A Hierarchical Fine-Grained Deep Learning Model for Automated Medical Coding

Joshua Carberry
Computer and Information Science Department
University of Massachusetts Dartmouth
Dartmouth, MA 02747, USA
jcarberry@umassd.edu

Haiping Xu
Computer and Information Science Department
University of Massachusetts Dartmouth
Dartmouth, MA 02747, USA
hxu@umassd.edu

***Abstract*—During a patient's visit, hospital staff record the patient's condition, including measurements, observations, diagnoses, and treatment. These records, known as doctor's notes, are usually kept in natural language and must be standardized for consistent communication during treatment and billing. Medical coding is the procedure of applying standardized, universal codes to doctor's notes for efficient and effective record keeping. Due to the complexity of healthcare and the breadth of natural languages, this has been a difficult task requiring significant training and resources. In this paper, we present a fine-grained, evidence-based hierarchical deep learning model for automated medical coding. Instead of treating the doctor's notes as a whole, the proposed method processes the coding of each diagnosis separately. For each diagnosis, the knowledge encoded in ontologies is used to extract semantically related sentences from the doctor's notes. Then, a hierarchical deep learning classifier processes the diagnosis and its semantically related sentences and generates the medical code prediction. This approach not only achieved promising results in our experiments, but also provided a high degree of explainability and trustworthiness in the results, which is a key factor in the adoption of medical solutions.**

Keywords—Hierarchical deep learning models, automated medical coding, natural language processing, ontologies

I. Introduction

In the healthcare industry, accurate and reliable record keeping is essential. Records such as doctor's notes describe the details of a patient's hospital visit, including observations, measurements, and procedures, and are key to the various steps in the healthcare process. For example, when treating a patient, it is important to review their medical history, including records of previous hospital visits. An accurate understanding of a patient's medical history can determine whether treatment is effective, ineffective or even dangerous. On the financial side, doctor's notes are used to identify a patient's diagnoses and the procedures carried out, and these records are used for billing and insurance purposes. If an error is made, it can lead to improper charges or insurance billing.

Doctor's notes are handwritten or typed documents used by a doctor to describe a patient's hospital visit. These documents are written in natural language and use specialized medical terminology. Since the doctor's notes are usually recorded during a patient's visit, they often contain abbreviations, shorthand, and even spelling errors. Therefore, doctor's notes are only considered a preliminary record of a hospital visit. When the doctor's notes are created, they must be annotated with a set of standardized diagnostic codes that indicate the precise medical diagnoses discussed in the notes. These standardized codes characterize patient visits more accurately than natural language and serve as a shared language that facilitates communications between doctors, hospitals, and financial institutions. In this study, we explore the use of the International Classification of Diseases (ICD) medical coding standards to annotate doctor's notes. ICD codes are very specific international standards for classifying diseases and include thousands of unique codes for annotation [1]. The procedure of annotating doctor's notes is known as medical coding and has traditionally been performed by doctors or trained medical coders. Automated solutions have the potential to reduce the workload associated with coding, allowing professionals to redirect their time and energy toward improving healthcare. They can also reduce human error in medical coding and prevent incorrect billing and treatment. Recent research efforts have approached medical coding as a classification task that returns a set of predicted medical codes given a doctor's notes instance. This classification task is typically solved as a multi-label classification, whereby the doctor's notes instance is processed by a deep learning classifier and all predicted medical codes are output at once. While these methods may exhibit good performance in classification tasks, they often lack proper emphasis on explainability and can be "black boxes" to users who are less familiar with deep learning. In an industry such as healthcare, which is built on trust, the lack of in-depth knowledge of such an important process can be a barrier to adoption.

In this paper, we present a fine-grained, evidence-based hierarchical deep learning model for automated medical coding. Unlike the multi-label approaches, which we call coarse-grained methods, our approach performs medical coding by solving a series of fine-grained single-label classifications. While doctor's notes are mostly unstructured, they usually contain a bulleted or numbered list of diagnoses. For each diagnosis in a doctor's notes instance, we extract only the semantically related sentences from the doctor's notes and generate a fine-grained data point for classification. Sentence extraction is performed semantically using medical ontologies that encode medical concepts and their various relationships. The resulting fine-grained data points are passed separately to a deep learning classifier, which outputs predicted medical codes corresponding to the fine-grained data points. Based on our preliminary research on automated medical coding [2], we introduce a hierarchical fine-grained deep learning model designed to improve the tractability of large label spaces, a key consideration in medical coding where thousands of unique codes can be used for medical code annotation. This hierarchical approach can not only improve medical coding performance, but also provides a human-understandable structure that reveals the classification path for a given input and contextualizes the model results.

II. Related Work

A significant amount of research effort has been invested into the development of automated medical coding systems. Early research efforts had little data for training and testing, and thus focused on less data-intensive procedures such as rule-based classification. An early approach to automated ICD coding for radiology reports was proposed by Goldstein and

their colleagues [3]. Their approach is to break down sentences into "lexical elements" and apply rules to make predictions. These rules can be derived by humans from a small number of examples and applied generally without the need for large volumes of training data. With the rise of electronic healthcare records (EHR), the availability of such data has gradually increased, leading to an increasing number of approaches based on machine learning. For example, Farkas and Szarvas attempted to automatically construct ICD-9-CM coding systems for radiology reports [4]. They managed to achieve comparable results with purely hand-crafted ICD-9-CM expert rules. Medori and Fairon described the architecture of an encoding system using machine learning [5]. They implemented a Naïve Bayes classifier to perform medical coding and showed that the extracted information to be coded is essential in the classification process. Following the introduction of initial machine learning approaches, the release of EHR datasets such as the Medical Information Mart for Intensive Care III (MIMIC-III) has significantly improved data accessibility and stimulated greater research interest [6]. This increased data availability has motivated researchers to explore more resource-intensive approaches. Consequently, they have delved into the problem using deep learning techniques to improve the performance of automated ICD coding. However, the large amount of text in the doctor's note data poses a challenge, as the large input can confuse a deep learning classifier, especially if it is biased toward a limited portion of the input. To address this challenge, some research efforts employing deep learning methods have focused on developing classifiers that can build representations at multiple levels of analysis, aiming to understand doctor's notes in terms of overarching concepts or even sentence-level complexity [2][7][8]. Other research efforts have addressed this issue by introducing a labeling attention mechanism in their deep learning methods to produce a different document interpretation for each unique label [9][10]. While these methods were effective and resulted in improved classification performance, they have significant drawbacks. Specifically, these methods perform coarse-grained ICD code prediction, where the classifier accepts the entire document and outputs a set of predicted ICD codes. This not only increases the difficulty of classification, but also makes it difficult for users to interpret and understand individual code predictions. In contrast, our hierarchical fine-grained approach can trace not only the classification steps for each diagnosis, but also the evidence (in the form of extracted sentences) used in each step. Our approach collects sentences semantically related to a given diagnosis based on human knowledge, thus providing the user with an exhaustive explanation for the classification results. This represents a major advantage in terms of explainability and transparency, which are valuable in a field as important and sensitive as healthcare.

There has also been previous work on the use of a hierarchical classifier in machine learning, which moves down the label hierarchy to individual class labels in a series of steps. A typical hierarchical classifier architecture uses the per-node local classifier approach, where each node represents a unique classification step performed by a classifier independent of other nodes. Based on the decision made by a particular node, further evaluations are carried out in the hierarchy using its child nodes until a label representing the decision of the entire hierarchical classifier is derived. The hierarchical classifier approach has been successful in a variety of classification tasks. Wang et al. proposed a hierarchical classification method for real-world document classification [11]. Their approach consists of a hierarchy of local rule-based classifiers that work together to produce document classifications. Marin et al. introduced a hierarchical model for classifying galaxy morphology based on geometric moments [12]. They used local classifiers to determine galaxy types, and the hierarchical design showed good performance improvement in the face of data imbalance and growing label space. Ramírez-Corona et al. proposed an approach called hierarchical multi-label classification (HMC) based on path evaluation [13]. They used a local classifier design to carry out multilabel classification on genomic data and showed that their approach works better when dealing with deep and populated hierarchies. Secker et al. implemented a hierarchy of local classifiers to categorize proteins by their unique functions [14]. Taking advantage of the modularity of the local classifier design, they investigated a hierarchy in which each local classifier is implemented using various machine learning methods, depending on which method provides the best performance. Daisey and Brown studied the effect of hierarchy design on the performance of multi-label classification tasks [15]. They noted that the improvement or degradation of the method depends heavily on the particular design pattern, evaluation strategy and training parameters. While these efforts do not approach the task of automated medical coding, they provide a basis for the general validity of hierarchical classification. As demonstrated in this paper, our approach utilizes a hierarchical fine-grained deep learning model to achieve automated medical coding that can be effectively interpreted.

An alternative to organizing local classifiers is to use a global classifier directly. Instead of arranging multiple local classifiers into a hierarchical structure, the hierarchical components can be built into a single classifier and trained as a single unit. Silla and Freitas introduced a global Naïve Bayes classifier for the protein function classification task [16]. Unlike the hierarchical classifiers, their global approach does not propagate the error to subsequent classification steps when one step goes wrong. Lawrence et al. proposed a hybrid neural network for human face recognition by combining local image sampling, self-organizing map (SOM) neural network and convolutional neural network [17]. They built hierarchical layers into a single convolutional neural network (CNN) to make it a global classifier. Although global classifier methods can streamline training and testing, and sometimes improve performance, they have some drawbacks compared to the hierarchical classifier approaches. In a hierarchical classifier approach, each local classifier can be individually designed, parameterized, tested, and trained, whereas this is not possible for a global classifier, which represents an entire hierarchy that cannot be disassembled and reassembled in the same way as the hierarchical classifiers. Local classifiers can be removed or reused in another hierarchy without retraining, whereas global classifiers must be retrained due to any architectural change, no matter how subtle. While our approach differs from the global classifier methods by using a hierarchical deep learning model for automated medical coding, we envision that in future work, parallel algorithms can be designed to support efficient training of hierarchical deep learning models. In this sense, our approach complements existing global classifier methods by providing a simple and efficient solution for supporting explainable automated medical coding.

III. A Hierarchical Deep Learning Model

A. Fine-Grained Medical Coding

In the medical coding task, we use a doctor's notes instance as input and output appropriate medical codes corresponding to the diagnoses listed in the doctor's notes. The doctor's notes instance contains the doctor's observations of a particular hospital visit, written in unstructured natural

language. Fig. 1 shows an example excerpted from a doctor's notes instance. In a typical coarse-grained approach, the entire document of the doctor's notes is fed into a multi-label classifier, which then outputs all predicted medical codes at once. In contrast to the multi-label coarse-grained approach, we use a fine-grained approach that decomposes the document and performs a series of single-label classifications to solve the medical coding problem.

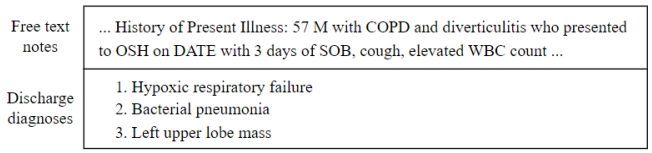

Fig. 1. An example excerpted from a doctor's notes instance.

In our fine-grained approach, we treat each diagnosis in a doctor's notes instance individually. Despite their mostly unstructured nature, a doctor's notes instance usually contains a bulleted list of discharge diagnoses. While these diagnoses are often not sufficient for medical coding, they provide a suitable starting point for further investigation. As shown in Fig. 2, for each listed diagnosis, we select and extract the sentences from the free text notes that are semantically related to the diagnosis. This resulted in a list of fine-grained data points containing a diagnosis and their semantically related sentences. Each fine-grained data point is then passed to the hierarchical classifier, which outputs a medical code prediction for that diagnosis. Note that the fine-grained data points can be processed sequentially or potentially in parallel. Once all fine-grained data points have been classified, we collect the predicted medical codes for different diagnoses into a single set and return them to the user.

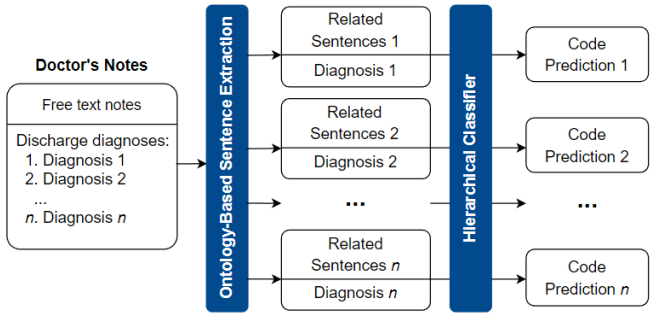

Fig. 2. An overview of the fine-grained approach for medical coding.

Compared to the coarse-grained approach, the fine-grained approach presents simpler single-label classification tasks. This is particularly important for medical coding problems, where thousands of unique codes can be applied. Fine-grained methods also expose key elements of the decision-making process to human understanding. In cases where the classifier outputs unexpected results, each predicted medical code can be traced back to the diagnosis as well as the relevant text extracted from the doctor's notes, which may help explain and contextualize the prediction or diagnose a model failure.

B. Knowledge-Based Sentence Extraction

In many machine learning scenarios, it is often necessary to prune the raw data so that the classifier accesses only the key information that is essential for classification. This pruning process serves a dual purpose. Firstly, it compresses the length of each data point, making the machine learning process more efficient. In our case, shortening data points not only improves the efficiency of the process, but also becomes necessary due to the use of a BERT-based classifier [18]. Given the architectural limitation that each data point can only consist of a maximum of 512 input tokens (including words, punctuation marks, and symbols), data reduction becomes crucial in order to accommodate more critical data elements within this threshold. Secondly, if we can identify the key elements of the input information that are used for classification, then compressing the input information into these key segments improves the reliability of the classification by eliminating more useless or even noisy information in a raw data point. In our fine-grained approach, data is reduced by extracting only the related sentences from doctor's notes. Specifically, for a given diagnosis, we focus on only the free text notes sentences that are semantically related to that diagnosis. With this approach, we can effectively minimize the size of input for automated medical coding. To determine which sentences may be relevant to a diagnosis, we rely on established knowledge of medical concepts and their interconnections. This knowledge allows us to scrutinize the relationship between the diagnosis and the concepts discussed in a given sentence to identify any concept overlap. Such an overlap indicates that the sentence has something in common with the diagnosis, suggesting that it contains information valuable for the classification task.

In recent years, graph-based knowledge representations such as ontologies have become increasingly popularity for a variety of applications. Graph-based representations treat concepts as nodes connected by edges representing relationships. In our approach, we use medical ontologies as a knowledge base for sentence extraction. Fig. 3 shows a partial medical ontology for the concept "congestive heart failure." As shown in the figure, the ontology can be encoded using ordered triples, e.g., the triple <*congestive heart failure*, *has_symptom*, *fatigue*> indicates that there is a relationship named "*has_symptom*" between the concept "*congestive heart failure*" and another concept "*fatigue*". We use such relationships to determine which medical concepts might be semantically related to a diagnosis.

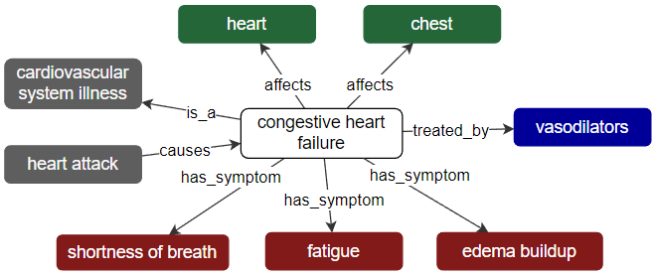

Fig. 3. Part of the medical ontology for concept "congestive heart failure".

Algorithm 1 presents the steps for generating a fine-grained data point δ, which consists of tokens selected from the diagnosis and semantically related sentences extracted from a doctor's notes instance Ξ.

Algorithm 1: Generate a Fine-Grained Data Point

Input: A diagnosis α from a doctor's notes instance Ξ, the free text notes Π in Ξ, a set of concepts C_α related to α, where $C_\alpha \subset$ medical ontology Φ
Output: fine-grained data point δ

1. Create a fine-grained data point δ with 512 token slots
2. Let string *text* be diagnosis α
3. **for each** sentence σ **in** Π:
4. Let C_σ be the set of concepts in Φ that are used in σ
5. Let *overlap* be the intersection of C_α and $C_{\sigma\xi}$
6. **if** $|overlap| > 0$:
7. Add sentence σ to *text*
8. Select tokens from *text* and add them to δ
9. **return** δ

C. Hierarchical Classification

Popular classification methods in machine learning are usually monolithic, with a single classifier performing the entire classification process. This approach is suitable for simple classification tasks where the classifier is sufficiently robust. However, as the complexity of classification increases, it becomes useful to decompose the classification process into multiple stages managed by specialized classifiers. For example, when identifying a specific breed of animal (e.g., a dog or cat breed), it is useful to first predict whether it is a dog or a cat. If the prediction favors the dog, we can rule out all feline possibilities, thus simplifying subsequent classification steps. Assuming the initial "dog" prediction holds, we can direct the partially classified animal to a specialized dog breed classifier for optimal accuracy. Partitioning the classification process into multiple subclassifiers reduces the local complexity of each classification stage. This division allows each subclassifier to focus on its specific, limited scope and thus deal more accurately with the reduced complexity.

Drawing on these concepts, we introduce a hierarchical fine-grained classifier designed to predict ICD codes for a given diagnosis by combining the results of multiple subclassifiers. Once we have generated a fine-grained data point based on the procedure described in Algorithm 1, we can pass that data point through our hierarchical classifier to predict a medical code. Unlike our previous approach using a monolithic classifier [2], we organize the individual classifiers into a hierarchy that progressively refines the fine-grained data point and improves specificity until a final medical code prediction is made. Similar to the practice of categorizing animal species ("dog" or "cat") before delving into animal breeds, it has proven advantageous to first classify diseases into their types or families. For example, a diagnosis of "influenza" might first be categorized as a respiratory disease and then further refined to a respiratory virus, ultimately resulting in a medical code prediction. As previously mentioned, the discrete subclassifiers assigned to each step focus on only a portion of their classification process, thereby reducing processing complexity. Fig. 4 illustrates the model architecture using an example of a hierarchical classifier with three subclassifiers. To predict the ICD code labels, the *TOP* classifier first determines whether the data point belongs to class *A* or class *B*, then it is sent to the appropriate subclassifier (Type *A* or *B* classifier) for further classification.

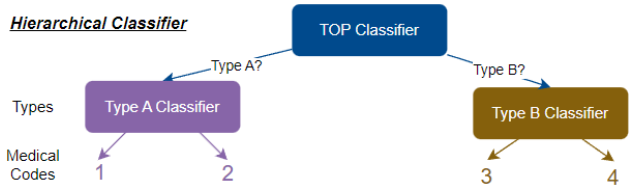

Fig. 4. An example hierarchical classifier with three subclassifiers.

Hierarchical classification provides two major advantages for automated medical coding tasks. First of all, it simplifies the classification problem by decomposing the label space into distinct subspaces. For example, in Fig. 4, we use only binary classification to deal with the medical coding task with 4 unique labels. As mentioned earlier, the growing label space is a particular concern for medical coding problems, and hierarchical classification can ensure that the label space for each subclassifier remains manageable. As more medical codes are considered, the hierarchical structure can be modularly extended to accommodate these codes, while the performance of a monolithic classifier may begin to suffer. In addition to performance issue, hierarchical classification enhances the explainability of the often "black box" deep learning aspect of the method by providing a human-understandable decision framework. A given prediction can be traced up the hierarchy to determine which classification path a data point was sent through, revealing the different steps that led to the final result. These details can help users understand the behavior of the model and provide additional insights on potential classification errors.

A hierarchical classifier can be formally defined as a tree of nodes, each of which is defined as a 3-tuple (*SC*, *CN*, *LA*), where *SC* is a subclassifier defined as a function that receives a fine-grained data point and outputs a list of classification confidences that represent the possibilities of belonging to the classes; *CN* is a list of child nodes that contain subclassifiers; and *LA* is the label associated with the input of the node. The leaf nodes at the bottom of the tree have no child nodes and contain the labels of the final output of the hierarchical classifier. Based on this definition, we can start from the *TOP* root node and progressively predict a given fine-grained data point through its child nodes, as shown in Algorithm 2.

Algorithm 2: Hierarchically Classify a Fine-Grained Data Point

Input: Fine-grained data point δ; a hierarchical classifier with root node *TOP*
Output: ICD code label β of data point δ

1. Let the current node ξ be the *TOP* root node
2. Initialize the ICD code label β of data point δ to *null*
3. **while** $\xi.CN \neq \emptyset$: // ξ is not a leaf node
4. Let *confidences* be an array of size $|\xi.CN|$
5. *confidences* = $\xi.SC(\delta)$ // i.e., the classification process
6. Let *type* be the child node ID with *max*(*confidences*)
7. $\xi = \xi.CN[type]$ // select the corresponding child node
8. Let ICD code label β be $\xi.LA$
9. **return** ICD code label β

Note that in the algorithm, the appropriate child node is selected based on the highest confidence of the classification results at each level of the classification hierarchy. The final ICD code label β of the given data point δ is determined by a leaf node that predicts a class with the highest confidence.

IV. Case Studies and Experimental Results

To demonstrate the feasibility of our new approach, we conducted experiments using MIMIC-III, a publicly available healthcare dataset that offers a large volume of deidentified patient records including doctor's notes and associated medical code labels. Training and testing for the following experiments were carried out on a workstation equipped with a NVIDIA GeForce RTX 2060 SUPER (8 GB VRAM), an Intel Core i7-9700 CPU, and 16 GB of main memory.

A. Experiments with a Small Code Set

As the label space expands, the performance gain from our previous monolithic approach to the current hierarchical approach shall become apparent. Essentially, the hierarchical classifier would exhibit greater resilience when handling classification tasks involving a larger number of unique ICD codes. However, when dealing with a reduced number of codes, the performance gain offered by the hierarchical classifier might not be significant. In this section, we substantiate this claim through our initial experiment on a small set of ICD codes. Specifically, we perform fine-grained evidence-based ICD coding for a subset of 7 heart-disease related ICD codes using both monolithic and hierarchical classification strategies. We expect the performance of the hierarchical approach to meet or exceed the performance of the monolithic method, which in our previous work has demonstrated exceptional performance when dealing with a

limited number of distinct labels or ICD codes [2]. As shown in Fig. 5 (a), in our experiments, the monolithic strategy uses a flat Long Short-Term Memory (LSTM) classifier, which predicts one of the 7 labels in just one step. On the other hand, the hierarchical classifier uses a series of subclassifiers for classification, each of which is responsible for a local decision that becomes part of the final prediction. In this particular example, the hierarchical design has only a marginal advantage due to the small size and low complexity of the 7-code set. Specifically, we added a subclassifier responsible for distinguishing between two highly similar codes C1: 401.1 *Benign hypertension* and C2: 401.9 *Unspecified essential hypertension*. The resulting hierarchy is shown in Fig. 5 (b). For each subclassifier in the hierarchy, we used a separate instance of BERT classifier [18].

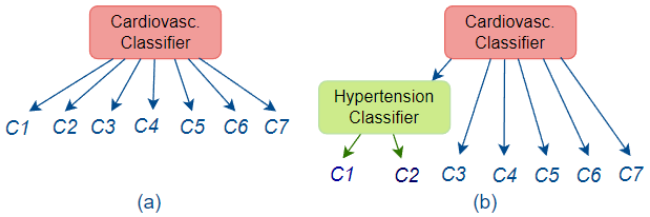

Fig. 5. (a) Monolithic classifier vs. b) hierarchical classifier for a 7-code set.

The classifiers were trained using a 5-fold cross validation method with 5 epochs and a learning rate of 1e-5. The cross-validation method was used to select the highest performing model training checkpoints. The results of the small code set experiments are shown in Table I, where precision, recall, and F1 score were calculated using macro-averages that consider all classes equally regardless of their sizes.

TABLE I. PERFORMANCE METRICS COMPARISON USING 7 CODES

Method	Accuracy	F1 Score	Precision	Recall
Monolithic	0.935	0.912	0.904	0.923
Hierarchical	0.961	0.937	0.964	0.921

As shown in the table, the performance of the former monolithic approach remains reasonable with a small code set, while the performance of the hierarchical classifier improves slightly or comes very close to that of the monolithic classifier.

B. *Experiments with a Large Code Set*

A major challenge in medical coding arises from the large label spaces involved in classifying a large number of unique codes. As with many classification tasks, automated medical coding becomes increasingly difficult as more and more labels are introduced. In this experiment, we have selected 40 common medical codes used in the MIMIC-III dataset. As in Section IV.A, we compared the previous monolithic classification method with our newly introduced hierarchical classification approach to show the advantages of our new approach when more unique ICD codes are involved in the classification task. The structure of our hierarchical classifier is shown in Fig. 6. This hierarchy, largely inspired by the existing hierarchy of ICD codes, subdivides the classification into several steps. First, the top-level classifier identifies the type of disease (e.g., *cardiovascular* and *respiratory*) that the diagnosis may involve. For *cardiovascular*, *mental*, and *digestive* disorders, this leads to a leaf node, which means that this is the final step in the classification; whereas for *respiratory* and *endocrine* disorders, the classification may go directly to the leaf node, which completes the classification, or it may continue on to the *chronic* or *fluid* classifiers. Each subclassifier is a BERT instance trained for its particular classification step. For subclassifiers with higher hierarchical levels such as *TOP*, the data sampling rate is reduced to minimize training time while maintaining performance. All subclassifiers were fine-tuned 5 epochs at a learning rate of 1e-5. We again use ~~an 80/20 training split and~~ the 5-fold cross validation method to diagnose model performance and select optimal training checkpoints. For the monolithic classifier, we used a single instance of BERT trained on all available data with the same training hyperparameters as described above.

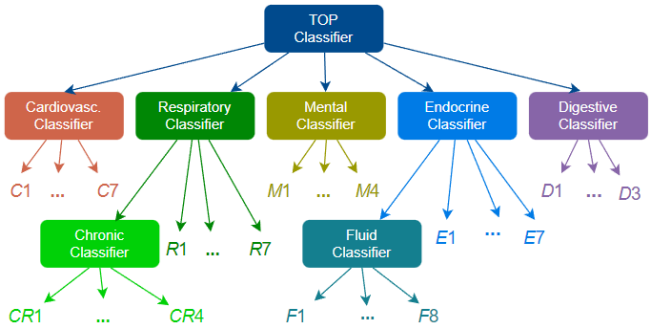

Fig. 6. Hierarchical classifier organization for a 40-code set.

Table II lists the performance metrics of the two classifiers. The main finding of this experiment is the improved performance of the hierarchical method, which significantly outperforms the monolithic classifier in all performance metrics. While the high accuracy reflects good overall classification performance, the F1 score is of particular interest in the performance analysis for two reasons. As macro-averaged values, the F1 scores consider all categories equally, regardless of the frequency of the categories (even the smallest categories are fairly reflected in the metric). Second, F1 score quantifies the incidence of false positives and false negatives, which is particularly important in medical coding tasks where false positives can lead to incorrect billing and false negatives can lead to incomplete documentation. The superiority of the hierarchical classifier in these respects is a good indication that the new model ensures greater robustness in classifying a large code set. Considering the high number of unique codes in the latest standards such as ICD-9, ICD-10, and ICD-11, our hierarchical approach is scalable and its advantage becomes critical in real coding situations, where a rational ICD coding procedure is especially important.

TABLE II. PERFORMANCE METRICS COMPARISON USING 40 CODES

Method	Accuracy	F1 Score	Precision	Recall
Monolithic	0.740	0.597	0.639	0.594
Hierarchical	0.927	0.893	0.932	0.866

C. *An Example of Classification Path*

In order to demonstrate the various steps of our new hierarchical approach using an example, we now trace the behavior of the method in predicting the ICD code for a single diagnosis. As shown in Fig. 7, we select the diagnosis "Metabolic acidosis" for the demonstration of automated ICD code prediction using our hierarchical approach.

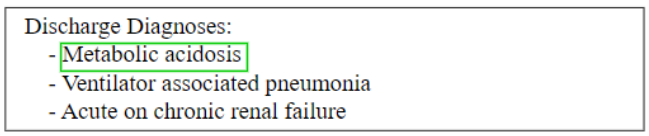

Fig. 7. Selected dignosis "Metabolic acidosis" for ICD code prediction.

To reason which sentences may be semantically related to the selected diagnosis, we must first derive a set of related concepts using the medical ontology as described in Section III.B. In this case, we derive a set of concepts related to "Metabolic acidosis" and extract sentences that discuss these

concepts in the free text of the doctor's notes, as shown in Fig. 8. The extracted sentences are then combined with the original diagnosis text "Metabolic acidosis" to form a fine-grained data point that is ready for classification.

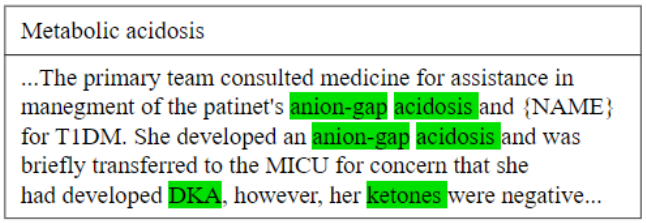

Fig. 8. Semantically related sentences extracted for "Metabolic acidosis".

Fig. 9 shows the classification path for the data point to arrive at the correctly predicted ICD code *276.2 Lactic Acidosis*. The *TOP* subclassifier first determines that the data point must belong to the category of *endocrine* diseases, and then the *Endocrine* subclassifier determines that it must belong to the subcategory of *fluid*-related endocrine diseases. Finally, the *Fluid* subclassifier predicts the final code 276.2.

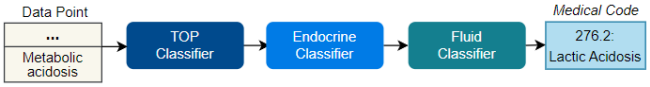

Fig. 9. An example classification path for predicting medical code.

V. Conclusions and Future Work

In this paper, we introduce a fine-grained, evidence-based hierarchical classification methodology customized for the task of automated ICD coding. Our approach involves assigning highly specific, standardized medical codes to patient diagnoses listed in a doctor's notes instance. Unlike other methods, we classify each diagnosis in a doctor's notes instance separately, forming a unique classification path for each. For each diagnosis, our approach uses medical knowledge to extract semantically related sentences from the free text of the doctor's notes to augment the input data, providing additional context for decision-making and human review. This data is passed through a hierarchical classifier that produces an ICD code prediction. Unlike a monolithic classifier that performs classification in a single step, our hierarchical classifier consists of subclassifiers, each of which is responsible for only one step in the classification process. The data point moves down through the hierarchy, receiving more detailed predictions, until a unique ICD code prediction is obtained. Through a number of comparative experiments, we demonstrate the improvement that the hierarchical classification approach brings over the monolithic approach used in previous work [2]. The experimental results show that the hierarchical classifier makes automated coding more robust to datasets containing a large number of unique ICD codes, which is an important consideration since real-world data may involve dozens, hundreds, or even thousands of unique codes, depending on the desired coverage.

In future work, we will explore more complex classifier designs to pursue higher performance and investigate general principles for effective hierarchical design in the context of automated ICD coding. The ultimate goal of this work is to design and implement larger classifiers for realistic ICD coding environments where very high code coverage may be required. In addition, we may consider different design architectures for different subclassifiers in the hierarchy. This may be necessary because the subclassifiers responsible for easier classification can be implemented using simpler but effective architectures to keep the computational cost more reasonable. Finally, the hyperparameters and training sets could also potentially be tuned across different subclassifiers to further increase the flexibility of our approach.

References

[1] World Health Organization, "International statistical classification of diseases and related health problems (ICD)," *Health Topics*, January 1, 2022. Retrieved on December 12, 2023 from: https://www.who.int/standards/classifications/classification-of-diseases

[2] J. Carberry and H. Xu, "Fine-grained ICD code assignment using ontology-based classification," In *Proceedings of the 2022 IEEE 23rd International Conference on Information Reuse and Integration for Data Science (IRI)*, San Diego, CA, USA, 2022, pp. 228-233.

[3] I. Goldstein, A. Arzumtsyan, and O. Uzuner, "Three approaches to automatic assignment of ICD-9-CM codes to radiology reports," *AMIA Annual Symposium Proceedings*, no. 1, 2007, pp.279-283.

[4] R. Farkas and G. Szarvas, "Automatic construction of rule-based ICD-9-CM coding systems," *BMC Bioinformatics*, vol. 9, suppl 3, no. S10, April 2008.

[5] J. Medori and C. Fairon, "Machine learning and features selection for semi-automatic ICD-9-CM encoding," In *Proceedings of the NAACL HLT 2nd Louhi Workshop Text Data Mining Health Documents*, Los Angeles, June 2010, pp. 84-89.

[6] A. E. W. Johnson, T. J. Pollard, L. Shen, L. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Scientific Data*, vol. 3, no. 1, 160035, 2016.

[7] T. S. Heo, Y. Yoo, Y. Park, B. Jo, K. Lee, and K. Kim, "Medical code prediction from discharge summary: document to sequence BERT using sequence attention," In *Proceedings of the 20th IEEE International Conference on Machine Learning and Applications* (ICMLA), Pasadena, CA, USA, 2021, pp. 1239-1244.

[8] M. Li, Z. Fei, F. Wu, Y. Li, Y. Pan, and J. Wang, "Automated ICD-9 coding via a deep learning approach," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, July-August 2019, vol. 16, no. 4, pp. 1193-1202.

[9] Y. Wu, Z. Chen, X. Yao, X. Chen, Z. Zhou, and J. Xue, "JAN: Joint attention networks for automatic ICD coding," *IEEE Journal of Biomedical and Health Informatics*, October 2022, vol. 26, no. 10, pp. 5235-5246.

[10] V. Mayya, S. S. Kamath, and V. Sugumaran, "LATA - Label attention transformer architectures for ICD-10 coding of unstructured clinical notes," In *Proceedings of the 2021 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, Melbourne, Australia, 2021, pp. 1-7.

[11] K. Wang, S. Zhou, and Y. He, "Hierarchical classification of real life documents," In *Proceedings of the 2001 SIAM International Conference on Data Mining (SDM01)*, Chicago, IL, USA, April 5-7, 2001, pp. 1-16.

[12] M. Marin, L. E. Sucar, J. A. Gonzales, and R. Diaz, "A hierarchical model for morphological galaxy classification," In *Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference* (FLAIRS 2013), St. Pete Beach, FL, USA, May 22-24, 2013, pp. 438-443.

[13] M. Ramírez-Corona, L. E. Sucar, and E. F. Morales, "Hierarchical multilabel classification based on path evaluation," *International Journal of Approximate Reasoning*, vol. 68, 2016, pp. 179-183.

[14] A. D. Secker, M. N. Davies, A. A. Freitas, J. Timmis, M. Mendao, and D. R. Flower, "An experimental comparison of classification algorithms for the hierarchical prediction of protein function," *Expert Update*, vol. 9, no. 3, 2007, pp. 17-22.

[15] K. Daisey and S. D. Brown, "Effects of the hierarchy in hierarchical, multi-label classification," *Chemometrics and Intelligent Laboratory Systems*, vol. 207, December 2020, p. 104177.

[16] C. N. Silla Jr. and A. A. Freitas, "A global-model naive bayes approach to the hierarchical prediction of protein functions," In *Proceedings of the 9th IEEE International Conference on Data Mining*, December 6-9, 2009, Miami Beach, FL, USA, pp. 992-997.

[17] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: a convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, 1997, pp. 98-113.

[18] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint, arXiv: 1810.04805, 2018.