International Journal of Semantic Computing

World Scientific
www.worldscientific.com

EXPLAINABLE ICD CODE ASSIGNMENT USING KNOWLEDGE-BASED SENTENCE EXTRACTION AND DEEP LEARNING

JOSHUA CARBERRY

Computer and Information Science Department, University of Massachusetts Dartmouth, 285 Old Westport Rd, Dartmouth, MA 02766, USA
jcarberry@umassd.edu

HAIPING XU

Computer and Information Science Department, University of Massachusetts Dartmouth, 285 Old Westport Rd, Dartmouth, MA 02766, USA
hxu@umassd.edu
http://www.cis.umassd.edu/~hxu

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Medical coding involves the assignment of standardized codes like those of the International Classification of Diseases (ICD) to patient records such as doctor's notes. Traditionally, medical coding has been performed by trained professionals and has incurred significant costs. Recent research efforts have produced good classification results, but often lack in explainability and trustability of the coding results. This paper introduces a novel fine-grained evidence-based approach for medical coding, which improves explainability and trustability by extracting text related to a given diagnosis based on existing ontologies. Then the given diagnosis along with the extracted sentences are treated as a fine-grained data point for deep training and prediction. Since the approach tracks verifiable human knowledge, the extracted sentences based on the knowledge can be used as evidence for ICD code classification. To demonstrate the effectiveness and efficiency of the approach, we used two subsets of the Medical Information Mart for Intensive Care III (MIMIC-III) dataset for case studies. The experimental results show that the classifier outperforms existing approaches and has a strong ability to distinguish between the different uses of similar terminologies.

Keywords: Automated medical coding; ICD code; ontologies; sentence extraction; deep learning.

1. Introduction

Healthcare is an essential and constantly growing field that has a significant and direct impact on each of our lives. According to the Centers for Medicare & Medicaid Services (CMS), U.S. healthcare spending is projected to grow faster than average GDP growth

during the 2022-2031 period, resulting in an increase in healthcare spending as a percentage of GDP from 18.3% in 2021 to 19.6% in 2031 [1]. One important aspect of healthcare is medical coding, which involves the assignment of well-defined codes for healthcare diagnoses, procedures, and medical services. These well-defined codes eliminate ambiguity and help standardize communication between healthcare organizations and external billing agencies such as insurance companies. The most widely used standards for medical codes, and the standards we will use in this paper, come from the International Classification of Diseases (ICD) [2]. These codes are critical in the healthcare process as they ensure invoices are paid properly and may serve as a reference for patients' future treatments. The assignment of these codes, however, can be a time-consuming and resource-intensive process. Due to the free-form nature of patient records, direct methods like keyword matching are rarely sufficient for medical coding. Often, a deeper understanding of healthcare and the code sets themselves is required to accurately code patient records. This means that anyone involved in medical coding must be well-educated and trained. Additionally, as the healthcare industry continues to grow, especially in the wake of global health crises such as the COVID-19 pandemic, the volume of patient records is staggering. For many hospitals, automating parts of the medical coding process could be an important step forward.

In recent years, the healthcare industry has continued to shift to the use of electronic healthcare records (EHRs), supported by government-funded incentives such as the HITECH Act, which rewards hospitals for moving to electronic records [3]. As more and more records are in digital formats, automation of healthcare processes, including medical coding, has become more tractable. This is due in part to the fact that text-based patient records can now be used directly in digital format for natural language processing. The main advantage of digital records, however, is that they can be more easily de-identified and shared amongst communities of researchers and professionals. The volume and shareability of EHRs has led to the release of large-scale de-identified datasets such as the Medical Information Mart for Intensive Care III (MIMIC-III), which includes over 40,000 patient records [4]. These datasets have enabled researchers to apply data-driven methods to dozens of healthcare-related tasks, including medical coding. Recently, deep-learning classifiers for medical coding have been successful and have produced state-of-the-art results in medical coding tasks [5, 6, 7]. However, these models, despite their high performance, tend to suffer in terms of explainability and trustability, because they often accept the entire text document of a given patient record and output one or more medical codes without further explanation. Clearly, this limited output is not conducive to trust between patients and healthcare providers using automated coding tools. In this paper, we propose a fine-grained, evidence-based approach that utilizes the loose structure of patient records to perform an intermediate evidence-gathering sentence extraction step prior to classification. Doctor's notes typically contain two important sections: a main section

containing completely free and natural text, where the doctor describes the patient's condition in plain language; and a discharge diagnosis section, where the doctor more systematically lists the diagnoses associated with the patient visit. Often, these systematically listed diagnoses contain valuable information but are not sufficient for medical coding. As a result, many approaches treat the diagnosis text as part of the document without giving special consideration [5, 6]. In our approach, we start with the listed diagnoses and the text semantically related to each diagnosis to form independent data points and classification tasks. For each diagnosis in a given patient record, we perform human knowledge-based sentence extraction to generate a unique view of the document that includes only the sentences discussing concepts related to the diagnosis. The extracted sentences are then passed to an attention-based deep learning classifier, which predicts the medical code for the diagnosis. This fine-grained evidence-based approach not only makes the classification task easier, but also allows us to trace the evidence used by the classifier through the extracted sentences, even further back to the original human knowledge used in extracting the sentences.

This work significantly extends our previously proposed automated medical coding approach for fine-grained ICD code assignment. In our prior research, we conducted initial investigations leveraging domain expertise to extract semantically related sentences from doctor's notes [8]. We utilized a Long Short-Term Memory (LSTM) artificial neural network to identify medical codes for diagnoses. In this paper, we present a formal framework for automated ICD code assignment using knowledge-based sentence extraction. Instead of using LSTM for classification, we adopted Bidirectional Encoder Representations from Transformers (BERT) to predict medical codes. We elucidate the process of fine-tuning a pretrained BERT model and thoroughly evaluate the classifier to demonstrate the effectiveness of our proposed approach.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 presents a formal framework for explainable ICD code assignment and provides the procedures for extracting sentences from doctor's notes that are semantically related to a diagnosis. Section 4 discusses the procedure of fine-tuning Med-BERT for automated medical coding. Section 5 presents the case studies and their analysis results. Section 6 concludes the paper and mentions future work.

2. Related Work

Research on automated medical code assignment to doctor's notes began with simple rule-based approaches. These systems make coding decisions based on rules specified by experts for ICD coding. For example, if a document contains a keyword, it is assigned the appropriate code, provided that some closely related keywords exist in its context. Goldstein et al. proposed a rule-based approach to automated medical coding that extracts lexical elements for further analysis in addition to searching for keywords [9]. In their

approach, lexical elements are used in rule-based systems to detect and represent negation, synonyms, and other linguistic features that help in decision making. Particularly in the early days of automatic coding research, rule-based systems allowed researchers to begin to address the problem without collecting large amounts of data. However, rule-based systems produce unsatisfactory results due to the rigidity of rules, which are insufficient to express complex natural language. With the addition and de-identification of electronic health records (EHRs), medical data becomes more widely available and the need to optimize and learn from training data becomes more feasible and popular. Researchers started to move from handmade rules to statistical methods. Farkas and Szarvas used decision tree models to supplement the expert-designed rules, thereby increasing the flexibility of the approach [10]. Medori and Fairon introduced a semi-automatic method for ICD coding using a Naïve Bayes classifier [11]. Later, in an influential paper, Perotte et al. demonstrated the potential of machine learning methods for this problem by applying SVM-based models to the problem of automatic coding [12]. While these methods provided the basis and rational for automatic ICD coding research, they have since been superseded by more sophisticated machine learning methods to provide higher performance. The approach presented in this paper incorporates some of the ideas from the original rule-based approach to improve comprehensibility while using deep learning models to improve performance. Fundamentally, a method based entirely on rules designed by experts and their results can be understood by humans, which is an advantage it has over deep learning methods. The proposed method adds human-understandable steps to deep learning classifiers, thereby refocusing on the explainability of the method results while gaining performance advantages of using more sophisticated classifiers.

Machine learning methods are receiving increasing attention as large EHR datasets have become publicly available. MIMIC-III, a public dataset containing the records of more than 40,000 inpatients at Beth Israel Deaconess Medical Center, provides a valuable source of labeled data for the research community to address the problem of automated ICD coding [4]. Utilizing this dataset and others that provide large amounts of training data, researchers have begun to explore deep learning methods to drive automatic coding performance to current levels. There has been a strong interest in designing a deep learning model with optimal performance on coding problems. Li et al. extended a straightforward convolutional neural network (CNN) for processing the doctor's notes text using a document-to-vector technique [5]. This technique allowed their approach to supplement the local features (words, phrases) of given doctor's notes with vectors representing global features (topics, main ideas). Other researchers have extensively explored the benefits of incorporating attention into coding performance. One popular form of attention is labeled attention, which allows a model to pay attention to differently labeled text. Shi et al. devised a model to achieve high performance using this kind of attention [6]. In addition, their model builds a representation from characters to words and word sequences, which makes

the method more robust to spelling errors and abbreviations in real doctor's notes. Baumel et al. first encoded sentences using a model based on gated recurrent units; they then use the sentence encodings as inputs to a deep learning classifier that utilizes attention to improve performance [13]. Wu et al. incorporated joint attention in their approach to provide attention at the word level and label level [7]. Following the original attention-based designs, large-scale language models have also been used and succeeded in ICD auto-coding problem. For example, many successful approaches have used Google's attention-based transformer model BERT, sometimes supplemented with labeled attention [14]. The strength of BERT lies not only in its self-attention-based architecture, but also in its pre-trained knowledge-based model. Biseda et al. used a version of BERT, called Clinical BERT, which was pre-trained on EHR data to perform ICD auto-coding [15]. To circumvent the issue of small input size in Clinical BERT, their approach produces encodings for one sentence at a time, and then use these sentence encodings as input to a CNN that generates code assignments. Heo et al. used a similar method, first generating encodings for sentences and then using them as elements in an input sequence for a sequential attention-based classifier [16]. Mayya et al. introduced labeled attention in a BERT-based model, demonstrating the potential of combining a large-scale language model, such as BERT, with new components or procedures [17]. While these approaches are key to improving performance in ICD automated coding tasks, they lack an appropriate emphasis on the understandability and interpretability of the classification results. In contrast, our fine-grained, evidence-based approach supports the ability to interpret the classification results, bridging the gap in human comprehensibility in healthcare settings.

Most existing approaches treat the task of automatic ICD code assignment as a multi-label classification problem. That is, given an instance of doctor's notes, these approaches view the doctor's notes as a single text input and output the ICD code assignments for the entire text. We refer to these approaches as "coarse-grained" because they deal with the classification problem at a higher level. Multi-label classification is generally more difficult than single-label classification. This is especially true as far as the ICD coding problem is concerned. That is, due to the extremely large number of labels (more than 1000 unique codes), the label space (2^n, where n is the number of labels) for multi-label classification quickly grows to be intractable when using larger code sets. The large imbalances found in medical diagnoses also make some codes grossly underrepresented. To make matters worse, many ICD codes are extremely similar, which further complicates the classification process. To address these difficulties, coarse-grained approaches often constitute highly complex deep learning models, whose main purpose is to improve classification performance. These are known as *black-box* methods that abstract important details of the coding process from users. Many of these approaches output only their coarse-grained ICD code predictions with no further explanations [5, 6, 15, 16]. Along with model complexity, this lack of explanations can hinder users' ability to contextualize and

understand the model's prediction behavior and the resulting codes. Ultimately, this can compromise the user's interpretation and usage of the results, potentially leading to incorrect coding. In addition to incorrect coding, this lack of explainability may also raise general concerns about the trustworthiness of the coding process. This is particularly important in the healthcare industry, where the use of any tool can have a significant impact on patient health and safety of patients, and must therefore be based on trust. In this paper, we outline a fine-grained approach that assigns ICD codes at the level of diagnoses, rather than at the level of the entire doctor's notes. This allows code prediction to be performed as a series of single-label classifications rather than a single multi-label classification, thus reducing the difficulty of classification and providing an opportunity to improve performance. Furthermore, our fine-grained evidence-based approach establishes a direct link between each automated ICD code assignment and the original diagnosis text, and gathers additional evidence from the free text of doctor's notes to support predictions. The collected evidence can also be returned to the user to provide context for automated code assignments and to clarify doubts or correct errors.

In addition to efforts to improve the classification performance of automated ICD coding tasks, much attention has been directed toward the usability and explainability of real-world coding applications. Montalvo et al. designed a user interface that provides a convenient way to interact with the underlying classification methods [18]. Some of the features implemented in this interface include selecting or modifying code assignments and allowing the user to highlight text to show its correspondence to the code. Siangchin and Samanchuen proposed to improve querying or browsing ICD database through the use of an interactive Chatbot [19]. The proposed system allows users to request code descriptions and explanations in plain language, which would greatly facilitate the work of medical coders. Besides ease of use, the incorporation of human knowledge can provide a human-understandable basis for the approach, thereby increasing its comprehensibility and credibility. Not only that, but the use of knowledge is often helpful for labels with low training data frequency (which is often the case with medical coding), as it can supplement the limited learning during training with some external knowledge. Lui et al. generated vector representations of ICD codes using the official ICD ontology [20]. These vector representations account for the groupings and relationships of ICD codes and represent their meaning in a vector space. Such work can complement our approach, especially in terms of matching text to ontology concepts. Teng et al. used the written English descriptions accompanying each ICD code to attend differently to doctor's notes as they passed through a deep learning classifier [21]. Almagro et al. used the SNOMED-CT clinical terminology as the knowledge base for their approach, which encodes doctor's notes and then matches them within SNOMED-CT using similarity in vector representation [22, 23]. Bai et al. utilized additional knowledge mined from Wikipedia to support model performance [24]. After conventional classification, their approach scores the similarity of

doctor's notes to various diagnostic descriptions on Wikipedia. Sonabend et al. mined text from five web sources to identify "concept-unique identifiers" that strongly indicates the presence of a specific ICD code [25]. Vector representations of these identifiers are compared to vector representations of doctor's notes and matched on the basis of similarity. Teng et al. extracted relevant concepts from the general knowledge graph Freebase, which were then used to apply attention to the output of a CNN [26]. Zeng et al. supplemented the code assignment task by learning on the relevant Medical Subject Heading (MeSH) task [27]. As with assigning ICD codes to doctor's notes, the MeSH task involves tagging documents with appropriate labels. That is, scientific articles must be labeled with medical topics and keywords to make them easier to organize and navigate. There is an overlap between the MeSH and ICD autocoding domains, so learning one task can help to improve the performance of the other. While more and more ICD coding datasets are being released as research progresses, there are still not enough training examples to learn thousands of codes. Techniques that incorporate external knowledge could be an important step in supplementing limited training data as we test our approach with more realistic and larger sets of codes. Our approach uses external knowledge in the form of ontologies; however, transfer learning techniques or other knowledge sources, such as some of the methods described above, can complement our approach and increase the efficiency of future work.

3. Fine-Grained Explainable ICD Code Assignment

3.1. *A Framework for Explainable ICD Code Assignment*

In this paper, we present a fine-grained evidence-based approach for assigning ICD codes to diagnoses in doctors' notes. In addition to the subject and admission identifier, doctors' notes contain two key elements for ICD code assignment. Namely, they include a list of diagnoses showing the main visit findings and a free text section that describes in more detail the doctor's impressions of the patient's condition. Several key definitions of ICD code assignment are now given as follows.

Definition 3.1 Doctor's Notes. Doctor's notes Θ is a 4-tuple (*SID*, *AID*, *SS*, *SD*), where *SID* is the subject identifier of the given patient; *AID* is the hospital admissions identifier of the hospital visit associated with the patient; *SS* is the list of sentences contained in the free text of the doctor's notes Θ; and *SD* is the set of diagnoses provided with the notes.

Definition 3.2 Coarse-Grained Data Point. A coarse-grained data point D_{coarse} associated with doctor's notes Θ is a 2-tuple (*SD*, *SS*), where *SD* and *SS* are $\Theta.SD$ and $\Theta.SS$, respectively. The coarse-grained data point treats the diagnosis and free text in the doctor's notes as a complete string required for machine learning.

Definition 3.3 ICD Code Set. An ICD code set Ω is a set of standardized medical codes that represent diseases or procedures along with their descriptions. A subset of Ω can be assigned to doctor's notes for medical coders and billers to report healthcare diagnoses and procedures.

Definition 3.4 Coarse-Grained Classification Model. A coarse-grained classification model M_{coarse} is defined as the following mapping function, where 2^{Ω} is the power set of the ICD code set Ω.

$$M_{\text{coarse}}: D_{\text{coarse}} \rightarrow 2^{\Omega}.$$

The objective of assigning ICD codes is to generate a list of medical codes that accurately reflect the doctor's findings following a patient visit. In our context, these ICD codes serve as precise identifiers for specific illnesses. They are organized according to standards and have a higher degree of specificity than typical natural language. These codes are organized hierarchically by category, and to find a specific code, one needs to traverse from one category to another, progressively narrowing down the classification. For instance, to obtain the code "487.0: Influenza with pneumonia," one must navigate from the broader category "460-519: Diseases of the Respiratory System" to the more specific category "480-488: Pneumonia and Influenza," where code 487.0 is located. In the coarse-grained approach, the training and prediction data points consist of the entire text of diagnoses and the doctor's free text notes, as defined in Definition 3.2. A coarse-grained data point serves as an input to a multi-label classifier that can output all code predictions simultaneously. However, this machine learning approach can be considered as a "black-box" as it obscures important details of the classification process. This lack of transparency hinders its adoption in sensitive areas such as healthcare and medicine. Without being fully explained to and trusted by physicians, patients and billers, these tools are unlikely to be successfully adopted.

To address the challenges posed by black-box methods and to enhance explainability of the classification process, we present a *fine-grained* code assignment approach that utilizes fine-grained data points for training and prediction. A few key definitions of our proposed explainable fine-grained approach are now given as follows.

Definition 3.5 Fine-Grained Data Point. A fine-grained data point D_{fine} associated with doctor's notes Θ is a 2-tuple (*DIAG*, *SRS*), where *DIAG* ∈Θ.*SD* is a single diagnosis and *SRS* ⊂ Θ.*SS* is a set of sentences that are semantically related to the diagnosis *DIAG* based on the given domain knowledge.

Definition 3.6 Fine-Grained Classification Model. A fine-grained classification model M_{fine} is defined as the following mapping function, which takes a fine-grained data point and outputs a single ICD code prediction from the ICD code set Ω.

$$M_{fine}: D_{fine} \rightarrow \Omega.$$

Figure 1 shows a general overview of the fine-grained explainable ICD code assignment method. Under the fine-grained approach, a coarse-grained data point associated with doctor's notes Θ is processed into k separate fine-grained data points, where k is the number of diagnoses in the coarse-grained data point. For each diagnosis $diag_i$, where $1 \leq i \leq k$, we extract a set of related sentences srs_i from the free text of doctor's notes, which contain only sentences that are semantically related to $diag_i$ based on the domain knowledge. The 2-tuple ($diag_i$. srs_i) is used as a fine-grained data point as an input to the fine-grained classifier. Note that the classifier can be run sequentially on each fine-grained data point d_i and generate an ICD code $code_i$ for d_i. Alternatively, we can run k fine-grained classifiers in parallel and generate the ICD codes in a more efficient manner. Note that it is possible for two fine-grained data points to map to the same medical code. The combined set of codes $\{code_1\} \cup \{code_2\} \cup \ \ldots \cup \{code_k\}$ constitute the ICD code prediction results for the doctor's notes Θ.

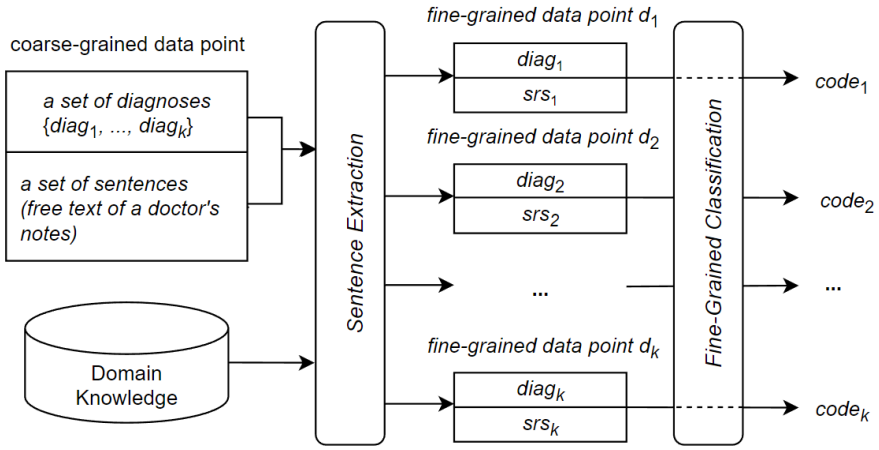

Fig. 1. A general overview of fine-grained explainable ICD code assignment.

Semantically extracted sentences serve two main purposes. Firstly, they provide vital information needed for accurate classification. Secondly, they serve as a pool of "evidence" for assigning a specific code to a diagnosis. Thus, the fine-grained explainable ICD code assignment methodology is firmly rooted in *evidence-based* principles. When necessary,

the set of semantically related sentences extracted for a particular diagnosis can be reviewed manually to gain insight into the information that contributes to the ICD code prediction process. This enhanced explainability gives users the ability to resolve any uncertainties or make manual adjustments as needed to ensure clarity and accuracy.

Note that the number of diagnoses k included in doctor's notes may not necessarily match the number of ICD codes required to fully code the doctor's notes. In other words, there are special situations where an ICD code is not represented in the given set of diagnoses, or a particular diagnosis may correspond to multiple codes. Consequently, manual intervention becomes necessary to ensure comprehensive code assignment in such cases. Given that the primary objective of the proposed research is to improve the accuracy and explainability of medical coding, special cases such as these are beyond the scope of this study.

3.2. *Knowledge-Based Sentence Extraction*

Knowledge-based sentence extraction is one fundamental step of the proposed fine-grained approach. In this step, we process doctor's notes separately for each diagnosis. Once an individual diagnosis is selected, the first task is to extract semantically related sentences to that diagnosis to form a fine-grained data point for fine-grained classification. We now provide a few key definitions for extraction of semantically related sentences.

Definition 3.7 Concept. A concept Λ is a 2-tuple (*CID*, *DES*), where *CID* is a concept identifier and *DES* is a description of the concept. The semantics of a concept must be unambiguously defined.

Definition 3.8 Ontology. An ontology Φ is a 2-tuple (*CON*, *REL*) representing domain knowledge as a directed graph. Φ.*CON* is a set of concepts, which are the vertices of the graph and Φ.*REL* is a set of relations encoded as a triple in the form <*concept*, *relation*, *concept*>, which are the edges of the graph.

Definition 3.9 Semantically Related Sentence. Given a domain ontology Φ, let C_s be a set of concepts defined in Φ related to sentence *s*, and C_{diag} be a set of concepts defined in Φ related to diagnosis *diag*. Sentence *s* is semantically related to *diag* under Φ if the two sets of concepts C_s and C_{diag} overlap, i.e., $|C_s \cap C_{diag}| > 0$.

In order to determine whether a sentence is semantically related to a diagnosis, we need some existing source of knowledge to encode medical concepts and their relations. Our approach utilizes ontologies as formal representations of human knowledge, from which we can identify the set of concepts C_s and C_{diag}. According to Definition 3.8, an ontology represents knowledge in the form of a directed graph of concepts and relations.

The relations in a given ontology follow a strict internal logic and can be used to reason about the related concepts. In the case of the medical ontologies suitable for this methodology, concepts can be diagnosable diseases, symptoms or treatments.

Due to the open-ended nature of ontology design, the implementation details of the related concept identification step vary for each ontology. In this paper, we use an example of a healthcare ontology Φ to demonstrate the related concept identification process. Fig. 2 shows a small portion of the ontology Φ used in this paper. The boxes represent concepts in Φ.*CON* of the following types: <disease>, <symptom>, <treatment>, and <body part>. Note that many concepts and their synonyms are omitted from the figure for simplicity. The example ontology features structural relations such as <superclass of> and some property relations such as <has symptom>, <has treatment>, and <affects body part>.

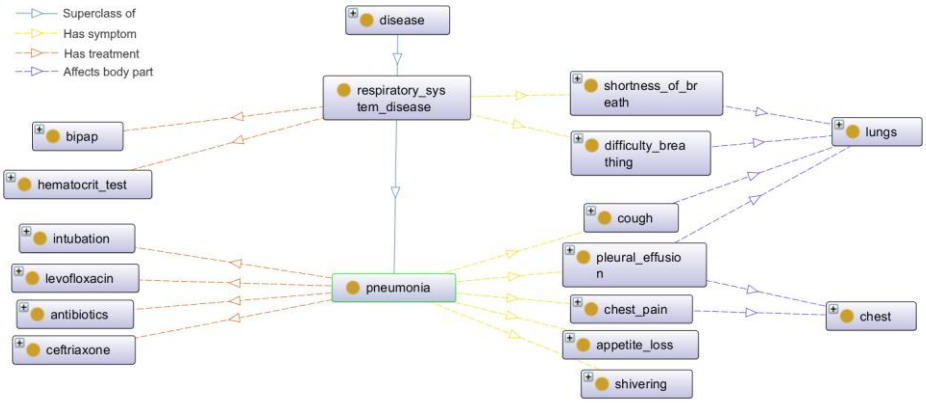

Fig. 2. An example medical ontology that can be used to generate related terms for the diagnosis “pneumonia”.

We use the relations defined in Φ.*REL* to reason about which concepts are related to a given diagnosis. Referring to Fig. 2, suppose our target diagnosis is “pneumonia.” Following the design of the ontology, we identify the concept “pneumonia” and then extend outward through the association relations of the concept "pneumonia" to get other related concepts. The first step is to find the diseases’ superclass “respiratory system disease” through the “superclass of” relation. In this example, we search all the superclasses up to the root class "disease" because it no longer contains any useful information. Let the set of concept “pneumonia” and all of its superclasses be C_{super}. Then, for each concept λ in C_{super}, we add λ and its related concepts that are associated with λ to C_{diag}. For example, the concept “cough” is added to C_{diag} because it is immediately related to the concept “pneumonia” by the property relation “has symptom.” We further identify concepts that have an “affects body part” relation with each identified symptom concept

and add them to C_{diag}. Finally, we identify all concepts that have "has treatment" relation with each concept in C_{super} and add them to C_{diag}. Algorithm 1 formalizes the procedure for generating semantic concepts related to a diagnosis.

Algorithm 1: Generating Semantically Related Concepts to a Diagnosis

Input: A diagnosis *diag*; a medical ontology Φ
Output: A set of related concepts C_{diag}

```
1.  Identify concept c_diag in Φ.CON for diag and its synonyms
2.  Initialize C_super and C_remaining to a set of concepts { c_diag }
3.  while C_remaining ≠ ∅
4.     remove a concept λ from C_remaining
5.     for each relation of type <superclass, "superclass of", λ> in Φ.REL
6.        if superclass is not the root class of Φ
7.           add superclass to C_super and C_remaining
8.  Initialize C_diag = ∅
9.  for each concept λ in C_super
10.    add λ to C_diag
11.    for each relation of type < λ, "has symptom", symptom> in Φ.REL
12.       add symptom to C_diag
13.       for each relation of type <symptom, "affects body part", part> in Φ.REL
14.          add part to C_diag
15.    for each relation of type < λ, "has treatment", treatment> in Φ.REL
16.       add treatment to C_diag
17. return C_diag
```

Once all related concepts to diagnosis *diag* under ontology Φ are obtained, C_{diag} can be used to extract semantically related sentences from Θ.*SS*, i.e., the list of sentences contained in the free text of doctor's notes Θ. For each sentence *s* in Θ.*SS*, we use a similar algorithm as defined in Algorithm 1 to generate the set of concepts C_s in Φ. According to Definition 3.9, sentence *s* is extracted as a semantically related sentence if there is an overlap of concepts in C_s and C_{diag}. Fig. 3 shows examples of sentences that are extracted to be used as evidence for the diagnosis "pneumonia" during the code assignment.

Detected **pleural cavity effusion**.
Visible **shivering** and **dyspnea** complicated speech.
Patient was put on **bipap** and retained overnight…

Fig. 3. Example sentences extracted for "pneumonia" using the related concepts under an ontology.

3.3. *Prioritization of Extracted Sentences by Sample Statistics*

The classifier used in this paper is based on the BERT architecture with an input limit of 512 tokens. This limit restricts the number of sentences that can be accommodated since many words occupy more than one token. When the input sequence exceeds 512 tokens, tokens beyond the 512th position are truncated and discarded. Thus, the accuracy of the classification depends heavily on which parts of the input are retained. To address this challenge, we use Z-scores to determine the most relevant concepts to be retained. Firstly, we compare the Z-scores of different samples with distinct concepts against a predefined threshold to identify and eliminate insignificant concepts. Secondly, we arrange the sentences in descending order based on their Z-scores. This reordering ensures that the most related sentences fit within the 512-token limit without being truncated.

In our approach, we count the number of occurrences of a concept c in random samples of the doctor's notes, deriving statistics that describe its frequency distribution in random samples. We then select the subset of notes associated with a particular diagnosis *diag*. We compare the occurrence of c within this special subset to its occurrence in random samples. If the special subset shows a much higher number of occurrences of c, it indicates that there is a relationship between *c* and the selection criteria using diagnosis *diag*. In this case, *c* may be more related to the diagnosis and therefore more useful for identifying *diag*.

Let x_c be a random variable denoting the number of occurrences of concept *c* in a doctor's notes. Let $x_{c,i}$ be the number of occurrences of concept *c* in doctor's notes Θ_i, where $0 \leq i \leq N$ and *N* is the total number of doctor's notes in a training dataset. The mean μ_{x_c} and standard deviation σ_{x_c} for x_c can be derived as in Eq. (1) and Eq. (2).

$$\mu_{x_c} = \frac{\sum_{i=1}^{N} x_{c,i}}{N}, \tag{1}$$

$$\sigma_{x_c} = \sqrt{\frac{\sum_{i=1}^{N} (x_{c,i} - \mu_{x_c})^2}{N-1}}, \tag{2}$$

With μ_{x_c} and σ_{x_c} established across the entire training data set, we can now start to reason about subsets or samples of the training data. Let $\bar{x}_{c,n}$ be a random variable representing the mean of x_c in a random sample of a given size n. According to the Central Limit Theorem, sample means $\bar{x}_{c,n}$ should be normally distributed given a sufficiently large sample size n. Under the assumption that a given $\bar{x}_{c,n}$ is normally distributed, the mean of sample means $\mu_{\bar{x}_{c,n}}$ and the standard deviation of sample means $\sigma_{\bar{x}_{c,n}}$ in the training data set are defined as in Eq. (3) and Eq. (4).

$$\mu_{\bar{x}_{c,n}} = \mu_{x_c} \tag{3}$$

$$\sigma_{\bar{x}_{c,n}} = \frac{\sigma_{x_c}}{\sqrt{n}} \tag{4}$$

These sample statistics show what a typical random sample set of size *n* looks like in terms of x_c (i.e., the number of occurrences of related concept *c* in a doctor's notes). Based on these sample statistics, we can check whether a particular sample set *S* of size *n* is also typical, or whether it is unlikely to be typical, or whether it is an outlier of the distribution. To do this, we can generate a *Z*-score for sample set *S* as in Eq. (5).

$$z_n = \frac{\bar{x}_{c,n} - \mu_{\bar{x}_{c,n}}}{\sigma_{\bar{x}_{c,n}}} \tag{5}$$

where $\bar{x}_{c,n}$ is the mean of x_c in sample set *S*.

Given a normal distribution, a *Z*-score gives a measure of an observation's distance from the mean, in terms of the distribution's standard deviation. This distance also corresponds to the unlikeliness of the observation: a *Z*-score with a high magnitude indicates an observed value far away from the mean, an unlikely outcome; while a *Z*-score close to zero indicates a value close to the mean, a likely outcome. Note that in our case, a *single observation* refers to the mean occurrences of a given concept across *a sample of doctor's notes*, not one individual instance. Thus, the higher the *Z*-score for a given sample, the more frequently the particular concept is used in that sample. With this in mind, we select a group of doctor's notes related to a particular concept and check if the usage of the concept is higher than normal. More specifically, we select all the notes related to a given diagnosis *diag* and generate a *Z*-score for each concept $c \in C_{diag}$. The generated *Z*-scores are stored in a HashMap *Z* such that *Z*(<*diag, c*>) gives the *Z*-score of concept *c* in the subset of doctor's notes containing *diag*, where <*diag, c*> is the encoding of diagnosis *diag* and concept *c*. Algorithm 2 formally describes the steps of this process. We first generate a list of all unique diagnoses and their associated concepts in the training dataset. Then, we count all related concepts in all doctor's notes to establish the population statistics μ_{x_c} and σ_{x_c}. Finally, we select the set of doctor's notes containing each diagnosis and compute a *Z*-score for each concept associated with the diagnosis. These scores are stored in a HashMap *Z* so that we can reorder the extracted sentences and eliminate unrelated ones for a diagnosis by looking up the *Z*-scores of the related concepts in a training or test data point. Note that if the *Z*-score for a diagnosis-concept pair (*diag, c*) is less than 0, this indicates that the concept typically occurs infrequently in doctor's notes containing *diag*; therefore, there is no need to record its *Z*-score, and only non-negative *Z*-scores are included in the HashMap. If new training data is introduced after the initial generation of HashMap *Z*, it can be easily updated to reflect the new language usage. Individual scores for existing related concepts can be recalculated following the same procedure as described above. In addition, if new concepts are introduced into the medical ontology used to derive the related concepts, new scores can also be computed and stored as new key-value pairs in the HashMap *Z*.

Algorithm 2: Generating HashMap *Z*

Input: Training set of doctor's notes *TS*, a medical ontology Φ
Output: HashMap *Z*, where *Z*(<*diag*, *c*>) is the Z-score of concept *c* in the subset of doctor's notes containing *diag*.

1. Initialize $C_{all} = \varnothing$ and $D_{all} = \varnothing$
2. **for each** doctor's notes Θ in *TS*
3. **for each** diagnosis *diag* in Θ.*SD*
4. $D_{all} = D_{all} \cup \{diag\}$
5. Invoke Algorithm 1 with inputs (*diag*, Φ) and return C_{diag}
6. **for each** concept *c* in C_{diag}
7. $C_{all} = C_{all} \cup \{c\}$
8. **for each** *c* in C_{all}
9. **for each** Θ in *TS*
10. Count the number of occurrences of *c* in Θ.*SS*
11. Calculate μ_{x_c} as in Eq. (1)
12. Calculate σ_{x_c} as in Eq. (2)
13. Initialize a HashMap *Z* with null <*key*, *value*> pair
14. **for each** *diag* in D_{all}
15. **for each** *c* in C_{diag}
16. Create a sample set S_{diag} of doctor's notes containing *diag*
17. Let *n* be the size of the sample set S_{diag}
18. Calculate $\mu_{\bar{x}_{c,n}}$ as in Eq. (3)
19. Calculate $\sigma_{\bar{x}_{c,n}}$ as in Eq. (4)
20. Calculate z_n for sample set S_{diag} as in Eq. (5)
21. **if** $z_n \geq 0$
22. Encode *diag* and *c* into key <*diag*, *c*>
23. Add (<*diag*, *c*>, z_n) to the HashMap *Z*
24. **return** *Z*

Concepts with higher Z_n scores appear more frequently in the subset of doctor's notes containing *diag* and can be more closely related to *diag*. Thus, we can now threshold and reorder the extracted sentences based on the *Z*-scores of the various concepts to ensure that the most closely related and useful words are emphasized within the constraints of the classifier. We score sentences based on the included concepts with the highest Z-scores and reorder all extracted sentences based on the sentence scores, with the highest scoring sentences coming first. Algorithm 3 provides a formal description of the sentence reordering process, provided that *Z*-scores for various diagnosis-concept pairs have been generated using Algorithm 2.

Algorithm 3: Reordering Extracted Sentences Based on their Z-scores

Input: A list of extracted sentences *SRS* that are semantically related to diagnosis *diag*, HashMap *Z*, acceptable Z-score threshold *t*, where $t > 0$
Output: A list of reordered semantically related sentences *SRS'*

1. Initialize *QSRS* to an empty priority queue
2. **for each** sentence *s* in *SRS*
3. Let C_s be a set of concepts mentioned in sentence *s*
4. Let $z_{max} = -1$
5. **for each** concept *c* in C_s
6. **if** $Z(<diag, c>)$ returns *null*, let $z_{diag} = 0$
7. **else** $z_{diag} = Z(<diag, c>)$
8. $z_{max} = max(z_{max}, z_{diag})$
9. **if** $z_{max} > t$
10. Insert sentence *s* into *QSRS* with priority z_{max}
11. Convert *QSRS* to a list of sentences *QRS'*
12. **return** *SRS'*

By extracting the list of sentences that are semantically related to diagnosis *diag* and reordering them according to importance, we can now form the fine-grained data point d_{fine} defined in Definition 3.5. The data point d_{fine} can be passed on to the classifier M_{fine} to predict the specific ICD code for *diag*. Meanwhile, the extracted semantically related sentences are provided to the user to resolve any doubts and address any concerns if they may have. In healthcare, where the stakes are very high not only from a financial perspective, but also from a health and safety perspective, this additional explainability in medical coding is crucial.

4. Fine-Tuning Med-BERT for Automated Medical Coding

To support automated medical coding, we developed a single-label classifier using Med-BERT, a pre-trained contextual embedding model on a structured electronic medical record dataset. Med-BERT is based on BERT, an attention-based encoding model with state-of-the-art performance in several areas of natural language processing. In traditional methods, words are separated and converted into word embeddings, which are then mapped into a space based on their meanings. However, traditional embedding methods can only map one representation for each word. In real languages, a word often has multiple meanings, and the appropriate meaning is determined by the context. This means that traditional word embedding methods create ambiguity in phrases such as “river bank” and “bank statement”, where the bank is given the same embedding despite having a different meaning in each phrase. BERT contains a series of attention-based encoders that allow words to “focus” on other words in the sequence, meaning that the interpretation of a word

is influenced by the context provided by other words in the sequence. This allows for a more flexible and human-like interpretation of text sequences.

Pre-trained models like Med-BERT incorporate existing knowledge generated during the pre-training phase, where tasks are designed to drive model learning. For example, Med-BERT was pretrained on 28 million patient records extracted from the Cerner Health Facts database. After the pre-training phase, these models can be "fine-tuned" in a downstream task to quickly learn specific problems using their existing general knowledge. In our case, the downstream task is the ICD code classification task described in Section 3.1. Given a data point consisting of a diagnosis and its semantically related sentences, our model must produce an ICD code assignment prediction. Although Med-BERT already knows a lot about medicine and healthcare, it must still learn how to apply this knowledge to any new task through this fine-tuning process, which is much faster than training from scratch. To generate single-label classifications, a classification head (a dense layer with one output for each ICD code to be predicted) needs to be attached to the end of the network and take advantage of Med-BERT's rich understanding of the input data.

During the fine-tuning process, the Med-BERT-based classifier is trained like a regular neural network, with losses calculated for predictions and backpropagated through the classification head into Med-BERT's transformers. As a result, the weights of the entire model are updated. Due to the severe data imbalance present among diagnoses and their corresponding codes, we use stratified random sampling to select the training/testing partition. In stratified random sampling, the number of samples chosen for each class is proportionate to the frequency of that class. Thus, smaller classes are not randomly underrepresented or excluded. During training, we use 5-fold cross-validation to diagnose model performance, with folds selected using stratified random sampling. We use focal loss as the loss function, which is a special type of loss function derived from the object detection problem in computer vision, where data imbalance is a central issue. The focal loss for multi-class single-label classification is formulated as (7).

$$L_F = -\sum_{i=1}^{n} t_i (1 - p_i)^{\gamma} \log(p_i), \tag{7}$$

where n is the number of classes (ICD codes) in the classification task, t_i is the truth value of class i (1 if the example belongs to class i and 0 if the example does not belong to class i), p_i is the predicted probability of the model for class i, and γ is a nonnegative focusing parameter that determines the degree of weight reduction for easy examples [28]. Focal loss underweights these well-classified "background" examples and places more emphasis on the poorly classified examples, allowing the learning to center around less frequent classes and improving results on imbalanced data.

For the BERT-based models, the ADAM optimizer was used with an initial learning rate of 5*e*-5 as suggested by the original work [14]. These models were fully trained after

3-5 epochs, as further training causes degradation of model performance due to overfitting, which includes "catastrophic forgetting" of applicable pre-trained knowledge [29]. After 5 training epochs, the models with the best validation performance were selected. Fig. 4 provides an overview of the classifier used for one diagnosis. Note that to simplify the diagram, the corresponding discharge diagnosis is not shown as part of the fine-grained data point d_{fine}.

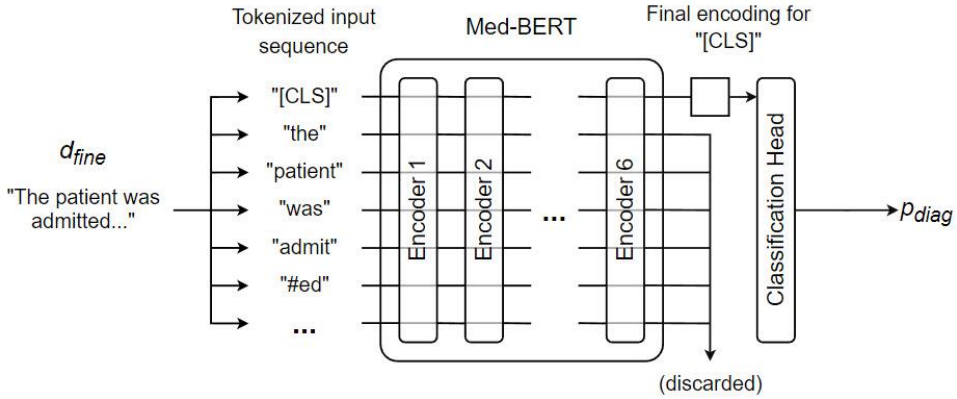

Fig. 4. A full procedure for classifying a fine-grained data point.

As shown in Fig. 4, the original input sequence is tokenized and passed through Med-BERT. The final encodings of all tokens are discarded except for the special "[CLS]" token. The embedding of the special "[CLS]" token is then used as input to the classification head, which outputs a single label, i.e., the ICD code prediction p_{diag} for the data point d_{fine}. While it is not shown in the figure, before tokenization, we need to preprocess the natural language in the input d_{fine}. In traditional natural language processing, the first step is to remove words with little or no information (i.e., removing stop words), and to reduce word inflections that unnecessarily increase vocabulary size (stemming or lemmatization). Unlike traditional models, BERT-based models are often able to strongly characterize and encode seemingly meaningless words based on context, thus making "useless word removal" unnecessary or even detrimental. In addition, BERT operates on "word pieces" rather than entire words. Whereas lemmatization reduces "cough" and "coughing" to the same lemma "cough", BERT's word-piece approach preserves the inflection of the second word by splitting it into two tokens "cough" and "#ing". Therefore, lemmatization is also not necessary with BERT. However, some traditional preprocessing methods still apply. For example, we remove weak punctuation marks such as commas, semicolons, and hyphens, and group numerical identifiers such as those for dates, hospitals, and patients.

Once the initial preprocessing is complete, we can tokenize the input text for use with Med-BERT. We split different words into different tokens based on spaces and punctuation marks. As mentioned earlier, inflected words are split into multiple tokens (e.g., “healthy” is split into “health” and “#y”). BERT also makes use of several special tokens that must be inserted. Firstly, the special “[CLS]” classification token is placed at the beginning of the sequence. The “[SEP]” sentence separator token is placed wherever strong punctuation marks (“.”, “!”, “?”) appear in the original sequence. Different BERT models take sequences of varying lengths as input. To accommodate Med-BERT’s fixed input size of 512 tokens, we add “[PAD]” tokens at the end of the sequence. Fig. 5 shows an example of the necessary tokenization. The example input is tokenized and assigned the aforementioned special tokens where necessary. Once this is done, the sequence of tokens can be fed into the model for prediction.

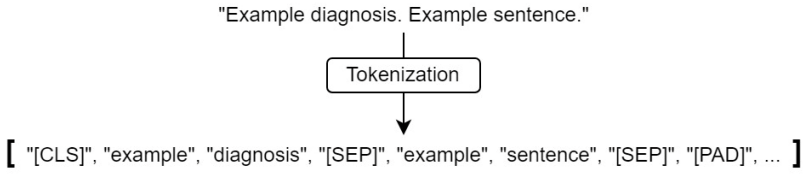

Fig. 5. An example showing the tokenization process required by BERT.

5. Case Studies

5.1 *Typical Use Cases for Training and Classification*

In this section, we first examine typical use cases of the method for training and classification. The data points generated following the steps described in Section 3.2 and their ICD code labels will be used to train the Med-BERT classifier to predict ICD codes. To demonstrate the efficacy of our approach on a variety of data, we present its performance on two datasets corresponding to different subsets of the ICD codes in the MIMIC-III dataset. We chose the set of doctor’s notes for cardiovascular diseases as the first dataset and the set of doctor’s notes for respiratory diseases as the second dataset. Tables 1 and 2 summarize the closely related codes used in the two datasets and the frequency of each code. Both datasets are severely imbalanced, containing one or more classes with extremely low frequencies. Furthermore, both datasets contain similar and related ICD codes, which can complicate classification even for human experts. For example, the respiratory disease dataset contains doctor’s notes that can be mapped to codes for two separate but similar asthma diagnoses: Code 493.90, for unspecified asthma and code 493.20, for chronic obstructive asthma.

Table 1. Frequencies of the cardiovascular disease ICD diagnosis codes.

ICD Code	Diagnosis Name	Frequency
401.9	Unspecified essential hypertension	19.117
427.31	Atrial fibrillation	12,122
428.0	Congestive heart failure, unspecified	11,689
414.01	Coronary atherosclerosis of native coronary artery	11,392
427.1	Paroxysmal ventricular tachycardia	1,635
426.0	Atrioventricular block	492
401.1	Benign hypertension	444

Table 2. Frequencies of the respiratory disease ICD diagnosis codes.

ICD Code	Diagnosis Name	Frequency
518.81	Acute respiratory failure	7,714
486	Pneumonia, organism unspecified	4,747
507.0	Pneumonitis due to inhalation of food or vomitus	3,845
511.9	Unspecified pleural effusion	2,746
496	Chronic airway obstruction	2,348
518.0	Pulmonary collapse	2,058
493.90	Asthma, unspecified type, unspecified	2,023
491.21	Obstructive chronic bronchitis with (acute) exacerbation	1,323
482.41	Methicillin susceptible pneumonia due to staphylococcus aureus	990
512.1	Iatrogenic pneumothorax	859
493.20	Chronic obstructive asthma, unspecified	752

All the training and testing tasks in the following sections were performed on the same machine configured with an NVIDIA GeForce RTX 2060 SUPER with 8 GB of VRAM, an Intel Core i7-9700 CPU, and 16 GB of main memory. The results of the test data were promising for both datasets, as shown in Table 3. Both models achieved accuracies in the mid-nineties. Precision and recall (denoting false-positive and false-negative rates, respectively) exceeded 90% on each dataset. This is especially important in healthcare, where false positives or false negatives can lead to incorrect billing or even inappropriate treatment. Based on precision and recall, both models have F1-scores in the mid-nineties. The F1-scores given are macro-averaged (all classes are equally weighted). We believe these metrics strongly suggest that the proposed approach achieves reliable coding performance even in the presence of real-world problems such as data imbalance and ambiguities arising from the classification of closely related codes. In the next section, we take a closer look at the classification of closely related codes using our approach.

Table 3. Performance of ICD code classifier on two datasets.

Data Set	F1-score	Recall	Precision	Accuracy
Cardiovascular	0.958	0.948	0.969	0.968
Respiratory	0.933	0.926	0.940	0.942

5.2 *Closely Related Medical Codes*

One of the common difficulties with medical coding is that a vague discharge diagnosis can result in a wrong code among a series of related codes. Even a neatly listed diagnosis may not be sufficiently specific and precise to achieve the level of specificity required to assign an ICD code. The main advantage of our approach is that it provides additional details to the classifier by extracting sentences that are semantically related to the diagnosis. With these added details, the classifier is more likely to make a correct classification, even if the original diagnosis is vague and inadequate.

Asthma is one of the diseases where we find this difficulty exists. Although there are variants of asthma with their own separate symptoms and etiology, and each has its own separate ICD code, doctors often use a simple "Asthma" diagnosis for these variants. This means that medical coders receive many "Asthma" diagnoses that correspond to different ICD codes. To effectively assist in coding, our approach should be able to cope with this situation. Fig. 6 illustrates an example of sentence extraction for two "Asthma" diagnoses that actually refer to two different ICD codes: 493.20 "Chronic obstructive asthma, unspecified" and 493.90 "Asthma, unspecified type, unspecified."

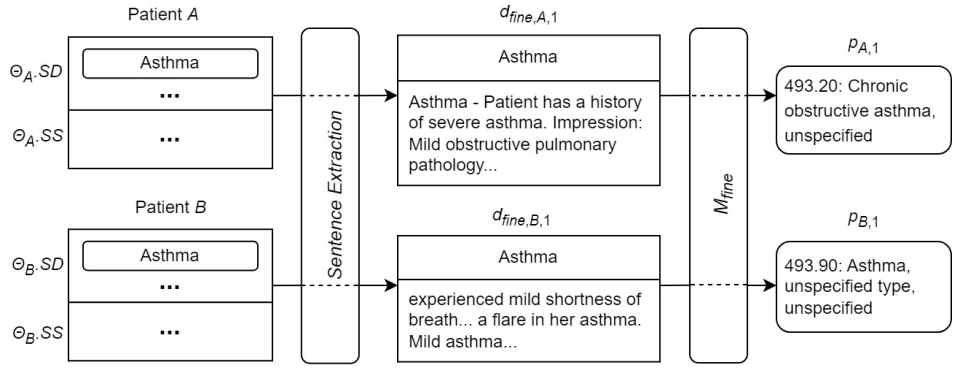

Fig. 6. Examples of sentences extraction for the vague diagnosis "Asthma", leading to different code assignments.

As shown in the figure, more details of the "Asthma" diagnoses are retrieved during the sentence extraction process and the classifier can differentiate between the same diagnoses to predict the corresponding ICD codes. There are two main differences between unspecified asthma and chronic obstructive asthma. That is, chronic obstructive asthma is persistent over a long period of time, whereas unspecified asthma may be intermittent short-term episodes; chronic obstructive asthma is caused by airway obstruction, whereas unspecified asthma may have other causes. Patient *A*'s records report a "history of severe asthma," which suggests that asthma is chronic. Patient *A*'s record also reports an

"obstructive pulmonary pathology", which suggests that there is an obstruction in the lungs. Combining this information, the classifier strongly suggests code 493.20 "Chronic obstructive asthma, unspecified." On the other hand, Patient *B*'s record shows a "flare" or acute episode of asthma with no mention of obstruction, suggesting that the classifier's prediction of code 493.90 "Asthma, unspecified type, unspecified" is the correct one. Clearly, this distinction is made possible by the extracted sentences that added additional information to the unclear "Asthma" diagnoses. However, the classification task remains challenging because both sets of extracted sentences contain the same terminologies (e.g., "asthma," "mild", and other respiratory terms, not shown in the figure). The experimental results show that BERT's attention architecture is advantageous in this case, where the same terms must be interpreted differently depending on the context.

5.3 *Comparisons with Related Approaches*

5.3.1 *Comparison with Black-Box Approaches*

The existing literature on automated ICD code assignment primarily features "black-box" approaches, which differ from the approach presented in this paper in two main ways. First, they treat the code assignment task as a multi-label classification task, where each data point is associated with one or more labels. The second difference is that these methods typically accept the entire text of a document (without sentence pruning/extraction) as input. To simulate a black-box classifier, we employ Med-BERT with a multi-label head to perform the classification, predicting all codes at once. Although the black-box classifier does not include sentence extraction, due to the 512-token input length limit of BERT, we discard sentences that are not related to *any* of the medical codes to be classified in order to fit a reasonable amount of useful information within the limited 512-token window. Given the extremely lengthy nature of the full documents, this was found to be necessary to achieve decent results without significant modifications to the basic classifier. Table 4 summarizes the performance of the model on the respiratory disease dataset described in Table 2.

Table 4. Performance metrics for the black-box classifier.

Accuracy	F1-score	Recall	Precision
0.450	0.509	0.417	0.697

Note that since the black-box classifier presented in this section treats ICD code assignment as a multi-label classification task, whereas our approach treats the problem as multiple single-label classifications, the metrics in Table 4 are fundamentally different from the metrics used in our approach and should not be directly compared. For example, in a single-label classification task, each prediction can be simply considered as correct or

incorrect for the purpose of calculating the accuracy. On the other hand, the predictions in a multi-label classification task may be partially correct, predicting the presence of some labels but not all the needed labels. Let L be the set of all labels in a multi-label classification task. Let L_{true} and L_{pred} be the sets of true and predicted labels, respectively, for a given problem example. To score the accuracy of this problem example, we first generate $|L|$-dimensional Boolean vectors Y_{true} and Y_{pred}, where each element indicates the presence or absence of a particular label. We then generate a third Boolean vector, $Y_{diff} = Y_{true}\ XOR\ Y_{pred}$, where each element in Y_{diff} is true if the corresponding elements in Y_{true} and Y_{pred} are different, and false if they are the same. This allows us to calculate the number of false predictions by counting the true elements in Y_{diff}. Finally, we divide the number of false predictions by the total number of labels $|L|$ to get the accuracy for the problem example. Figure 7 shows an example of calculating Y_{diff} when $L = \{A, B, C\}$, $L_{true} = \{A, C\}$ and $L_{pred} = \{A, B\}$.

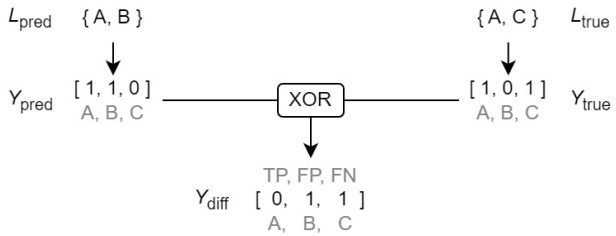

Fig. 7. An example of generating vector Y_{diff} to determine model performance for a given multi-label prediction.

Although a direct comparison between the two approaches is not possible, our analysis of the black-box models indicates that they generally have poor performance. While the state-of-the-art multi-label classifiers applied to ICD coding may perform better, the lower performance we observed suggests that the difficulty of the problem increases when using coarse-grained multi-label classification. Using the same model (Med-BERT) and training parameters, our proposed fine-grained single-label classification approach shows much better performance, which leads us to believe that fine-grained approach is a more efficient means of solving the ICD coding problem. In addition to the performance difference, our method provides additional outputs that are important for the coding process. Black-box methods only return code predictions. However, our approach produces not only code predictions, but also the diagnosis text and semantically related sentences extracted for each individual code. Thus, our approach provides the user with additional explainability, which is particularly important in the healthcare domain. Note that important distinction that enables our approach to improve explainability is the intermediate human knowledge-driven and human-understandable sentence extraction step.

5.3.2 *Comparison to LSTM-Based Classifier*

In our previous work, we performed the classification step using LSTM [8]. In order to quantify and analyze the advantages of the Med-BERT classifier in our approach, we compared the performance of Med-BERT and LSTM on a reduced version of the respiratory disease dataset shown in Table 1. That is, the number of examples per medical code is reduced to 25% of the original to highlight the advantage of Med-BERT with limited training data. Furthermore, since we previously observed that the performance of LSTM degrades rapidly with the introduction of new codes (especially closely related codes), we decided to test each model in a given trial with n medical codes, where $1 \leq n \leq 11$. The 11 codes were added in the order shown in Table 5, with some codes marked in bold. These bold codes are closely related to some codes introduced previously. For example, code 5, 482.41 Methicillin susceptible pneumonia, is closely related to previous code 2, 486 Pneumonia, organism unspecified. The order of codes generally follows the original frequency order, but with some closely related codes switched into more demonstrative positions.

Table 5. Frequency and order of addition of the 11 respiratory codes used to compare the Med-BERT and LSTM classifiers.

#	ICD Code	Diagnosis Name	Frequency
1	518.81	Acute respiratory failure	1,928
2	486	Pneumonia, organism unspecified	1186
3	507.0	Pneumonitis due to inhalation of food or vomitus	961
4	511.9	Unspecified pleural effusion	686
5	**482.41**	**Methicillin susceptible pneumonia due to staphylococcus aureus**	**221**
6	496	Chronic airway obstruction	587
7	518.0	Pulmonary collapse	514
8	493.90	Asthma, unspecified type, unspecified	505
9	**493.20**	**Chronic obstructive asthma, unspecified**	**185**
10	491.21	Obstructive chronic bronchitis with (acute) exacerbation	315
11	512.1	Iatrogenic pneumothorax	213

For any number of codes n, where $1 \leq n \leq 11$, the LSTM was trained and tested for 10 trials. Due to time constraints, the Med-BERT classifier was only trained and tested for 5 trials in each case. Fig. 9 shows the macro-F1-score and accuracy metrics for each classifier on different number of codes n.

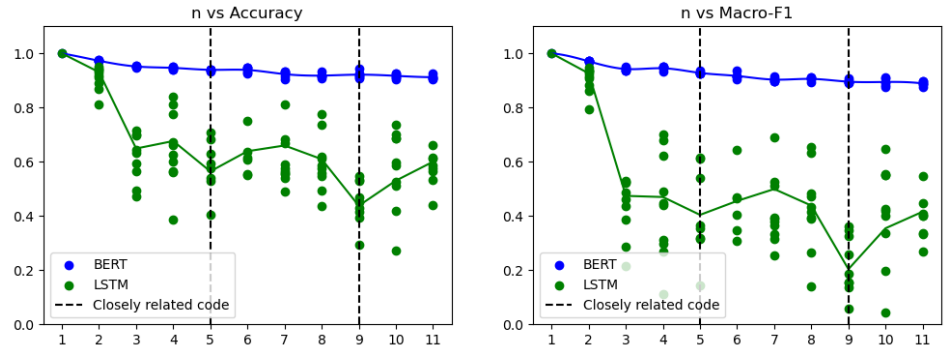

Fig 9. Accuracies and macro-F1 scores of LSTM and Med-BERT using different number of codes *n*.

The pre-trained knowledge packaged with BERT-based models like Med-BERT allows it to excel even when training examples are scarce [14, 29]. As shown in Fog. 9, despite the reduced dataset size, the Med-BERT classifier still exhibits strong performance, with accuracy and F1-scores mostly above 90%. On the other hand, once the classification becomes more complex, LSTM suffers immediately. Since an LSTM must be trained from scratch, it relies heavily on a rich set of training examples that are not always available in the real world. We observe that the performance of LSTM especially suffers when closely related codes are introduced. As shown in the figure, closely related codes pose challenges to the good performance of the LSTM, and only the addition of easily categorizable codes may bring the overall performance back up. Unsurprisingly, we do not observe this pattern in Med-BERT's metrics, as BERT's attention-based transformer architecture is much stronger at separating these linguistically similar, but slightly different related codes. Meanwhile, LSTM more often confuses these less frequent codes with the dominant related codes. Figs. 10 and 11 show examples of obfuscated codes whose inputs were correctly classified by the BERT-based classifier and misclassified by LSTM.

> The patient is a 75-year-old male with a history of COPD, asthma… who presented… with increased shortness of breath, **expectoration of sputum**, and **wheezing**. Asthma.

Fig. 10. Except from an example input labelled with code 493.20 Chronic obstructive asthma.

> Vancomycin was started as her catheter tip revealed **staph** on culture. Ceftriaxone was started for pneumonia. Ceftriaxone and clindamycin were continued for pneumonia…

Fig. 11. Excerpt from an example input labeled with code 482.41 Methicillin susceptible pneumonia due to staphylococcus aureus.

Fig. 10 illustrates the case of a patient suffering from chronic obstructive asthma with code 493.20. The Med-BERT classifier was able to contextualize the various mentions of "asthma" with specific symptoms "wheezing" and "expectoration of sputum" (indicating that the patient has been coughing up substances that obstruct the airways), which are two specific symptoms that are more closely related to chronic obstructive asthma. On the other hand, the LSTM seems to focus on the repeated use of the word "asthma" that always uses the same word embedding. It is likely that because this word occurs more frequently with the dominant class 493.90 Unspecified asthma (simply due to its high frequency in the dataset), the LSTM incorrectly predicts this code.

Fig. 11 shows an example of LSTM's failure in predicting code 482.41 Methicillin susceptible pneumonia due to staph aureus. Note that "staph" is mentioned just before the sentences where "pneumonia" is mentioned. BERT classifies this data point correctly, where the mention of the staph virus is clearly a hint for code 482.41. However, perhaps due to the LSTM's limited short-term memory when processing full text, it was unable to utilize this information to correctly classify pneumonia caused by staph, and instead predicted the generic code 486, Pneumonia, organism unspecified, that is highly frequent. This example once again demonstrates the strength of BERT in contextualizing generic and shared terms such as "pneumonia" with surrounding information that may significantly alter the interpretation of these terms.

6. Conclusions and Future Work

In this paper, we introduced a fine-grained evidence-based approach for automatically assigning ICD codes to patient discharge summary records. This approach differs from traditional "black-box" methods by including an intermediate human-understandable sentence extraction step, which improves the explainability and simplicity of the classification process. This approach starts with the diagnoses listed in the documents and extracts sentences semantically related to each diagnosis using a body of human knowledge such as an ontology. As a result, code assignments can be traced back not only to the evidence in the document, but also to the human knowledge used to extract the evidence. This additional information helps to improve the credibility of predictions, a key factor in healthcare where liability is always a concern and tends to complicate the adoption of AI tools. In addition to increased explainability, the proposed approach offers significant performance advantages over the "black-box" methods and the classifier used in our previous work.

In future work, we plan to demonstrate of how the proposed approach can be used for larger ICD code sets. To accommodate these larger code sets, we will incorporate our proposed method in a hierarchical classifier that can classify diagnoses with increasing specificity. In the hierarchical classifier, the top classifiers may determine the general type of disease, the intermediate classifiers may determine the families of diseases, and the final

classifiers determine the individual ICD codes. In addition to demonstrations on larger code sets, future work will continue to improve the classifier by incorporating some of the more sophisticated mechanisms that make black-box approaches effective [7], [15-17], [21-23], and by improving the sentence extraction step with enhanced extraction methods including new approaches for concept identification and the creation of a completely new knowledge base that better match the hierarchy of ICD codes.

Acknowledgements

We thank the editors and all anonymous referees for their valuable time in reviewing this paper. We also thank UMass Dartmouth for their financial support to the first author of this paper in completing this work.

References

[1] CMS, *National Health Expenditure (NHE) Fact Sheet*, Centers for Medicare & Medicaid Services (CMS), 2023. Retrieved from https://www.cms.gov/research-statistics-data-and-systems/statistics-trends-and- reports/nationalhealthexpenddata/nhe-fact-sheet. [Accessed: 01-Mar-2023].

[2] WHO, International Classification of Diseases (ICD), World Health Organization (WHO), 2023. Retrieved from https://www.who.int/standards/classifications/classification-of-diseases. [Accessed: 01-Mar-2023].

[3] HIPAA, *What is the HITECH Act?* The HIPAA Journal, January 2023. Retrieved from https://www.hipaajournal.com/what-is-the-hitech-act/. [Accessed: 01-Mar-2023].

[4] 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*, **3** (1) 160035 (2016).

[5] 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*, **16** (4) (2019) 1193-1202, doi: 10.1109/TCBB.2018.2817488.

[6] H. Shi, P. Xie, Z. Hu, M. Zhang and E. Xing, Towards automated ICD coding using deep learning, *arXiv preprint*, arXiv:2008.10492, 2017.

[7] 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*, **26** (10) (2022) 5235-5246, doi: 10.1109/JBHI.2022.3189404.

[8] J. Carberry and H. Xu, Fine-grained ICD code assignment using ontology-based classification, in *Proc. 2022 IEEE 23rd International Conference on Information Reuse and Integration for Data Science (IRI'22)*, San Diego, CA, USA, 2022, pp. 228-233, doi: 10.1109/IRI54793.2022.00058.

[9] I. Goldstein, A. Arzumtsyan and O. Uzuner, Three approaches to automatic assignment of ICD-9-CM codes to radiology reports, *AMIA Annual Symposium Proceedings*, **2007** (1) (2007) 279-283.

[10] R. Farkas and G. Szarvas, Automatic construction of rule-based ICD-9-CM coding systems, *BMC Bioinformatics*, **9** (Suppl 3) S10, 2008.

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

[12] A. Perotte, R. Pivovarov, K. Natarajan, N. Weiskopf, F. Wood and N. Elhadad, Diagnosis code assignment: models and evaluation metrics, *Journal of the American Medical Informatics Association*, **21** (2) (2014) 231-237.

[13] T. Baumel, J. Nassour-Kassis, R. Cohen, M. Elhadad and N. Elhadad, Multi-label classification of patient notes a case study on ICD code assignment, *arXiv preprint*, arXiv:1709.09587, 2017.

[14] 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.

[15] B. Biseda, G. Desai, H. Lin and A. Philip, Prediction of ICD codes with clinical BERT embeddings and text augmentation with label balancing using MIMIC-III, *arXiv preprint*, arXiv:2008.10492, 2020.

[16] 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 *Proc. 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Pasadena, CA, USA, 2021, pp. 1239-1244, doi: 10.1109/ICMLA52953.2021.00201.

[17] V. Mayya, S. S. Kamath and V. Sugumaran, LATA - label attention transformer architectures for ICD-10 coding of unstructured clinical notes, in *Proc. 2021 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, Melbourne, Australia, 2021, pp. 1-7, doi: 10.1109/CIBCB49929.2021.9562815.

[18] S. Montalvo, M. Almagro, R. Martínez, V. Fresno, S. Lorenzo, M. C. Morales, B. González, J. Álamo and A. García-Caro, Graphical user interface for assistance with ICD-10 coding of hospital discharge records, in *Proc. 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Madrid, Spain, 2018, pp. 2786-2788, doi: 10.1109/BIBM.2018.8621420.

[19] N. Siangchin and T. Samanchuen, Chatbot implementation for ICD-10 recommendation system, in *Proc. 2019 International Conference on Engineering, Science, and Industrial Applications (ICESI)*, Tokyo, Japan, 2019, pp. 1-6, doi: 10.1109/ICESI.2019.8863009.

[20] S. J. Lui, X. Cheng and S. Krishnaswamy, Inductive representation learning of multiple ICD codes for healthcare, in *Proc. 2022 IEEE 17th International Conference on Control & Automation (ICCA)*, Naples, Italy, 2022, pp. 498-503, doi: 10.1109/ICCA54724.2022.9831933.

[21] F. Teng, Z. Ma, J. Chen, M. Xiao and L. Huang, Automatic medical code assignment via deep learning approach for intelligent healthcare, *IEEE Journal of Biomedical and Health Informatics*, **24** (9) (2020) 2506-2515, doi: 10.1109/JBHI.2020.2996937.

[22] M. Almagro, R. Martínez-Unanue, V. Fresno, S. Montalvo and H. Tissot, ICD-10 coding based on semantic distance: LSI_UNED at CLEF eHealth 2020 Task 1, *CLEF (Working Notes)*, 2020.

[23] SNOMED, *SNOMED International Leading Healthcare Terminology, Worldwode*, 2023. Retrieved from https://www.snomed.org/. [Accessed: 15-Feb-2023].

[24] T. Bai and S. Vucetic, Improving medical code prediction from clinical text via incorporating online knowledge sources, in *Proc. World Wide Web Conference*, San Francisco, CA, USA, 2019, pp. 72-82.

[25] A. Sonabend W, W. Cai, Y. Ahuja, A. Ananthakrishnan, Z. Xia, S. Yu and C. Hong, Automated ICD coding via unsupervised knowledge integration (unite), *International Journal of Medical Informatics*, **139** (2020) 104135.

[26] F. Teng, W. Yang, L. Chen, L. F. Huang and Q. Xu, Explainable prediction of medical codes with knowledge graphs, *Frontiers in Bioengineering and Biotechnology*, **8** (867) (2020) 1-11.
[27] M. Zeng, M. Li, Z. Fei, Y. Yu, Y. Pan and J. Wang, Automatic ICD-9 coding via deep transfer learning, *Neurocomputing*, **324** (2019) 43-50.
[28] T. Lin, P. Goyal, R. Girschick, K. He and P. Dollár, Focal loss for dense object detection, *arXiv preprint*, arXiv:1708.02002, 2018
[29] C. Sun, X. Qui, Y. Xu and X. Huang, How to fine-tune BERT for text classification? *arXiv preprint*, arXiv:1905.05583, 2019.