to identify the optimal settings for performance. All models were trained for a maximum of 25 epochs. Early stopping was applied with a patience of 3 epochs. In our experiments, due to GPU memory limitations, we used a batch size of 1 for training. To effectively simulate a larger batch size and ensure stable gradient updates, we employed gradient accumulation with 4 accumulation steps. This approach allowed us to achieve an effective batch size of 4 while staying within the memory constraints of the hardware.

4 RESULTS

In this section, we present the results of our experiments using the MIMIC-IV v2.2 dataset. The findings are organized into three main areas: the results of data analysis, the performance of the fine-tuned T5 models, and the uncertainty estimation using MCD method.

4.1 RESULTS OF DATA ANALYSIS

In this section, we present an analysis of the MIMIC-IV-ICD-10 dataset, focusing on various aspects of the data. The dataset contains *diagnoses* coded according to ICD-10.

Tabel 2 provides some statistics on the MIMIC-IV-ICD-10 dataset.

Table 2: MIMIC-IV-ICD-10 Dataset Statistics

MIMIC-IV-ICD-10	Value
Unique ICD-10 Codes	16,156
Patients	80,213
Clinical Notes	122,300
Min. / Median / Avg. / Max. words per clinical note	62 / 1492 / 1597.37 / 7754
Min. / Avg. / Max. codes per clinical note	1 / 14.44 / 39

In Figure 3, we illustrate the frequency distribution of the top 50 ICD-10 codes, with a few codes like E785 and I10 occurring far more frequently than others. This imbalance poses a challenge for models, as they may become biased toward predicting the more common codes, making it harder to correctly predict less frequent.

In Figure 4, we show the distribution of the number of ICD-10 codes assigned to clinical notes. The wide variation in the count of assigned codes, ranging from fewer than 5 to over 35, highlights the complexity of the task. This variation increases the challenge for models, as they need to handle both short and long code sequences, making the problem more harder to generalize across notes with different code lengths.

Figure 5 presents the text length distribution of clinical notes, indicating that most notes fall within the 500 to 3000-word range.

We analyzed the correlation between the word count of clinical notes and the number of ICD-10 codes assigned. The correlation analysis yielded a moderate positive correlation of 0.499, indicating that longer clinical notes tend to have more assigned codes. This trend is

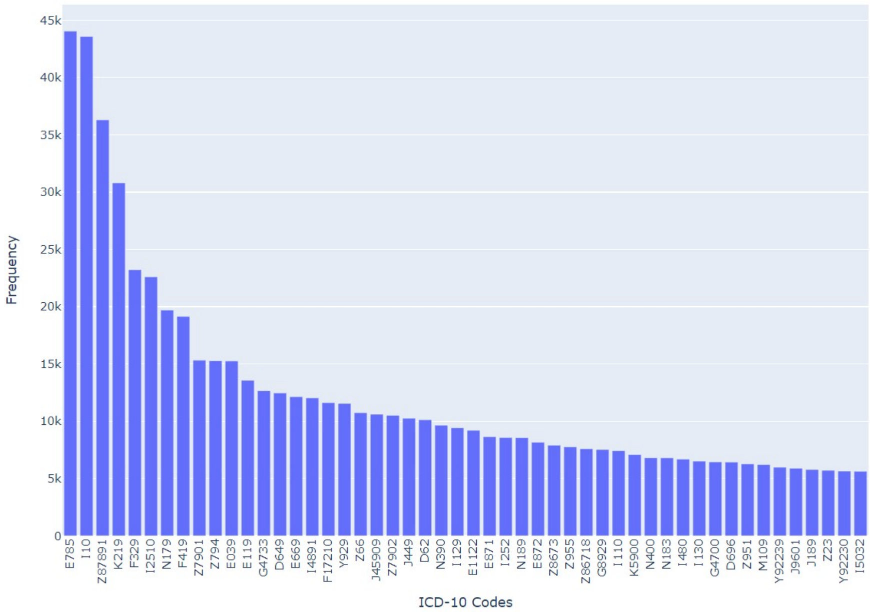


Figure 3: Frequency distribution of the top 50 ICD-10 codes

further illustrated in Figure 6, which displays a box plot of ICD-10 codes across different word count ranges. As shown, shorter clinical notes (0-500 words) generally have fewer assigned codes, while longer notes (3000-4000 words) are associated with a higher number of codes, indicating that as the word count increases, the median number of assigned ICD-10 codes also rises.

4.2 EXPERIMENTAL RESULTS

This section presents the results of our experiments designed to evaluate the performance of various approaches to ICD coding. Specifically,

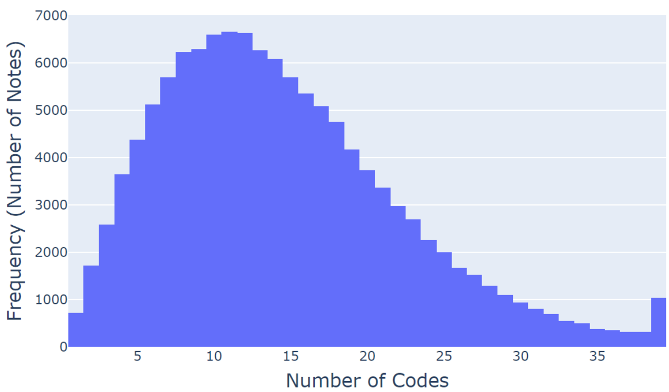


Figure 4: Distribution of the counts of ICD-10 codes in the clinical notes

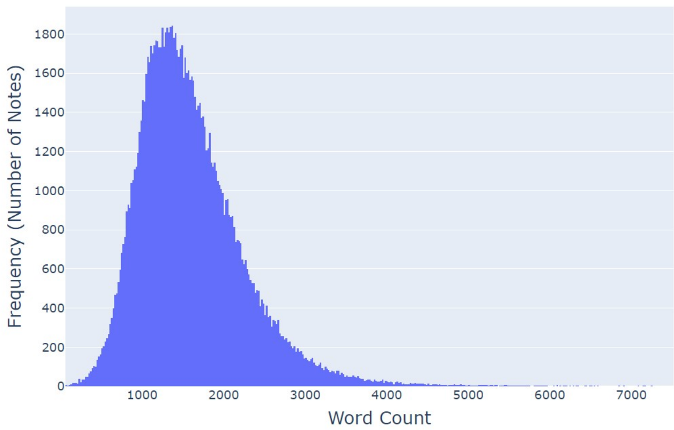


Figure 5: Text length distribution of clinical notes in the MIMIC-IV-ICD-10 dataset. The histogram illustrates the frequency of clinical notes categorized by word count

we compare the efficacy of different LLMs and assess the impact of input length and different chunking strategies on the accuracy and efficiency of ICD coding predictions. Additionally, we explore the effects of using differentiable versus non-differentiable distance-based loss functions on model training and performance.

4.2.1 *Comparison of LLMs and Input Lengths for ICD Coding*

In this work, we incorporate the hierarchical structure of ICD codes into a T5-base model. The decision to use T5-base was guided by a series of experiments where we fine-tuned different LLMs for ICD coding. These experiments explored various input lengths: 1024, 2048, and 2500 tokens. Importantly, we used the full length of ICD codes without truncation or preprocessing. Due to resource limitations and the larger parameter count of BART compared to T5, we fine-tuned BART only for an input length of 1024. To overcome this limitation, we employed LoRA (Low-Rank Adaptation) to fine-tune both T5-base and BART for an input length of 2500 tokens.

The results in table 3, measured by F1 score, demonstrated that T5-base with an input length of 2500 tokens achieved the best performance. Based on this outcome, we selected T5-base with an input length of 2500 tokens for further experiments to incorporate the hierarchical structure of ICD codes.

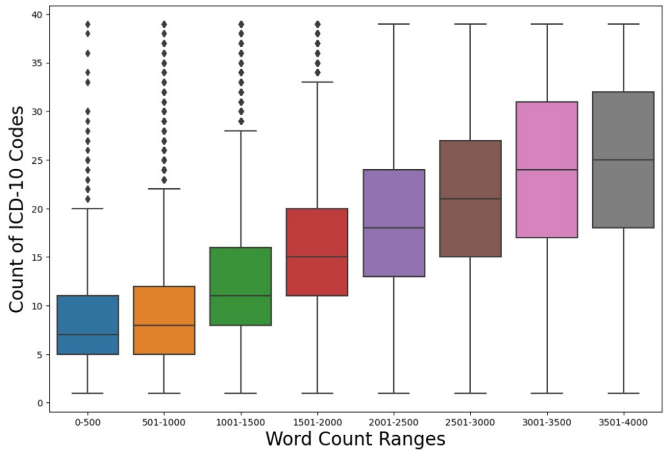


Figure 6: Box plot of ICD-10 codes assigned across different word count ranges. The plot shows the distribution of ICD-10 codes assigned to clinical notes

Table 3: Performance comparison of different LLMs and configurations for ICD coding. T5-base with an input length of 2500 achieved the highest F1 score

MODEL	INPUT LENGTH	F1 SCORE
T5-base	2500	**0.473**
T5-base	2048	0.46
T5-base	1024	0.397
Flan-T5-base	2500	0.462
LoRA-T5-base	2500	0.457
LoRA-BART	2500	0.443

4.2.2 *Comparison of Different Chunking Strategies for ICD Coding*

To further evaluate the impact of text chunking on the performance of the model, we conducted two additional experiments using the T5-base model with a fixed input length of 2500 tokens. This setting was selected because T5-base with 2500 tokens achieved the best performance, as shown in 3, where the input included the first 2500 tokens of the clinical note.

Building upon this, we explored how different sections of the clinical note might affect the model's ICD coding accuracy. In the first additional experiment, we used the last 2500 tokens of the clinical note, while in the second experiment, we selected the middle 2500 tokens. Both experiments involved truncating or padding the notes to maintain the 2500-token length. For clarification, if a clinical note is shorter than 2500 tokens, it is padded to reach the required length of 2500 tokens in all three cases. If the clinical note is longer than 2500 tokens, for example, 3000 tokens, different truncation strategies are applied based on the chunking method. For the first chunk, the first 2500 tokens are retained, and the remaining 500 tokens at the end are truncated. For the last chunk, the last 2500 tokens are retained, and the initial 500 tokens are truncated. For the middle chunk, 250 tokens are truncated from both the beginning and the end, leaving the middle 2500 tokens intact.

As shown in Table 4, the middle chunk achieved the highest performance, as measured by F1 score. Consequently, we used the middle chunk of the clinical notes in all subsequent experiments.

Table 4: Performance comparison of different chunking strategies for ICD coding using T5-base with an input length of 2500 tokens.

CHUNKING STRATEGY	F1 SCORE
First 2500 Tokens	0.473
Last 2500 Tokens	0.476
Middle 2500 Tokens	**0.481**

4.2.3 *Comparison of Differentiable and Non-Differentiable Distance-Based Loss Functions*

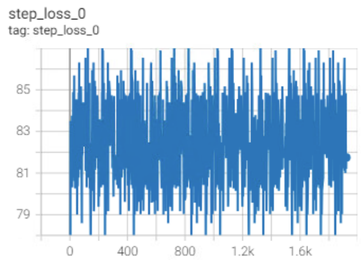
In this experiment, we evaluate the impact of using a differentiable versus a non-differentiable distance-based loss function during model training. As previously discussed in section 3.6.1, one of the primary challenges in integrating a distance-based loss into the training process is ensuring that the loss function remains differentiable, as non-differentiable functions can disrupt the gradient flow and hinder model optimization.

To assess this, we conducted two training runs: one using a differentiable surrogate for the distance metric and another using the original non-differentiable distance metric. For both experiments, ICD codes were truncated to a length of 4 and the scaling factor λ was fixed at 10 across all hierarchical levels.

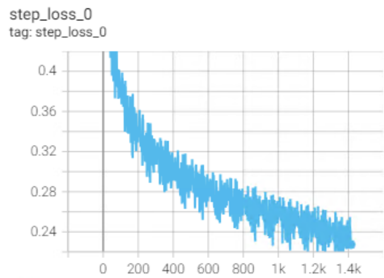
- Non-differentiable distance-dased loss: In this experiment, we used the original proposed distance loss (Eq. 2), where the minimum function is non-differentiable, along with a non-differentiable distance metric (Eq. 3).
- Differentiable distance-based loss: In this experiment, we used a differentiable approximation of both the distance metric (Eq.

6) and the minimum function (Eq. 5). Specifically, we replaced the non-differentiable distance metric with a differentiable one and applied a smooth minimum function to approximate the non-differentiable minimum operation.

The training curves for both configurations are shown in Figure 7. The model trained with the non-differentiable distance-based loss (Figure 7a) exhibited significant fluctuations in the training curve, reflecting inconsistent gradient updates and unstable optimization. In contrast, the model trained with the differentiable distance-based loss (Figure 7b) displayed a smooth and steady decrease in loss across training steps, indicating stable convergence and effective optimization.



(a) Training curve with non-differentiable distance-based loss.



(b) Training curve with differentiable distance-based loss.

Figure 7: Training dynamics with non-differentiable and differentiable distance-based loss functions.

This experiment underscores the importance of using a differentiable loss function for training, as it ensures smooth gradient flow, allowing the model to learn more effectively. The fluctuations observed with the non-differentiable distance-based loss suggest that, without smooth approximations, the optimization process becomes erratic, potentially leading to poor performance.

These results provide empirical evidence supporting the use of differentiable distance-based loss functions to maintain smooth gradient flow and enhance training effectiveness.

4.2.4 *Experiments with Distance-Based Loss Function Alone*

As part of our experiments, we trained the model using only the distance-based loss function (Eq. 5), excluding the cross-entropy loss. In this setup, the model failed to predict any ICD-10 codes. This behavior arises from the objective of the distance-based loss function, which minimizes differences between predicted and ground truth labels. When no labels are predicted, the loss evaluates to zero, as there are no discrepancies to penalize. Without cross-entropy loss to encourage accurate label predictions, the model defaults to the trivial solution of predicting nothing to minimize the loss.

In a subsequent experiment, we modified the loss function to impose a significant penalty when the model failed to predict any ICD-10 codes. This adjustment led the model to predict exactly one code per input. This behavior reflects the model's attempt to avoid the penalty for no predictions while minimizing the risk of additional loss from incorrect predictions. By selecting a single code, the model achieved a minimal-effort compromise, satisfying the penalty constraint without optimizing for accuracy or completeness.

These results highlight the necessity of combining the distance-based loss with cross-entropy loss to achieve effective learning and accurate predictions.

4.3 RESULTS OF FINE-TUNING T5 ON ICD CODING

We compared the performance of T5-base model and the hierarchically informed T5-HI model using several metrics, as described in section 3.8. Our data was split into training, validation, and testing sets, with sizes of 89,077, 13,375, and 19,785 samples, respectively, as mentioned in section 3.3.3.

For T5-HI model, we set the hyperparameter α for the loss function (Eq. 7) and the decay factor λ for the distance metric (Eq. 6) to 5 and 10, respectively. These values were selected through manual hyperparameter tuning on the validation set, as grid search was impractical due to the long training times. We tested λ values of 4, 5, and 10, and α values of 1, 5, 7, and 10. The chosen combination provided the best balance between minimizing TD and achieving a high F1 micro score. The final model performances, reported in this work, were evaluated on the unseen test set. The criteria for selecting the best hyperparameters were based on optimizing two primary metrics: TD and F1 micro score.

To calculate the Confidence Intervals (CI) for each metrics, we used a bootstrapping method. We resampled the entire dataset with replacement for 100 iterations and calculated the respective metric for each resampled dataset. The confidence intervals were then determined by computing the 5th and 95th percentiles of the bootstrap distributions.

Table 5 represents the results. These results indicate that although the standard T5-base model achieves a statistically significant higher

F1 micro score (p-value < 0.001), T5-HI model shows notable improvements in other metrics. Specifically, T5-HI has a lower TD, suggesting better alignment with ground truth codes. Furthermore, T5-HI shows fewer irrelevant chapter, block, and third-level predictions compared to T5-base. This suggests that T5-HI makes more accurate predictions across the hierarchical levels, from broad chapter classifications to detailed third-level codes.

Table 5: Results of comparison between the T5-base and T5-HI models on the test dataset. While T5-base achieves a higher F1 score, T5-HI demonstrates better performance with lower TD and fewer irrelevant predictions across other metrics.

METRIC	T5-BASE	T5-HI
F1 micro (95% CI)	**0.487 (0.484, 0.492)**	0.471 (0.467, 0.474)
TD	7,270	**4,950**
Mean TD ((95% CI)	0.36 (0.26,0.45)	**0.25 (0.19, 0,34)**
IIC	16,267	**13,480**
Mean IIC ((95% CI)	0.82 (0.58, 1.16)	**0.68 (0.42, 0.96)**
IIB	27,548	**22,202**
Mean IIB ((95% CI)	1.39 (0.98, 1.86)	**1.12 (0.74, 1.56)**
IIT	18,950	**16,237**
Mean IIT ((95% CI)	0.96 (0.62, 1.34)	**0.82 (0.52, 1.18)**

Figure 8 illustrates the cumulative distances between predicted ICD codes and ground truth codes for each clinical note in the testing dataset for both the T5-base and T5-HI models. Notably, the predicted codes of the T5-HI model are closer to the ground truth, as shown by the lower distance curve compared to the T5-base model.

Additional analysis of ICD chapter-level predictions is detailed in Table 6, which presents the F1 scores for each chapter as predicted by both T5-HI and T5-base models. CI for the F1 scores were estimated using a bootstrapping approach, where we resampled the dataset 100 times with replacement.

Overall, T5-HI model demonstrates higher F1 scores in several chapters compared to the T5-base model, which may be attributed to its hierarchical structure enhancing its predictive capability.

The F1 scores also reflect the frequency of the chapters in the training dataset. Chapters with higher frequencies generally show better F1 scores, as seen in the strong performance of chapters such as E, I, and C. In contrast, chapter P, which has a low frequency in the training dataset (only 22 cases), exhibits an F1 score of 0. This suggests that

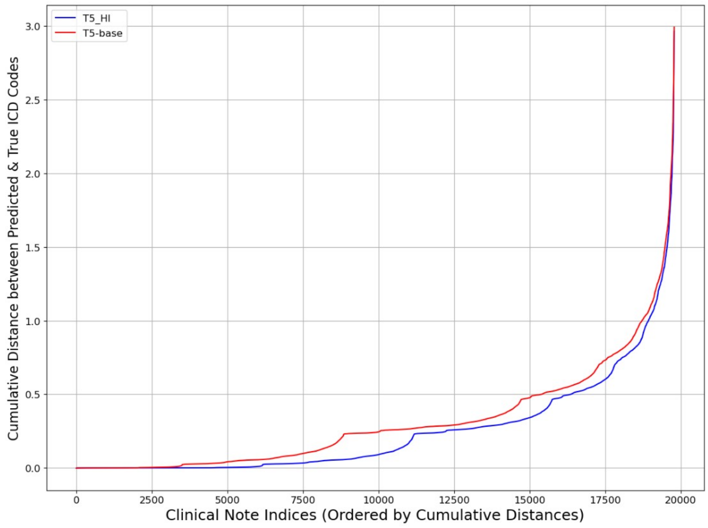


Figure 8: This figure illustrates the cumulative distances between predicted and ground truth ICD codes for clinical notes in the testing dataset. The x-axis displays the sorted indices of clinical notes, ordered by their cumulative minimum distances, while the y-axis represents the cumulative minimum distance between predicted and ground truth ICD codes. Lower cumulative distances indicate closer matches to the ground truth codes, highlighting the performance differences between the two models.

chapter P is underrepresented, leading to the model's inability to predict codes associated with it. Chapter S is also not represented in the predictions. Upon investigation, we found that the ICD codes from chapter S in our dataset contained letters in positions other than the first character. Due to our preprocessing step in section 3.3.2, which ensures all ICD-10 codes follow the standard format (beginning with a letter followed by numbers), any codes containing letters beyond the first position were removed. As a result, all the codes from chapter S were excluded from the dataset, explaining their absence in the predictions.

In Figure 9, we illustrate the counts of irrelevant ICD chapter predictions for both models. The results indicate that T5-HI predicts fewer codes with irrelevant chapters compared to T5-base. Notably, chapter P was not presented in the figure, indicating that the models did not predict any codes for this chapter due to its extremely low frequency in the training dataset. which prevented the models from learning this chapter.

Figures 10, 11, and 12 compare the distributions of irrelevant chapter (IIC), block (IIB), and third-level (IIT) codes predicted by both models, respectively. Each figure illustrates that the T5-HI model results in more clinical notes where the irrelevant chapter, block, or

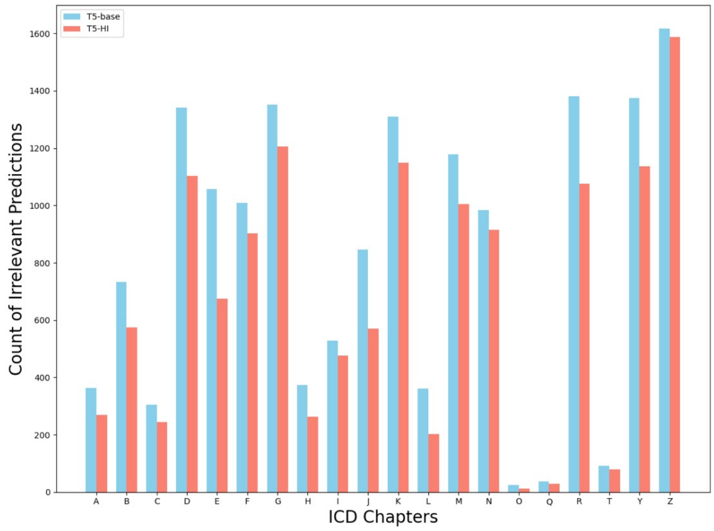


Figure 9: Comparison of irrelevant ICD chapter predictions between T5-base and T5-HI models.

third-level predictions are 0. This indicates that T5-HI is more effective at minimizing irrelevant predictions across the first three levels of ICD codes.

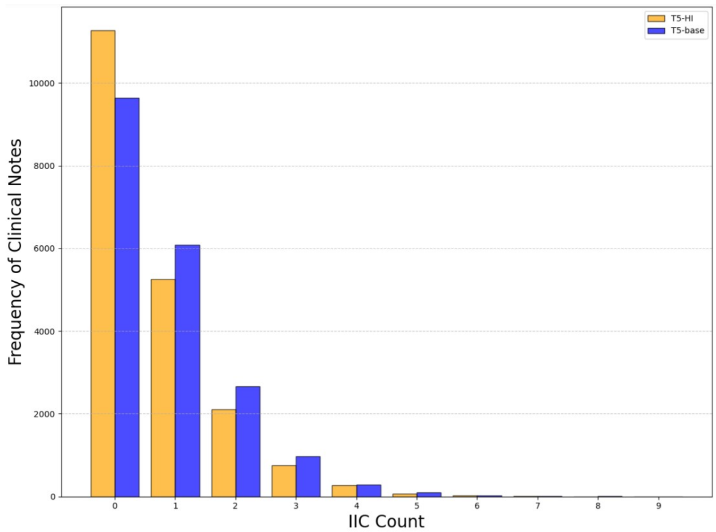


Figure 10: Distribution of irrelevant chapter predictions (IIC) between T5-base and T5-HI models.

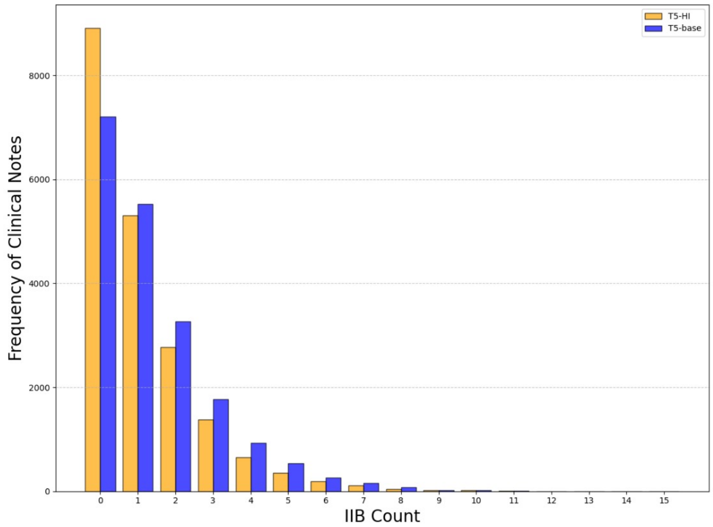


Figure 11: Distribution of irrelevant block predictions (IIB) between T5-base and T5-HI models.

4.4 UNCERTAINTY RESULTS IN ICD CODE PREDICTION USING MCD METHOD

To evaluate uncertainty in ICD-10 code predictions, we applied MCD with 25 inference passes on a randomly selected subset of 1000 clinical notes from the test data.

For each ICD-10 code predicted across these samples, we calculated the mean probability by aggregating the predicted probabilities of the code over the 25 inference passes for each sample. The mean probability for each code was determined by dividing the total probability of that code across all passes by the number of samples where the code was predicted.

The uncertainty for each ICD code was then calculated as the variance of the predicted probabilities across the 25 passes for each sample:

$$\text{Uncertainty} = \sigma_l^2 = \frac{1}{N}\sum_{i=1}^{N}(p_l(i) - \mu_l)^2$$

where μ_l is the mean predicted probability for label l, and N is the total number of passes.

We calculated the uncertainty for a total of 833 predicted ICD codes across the sample set. Higher uncertainty values indicate that the model is less confident in its predictions for those codes, while lower

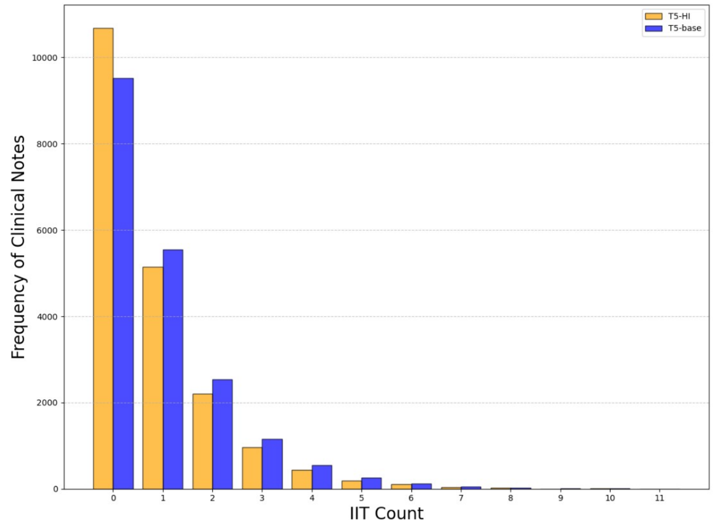


Figure 12: Distribution of irrelevant third-level predictions (IIT) between T5-base and T5-HI models.

values suggest higher confidence.

To assess the relationship between the uncertainty of ICD code predictions and their frequency in the training dataset, we calculated the Spearman correlation coefficient. The analysis resulted in a Spearman correlation of -0.41, indicating a moderate inverse relationship between these two variables. This negative correlation suggests that, generally, codes that are more frequently represented in the training data tend to have lower uncertainty in their predictions. In contrast, less frequently encountered codes are associated with higher levels of uncertainty.

In Tables 7 and 8, we present the uncertainty levels of the top 10 frequent and low-frequency ICD codes, respectively. The results align with the correlation analysis and indicate a moderate inverse relationship between the frequency of ICD codes in the training data and the uncertainty associated with their predictions. As the frequency of a code increases, the uncertainty tends to decrease. However, it is important to note that the moderate correlation does not imply a significant direct effect, high frequency does not necessarily guarantee low uncertainty (less than 0.5).

To further illustrate the model's output and conduct a confidence check, we performed uncertainty estimation using MCD on a single randomly selected clinical note. The ground truth ICD-10 codes for this note were H409, R339, I10, I251, Z952, Y831, Z867, C61, Z951, and E785. After applying MCD with 25 inference passes, the model predicted a range of ICD-10 codes, including B956, E785, H409, I10, I251, and others.

The output presented in Figure 13 highlights these predicted ICD-10 codes alongside their corresponding uncertainty values. Codes exhibiting high or medium uncertainty (e.g., those with uncertainty values greater than 0.5) indicate areas where the model's confidence is lower. Such predictions may require further review by human coders to ensure accuracy and alignment with the clinical context.

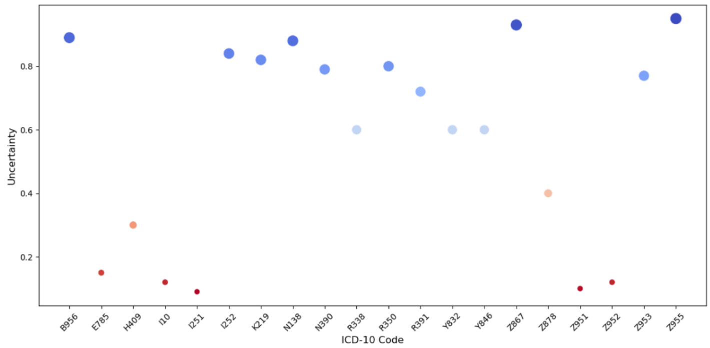


Figure 13: Predicted ICD-10 codes for a randomly selected clinical note, displaying their corresponding uncertainty values. Larger markers indicate higher uncertainty values, and a color gradient from dark red (low uncertainty) to dark blue (high uncertainty) emphasizes the level of uncertainty associated with each code.

Table 6: F1 scores for ICD chapter predictions by T5-HI and T5-base models. This table presents the F1 scores for each ICD chapter, comparing the performance of the T5-HI and T5-base models. The chapter frequencies in the training dataset are also included, highlighting the relationship between the frequency of chapters and their predictive accuracy.

Chapter	T5-HI Mean F1 (95% CI)	T5-base Mean F1 (95% CI)	Chapters' Freq.
A	**0.63 (0.61-0.65)**	0.51 (0.49-0.53)	8,805
B	0.54 (0.52-0.55)	**0.55 (0.53-0.56)**	19,164
C	**0.76 (0.75-0.77)**	0.73 (0.71-0.74)	25,278
D	0.73 (0.72-0.74)	**0.74 (0.73-0.74)**	58,585
E	**0.92 (0.91-0.92)**	0.89 (0.89-0.90)	153,774
F	**0.74 (0.73-0.75)**	0.72 (0.71-0.73)	71,770
G	**0.73 (0.72-0.73)**	0.69 (0.69-0.70)	52,869
H	**0.53 (0.51-0.55)**	0.49 (0.46-0.51)	9,438
I	0.85 (0.85-0.86)	0.85 (0.85-0.86)	182,259
J	**0.79 (0.78-0.79)**	0.78 (0.77-0.79)	50,081
K	**0.80 (0.80-0.81)**	0.79 (0.78-0.80)	86,635
L	**0.58 (0.56-0.60)**	0.53 (0.51-0.54)	14,214
M	**0.62 (0.61-0.63)**	0.61 (0.60-0.62)	41,845
N	0.83 (0.83-0.84)	0.83 (0.82-0.84)	60,024
O	**0.87 (0.85-0.90)**	0.85 (0.82-0.88)	4,379
P	0.00 (0.00-0.00)	0.00 (0.00-0.00)	22
Q	**0.30 (0.26-0.35)**	0.17 (0.13-0.21)	2,484
R	**0.68 (0.67-0.69)**	0.63 (0.63-0.64)	88,458
T	0.27 (0.22-0.31)	**0.33 (0.29-0.38)**	1,778
Y	0.46 (0.45-0.48)	**0.47 (0.45-0.48)**	41,568
Z	**0.92 (0.92-0.92)**	0.89 (0.89-0.90)	213,622

Table 7: Top 10 frequent ICD Codes in Training Data. This table presents the ICD codes with their corresponding uncertainty and frequency in the training dataset, highlighting the relationship between code frequency and prediction uncertainty.

Code	Uncertainty	Frequency
E785	0.32	32,176
Z878	0.45	27,117
K219	0.50	22,374
I251	0.47	18,390
Z790	0.53	18,195
F329	0.66	16,602
N179	0.58	14,377
F419	0.55	13,693
Z867	0.62	13,354
Z794	0.65	11,005

Table 8: Top 10 low frequent ICD codes in training data. This table presents the ICD codes with their corresponding uncertainty and frequency in the training dataset, illustrating the patterns of uncertainty associated with less frequently encountered codes.

Code	Uncertainty	Frequency
O480	0.91	20
O420	0.96	23
O441	0.79	24
O988	0.92	26
O621	0.88	28
Z302	0.94	32
O770	0.92	34
O133	0.92	34
O210	0.48	38
E849	0.96	39

5 DISCUSSION

Our findings highlight the potential of integrating the hierarchical structure of ICD-10 codes within the model to reduce prediction errors, particularly by avoiding irrelevant disease classifications. This approach enhances the ability of LLMs, specifically T5, to automate the task of ICD-10 code assignment from clinical notes. By incorporating the ICD-10 hierarchy, we reduce clinically irrelevant predictions.

5.1 KEY EXPERIMENTAL DECISIONS AND INSIGHTS

Several critical decisions were made during the experimental process that shaped the final model. One of the most significant was the decision to incorporate the hierarchical structure of ICD-10 codes. This integration aimed to improve the model's understanding of the relationships between broader disease categories and specific diagnoses. An essential aspect of this decision was the introduction of a distance-based loss function, which penalizes predictions based on their distance within the ICD-10 hierarchy. However, during development, it became evident that a non-differentiable distance-based loss function would disrupt the gradient flow, leading to unstable training. This realization prompted an experiment comparing the non-differentiable loss with a differentiable approximation using a smooth minimum function. The results in section 4.2.3 clearly indicated that the non-differentiable loss caused significant fluctuations in training, highlighting the importance of maintaining smooth gradient flow.

Consequently, the loss function was modified to incorporate a differentiable approximation of the distance metric and the minimum operation. This change resulted in more stable training, as continuous gradients facilitated smoother optimization. The final decision to use a differentiable loss function proved crucial, as it enabled the model to effectively leverage the hierarchical relationships between ICD-10 codes, improving prediction accuracy and model stability.

5.2 HIERARCHICAL STRUCTURES AND PREDICTION ACCURACY

Although the standard T5-base model achieved a higher F1 micro score, the key advantage of the T5-HI model was its reduction in irrelevant predictions at the chapter, block, and third-level stages. As shown in Table 5, T5-HI model demonstrates a notable improvement in minimizing these irrelevant predictions compared to T5-base model. This supports the hypothesis that leveraging the hierarchical nature of ICD-10 codes enhances the model's ability to predict finer-level codes more accurately. These findings are also consistent with previous research, which highlights how hierarchical structuring improves a model's ability to distinguish between closely related diagnostic categories [35].

Additionally, T5-HI model showed better alignment with ground truth codes, as showed by a lower TD in Figure 8. This indicates that the T5-HI model generates predictions that are closer to the actual ICD codes, which is crucial for reducing clinically irrelevant diagnoses. So while the F1 score is an important metric, the reduction in irrelevant predictions achieved by the T5-HI model, despite a statistically lower F1 score ($p = 0.001$), reflects a valuable trade-off between accuracy and clinical relevance.

5.3 UNCERTAINTY IN PREDICTIONS

The uncertainty analysis using the MCD method provides important insights into how reliable the ICD-10 code predictions are. The moderate inverse Spearman correlation coefficient of -0.38 indicates that more frequently represented codes in the training data generally exhibit lower uncertainty in their predictions. This suggests that frequent codes are learned more effectively by the model, leading to higher confidence in their predictions.

In contrast, infrequent codes tend to show greater uncertainty, highlighting the model's challenges in accurately predicting these codes due to limited training examples. While the correlation suggests a trend, it's important to note that high frequency does not always guarantee low uncertainty, as some frequent codes may still exhibit high uncertainty.

Furthermore, the ability of MCD to provide meaningful uncertainty estimates depends significantly on the model's generalization capability. If the model is biased or overconfident, it may produce low uncertainty values for incorrect predictions, leading to a misrepresentation of its confidence. This limitation underscores the need for robust model training to reduce biases and improve generalization, ensuring that uncertainty estimates accurately reflect the true reliability of predictions.

These insights suggest that enhancing training data with more examples of infrequent codes could improve prediction reliability and reduce uncertainty. Additionally, addressing model biases and ensuring better generalization is critical to maximize the effectiveness of uncertainty estimation techniques like MCD.

5.4 CLINICAL IMPLICATIONS

The findings of this study have notable clinical implications, underscoring the potential of advanced machine learning techniques in addressing key challenges in ICD coding. By leveraging the hierarchical structure of ICD-10 codes, the T5-HI model effectively reduced irrelevant predictions across multiple levels of the coding hierarchy. Although the model exhibited a lower F1 score compared to the baseline, its ability to minimize clinically irrelevant codes highlights its practical utility in healthcare applications.

The improved precision can streamline coding workflows, reduce the administrative burden on healthcare professionals, and minimize error correction time. It also enhances data quality, supporting better clinical decision-making, resource allocation, and healthcare management.

These advancements can optimize ICD coding efficiency, reduce manual verification costs, and help healthcare systems better utilize medical data for operational and research purposes.

6 CONCLUSION

In this work, we explored the application of LLMs, particularly the T5 model, for automating the assignment of ICD-10 codes from clinical notes. The focus was on leveraging the hierarchical structure of ICD codes and integrating it into the T5-base model to reduce irrelevant code assignments.

Our findings demonstrate the significant advantages of incorporating hierarchical information into ICD coding. The enhanced T5-HI model showed notable improvements in reducing irrelevant predictions, especially at the first three levels of ICD-10 codes. Although T5-HI model resulted in a statistically significant lower F1 score compared to T5-base model (p-value < 0.001), this trade-off must be understood in context. While the F1 score is an important metric, it may not fully capture the benefits of hierarchical modeling, where minimizing irrelevant predictions is critical for clinical relevance. The lower F1 score does not reduce the model's focus on generating clinically relevant predictions, as reflected by its improved performance on other metrics like TD and fewer irrelevant predictions across hierarchical levels. In practical healthcare applications, reducing irrelevant classifications might be more beneficial than achieving a higher F1 score, which could make this model more effective for real-world use.

In addition, the integration of uncertainty estimation via MCD further enhanced the utility of the T5-HI model by enabling the identification of predictions with higher uncertainty, thus highlighting cases where human oversight may still be necessary.

Future work should focus on applying T5-HI model to larger and more diverse datasets, including data from various healthcare systems and populations, to ensure its generalizability in real-world settings. Addressing complex and rare diagnoses remains a challenge, and future improvements could be achieved through multi-task learning or graph-based methods, which can better capture relationships between different diagnoses across ICD chapters. Additionally, integrating contextual information, such as patient history and demographics, may improve the model's ability to handle more complex cases with greater accuracy. Further research should focus on incorporating uncertainty estimation techniques into the training process to improve overall model performance. Fine-tuning the hierarchical integration will also be essential for improving F1 scores while preserving the relevance of predictions. We also aim to explore the incorporation of the hierarchical structure of ICD codes into non-generative models, expanding beyond our current focus on the generative T5 model.

[1] Marilisa Amoia, Frank Diehl, Jesus Gimenez, Joel Pinto, Raphael Schumann, Fabian Stemmer, Paul Vozila, and Yi Zhang. Scalable wide and deep learning for computer assisted coding. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 1–7. Association for Computational Linguistics. doi: 10.18653/v1/N18-3001. URL http://aclweb.org/anthology/N18-3001.

[2] Awais Ashfaq, Markus Lingman, Murat Sensoy, and SÅawomir Nowaczyk. DEED: DEep evidential doctor. 325:104019. ISSN 00043702. doi: 10.1016/j.artint.2023.104019. URL https://linkinghub.elsevier.com/retrieve/pii/S0004370223001650.

[3] Sheikh Shams Azam, Manoj Raju, Venkatesh Pagidimarri, and Vamsi Chandra Kasivajjala. CASCADENET: An LSTM based deep learning model for automated ICD-10 coding. In Kohei Arai and Rahul Bhatia, editors, *Advances in Information and Communication*, volume 70, pages 55–74. Springer International Publishing. ISBN 978-3-030-12384-0 978-3-030-12385-7. doi: 10.1007/978-3-030-12385-7_6. URL http://link.springer.com/10.1007/978-3-030-12385-7_6. Series Title: Lecture Notes in Networks and Systems.

[4] MarÃlia Barandas, Lorenzo Famiglini, Andrea Campagner, Duarte Folgado, Raquel SimÃ£o, Federico Cabitza, and Hugo Gamboa. Evaluation of uncertainty quantification methods in multi-label classification: A case study with automatic diagnosis of electrocardiogram. 101:101978. ISSN 15662535. doi: 10.1016/j.inffus.2023.101978. URL https://linkinghub.elsevier.com/retrieve/pii/S1566253523002944.

[5] AtÄ±lÄ±m GunesBaydin, Barak APearlmutter, Alexey AndreyevichRadul, andJeï¬rey Mar
asurvey.

[6] David Bertoin, Jerome Bolte, Sebastien Gerchinovitz, and Edouard Pauwels. Numerical inï¬uenceofReLUâ(0)onbackpropagation.

[7] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, and Tom Henighan. Language models are few-shot learners.

[8] Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. HyperCore: Hyperbolic and co-graph representation for automatic ICD coding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3105–3114. Association for Computational Linguistics, . doi: 10.18653/v1/2020.acl-main.282. URL https://www.aclweb.org/anthology/2020.acl-main.282.

[9] Pengfei Cao, Chenwei Yan, Xiangling Fu, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. Clinical-coder: Assigning interpretable ICD-10 codes to chinese clinical notes. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 294–301. Association for Computational Linguistics, . doi: 10.18653/v1/2020.acl-demos.33. URL https://www.aclweb.org/anthology/2020.acl-demos.33.

[10] Lorenzo Ciampiconi, Adam Elwood, Marco Leonardi, Ashraf Mohamed, and Alessandro Rozza. A survey and taxonomy of loss functions in machine learning. URL http://arxiv.org/abs/2301.05579.

[11] Joakim Edin, Alexander Junge, Jakob D. Havtorn, Lasse Borgholt, Maria Maistro, Tuukka Ruotsalo, and Lars MaalÃže. Automated medical coding on MIMIC-III and MIMIC-IV: A critical review and replicability study. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2572–2582. doi: 10.1145/3539618.3591918. URL http://arxiv.org/abs/2304.10909.

[12] Martin Engilberge, Louis Chevallier, Patrick Perez, and Matthieu Cord. SoDeep: A sorting deep net to learn ranking loss surrogates. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10784–10793. IEEE. ISBN 978-1-72813-293-8. doi: 10.1109/CVPR.2019.01105. URL https://ieeexplore.ieee.org/document/8953213/.

[13] Mingzhou Fu, Yu Yan, Loes M Olde Loohuis, and Timothy S Chang. Defining the distance between diseases using SNOMED CT embeddings. 139:104307. ISSN 15320464. doi: 10.1016/j.jbi.2023.104307. URL https://linkinghub.elsevier.com/retrieve/pii/S153204642300028X.

[14] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning.

[15] Dominic Girardi, Sandra Wartner, Gerhard Halmerbauer, Margit EhrenmÃŒller, Hilda Kosorus, and Stephan Dreiseitl. Using concept hierarchies to improve calculation of patient similarity. 63: 66–73. ISSN 15320464. doi: 10.1016/j.jbi.2016.07.021. URL https://linkinghub.elsevier.com/retrieve/pii/S1532046416300752.

[16] Chao-Wei Huang, Shang-Chi Tsai, and Yun-Nung Chen. PLM-ICD: Automatic ICD coding with pretrained language models, . URL http://arxiv.org/abs/2207.05289.

[17] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. ClinicalBERT: Modeling clinical notes and predicting hospital readmission, . URL http://arxiv.org/abs/1904.05342.

[18] Zheng Jia, Xudong Lu, Huilong Duan, and Haomin Li. Using the distance between sets of hierarchical taxonomic clinical concepts to measure patient similarity. 19(1):91. ISSN 1472-6947. doi: 10.1186/s12911-019-0807-y. URL https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-019-0807-y.

[19] Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li-wei H. Lehman, Leo A. Celi, and Roger G. Mark. MIMIC-IV, a freely accessible electronic health record dataset. 10(1):1, . ISSN 2052-4463. doi: 10.1038/s41597-022-01899-x. URL https://www.nature.com/articles/s41597-022-01899-x.

[20] Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. 3(1):160035, . ISSN 2052-4463. doi: 10.1038/sdata.2016.35. URL https://www.nature.com/articles/sdata201635.

[21] Pravneet Kaur, Gautam Siddharth Kashyap, Ankit Kumar, Tabrez Nafis, Sandeep Kumar, and Vikrant Shokeen. From text to transformation: A comprehensive review of large language models' versatility. .

[22] Rajvir Kaur, Jeewani Anupama Ginige, and Oliver Obst. AI-based ICD coding and classification approaches using discharge summaries: A systematic literature review. 213:118997, . ISSN 09574174. doi: 10.1016/j.eswa.2022.118997. URL https://linkinghub.elsevier.com/retrieve/pii/S0957417422020152.

[23] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision?

[24] Byung-Hak Kim and Varun Ganapathi. Read, attend, and code: Pushing the limits of medical codes prediction from clinical notes by machines.

[25] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181. URL http://aclweb.org/anthology/D14-1181.

[26] Youngwoo Kim, Cheng Li, Bingyang Ye, Amir Tahmasebi, and Javed Aslam. Supervised learning in the presence of noise: Application in ICD-10 code classification. URL http://arxiv.org/abs/2103.07808.

[27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. URL http://arxiv.org/abs/1412.6980.

[28] Bevan Koopman, Guido Zuccon, Anthony Nguyen, Anton Bergheim, and Narelle Grayson. Automatic ICD-10 classification of cancers from free-text death certificates. 84(11):956–965. ISSN 13865056. doi: 10.1016/j.ijmedinf.2015.08.004. URL https://linkinghub.elsevier.com/retrieve/pii/S1386505615300289.

[29] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles.

[30] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. 36(4): 1234–1240. ISSN 1367-4803, 1367-4811. doi: 10.1093/bioinformatics/btz682. URL https://academic.oup.com/bioinformatics/article/36/4/1234/5566506.

[31] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://www.aclweb.org/anthology/2020.acl-main.703.

[32] Fei Li and Hong Yu. ICD coding from clinical text using multi-filter residual convolutional neural network. 34(5):8180–8187. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v34i05.6331. URL https://ojs.aaai.org/index.php/AAAI/article/view/6331.

[33] Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Pre-trained language models for text generation: A survey. 56(9): 1–39. ISSN 0360-0300, 1557-7341. doi: 10.1145/3649449. URL https://dl.acm.org/doi/10.1145/3649449.

[34] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353.

[35] Leibo Liu, Oscar Perez-Concha, Anthony Nguyen, Vicki Bennett, and Louisa Jorm. Automated ICD coding using extreme multi-label long text transformer-based models. 144:102662, . ISSN 09333657. doi: 10.1016/j.artmed.2023.102662. URL https://linkinghub.elsevier.com/retrieve/pii/S0933365723001768.

[36] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning, . URL http://arxiv.org/abs/1605.05101.

[37] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. BioGPT: generative pre-trained transformer for biomedical text generation and mining. 23(6):bbac409. ISSN 1467-5463, 1477-4054. doi: 10.1093/bib/bbac409. URL https://academic.oup.com/bib/article/doi/10.1093/bib/bbac409/6713511.

[38] Aurelie Mascio, Zeljko Kraljevic, Daniel Bean, Richard Dobson, Robert Stewart, Rebecca Bendayan, and Angus Roberts. Comparative analysis of text classification approaches in electronic health records. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 86–94. Association for Computational Linguistics. doi: 10.18653/v1/2020.bionlp-1.9. URL https://www.aclweb.org/anthology/2020.bionlp-1.9.

[39] Susan McBride, Mari Tietze, Catherine Robichaux, Liz Stokes, and Eileen Weber. Identifying and addressing ethical issues with use of electronic health records. 23(1). ISSN 1091-3734. doi: 10.3912/OJIN.Vol23No01Man05. URL https://ojin.nursingworld.org/MainMenuCategories/ANAMarketplace/ANAPeriodicals/OJIN/TableofContents/Vol-23-2018/No1-Jan-2018/Identifying-and-Addressing-Ethical-Issues-EHR.html.

[40] J. H. Moltz, A. Hansch, B. Lassen-Schmidt, B. Haas, A. Genghi, J. Schreier, T. Morgas, and J. Klein. Learning a loss function for segmentation: A feasibility study. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 357–360. IEEE. ISBN 978-1-5386-9330-8. doi: 10.1109/ISBI45749.2020.9098557. URL https://ieeexplore.ieee.org/document/9098557/.

[41] Ghulam Mujtaba, Liyana Shuib, Norisma Idris, Wai Lam Hoo, Ram Gopal Raj, Kamran Khowaja, Khairunisa Shaikh, and Henry Friday Nweke. Clinical text classification research trends: Systematic literature review and open issues. 116:494–520. ISSN 09574174. doi: 10.1016/j.eswa.2018.09.034. URL https://linkinghub.elsevier.com/retrieve/pii/S0957417418306110.

[42] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. Explainable prediction of medical codes from clinical text. URL http://arxiv.org/abs/1802.05695.

[43] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. URL http://arxiv.org/abs/2307.06435.

[44] Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and NoÃ©mie Elhadad. Diagnosis code assignment: models and evaluation metrics. 21(2):231–237. ISSN 1067-5027, 1527-974X. doi: 10.1136/amiajnl-2013-002159. URL https://academic.oup.com/jamia/article-lookup/doi/10.1136/amiajnl-2013-002159.

[45] Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. Machine learning in medicine. 380(14):1347–1358. ISSN 0028-4793, 1533-4406. doi: 10.1056/NEJMra1814259. URL http://www.nejm.org/doi/10.1056/NEJMra1814259.

[46] Colin Raï¬el, $NoamShazeer, AdamRoberts, KatherineLee, SharanNarang, MichaelM$
$to - texttransformer$.

[47] David E Rumelhart, Richard Durbin, Richard Golden, and Yves Chauvin. Backpropagation: The basic theory.

[48] Mary Jane C. Samonte, Bobby D. Gerardo, and Ruji P. Medina. Towards enhanced hierarchical attention networks in ICD-9 tagging of clinical notes. In *Proceedings of the 3rd International Conference on Communication and Information Processing*, pages 146–150. ACM. ISBN 978-1-4503-5365-6. doi: 10.1145/3162957.3163030. URL https://dl.acm.org/doi/10.1145/3162957.3163030.

[49] Henning Schafer and Christoph M Friedrich. Multilingual ICD-10 code assignment with transformer architectures using MIMIC-III discharge summaries.

[50] Haoran Shi, Pengtao Xie, Zhiting Hu, Ming Zhang, and Eric P. Xing. Towards automated ICD coding using deep learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1066–1076. doi: 10.18653/v1/P18-1098. URL http://arxiv.org/abs/1711.04075.

[51] Congzheng Song, Shanghang Zhang, Najmeh Sadoughi, Pengtao Xie, and Eric Xing. Generalized zero-shot text classification for ICD coding. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 4018–4024. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-6-5. doi: 10.24963/ijcai.2020/556. URL https://www.ijcai.org/proceedings/2020/556.

[52] Fei Teng, Yiming Liu, Tianrui Li, Yi Zhang, Shuangqing Li, and Yue Zhao. A review on deep neural networks for ICD coding. pages 1–1. ISSN 1041-4347, 1558-2191, 2326-3865. doi: 10.1109/TKDE.2022.3148267. URL https://ieeexplore.ieee.org/document/9705116/.

[53] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. URL http://arxiv.org/abs/2109.01652.

[54] Zhichao Yang, Sunjae Kwon, Zonghai Yao, and Hong Yu. Multi-label few-shot ICD coding as autoregressive generation with prompt. URL http://arxiv.org/abs/2211.13813.

[55] Rui Zhao, Ruqiang Yan, Zhenghua Chen, Kezhi Mao, Peng Wang, and Robert X. Gao. Deep learning and its applications to machine health monitoring. 115:213–237. ISSN 08883270. doi: 10.1016/j.ymssp.2018.05.050. URL https://linkinghub.elsevier.com/retrieve/pii/S0888327018303108.

[56] Arman Zharmagambetov, Brandon Amos, Aaron Ferber, Taoan Huang, Bistra Dilkina, and Yuandong Tian. Landscape surrogate: Learning decision losses for mathematical optimization under partial information.

HALMSTAD
UNIVERSITY

PO Box 823, SE-301 18 Halmstad
Phone: +35 46 16 71 00
E-mail: registrator@hh.se
www.hh.se