Automated Extraction of ICD-10 Diagnosis Codes from Clinical Notes

Stanford CS224N Custom Project

Rishi Verma
Department of Computer Science
Stanford University
rishirv@stanford.edu

Arjun Jain
Department of Computer Science
Stanford University
arjunj@stanford.edu

Devanshu Ladsaria
Department of Computer Science
Stanford University
devanshu@stanford.edu

Abstract

ICD-10 codes are essential in healthcare for standardizing medical documentation, streamlining communication among healthcare professionals, ensuring accurate billing, and enabling downstream clinical prediction. Currently, ICD-10 diagnosis codes are manually assigned by healthcare administrators which is labour intensive and error-prone. In this paper, we aim to automate the extraction of ICD-10 diagnosis codes from free-text clinical notes, making the coding process more efficient and accurate. To tackle this task, we extract key sections from the clinical notes and apply two methodologies: 1) finetuning pre-trained ClinicalBERT with a linear classification layer and 2) finetuning ClinicalBERT on text similarity between code definitions and clinical note embeddings. Both models are trained and evaluated on clinical notes and corresponding ICD-10 codes from MIMIC-IV dataset, a comprehensive electronic health records database. Our experimentation reveals that both models significantly beat baselines, with the finetuned ClinicalBERT + Linear Layer presenting the best results. These results underscore the efficacy of our system in automating the extraction of ICD-10 codes from clinical notes, paving the way for improved efficiency and accuracy in healthcare documentation.

1 Key Information

- **External Mentor**: Jeff Choi (Stanford General Surgery Resident)
- **Team Contributions**: Arjun handled all implementation and experimentation for finetuning ClinicalBERT with the linear classification layer. Rishi handled implementation and experimentation of baselines and text similarity classifer. Devanshu handled all data preprocessing, sectioning of clinical notes, and ran experiments. All authors contributed to the paper equally. All code was implemented from scratch.

2 Introduction

The International Classification of Diseases, 10th edition, also known as ICD-10, is a system for classifying medical procedures and diagnoses. This taxonomy is used internationally and is maintained by the World Health Organization (WHO). These codes are crucial in healthcare for a multitude of reasons. Firstly, they enable standardized documentation of medical diagnoses, ensuring clarity and accuracy in patient records. Secondly, they facilitate streamlined communication among healthcare professionals, allowing for effective care coordination and treatment planning.

Additionally, ICD codes play a pivotal role in billing and reimbursement processes, ensuring that healthcare providers receive appropriate compensation for their services. Moreover, these codes serve as a foundation for healthcare data analysis, enabling researchers to identify trends, monitor public health, and drive evidence-based decision-making (Hirsch et al., 2016).

Currently, the task of ICD coding relies on manually trained coders, who spend an average of 34 minutes assigning codes for each patient (Teng et al., 2023). However, the accuracy and efficiency of this manual process often fall short in real-world applications. It is susceptible to errors stemming from various uncontrollable factors, including discrepancies in patient discharge summaries and variations among coders or healthcare facilities. These errors can result in incorrect billing, refusals of health insurance reimbursements, and inadequate payments (Teng et al., 2023).

Given the significance of ICD coding in healthcare settings and the inefficiency and inaccuracy of the current manual process, there is a growing interest in automating the extraction of ICD codes from clinical documentation. Automated extraction utilizes natural language processing and machine learning algorithms to analyze and interpret clinical notes to identify and assign relevant ICD codes.

3 Related Work

There have been a number of works studying Automated Medical Coding, which have been surveyed by Ji et al. (2022). This paper highlights that this task is challenging because medical notes often use a completely different structure and terminology than prototypical examples of English language. For one, medical texts use abbreviations, acronyms, and jargon that can seem incomprehensible to those not trained professionally. In addition, medical texts are lengthy, unstructured, noisy, and don't correspond one-to-one with ICD codes. We present several key works below.

Choi et al. (2023) develop and validate TraumaICDBERT, a NLP algorithm to predict injury ICD-10 diagnosis codes from trauma tertiary survey notes. This model has two key limitations. Their model is only trained on clinical notes from trauma settings, and is only trained on a small dataset of 3,478 data points. In contrast, our approach overcomes these limitations by leveraging the MIMIC-IV dataset, a comprehensive medical dataset that offers a diverse range of clinical notes across various medical specialties and a substantially larger volume of data.

The authors of "ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission" use bidirectional encoder representations to predict 30-day hospital readmission at various time points Huang et al. (2019). They use both discharge summaries and the first few days of notes in the intensive care unit to train their model . However, they do not use other additional information like patient's history and outpatient doctor notes. Our approach uses these additional pieces of information to predict ICD-10 codes, allowing us to predict ICD-10 codes for non-hospitalized patients which can be later used in other downstream applications.

Lastly, in "Predicting ICD-9 Codes Using Self-Report of Patients," the authors apply a Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) to develop a model for predicting ICD-9 codes Singaravelan et al. (2021). Instead of using free text clinical data or lab reports, the paper only uses self-reported information like signs and symptoms of a patient to predict ICD-9 codes, which can then be used for applications like self-diagnosis and disease prevention. The authors only predict ICD-9 codes, which jave been deprecated and replaced by more uniform and more accurate ICD-10 codes. This approach can also result in incorrect diagnoses, since it only replies on a patient's own report. Building on top of this, using free-text clinical notes augmented with self-reported information, we can allow for a holistic self-diagnosis tool, along with a medical aid for professionals.

4 Dataset and Preprocessing

4.1 Task Overview and MIMIC-IV Database

Our task is to parse free-text clinical notes and automatically extract ICD-10 codes. This is a multi-label binary classification problem, as a patient can have multiple diagnoses, and thus be assigned multiple codes. For this task, we leverage MIMIC-IV, a database of electronic health records collected over 2009-2019 from the Beth Israel Deaconess Medical Center (Johnson et al., 2023). MIMIC-IV contains over 330,000 de-identified, free-text clinical notes and their corresponding ICD-10 diagnosis codes.

Table 1 below displays an example input and output. Due to restrictions on publicly sharing the clinical notes in MIMIC-IV, we present an example of a clinical note along with its corresponding truncated ICD-10 codes (truncated) from Choi et al. (2023) in Table 1. Note that the notes from our dataset are significantly longer, averaging over 1,000 tokens.

Note	ICD-10 Code
1. *** stab wounds to L back with retained stabbing implement 2. Descending thoracic aortic injury s/p primary repair 3. L hemopneumothorax s/p chest tube x2 to wall suction 4. Small R pneumothorax managed expectantly 5. Grade 2-3 liver lac managed non-op 6. R ***th rib Fx 7. LLL lung contusion 8. L forehead lac	S01.8 : Open wound of other parts of head S22.3 : Fracture of one rib S27.2 : Traumatic hemopneumothora x S27.3 : Other and unspecified injuries of lung S36.1 : Injury of liver and gallbladder and bile duct

Table 1: Example Input and Output. Note that all 8 items in the list are part of a single input, and all 5 codes are part of the observed output. (Choi et al., 2023).

4.2 Code Selection

ICD-10 codes vary in length, with each additional character providing increased specificity. Following Huang et al. (2019), we truncate each code to the first 4 characters to balance specificity and predictive ability.

The dataset is highly imbalanced, with a long-tail of over 5,000 ICD-10 codes codes with a frequency of 1, so we limit the data to the most frequent ICD-10 codes. This is in accordance with Nguyen et al. (2023).

Specifically, we create two datasets: one with 50 most frequent ICD-10 codes ("top50") and one with 200 most frequent ICD-10 codes ("top200"). We chose to include top 200 in addition to top 50 because it represents a more challenging task of predicting less frequent codes, while still not being totally imbalanced.

4.3 Stratified Sampling

For computational reasons, we randomly subsampled 50,000 examples from the dataset. Then, we used stratified sampling to define train, valid, and test splits. Although the MIMIC-IV dataset has predefined train, validation, and test splits, Edin et al. (2023) noted that a large percentage of codes in the training set are completely absent in the test set. In addition, the test set is small, only comprising of 5% of examples. We define new splits of the dataset to better and more fairly assess our model's performance on this unbiased dataset. We use iterative stratified sampling, an algorithm developed by Sechidis et al. (2011) for stratification of multilabel data, which addresses the issue of random sampling resulting in sparse examples in test datasets. This algorithm greedily assigns datapoints to the three datasets to ensure equitable distributions of labels. We apply an approximate 80% train, 10% validation, and 10% test split, which resulted in **38,730 training examples**, **5,565 for validation**, and **5,705 held out for testing**. These splits were equal for both the top50 and top200 datasets.

4.4 Extracting Key Sections from Clinical Notes

The free-text clinical notes are on average 1,122 tokens long. From our initial experimentation, we observed that the performance of ClinicalBERT on these notes was limited by its maximum sequence length of 128 tokens, and simply using the first 128 tokens failed to capture the content of the note relevant to ICD-10 code classification.

To address this limitation, we decided to extract three key sections from each clinical note, we take 2 chunks of 128 tokens from each section and concatenate them. MIMIC-IV has multiple sections that includes patient history, discharge summaries, patients notes, social/family history, test results, etc. To identify which sections to choose, we ran a logistic regression model on the sections and picked 3 sections with the highest correlation with the ICD-10 diagnoses codes. The 3 sections are: "History of Present Illness", "Past Medical History", and "Physical Exam".

Extracting these sections from each note proved to be quite difficult because the notes vary significantly from patient to patient. There were no consistent delimiters which could be used to break up these sections, and their lengths varied from 10 tokens to over 1000 tokens. After manual screening of over a hundred patient records, a hierarchical approach was developed to make these sections. For example, in the case of physical exam, over seventy-five percent of records were delimited by "discharge:". Using this strategy, sectioning of the notes was possible. Due to the inconsistent nature of the data, this approach was not perfect, resulting in certain sections containing a part of the next subsection, but was simpler than a segmentation model. However, since we are using a maximum of 256 tokens per section (2 128-token chunks per section), we were able to mitigate this situation.

5 Approach

5.1 Baseline Models

Previous work, such as TraumaICDBERT by Choi et al. (2023), is limited to specific categories of patients, like trauma patients, and is not generalizable to other medical settings. As we expand the scope of our models, their results are not a suitable baseline. Instead, we formulate a set of naive classifiers as baselines. To tackle multi-label classification, we use a technique called binary relevance (Zhang et al., 2018), which independently trains a binary classifier for each possible label. We first generate an embedding for each discharge summary using TF-IDF scores, a standard data-mining approach which treats documents in a Bag-of-Words representation and identifies the distribution of important words. The resulting vectors are essentially augmented frequency counts for these words. Then, we apply three standard methods for binary classification: Logistic Regression, Support Vector Classifiers, and K-Nearest Neighbor Classifiers. Apart from binary relevance, we also tried to apply LLMs such as Llama and Mistral to extract ICD-10 codes from discharge summaries, However, due to restrictions on sharing credentialed data with third parties and the inability of the LLMs to reliably output a consistent format, we were unable to generate this baseline.

5.2 Finetuning ClinicalBERT with Linear Classification Layer

For this approach, we finetuned a pre-trained BERT model called ClinicalBERT (Alsentzer et al., 2019). The base model of ClinicalBERT is BioBERT, a language model pretrained for biomedical text mining (Lee et al., 2019). ClinicalBERT finetunes BioBERT on discharge summaries from MIMIC-III, a smaller dataset similar to MIMIC-IV from the same hospital. We chose ClinicalBERT because it has a semantic understanding of medical terminology through its base model and it has a syntactic understanding of clinical notes through its pretraining. The maximum sequence length of ClinicalBERT is 128 tokens.

To train ClinicalBERT, we first concatenate the preprocessed sections of each clinical note and tokenize them, and then split this tensor into six chunks of 128 tokens (2 chunks per section). We process each chunk separately with ClinicalBERT, generating contextualized embeddings. Next, we combine these embeddings by concatenating them, which ensures that the information of each chunk is preserved (embedding dimension= $6 \cdot 128 = 768$). Finally, we pass these embeddings through a fully-connected linear classification layer that has input dimension 768 and output dimension equal to the number of output labels (50 or 200). For the forward and backward passes, we use binary cross-entropy loss because our task is multi-label classification and binary cross-entropy loss is well-suited for tasks where each example can belong to multiple classes simultaneously.

5.3 Text Similarity Classifier

A clear limitation of NLP-based classifiers is their dependence on a predetermined set of ICD-10 codes. When we expanded our set of codes considered from the top 50 to top 200 codes, we had to fine-tune ClinicalBERT from scratch, with a new linear layer for classification (described above) As a result, these models cannot *generalize* to rare codes or if the classification system gets modified. The ICD-10 codes are updated yearly, and models would require frequent maintenance to handle the changing label set.

Instead, we note that each ICD-10 code has an official definition, which can aid the model in information retrieval, rather than learning contextual associations between the textual inputs and codes from scratch. We propose a novel text similarity classifier embeds each discharge summary using the ClinicalBERT model and each ICD-10 code using their official definitions. We use cosine similarity between these embeddings to generate a classification score, under the assumption that the description of an ICD code and a patient diagnosed with that code will be similar. We thus convert our classification task into a semantic similarity task.

Our approach is similar to that of Sentence BERT (Reimers and Gurevych, 2019), which is derived from BERT but generates sentence, rather than token-level embeddings. This approach has noted state-of-the-art results using cosine similarity between text embeddings for semantic similarity tasks. In the absence of a Sentence BERT model fine-tuned for clinical texts, we pool over the token embeddings generated by BERT, as done by Zhang et al. (2019). Because of the limited maximum sequence length of our base model, we first pool over the token embeddings for each section to generate section embeddings. We then generate a final embedding for the clinical note by pooling over all the sections.

6 Experiments

6.1 Evaluation Methods

We use standard metrics for multi-label classification, as there are 5.8 ICD-10 codes on average for each clinical note. We average AUROC and F1 scores (Grandini et al., 2020) over each class (macro), over each datapoint (micro), and weighted classes (weighted). We also compute Precision and Recall $@k$ ($k \in \{5, 10, 20\}$). These measure precision and recall among the k ICD-10 diagnosis codes with the highest predicted probabilities.

To compute these metrics, we need to threshold the continuous probability scores produced by the model to produce binary predictions. Instead of relying on a fixed threshold, which may not generalize well across different datasets and model architectures, we opt for a dynamic threshold selection approach. We iterate over a range of thresholds from 0 to 1 with a step size of 0.01 to explore threshold space. We maximize the weighted F1 score, a metric that balances precision and recall to trade-off between correctly identifying positive instances and capturing all positive instances. This adaptive thresholding method calibrates our models, ensuring each model predicts a similar number of labels for each text example.

6.2 Hyperparameter Tuning

For both approaches, we use the Adam optimizer. To optimize the learning rate for finetuning of ClinicalBERT, we conducted hyperparameter tuning of the learning rate. On a smaller subset of the dataset (train: 10k examples, validation: 1k examples, test: 1k examples), we finetuned ClinicalBERT for 5 epochs and with four different learning rates: $1E-5$, $5E-5$, $7.5E-5$, $1E-4$. For each learning rate, after ClinicalBERT model was trained, the model was then evaluated on the validation dataset, optimizing for weighted F1 score. The results of the hyperparameter tuning are presented in Figure 1.

From the results in Figure 1, we see that the weighted F1 score slightly increases at a learning rate of $5.0E-5$, slightly drops at learning rate of $7.5E-5$, and then significantly drops as the learning rate increases past that point. We thus choose $5.0E-5$ as our optimal learning rate. We present all the other parameters of our models below:

6.3 Experiment Details

We first trained and evaluated the baseline, the ClinicalBERT + Linear Layer model, and the Text-Similarity Classifier, on the "top50" dataset.

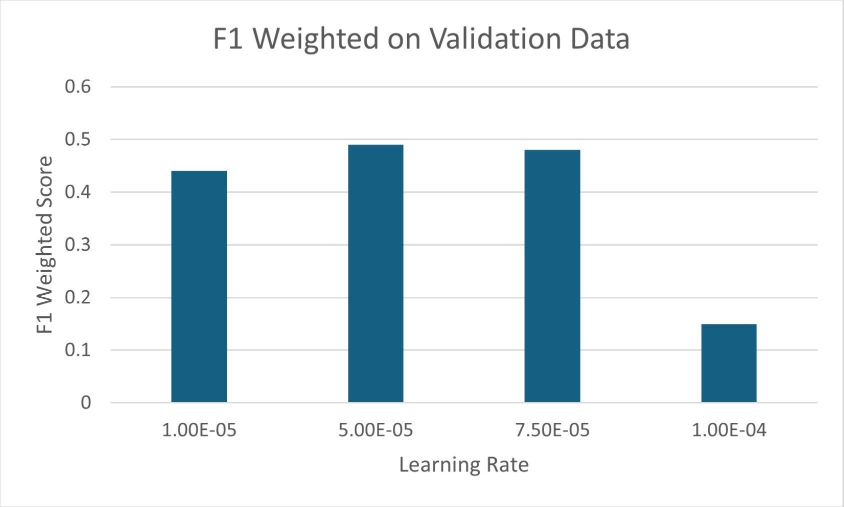

Figure 1: Learning Rate Optimization

Parameters of ClinicalBERT + Linear Layer Model: vocab_size=28996, max_sequence_length=128, embedding_dimension=768, num_hidden_layers=12, num_parameters=108,387,122, learn_rate=5e-5, batch_size=20, num_epochs=5, dropout_rate=0.10, train_time=6hrs.

Parameters of Text-Similarity Classifier: vocab_size=28996, max_sequence_length=128, embedding_dimension=768, num_hidden_layers=12, num_parameters=108,348,722, learn_rate=5e-5, batch_size=12, num_epochs=5, dropout_rate=0.10, train_time=5hrs.

The Text Similarity Classifier required a smaller batch size, due to the increased memory capacity of having to backpropagate across both the text embeddings and the definition. In addition, due to these memory limitations, we train the model to maximize cosine similarity with all positive examples (true codes for the text) and minimize for 10 randomly sampled negative examples.

We then trained and evaluated all models on the "top200" dataset.

Parameters of text similarity classifier model: vocab_size=28996, max_sequence_length=128, embedding_dimension=768, num_hidden_layers=12, num_parameters=108,348,722, learn_rate=5e-5, batch_size=12, num_epochs=5, dropout_rate=0.10, train_time=5.5hrs.

Parameters of ClinicalBERT + linear layer model: vocab_size=28996, max_sequence_length=128, embedding_dimension=768, num_hidden_layers=12, num_parameters=108,502,322, learn_rate=5e-5, batch_size=20, num_epochs=5, dropout_rate=0.10, train_time=7hrs.

	AUROC			F1		
	Micro	Macro	Weighted	Micro	Macro	Weighted
Logistic Regression	77.2	71.2	70.3	37.9	31.2	36.0
KNN Classifier	64.6	58.5	58.4	27.7	23.3	29.3
Support Vector Classifier	77.2	71.2	70.3	37.9	31.7	36.3
Fine-Tuned ClinicalBERT	**86.0**	**82.7**	**82.4**	**53.0**	**46.0**	**51.5**
Text-Similarity ClinicalBERT	82.5	78.7	76.8	44.5	38.7	43.1

Table 2: AUROC and F1 Scores for top50 Dataset

7 Analysis

7.1 Quantitative Results

We presents the results of our experiments on the top50 dataset in Tables 2 and 3, and the results on the top200 dataset in Tables 4 and 5. The Fine-Tuned ClinicalBERT model refers to the classifier

	Precision		Recall	
	5 Codes	10 Codes	5 Codes	10 Codes
Logistic Regression	37.4	29.2	35.2	53.8
KNN Classifier	26.7	21.5	25.0	39.2
Support Vector Classifier	37.6	29.2	35.3	53.8
Fine-Tuned ClinicalBERT	**51.9**	**37.0**	**49.9**	**67.8**
Text-Similarity ClinicalBERT	41.6	32.9	40.8	61.9

Table 3: Precision and Recall for top50 Dataset

	AUROC			F1		
	Micro	Macro	Weighted	Micro	Macro	Weighted
Logistic Regression	70.7	63.8	62.0	16.1	15.4	17.8
KNN Classifier	57.2	50.9	50.8	9.1	7.5	10.1
Support Vector Classifier	70.6	63.8	62.0	16.1	15.6	17.9
Fine-Tuned ClinicalBERT	**84.5**	**78.5**	**78.4**	**35.2**	**20.5**	30.7
Text-Similarity (50-train)	76.7	72.0	75.5	23.2	17.8	31.4
Text-Similarity (200-train)	79.6	76.8	77.8	25.5	19.8	**31.5**

Table 4: AUROC and F1 Scores for top200 Dataset

from Section 5.2 and the Text-Similarity Clinical BERT refers to the classifier from Section 5.3. We detail the difference between the 50-train and 200-train versions of the Text-Similarity models in Section 7.3.

We note that the Logistic Regression and Support Vector Classifiers have similar results, while the KNN classifier performs significantly worse. This is unsurprising - we expect significant variation across the notes, and that the content test notes will not significantly resemble the train or validation notes. Across almost all metrics and both datasets, the ClinicalBERT + Linear Layer model performs the best. The Text-Similarity model beats the baselines, and has performance approaching that of the ClinicalBERT + Linear Layer, but still underperforms. This matches with our expectations comparing the both models. The ClinicalBERT + Linear Layer only needs to consider the top 50 or 200 codes, and can thus adapt its embeddings specifically for that set of codes. The Text-Similarity model has a harder task: it must simultaneously adapt the embeddings for the definitions and the text embeddings, and must maximize the cosine similarity of same text embedding with a large number of definition embeddings. We hypothesize that performance will improve using an MLP to generate a score from the two embeddings, rather than simply cosine similarity.

7.2 Baseline Model Analysis and Discriminative Ability

We notice that the baseline models performed significantly better than we expected. Given that the baseline models consist of naive classifiers trained on bag of word embeddings for the text, we expected them perform poorly given the nuanced context behind clinical classification. Investigating the coefficients of one of the baselines (logistic regression classifier), we notice that they place weights on a select few vocabulary items. For example, the classifier for J960 (acute respiratory failure) has weight greater than 1 on the presence of words like “acute,” “breath,” “cough,” “dyspnea” (meaning labored breathing), and “shortness.”

So, the strong performance of baseline models can be attributed to the diverse nature of the top 50 ICD-10 codes, which cover a wide range of medical conditions. This diversity inadvertently made the classification task easier for the baselines by providing a broad cross-section of patients. For example, since few codes directly relate to breathing problems, terms like "dyspnea" can easily lead to a prediction of acute respiratory failure. Baseline models, with their limited feature set, could then more easily learn the importance of specific vocabulary items for each classification unlike BERT-based models, which consider the entire text and use shared embeddings for all ICD codes.

A more clinically relevant metric would be to compare the model’s ability to distinguish *similar* ICD codes. So, we reevaluated our models on 4 codes corresponding to variations of "type2 diabetes mellitus": E112 (with kidney complications), E114 (with neurological complications), E116 (with

	Precision		Recall	
	5 Codes	10 Codes	5 Codes	10 Codes
Logistic Regression	24.3	16.4	22.1	17.7
KNN Classifier	12.6	8.4	12.2	5.9
Support Vector Classifier	24.3	16.4	22.2	17.8
Fine-Tuned ClinicalBERT	**41.6**	**31.4**	**26.2**	**37.9**
Text-Similarity (50-train)	28.2	22.8	18.4	28.5
Text-Similarity (200-train)	28.7	23.6	19.3	30.4

Table 5: Precision and Recall Scores for top200 Dataset

	AUROC			F1		
	Micro	Macro	Weighted	Micro	Macro	Weighted
Logistic Regression	66.4	63.1	61.9	20.8	19.1	19.9
Fine-Tuned ClinicalBERT	**90.4**	**90.3**	**89.9**	**55.4**	**54.0**	**55.3**
Text-Similarity ClinicalBERT	87.6	87.4	86.6	42.9	42.1	43.5

Table 6: Evaluation of 50-code Models on just codes E112, E114, E116, and E119.

other specified complications), and E119 (without complications).The results of this experiment are presented in Table 6. Because of the similarity across baselines, we only present logistic regression. **These results show where transformer-based models shine**. While the strength of our baselines comes from key vocabulary elements distinguishing substantially different codes, transformer-based models have a much stronger discriminative ability to distinguish similar codes.

7.3 Text-Similarity Zero-Shot Learning

As described in approaches, the text-similarity model has the theoretical ability to generalize. Because there is no additional linear layer, only the weights for the base BERT model are fine-tuned to generate high-quality embeddings. Thus, if definitions of some ICD-10 codes change, new codes are added, or even a new standard like ICD-11 is released, the model should be able to understand the modified codes just from their definition, without any additional training. To assess this, we re-evaluated the Text-Similarity model trained only on top-50 code dataset ("50-train") on the top-200 code test set, despite the model never having seen 150 of the 200 classes which it is expected to predict. Incredibly, the model's results do not degrade significantly. Though the text-similarity model is still far from the ClinicalBERT classifier, the 50-train version still significantly beats baselines, and reports scores only marginally lower than the 200-train version of the model. Our model displays zero-shot learning, which to the best knowledge of the authors, is a first for Automated Medical Coding. We hypothesize that the Text-Similarity model will demonstrate better performance when there are much larger numbers of ICD-10 codes, including rare codes which may be difficult for the ClinicalBERT + Linear Layer model, such as with extremely rare codes that only appear a single time in the dataset.

8 Conclusion and Future Work

ICD-10 codes are extensively used in healthcare industry from billing, insurance claims to medical communication. Currently, trained hospital administrators manually assign these codes for each patient. This process is extremely expensive and requires significant amount of time and labour. Using free-text clinical notes, we finetuned pre-trained Clinical-BERT with a linear classification layer. We saw promising results, particularly on hard-to-differentiate codes, validating an NLP approach to automate ICD-10 coding. In addition, a text-similarity classifier demonstrates zero-shot learning, a new result in the domain of medical coding. This can result in a step towards a faster, efficient and a more affordable health care industry.

Further work can be done to use a patient's self-reported data like signs and symptoms. This additional detail with the patients history, can allow us to predict future health concerns and can be used for preventative measures. In addition, we hope to consider larger portions of clinical notes beyond the three primary sections identified, and to evaluate both models on rarer codes.

References

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.

Jeff Choi, Yifu Chen, Alexander Sivura, Edward B Vendrow, Jenny Wang, and David A Spain. 2023. Traumaicdbert, a natural language processing algorithm to extract injury icd-10 diagnosis code from free text. *Annals of Surgery*, pages 10–1097.

Joakim Edin, Alexander Junge, Jakob D Havtorn, Lasse Borgholt, Maria Maistro, Tuukka Ruotsalo, and Lars Maaløe. 2023. Automated medical coding on mimic-iii and mimic-iv: A critical review and replicability study. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2572–2582.

Margherita Grandini, Enrico Bagli, and Giorgio Visani. 2020. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*.

JA Hirsch, G Nicola, G McGinty, RW Liu, RM Barr, MD Chittle, and L Manchikanti. 2016. Icd-10: history and context. *American Journal of Neuroradiology*, 37(4):596–599.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*.

Shaoxiong Ji, Wei Sun, Hang Dong, Honghan Wu, and Pekka Marttinen. 2022. A unified review of deep learning for automated medical coding. *arXiv preprint arXiv:2201.02797*.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. 2023. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Thanh-Tung Nguyen, Viktor Schlegel, Abhinav Kashyap, Stefan Winkler, Shao-Syuan Huang, Jie-Jyun Liu, and Chih-Jen Lin. 2023. Mimic-iv-icd: A new benchmark for extreme multilabel classification. *arXiv preprint arXiv:2304.13998*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2011. On the stratification of multi-label data. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part III 22*, pages 145–158. Springer.

Anandakumar Singaravelan, Chung-Ho Hsieh, Yi-Kai Liao, and Jia-Lien Hsu. 2021. Predicting icd-9 codes using self-report of patients. *Applied Sciences*, 11(21):10046.

Fei Teng, Yiming Liu, Tianrui Li, Yi Zhang, Shuangqing Li, and Yue Zhao. 2023. A review on deep neural networks for icd coding. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4357–4375.

Min-Ling Zhang, Yu-Kun Li, Xu-Ying Liu, and Xin Geng. 2018. Binary relevance for multi-label learning: an overview. *Frontiers of Computer Science*, 12:191–202.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.