Fusion: Towards Automated ICD Coding via Feature Compression

Junyu Luo
Pennsylvania State University
junyu@psu.edu

Cao Xiao
Amplitude
danica.xiao@amplitude.com

Lucas Glass
IQVIA
lucas.glass@iqvia.com

Jimeng Sun
UIUC
jimeng@illinois.edu

Fenglong Ma*
Pennsylvania State University
fenglong@psu.edu

Abstract

ICD coding aims to automatically assign International Classification of Diseases (ICD) codes from unstructured clinical notes or discharge summaries, which saves human labor and reduces errors. Although several studies are proposed to solve this challenging task, none distinguishes the importance of different phrases with a word window. Intuitively, informative phrases should be useful for the prediction. This paper proposes a feature compressed ICD coding model named Fusion to address this issue. In particular, we propose an attentive soft-pooling approach to compress the sparse and redundant word representations into informative and dense ones as local features. Next, we use the key-query attention mechanism for modeling the inner relations among local features to generate the global features, which are further used to predict ICD codes. Experiments on two widely used datasets demonstrate that Fusion is comparable with baselines. We also find that none of the state-of-the-art approaches significantly perform better than others. Thus, automated ICD coding is still a challenging task.

1 Introduction

The International Classification of Diseases (ICD) coding system helps standardize the recording of diagnoses and treatments assigned to patients by medical professionals in the world. These ICD codes are generated from massive unstructured clinical notes. However, manual code assignments is labor-intensive and prone to errors. Thus, automatic ICD code assignment becomes an urgent need in the healthcare domain.

Traditional machine learning methods (Larkey and Croft, 1996) tried to tackle this task based on feature extraction. However, it does not work well since clinical notes are noisy and complex. Recently, deep learning-based approaches (Cao et al., 2020; Xie et al., 2019; Li and Yu, 2020; Mullenbach et al., 2018) are proposed to improve its performance. Among others, convolutional methods (Cao et al., 2020; Xie et al., 2019; Li and Yu, 2020; Mullenbach et al., 2018) outperform other approaches. Besides, some studies try to incorporate external information to further improve the performance (Cao et al., 2020; Xie et al., 2019). However, they still suffer the following issues.

- **Redundant Information Deduction**. The clinical notes are noisy and complex, where only some key phrases are highly related to the coding. However, convolutional methods treat all the word windows equally, ignoring that different words have different importance and should be weighted differently within word windows. Besides, the sliding windows used in the convolutional methods produce a lot of redundant information. Thus, it is important to reduce the non-informative and redundant information and distinguish the contributions of different convolutional features.

- **Interactions among Local Features**. Most existing approaches such as MultiResCNN (Li and Yu, 2020) only use the local features for coding obtained using different filters. However, they ignore the importance of interactions among different local features. For example, *sleep apnoea (OSA)* and *insomnia* are related to *hypertension* and *ischaemic heart disease* (Harrison and Wood, 1949). Thus, combining different local features may discover new useful patterns to improve coding.

To tackle these issues, we propose a **f**eat**u**re compre**s**sed **I**CD c**o**di**n**g model named Fusion, which can automatically compress the local fea-

* Corresponding author.

tures and further learn global features to enhance the coding performance. In particular, Fusion uses an LSTM network to stack the segments of Bert embeddings from truncated clinical notes as iputs, and takes an attention-based soft-pooling approach to compress local features learned by word convolutions, passing residual convolution blocks. By aggregating all the local features from different convolutional filters, Fusion then applies key-query attention mechanism to model interactions among local features and obtain global ones. A code-wise attention mechanism is then used to learn a feature vector associated with each ICD code. This vector is finally used to make a prediction. Experiments on two public datasets show that Fusion outperforms state-of-the-art baselines over five evaluation metrics. Moreover, we find that none of the existing approaches outperforms others on the MIMIC-III dataset. Thus, automated ICD coding is still an open challenge.

2 Related Work

Traditional machine learning models have been applied to automatically extract ICD codes using the hand-crafted feature vectors as the inputs (Larkey and Croft, 1996; Gundersen et al., 1996; Franz et al., 2000; Pestian et al., 2007; Farkas and Szarvas, 2008). However, they did not achieve satisfactory performance due to the difficulty of extracting useful features from complex and noisy clinical notes. Deep learning models have shown their superiority for this task, including recurrent-based deep models (Shi et al., 2017; Li et al., 2018; Xu et al., 2019) and convolution-based models (Kim, 2014; Mullenbach et al., 2018; Cao et al., 2020; Li and Yu, 2020). In general, convolutional models perform better than recurrent-based ones. Several studies try to incorporate advanced pretrained language model BERT (Devlin et al., 2019), ICD code descriptions (Wang et al., 2018; Mullenbach et al., 2018; Xie and Xing, 2018; Li and Yu, 2020), ICD code structure (Wang et al., 2020; Cao et al., 2020), and knowledge graph (Cao et al., 2020; Xie et al., 2019) to improve the performance.

3 Model

The goal of automated ICD coding is to predict a set of unique ICD codes $\mathcal{Y}$ from the code set $\mathcal{C} = \{c_1, c_2, \cdots, c_s\}$ when given clinical note $D = \{w_1, w_2, \cdots, w_n\}$, where $\mathcal{Y} \subseteq \mathcal{C}$, s is the number of unique ICD codes, and n is the number of words in D. This task is challenging since s is very large, which is over 15,000 for ICD-9 codes and 60,000 for ICD-10 codes, respectively. Besides, extensive noisy information exists in the clinical note D.

To solve these challenges, we propose a feature denoised model (Fusion) for automated ICD coding as shown in Figure 1. This model consists of six modules: the Input layer, the RoBert layer, the compressed convolutional layer, the feature aggregation layer, the code-wise attention layer, and the prediction layer. Next, we introduce the details of each module in the following subsections.

3.1 Input Layer

We take the clinical note $D = \{w_1, w_2, \cdots, w_n\}$ as the model input. For each unique word w_i, word2vec (Mikolov et al., 2013) is used to pretrain its embedding, which is denoted as $\mathbf{e}_i$, a d_e-dimensional embedding. Thus, the input of Fusion is a matrix $\mathbf{D} = \{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n\}$.

3.2 Compressed Convolutional Layer

Given the input data $\mathbf{D}$, the compressed convolutional layer aims to learn dense and informative word representations, which are further used to learn the clinical note representation. In particular, we first use convolutional neural networks (CNN) to learn word representations and then propose an attention-based soft-pooling approach to compress those representations. Finally, residual convolution blocks (He et al., 2016) are introduced as MultiResCNN (Li and Yu, 2020) on top of the compressed features.

3.2.1 Word Convolution

CNNs are powerful for text classification tasks (Kim, 2014) that they have multiple filters with different kernel sizes (i.e., word windows) to capture diverse patterns. Let m be the number of filters. The kernel of each filter f_i is denoted as k_i. Thus, we can apply m different 1-dimensional convolutions on the input data $\mathbf{D}$. For the i-th filter, we have

$$\mathbf{x}_j^i = conv(\{\mathbf{e}_j, \mathbf{e}_{j+1}, \cdots, \mathbf{e}_{j+k_i-1}\}; \mathbf{W}_x^i), \quad (1)$$

where $conv(\cdot;\cdot)$ represents the 1-dimensional convolutional operation, and $\mathbf{W}_x^i$ denotes the learned parameter.

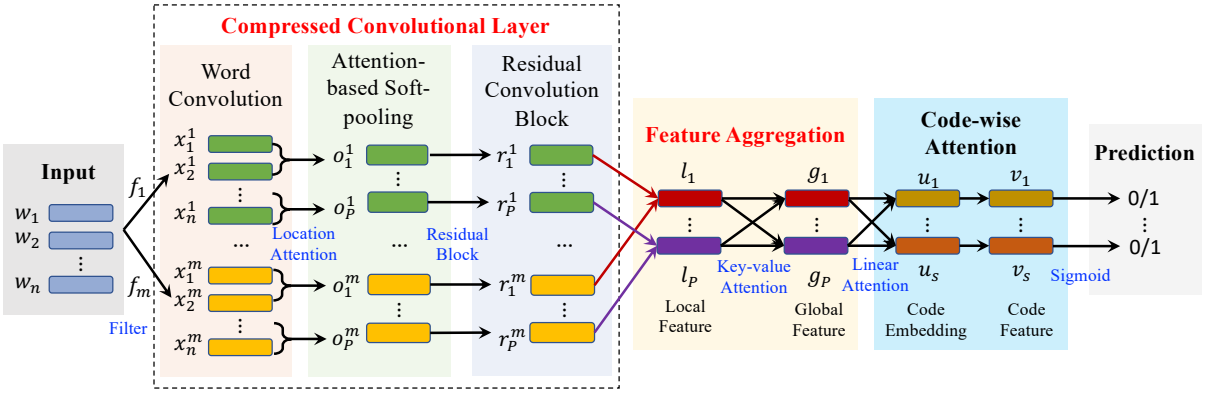

Figure 1: Overview of the proposed Fusion.

3.2.2 Attention-based Soft-pooling

The word convolutional operation uses sliding windows, which produces redundant information existing in adjacent word representations. Thus, to remove such information, we propose to compress word representations learned by Eq. (1) via an attention-based soft-pooling operation.

Given a word w_j, its neighboring words $\{w_{j+1}, \cdots, w_{j+g-1}\}$, and the corresponding filter f_i, we first learn the local-based attention scores $\alpha_j^i = \mathbf{W}_\alpha^i \mathbf{x}_j^i + b$ with softmax function, i.e., $[\beta_j^i, \cdots, \beta_{j+g-1}^i] = softmax([\alpha_j^i, \cdots, \alpha_{j+g-1}^i])$, where $\mathbf{W}_\alpha^i$ and b are learnable parameters. Then we conduct attention-based soft-pooling on the g words and obtain the compressed representation as in Eq. (2).

$$\mathbf{o}_p^i = \sum_{q=j}^{j+g-1} \beta_q^i \mathbf{x}_q^i \tag{2}$$

In such a way, the whole n word representations learned by Eq. (1) will be replaced by $P = \lceil \frac{n}{g} \rceil$ new representations, i.e., $\{\mathbf{o}_1^i, \mathbf{o}_2^i, \cdots, \mathbf{o}_P^i\}$. In such a way, we can reduce the number of word representations and obtain more dense ones.

3.2.3 Residual Convolution Block

For each filter f_i, we now have a denoised matrix $\{\mathbf{o}_1^i, \mathbf{o}_2^i, \cdots, \mathbf{o}_P^i\}$ that represents the input $\mathbf{D}$. To avoid vanishing gradients and train the model easier, we also introduce residual blocks on top of the compressed features. In particular, we replace the batch norm layer with the group norm layer. Let a denotes the number of residual blocks, and we have $\mathbf{r}_p^i = ResidualBlcok(\{\mathbf{o}_p^i, \cdots, \mathbf{o}_{p+a-1}^i\})$.

3.3 Feature Aggregation Layer

Since m filters are used to obtain m compressed features, we concatenate them together as the local features, i.e., $\mathbf{l}_p = [\mathbf{r}_p^1, \cdots, \mathbf{r}_p^m]$. Then the whole document can be represented by a matrix $\mathbf{D}_l = \{\mathbf{l}_1, \mathbf{l}_2, \cdots, \mathbf{l}_P\}$. However, such an aggregation only takes local information into account but ignores the interactions with the remaining words.

Thus, we propose to use the key-query attention mechanism (Vaswani et al., 2017) to learn a global feature representation for each compressed word window. Thus, we have the global features $\mathbf{D}_g = [\mathbf{g}_1, \cdots, \mathbf{g}_P] = attention([\mathbf{l}_1, \cdots, \mathbf{l}_P])$.

3.4 Code-wise Attention Layer

Due to a large number of labels, directly using the global features $\mathbf{D}_g$ to make predictions may not perform well. Thus, we use a code-wise attention layer to generate a matching vector for each ICD code used to make a prediction. Let $\mathbf{u}_k$ represent the embedding of the k-th ICD code, i.e., c_k. Then we calculate the attention weights on all the global features using $\mathbf{u}_k$, i.e., $[\gamma_1^k, \cdots, \gamma_P^k] = softmax([\mathbf{u}_k \mathbf{g}_1, \cdots, \mathbf{u}_k \mathbf{g}_P])$. Then the code-wise vector can be obtained by $\mathbf{v}_k = \sum_{p=1}^P \gamma_p^k \mathbf{g}_p$.

3.5 Prediction Layer

Using the code-wise vector $\mathbf{v}_k$, we can make a prediction using the sigmoid function, i.e.,

$$\tilde{y}_k = (1 + \exp(\mathbf{w}_k^\top \mathbf{v}_k))^{-1}, \tag{3}$$

where $\mathbf{w}_k$ is the learnable parameter vector. Finally, cross-entropy loss function on a specific clinical note $\mathbf{D}$ is used to optimize the proposed model.

$$\mathcal{L} = -\sum_{k=1}^{s} (y_k \log(\tilde{y}_k) + (1 - y_k) \log(1 - \tilde{y}_k)). \tag{4}$$

Dataset		MIMIC-III 50					MIMIC-III Full				
Setting	Model	AUC		F1		P@N	AUC		F1		P@N
		Macro	Micro	Macro	Micro	5	Macro	Micro	Macro	Micro	8
Note Only	Fusion	**0.909**	**0.933**	**0.619**	**0.674**	**0.647**	**0.915**	**0.987**	0.083	**0.554**	**0.736**
	C-MemNN	0.833	–	–	–	0.420	–	–	–	–	–
	C-LSTM-ATT	–	0.900	–	0.532	–	–	–	–	–	–
	CAML	0.875	0.909	0.532	0.614	0.609	0.895	0.986	0.088	0.539	0.709
	DR-CAML	0.884	0.916	0.576	0.633	0.618	0.897	0.985	0.086	0.529	0.690
	MultiResCNN	0.899	0.928	0.606	0.670	0.641	0.910	0.986	0.085	0.552	0.734
Note + Ontology	HyperCore	0.895	0.929	0.609	0.663	0.632	0.930	0.989	0.090	0.551	0.722
	MSATT-KG	0.914	0.936	0.638	0.684	0.644	0.910	0.992	0.090	0.553	0.728

Table 1: Experiment results on MIMIC-III 50 and MIMIC-III Full datasets.

4 Experiment

4.1 Datasets

We conduct experiments on two public datasets MIMIC-III 50 and MIMIC-III Full (Johnson et al., 2016) to extract ICD-9 codes from discharge summaries. We use the same setting as previous works (Mullenbach et al., 2018; Shi et al., 2017; Li and Yu, 2020; Cao et al., 2020). The MIMIC-III 50 dataset contains the top 50 most frequent codes, 8,067, 1,574, and 1,730 discharge summaries for training, development, and testing, respectively. The MIMIC-III Full dataset consists of 8,921 codes, 47,719, 1,631, and 3,372 discharge summaries for training, development, and testing, respectively. The number of labels on the MIMIC-III Full dataset is significantly greater than that on the MIMIC-III 50 dataset, making the task more difficult.

4.2 Metrics and Parameter Settings

We follow previous work (Mullenbach et al., 2018) and use Micro Macro AUC (area under the ROC), Micro Macro F1, and Precision@K scores as metrics. For MIMIC-III 50, we report Precision@5 (P@5) and P@8 for MIMIC-III Full. We use the same parameter setting as MultiResCNN (Li and Yu, 2020)[1], and set g as 2 in our experiments, i.e., compress two features together.

4.3 Baselines

Existing studies either only take clinical notes as the inputs or incorporate external information, working with notes to enhance the performance. Our work belongs to the first category. For the "note only" category, we employ C-MemNN (Prakash et al., 2017), C-LSTM-ATT (Shi et al., 2017), CAML (Mullenbach et al., 2018), DR-CAML (Mullenbach et al., 2018), and MultiResCNN (Li and Yu, 2020) as baselines. We also use HyperCore (Cao et al., 2020), and MSATT-KG (Xie et al., 2019) as baselines, which incorporate the ICD code ontology to enhance the performance. Since all the approaches use the same settings, we directly use the results reported in the original papers.

4.4 Performance Analysis

Table 1 shows the experimental results of all approaches in terms of different metrics. We can observe that overall Fusion outperforms all the baselines in the "Note Only" setting on both the MIMIC-III 50 and MIMIC-III datasets in terms of all the metrics. These results clearly demonstrate the effectiveness of the proposed feature compression and aggregation approaches for the automated ICD coding task.

Although HyperCore and MSATT-KG incorporate external information to improve the performance, the performance of Fusion is still comparable. On the MIMIC-III Full dataset, our model is even better at F1 Micro and P@N scores. On the MIMIC-III 50 dataset, Fusion also achieves the highest P@N score without using any additional knowledge. We also can observe that on the MIMIC-III Full dataset, none of the methods can be significantly better than others. The reason may be that all the models cannot be trained sufficiently with the huge number of ICD code labels on noisy, sparse, and unstructured medical clinical notes, which makes this task more challenging.

4.5 Ablation Study

In this section, we remove parts of the full Fusion model to validate the contribution of each individual module. Table 2 shows the ablation study results. "MaxPool" means replacing our soft-pooling layer with the traditional max-pooling layer. As

[1] https://bit.ly/3opDmjM

Dataset	MIMIC-III 50					MIMIC-III Full				
Model	AUC		F1		P@N	AUC		F1		P@N
	Macro	Micro	Macro	Micro	5	Macro	Micro	Macro	Micro	8
Fusion	0.909	0.933	0.619	0.674	0.647	0.915	0.989	0.088	0.554	0.736
MaxPool	0.908	0.929	0.624	0.669	0.639	0.900	0.986	0.081	0.552	0.726
DocLevel	0.853	0.886	0.477	0.556	0.551	-	-	-	-	-

Table 2: Ablation experiment results on MIMIC-III 50 and MIMIC-III Full datasets.

Model	AUC		F1		P@N
	Macro	Micro	Macro	Micro	5
Fusion+	0.931	0.950	0.683	0.725	0.679
Fusion	0.909	0.933	0.619	0.674	0.647

Table 3: Performance evaluation with extra data.

shown in Table 2, the results drop on all metrics except the F1 Macro, which indicates the importance and benefits of using the proposed soft-pooling layer. Max-pooling will lose part of critical information during the compression and is not differentiable. With soft-pooling, the key information can be better preserved during the compression process, since the selection process is guided by the gradient.

"DocLevel" refers to replacing the code-wise attention layer with the single document-level attention. The attention is based on the document feature, and all codes use the same attention weight during the prediction instead of calculating code-specific attention weights. Thus, all codes will use the same feature for the prediction. In such a way, much unrelated information will also be kept. For example, we do not want to preserve the heart-failure-related information while predicting the COPD code. As a result, most scores drop significantly compared to the original design. The introduction of the code-specific attention makes it possible that the predictor can dynamically adjust the attentions based on the cases. Thus, the redundant information can be better removed with our design.

4.6 Model Training with Extra Data

In this section, we aim to validate whether using extra data can improve the performance of the proposed Fusion. Towards this end, we use the training data of the MIMIC-III full dataset to train the model to predict the top 50 most frequent codes. The testing set is the same as that of the MIMIC-III 50 dataset. Table 3 shows the results. We can observe that using extra training data significantly improves the performance of the ICD coding task.

5 Conclusion

In this paper, we propose Fusion for the automated ICD coding task. In particular, Fusion uses RoBert to embed the notes, focuses on compressing redundant feature information, distinguishing the importance of adjacent phrases, and considering interactions among local features. We conduct experiments on two widely-used datasets to show the effectiveness of Fusion in terms of five evaluation metrics. From experimental results on the MIMIC-III Full dataset, we find that automated ICD coding is still challenging due to the noisy data and a large number of ICD code labels.

References

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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 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 (1)*.

Richárd Farkas and György Szarvas. 2008. Automatic construction of rule-based icd-9-cm coding systems. In *BMC bioinformatics*, volume 9, page S10. Springer.

Pius Franz, Albrecht Zaiss, Stefan Schulz, Udo Hahn, and Rüdiger Klar. 2000. Automated coding of diagnoses–three methods compared. In *Proceedings of the AMIA Symposium*, page 250. American Medical Informatics Association.

Michael L Gundersen, Peter J Haug, T Allan Pryor, Rudy van Bree, Spence Koehler, Kay Bauer, and Brenda Clemons. 1996. Development and evaluation of a computerized admission diagnoses encoding system. *Computers and Biomedical Research*, 29(5):351–372.

CV Harrison and Paul Wood. 1949. Hypertensive and ischaemic heart disease: A comparative clinical and pathological study. *British heart journal*, 11(3):205.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, 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):1–9.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Leah S Larkey and W Bruce Croft. 1996. Combining classifiers in text categorization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 289–297.

Fei Li and Hong Yu. 2020. Icd coding from clinical text using multi-filter residual convolutional neural network. In *AAAI*, pages 8180–8187.

Min Li, Zhihui Fei, Min Zeng, Fang-Xiang Wu, Yaohang Li, Yi Pan, and Jianxin Wang. 2018. Automated icd-9 coding via a deep learning approach. *IEEE/ACM transactions on computational biology and bioinformatics*, 16(4):1193–1202.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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

John Pestian, Chris Brew, Pawel Matykiewicz, Dj J Hovermale, Neil Johnson, K Bretonnel Cohen, and Wlodzislaw Duch. 2007. A shared task involving multi-label classification of clinical free text. In *Biological, translational, and clinical language processing*, pages 97–104.

Aaditya Prakash, Siyuan Zhao, Sadid Hasan, Vivek Datla, Kathy Lee, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2017. Condensed memory networks for clinical diagnostic inferencing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

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*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint embedding of words and labels for text classification. *arXiv preprint arXiv:1805.04174*.

Shanshan Wang, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jian-Yun Nie, Jun Ma, and Maarten de Rijke. 2020. Coding electronic health records with adversarial reinforcement path generation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 801–810.

Pengtao Xie and Eric Xing. 2018. A neural architecture for automated icd coding. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1066–1076.

Xiancheng Xie, Yun Xiong, Philip S Yu, and Yangyong Zhu. 2019. Ehr coding with multi-scale feature attention and structured knowledge graph propagation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 649–658.

Keyang Xu, Mike Lam, Jingzhi Pang, Xin Gao, Charlotte Band, Piyush Mathur, Frank Papay, Ashish K Khanna, Jacek B Cywinski, Kamal Maheshwari, et al. 2019. Multimodal machine learning for automated icd coding. In *Machine Learning for Healthcare Conference*, pages 197–215. PMLR.