Analyzing Code Embeddings for Coding Clinical Narratives

Wei Shi[1] **Jiewen Wu**[2*] **Xiwen Yang**[3] **Nancy F. Chen**[1]

Ivan Ho Mien[1,4] **Jung-jae Kim**[1] **Pavitra Krishnaswamy**[1]

[1]Institute for Infocomm Research, A*STAR, Singapore [2]Huawei Technologies Co., Ltd, China

[3]A*STAR Artificial Intelligence Initiative, Singapore [4]National Neuroscience Institute, Singapore

{shi_wei, nfychen, Ivan_Ho, jjkim, pavitrak}@i2r.a-star.edu.sg

wujiewen@huawei.com xiwenyang22@gmail.com

Abstract

Medical professionals review clinical narratives to assign medical codes as per the International Classification of Diseases (ICD) for billing and care management. This manual process is inefficient and error-prone as it involves a nuanced one-to-many mapping. Recent works on automated ICD coding learn mappings between low-dimensional representations of the reports and the codes. While they propose novel neural networks for encoding varied types of information about the codes, it is unclear as to what information in the medical codes is helpful for performance improvement and why. Here, we compare different ways to represent, or embed, the codes based on their textual, structural and statistical characteristics, using a single deep learning baseline model in quantitative evaluations on discharge reports from the MIMIC-III Intensive Care Unit database. We also qualitatively analyse the nature of the cases that benefit most from the code embeddings and demonstrate that code embeddings are important for predicting ambiguous and oblique codes.

1 Introduction

Free-text clinical narratives contain the majority of information pertaining to patient state, disease progression and care management. Following a patient encounter, the text reports from the visit are codified by representing the key diagnoses and procedures according to the International Classification of Diseases (ICD) system (Medicode (Firm), 1996). The resulting ICD codes are used for a variety of diagnostic, billing, epidemiology and research purposes (Bach and First, 2018; Feder et al., 2018; Alsentzer et al., 2019).

The process of ICD coding, i.e., mapping clinical text reports to ICD codes, is challenging. It involves processing diverse domain-specific text with large vocabulary and significant irrelevant content to make a nuanced choice of a small set of codes from a high-dimensional taxonomy of 15,000 ICD codes. Hence, manual ICD coding tends to be time-intensive, costly, and error-prone (Lang, 2007; Shi et al., 2017; Xie and Xing, 2018), and there is great interest in automated ICD coding methods.

Previous works on automated ICD coding have employed conventional rule-based or machine learning methods (Larkey and Croft, 1996; Farkas and Szarvas, 2008; Perotte et al., 2014). Recently, deep learning methods (Baumel et al., 2017; Xie and Xing, 2018; Nie et al., 2018; Mullenbach et al., 2018; Vu et al., 2020; Cao et al., 2020; Teng et al., 2020; Yuan et al., 2020) have achieved leading-edge performance. Of these, the best performing deep learning approaches typically employ attention mechanisms to use representations of the ICD codes to guide the model's predictions. However, the specific representations of the ICD codes used vary from code textual descriptions (Mullenbach et al., 2018) and code hierarchy (Vu et al., 2020; Cao et al., 2020) to code co-occurrences (Cao et al., 2020) and graph of medical entities associated with codes (Teng et al., 2020; Yuan et al., 2020). Yet, it is unclear which ICD code representation is most effective, what types of cases would benefit from these representations, and why.

Addressing these gaps requires comparing different code embeddings within one united framework. We introduce a simple attention mechanism to leverage varied statistical, textual, structural representations of ICD codes and enhance a predefined baseline clinical notes classifier. We use discharge reports within the benchmark MIMIC-III Intensive Care Unit database (Johnson et al., 2016) for comparative evaluation, and perform extensive experiments to characterize effects of dif-

This work was done while the author was at A*STAR.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4665–4672
August 1–6, 2021.

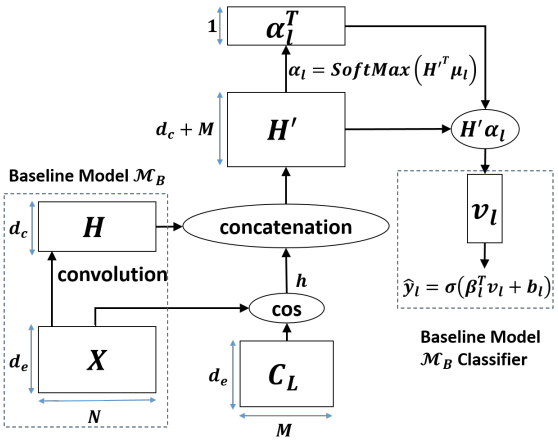

Figure 1: Proposed model architecture

ferent code embeddings on prediction performance. Quantitative results show that our proposed attention mechanism (a) enables 7-9% micro-F-1 boost over the baseline classifier, and (b) performs at least as accurately as more advanced two-level attention, hyperbolic embedding or graph convolutional network approaches. We further perform qualitative analyses and show that our attention network enables large improvements when the coding task is more ambiguous or nuanced. Our approach and findings offer practical means to enhance performance in nuanced text classification tasks.

2 Methods

The task entails mapping a given free-text discharge report to a set of ICD codes. This is a multi-label text classification problem. We propose an approach for learning varied textual, structural and statistical representations of the ICD codes (i.e., code embeddings), and employing them to enhance performance of a given baseline model.

2.1 Attention to Code Embeddings

Figure 1 illustrates the architecture. We start with a given baseline model $\mathcal{M}_B$ based on a convolutional layer. $\mathcal{M}_B$ takes word embeddings $X \in \mathbb{R}^{d_e \times N}$ of words in a given report as input and learns to generate their hidden representation $H \in \mathbb{R}^{d_c \times N}$, where N is the length of input medical narratives after padding, d_e is the input embedding size, and d_c is the number of filters. We propose code embeddings $C_L \in \mathbb{R}^{d_e \times M}$ as an auxiliary for $\mathcal{M}_B$, where M is the number of ICD codes. We compute the cosine similarity between X and C_L, and denote the result as h. We then compute per-label attention weights α_l as follows:

$$\alpha_l = \text{SoftMax}(\begin{bmatrix} H \\ h \end{bmatrix}^{\mathrm{T}} \mu_l), \quad (1)$$

where $[\ ]$ indicates concatenation (denoted as H') and $\mu_l \in \mathbb{R}^{d_c+M}$ is a vector parameter for label l. Weights α_l denote attention from the note representation to the code representation for label l and can be used to enhance performance of $\mathcal{M}_B$ (e.g., as shown in Figure 1). Then we apply the attention weights α_l to H' to get the final representation v_l of an input report corresponding to label l. We also adopt the same classifier as $\mathcal{M}_B$, which uses a linear layer and a sigmoid transformation, as illustrated in the right dotted box, where β_l and b_l are the weight and bias of the classification layer for label l, respectively. $\hat{y}_l$ is the binary classification probability that X belongs to l.

We set both the word embeddings for text input and the code embeddings for ICD codes as non-trainable to give the best performance. Our proposed method introduces only a small number of learnable parameters for labels.

2.2 Code Embeddings

Each ICD code has a unique identifier and a text description and is structurally situated in a tree hierarchy. Further, based on the reports labelled with any given ICD code, we can obtain sample statistics of the code usage. We propose to learn embeddings or representations that capture the above textual, structural, and statistical characteristics of ICD codes, as described below.

Textual code embeddings are obtained by either (a) averaging word vectors (Mikolov et al., 2013) of the words in the description of a code (denoted as CE-w2v) or (b) learning the contextual representation of the code description with BERT (Devlin et al., 2019) (denoted as CE-BERT). For CE-w2v, we use gensim[1] to train the word vectors with discharge reports. For CE-BERT, we use Keras BERT[2] uncased large model to get contextualized word representations, apply max pooling to all the word representations and then add a linear layer for dimension reduction to get code representations with 100 dimensions. This is integrated end-to-end into our model.

Structural code representations leverage the

[1] https://github.com/RaRe-Technologies/gensim/blob/develop/gensim/models/word2vec.py

[2] https://github.com/CyberZHG/keras-bert

ICD tree hierarchy. We capture the parent-child and sibling relations in triples (e.g., (401, *ParentOf*, 401.1) or (401.1, *SiblingOf*, 401.9). We then feed the triples into a knowledge graph embedding approach such as TransR (Lin et al., 2015) (denoted by CE-TransR).

Statistical code representations are learned from the sample statistics between ICD codes and the discharge reports from the training dataset. We designate the embedding of code l as the weighted average of word vectors, as follows:

$$C_l = \frac{1}{N}\sum_{i=1}^{N} v_{w_i} \sum_{d \in docs(l)} tf(w_i, d), \qquad (2)$$

where the weight of a word vector v_{w_i} is proportional to the sum of term frequencies of the word in the notes that are labelled with the code l; N indicates the size of the dataset vocabulary; docs(l) refers to the set of notes associated with code l; and tf(w_i, d) is the function that returns the term frequency of the word w_i in document d. We apply smoothing with increasing all word counts by one, and denote resulting embeddings as CE-Stat.

3 Experiments

We follow the recent state-of-the-art (SOTA) ICD coding studies and perform experiments on the benchmark Medical Information Mart for Intensive Care-III (MIMIC-III) dataset (Johnson et al., 2016). Specifically, we implement our proposed code embeddings (denoted as CE-xxx) atop the popular CAML baseline (Mullenbach et al., 2018). Note that our approach is amenable to any baseline of choice.

Data: Like previous works (Mullenbach et al., 2018; Vu et al., 2020), we focus on multi-label classification task of mapping the discharge reports in the MIMIC-III dataset to ICD codes. Preprocessing details are listed in Appendix A.1. The resultant preprocessed dataset, termed as FULL, has over 52,700 discharge reports associated with subsets of over 8,929 ICD codes (unlike the 8,921 ICD codes reported in prior works). We evaluated our approach on the FULL dataset.

As our focus was to understand what information in ICD codes enables performance improvement, we also investigated whether and to what extent the choice of a code subset affects performance. Therefore, we created new subsets of MIMIC-III (termed sub-datasets) for further evaluation. Specifically, we selected the top k frequent ICD codes in the FULL MIMIC-III dataset and collated the subset of discharge reports tagged with at least one of the top k frequent codes. We term the sub-datasets as Top-k for k=20, 50, 100 and 300.

Finally, we also evaluated our approach on the more widely used subselection of top-50 codes (termed as Top-50$^+$) (Shi et al., 2017). We note that the Top-50$^+$ dataset is much smaller than the other Top-k and FULL datasets because it excludes reports without associated diagnosis descriptions. The detailed breakdown of the dataset sizes and splits are showed in Appendix A.2.

Evaluations: We evaluate performance against two baseline models (i.e., $\mathcal{M}_B$): (a) CAML which uses a per-label attention mechanism within a convolutional neural network (CNN) classifier and (b) DR-CAML which uses code embeddings to constrain the learned model parameters of CAML (Mullenbach et al., 2018). We provide all parameters and model tuning details of the proposed method in Appendix A.3. We follow prior works and report micro-F1 to evaluate model performance, and showcase detailed comparisons for other common metrics. For each experiment, we report averages from 3 independent runs.

Comparative Results on Top-k Sub-Datasets: Table 1 shows the performance of our CE approach compared with baselines on the 5 MIMIC-III sub-datasets. Our CE approach (any embedding type) outperforms the baselines in all the Top-k sub-datasets. We observe that CE-w2v, CE-BERT and CE-TransR lead to slightly better performance than CE-Stat. CE also obtains comparable results on the FULL dataset compared to the baselines. As prior works did not focus on understanding the relation between information in the codes and model performance, there are no reported results on our Top-20, Top-50, Top-100, and Top-300 datasets. Thus, we only compared with baselines in Table 1.

We highlight that our experiments on the FULL dataset were limited by the memory size of the GPUs used. To address this, we reduced batch size of our method (from 128 to 16) and also applied a linear layer to reduce the number of dimensions (M) from the number of FULL codes to 50. Consequently, for the FULL dataset, our CE approach does not improve over baselines and SOTA. However, as our results indicate ability to consistently improve over baselines for different datasets, we posit that increasing batch size and allowing attention to focus on all the FULL codes would enable

Data size	Top-20	Top-50	Top-100	Top-300	FULL†
Baseline (CAML)	0.681	0.641	0.599	0.555	**0.520**
DR-CAML	0.668	0.641	0.584	0.543	0.509
CE-w2v	0.768	**0.713**	0.684	**0.635**	0.502
CE-BERT	**0.775**	0.710	**0.689**	0.622	0.518
CE-TransR	0.765	0.710	**0.689**	0.623	0.507
CE-stat	0.765	0.711	0.688	0.614	0.500

Table 1: Evaluation results of all data on micro-F1. The default batch size is 128, while † uses 16 as batch size due to memory limit.

	AUC		F1		
Model	Macro	Micro	Macro	Micro	Precision@5
Yuan et al. (2020)	-	-	-	-	0.635
Teng et al. (2020)	-	0.933	-	0.692	0.653
Vu et al. (2020)	**0.925**	**0.946**	**0.666**	**0.715**	**0.675**
Cao et al. (2020)	0.895	0.929	0.609	0.663	0.632
CE-Best	0.914	0.937	0.637	0.694	0.652

Table 2: Results on MIMIC-III for the most frequent 50 labels (Top-50$^+$). Based on the performance, CE-Best corresponds to CE-w2v.

our approach to perform comparably with SOTA.

Comparisons with SOTA on Top-50$^+$: As the Top-50$^+$ benchmark is the common dataset evaluated in all SOTA works, we tabulate the results of our proposed approach on the Top-50$^+$ dataset in relation to previously published SOTA results in Table 2. We observe that our approach outperforms all previous methods in terms of macro-/micro-averaged F1 and AUC, except for Vu et al. (2020) (Vu et al., 2020). The performance of Vu et al. (2020) (Vu et al., 2020) is slightly higher than ours, as they use a model based on bidirectional long short-term memory (Bi-LSTM) (Hochreiter and Schmidhuber, 1997) with a similar but more complex attention mechanism. While we also ran experiments with Bi-LSTMs, we found that they tend to be computationally intensive and often did not converge, and thus focused on the more practical CNNs. We further tried to combine code embeddings of different kinds (e.g. CE-w2v + CE-TransR) to see if there is any synergistic effect, but found that no such combination led to performance improvement. We report results of the combination experiments on Top-50$^+$ in Appendix A.4.

4 Qualitative Analysis

To dissect gains of the code representations, we performed qualitative analyses on the Top-50$^+$ test results.

Data Selection: For each CE embedding, we computed the per-code micro-F1 gains over baseline CAML, summed the gains across all the CE embeddings, and rank-ordered the ICD codes by total micro-F1 gain. Next, we selected the 10 codes with the highest gains over baseline (CE $\gg$ baseline) and also the 10 codes with the least gains over baseline (CE $\approx$ baseline). For the first selection (those with the highest gains over baseline), our 4 CE methods typically improve over the baseline. For the second selection of the 10 lowest gain codes, CE is almost always as good as the baseline in these cases. Specifically, out of all discharge summaries for the second selection, the baseline outperforms all 4 CE methods in only 0.2% of cases and 2 out of 4 CE methods in only 1.2% of cases. Hence, we term this second selection as "CE $\approx$ baseline". For qualitative review, we randomly sampled 5 cases corresponding to each of these 20 codes from the Top-50$^+$ testset and obtained 100 cases.

Review Procedure: All qualitative analyses were performed independently by two clinical reviewers. After analysis, the two reviewers discussed to arrive at consensus. First, for each of the 20 codes selected, reviewers considered the ICD coding guidelines and assessed whether they fall into medical, procedural, or surgical categories. Next, for each of the 100 cases selected, reviewers read the discharge reports and marked out reports that did not have any viable information relating to the code assignment for exclusion from further analysis. Second, for reports deemed viable, the

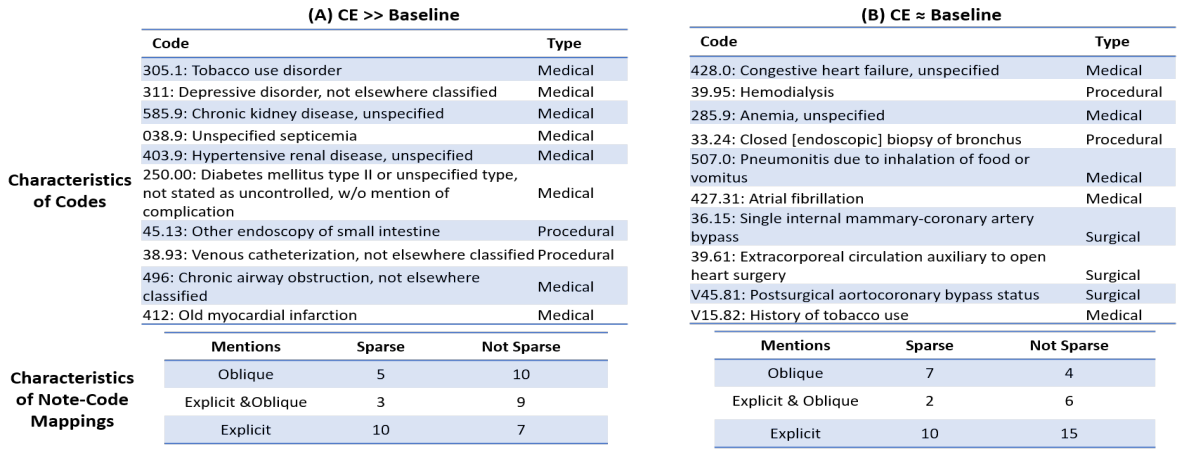

Figure 2: Results of Qualitative Analysis

reviewers assessed whether the reports explicitly delineated the codes (e.g., word-to-word match with code description or synonymous mentions) or contained information that more obliquely relates to the codes (e.g., mentions which might lead a domain expert with specialized knowledge to indirectly infer the code). Third, reviewers further indicated whether the mentions were sparse (1-2 circumscribed mentions) or not (several mentions or extensive sections relating to the code). Finally, reviewers marked out whether the reports had diverse expressions linking to the codes.

Qualitative Analysis Results: Figure 2 details the results. Comparing the code characteristics, we observe that codes where CE gains more tend to (a) have descriptions that include "unspecified" or "not elsewhere classified" and (b) fall into the medical category. In contrast, codes where CE does not gain much tend to be more procedural or surgical in nature. Next, comparing characteristics of the mappings between the notes and the codes, we observe that cases where CE gains more tend to have more oblique mentions; while cases where CE does not gain much tend to have more explicit mentions. This suggests that code embeddings may provide more gains in cases where the discharge reports more obliquely correspond to the code. We detail more in Appendix A.5 and A.6 by providing excerpts from 2 exemplar cases and also showing that CE enables strong Micro-F1 gains on the oblique codes (codes with descriptions including "unspecified" or "not elsewhere classified") of the FULL dataset.

We found that the numbers of cases with sparse mentions were similar for the cases where CE gained more vs. less over baseline. That said, the reviewers did observe that codes such as "Tobacco use disorder" were largely associated with sparse mentions and these kinds of cases were more likely to be accurately predicted with CE than with the baseline. We also note that the cases corresponding to higher gains for CE tended to have more diversity in expression.

5 Conclusions and Future Work

We proposed and characterized methods to leverage representations that capture statistical, textual, and structural properties of medical codes for clinical report coding. We implemented the proposed method on a simple but efficient baseline system and demonstrated substantial performance improvements in micro-F1. Additionally, we performed qualitative evaluation studies to show that our method is more useful in cases when the code prediction task is more ambiguous or nuanced. Future work will experiment with more general datasets and enhancements of the attention network to further improve performance.

Acknowledgments

This research was supported by funding for Digital Health and Deep Learning from the Institute for Infocomm Research (I2R) and the Science and Engineering Research Council (Project No. A1818g0044), A*STAR, Singapore. This work was also supported by resources and infrastructure provided by the A*STAR AI Programme.

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*.

Bo Bach and Michael B First. 2018. Application of the icd-11 classification of personality disorders. *BMC psychiatry*, 18(1):1–14.

Tal Baumel, Jumana Nassour-Kassis, Michael Elhadad, and Noémie Elhadad. 2017. Multi-label classification of patient notes a case study on ICD code assignment. *ArXiv*, abs/1709.09587.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020. HyperCore: Hyperbolic and Co-graph Representation for Automatic ICD Coding. In *ACL*, pages 3105–3114.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, pages 4171–418.

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

Shelli L Feder, Nancy S Redeker, Sangchoon Jeon, Dena Schulman-Green, Julie A Womack, Janet P Tate, Roger J Bedimo, Matthew J Budoff, Adeel A Butt, Kristina Crothers, et al. 2018. Validation of the icd-9 diagnostic code for palliative care in patients hospitalized with heart failure within the veterans health administration. *American Journal of Hospice and Palliative Medicine®*, 35(7):959–965.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):160035.

Dee Lang. 2007. *CONSULTANT REPORT - Natural Language Processing in the Health Care Industry*. Cincinnati Children's Hospital Medical Center.

Leah S. Larkey and W. Bruce Croft. 1996. Combining classifiers in text categorization. In *SIGIR*, pages 289–297.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*, pages 2181–2187.

Medicode (Firm). 1996. *ICD-9-CM: International classification of diseases, 9th revision, clinical modification*. Medicode, Salt Lake City, Utah.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*, pages 3111–3119.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *NAACL-HLT*, pages 1101–1111.

A Nie, A Zehnder, R L Page, Y Zhang, A L Pineda, M A Rivas, C D Bustamante, and J Zou. 2018. DeepTag: inferring diagnoses from veterinary clinical notes. *NPJ Digit Med.*, page 60.

A Perotte, R Pivovarov, K Natarajan, N Weiskopf, F Wood, and N Elhadad. 2014. Diagnosis code assignment: models and evaluation metrics. *J Am Med Inform Assoc.*, 21(2):231–237.

Haoran Shi, Pengtao Xie, Zhiting Hu, Ming Zhang, and Eric P Xing. 2017. Towards automated icd coding using deep learning. *arXiv preprint arXiv:1711.04075*.

Fei Teng, Wei Yang, Li Chen, Lufei Huang, and Qiang Xu. 2020. Explainable Prediction of Medical Codes With Knowledge Graphs. *Frontiers in Bioengineering and Biotechnology*, 8:867.

Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. 2020. A Label Attention Model for ICD Coding from Clinical Text. In *IJCAI*, pages 3335–3341.

Pengtao Xie and Eric Xing. 2018. A neural architecture for automated ICD coding. In *ACL*, pages 1066–1076.

Quan Yuan, Jun Chen, Chao Lu, and Haifeng Huang. 2020. The Graph-based Mutual Attentive Network for Automatic Diagnosis. In *IJCAI*, pages 3393–3399.

A Appendix

A.1 Data Preprocessing

We preprocess discharge reports following CAML. By retaining a maximum of 2,500 words for each summary, we obtain a vocabulary of about 52,000 words. We found there is a minor parsing error in CAML data preprocessing. When CAML read diagnosis and procedure codes from MIMIC III PROCEDURES_ICD.csv and DIAGNOSES_ICD.csv using Python function pandas.read_csv(), the data type of codes used is numpy.int64. In fact, the data type should be str. We correct this error by indicating the data type. For full codes, the number of common codes contained in our produced codes and CAML codes is 8706. For top-50 codes, only one code is different.

A.2 Dataset Splits and Details

Data size	Top-20	Top-50	Top-100	Top-300	FULL	Top-50$^+$
Train	42,590	44,804	46,458	47,285	47,724	8,067
Dev.	1,471	1,574	1,600	1,622	1,632	1,574
Test	3,054	3,242	3,291	3,352	3,373	1,730

Table 3: Data splits details

Table 3 shows the number of discharge summaries contained in the training, development and test data for all the top-k, full and Top-50$^+$ data. We can see through sub-selection, Top-50$^+$ is much smaller than the other data.

A.3 Parameters and Model Tuning

Our code embeddings introduce extra training parameters due to the changes in attention structure. For a Top-k dataset, $2k^2$ more parameters are added. For small k, this number is negligible. For greater k, such as the FULL dataset, we add one fully-connected layer after h to reduce the first k in $2k^2$ to a fixed number, so that the number of the introduced parameters is smaller than the number of original parameters of CAML.

We also tuned the batch size and learning rate to enhance performance. For top-20/50/100/300 data, we use a fixed batch size of 128 in all our models. For Top-50$^+$ and FULL, we use a fixed batch size of 16 in all our models. Due to the expensive GPU memory cost in cosine matrix computation and the large number of added feature maps, we add a linear layer to reduce the size of cosine matrix. For all datasets, we set learning rate to 0.001. For CAML based methods, we use the settings from (Mullenbach et al., 2018).

A.4 Detailed Evaluation on Top-50$^+$

We report deeper analyses on the Top-50$^+$ benchmark in Table 4. We first assess whether CE improves performance over CAML by adding more features to the representation of a discharge report. Specifically, we add 50 filters (thus enhancing number of features) to those used in CAML and DR-CAML, and denote the revised models as CAML_add and DR-CAML_add. We observe that the additional filters offer limited improvements in comparison with the CE approach (any embedding). This suggests that our CE approach may not just be adding more features to improve performance. Next we assess if combining the different CE embeddings would enable even better performance. We experiment with several combinations of our different code embeddings: (a) CE+WT combining CE-w2v and CE-TransR, (b) CE+WTS combining CE-w2v, CE-TransR and CE-Stat, and (c) CE+BTS combining CE-BERT, CE-TransR and CE-Stat. We observe that the combinations of CE embeddings do not improve the performance much over individual CE embeddings. This suggests that dot product of discharge summary representation with concatenation of multiple code representations may not have synergistic effects.

	AUC		F1		
Model	Macro	Micro	Macro	Micro	P@5
CAML	0.870	0.913	0.521	0.614	0.612
DR-CAML	0.870	0.906	0.541	0.612	0.606
CAML_add	0.884	0.920	0.547	0.626	0.621
DR-CAML_add	0.878	0.916	0.546	0.618	0.616
CE-w2v	**0.914**	**0.937**	0.637	**0.694**	0.652
CE-BERT	0.913	0.936	0.638	0.692	0.651
CE-TransR	0.913	**0.937**	0.636	0.693	**0.654**
CE-Stat	0.911	0.936	0.633	0.687	0.651
CE+WT	**0.914**	**0.937**	0.640	0.693	0.647
CE+WTS	0.912	0.935	**0.649**	0.689	0.650
CE+BTS	0.912	0.936	0.644	0.692	0.650

Table 4: Results on MIMIC-III, 50 labels (Top-50$^+$). P@5 means precision at 5.

A.5 Constrained Evaluations on FULL

Method	Micro-F1
CAML	0.106
DR-CAML	0.106
CE-w2v	0.169
CE-BERT	0.119
CE-TransR	0.114
CE-Stat	0.122

Table 5: Micro-F1 on Oblique Codes of Full

We look into the oblique codes in the testing data of FULL. We select the codes of which the code descriptions containing keywords from [“unspecified”, “not elsewhere classified”, “other”]. Table 5

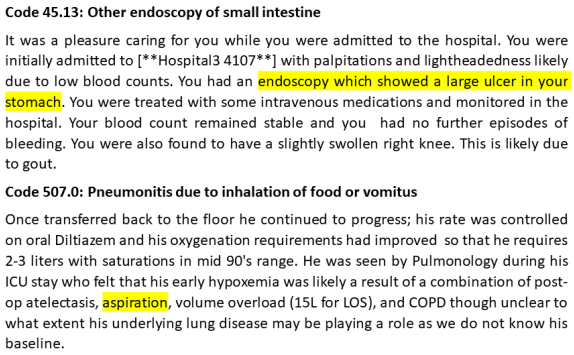

Figure 3: Illustrations of Oblique Coding Cases

shows the Micro-F1 scores of our code embedding methods compared with the baseline methods. From the table, we can see our methods perform better on the oblique codes, especially CE-w2v.

A.6 Case Illustrations

To provide richer insight on the qualitative analysis, we provide two case illustrations, shown in Figure 3. In both cases, the indicated ground truth codes were missed by the baseline but predicted correctly by our CE approach. In the first case (i.e., code 45.13), there are synonym mentions of "EGD" in the Major Surgical Procedure, Images, and Brief Hospital Course subsections of the report. However, indirect phrases on the type of endoscopy performed in the Discharge Instructions imply that this is specifically a case of upper gastrointestinal endoscopy, which leads to the said code assignment. In the second case (i.e., code 507.0), there are no explicit mentions of pneumonitis with vomitus anywhere in the discharge report. However, there is only one oblique mention of "aspiration" without the word pneumonia or its equivalent. As this code is also often termed as "aspiration pneumonia" in medical parlance, the oblique mention ties down the link between the report and the said code assignment.