BEnTo: Transformers Language Models for ICD Coding

Haau-Sing Li,[1] **Kyunghyun Cho,**[1,2] **and Narges Razavian**[3,4]
[1]Center for Data Science, [2]Dept. of Computer Science, [3]Dept. of Population Health, [4] Dept. of Radiology
New York University
{xl3119, kyunghyun.cho}@nyu.edu, narges.razavian@nyulangone.org

Abstract

Automatic ICD coding has attracted much attention from the research community as it saves time and labor for the exhausting work. Previous work on ICD coding utilizes Convolutional Neural Network (CNN) or Recurrent Neual Network (RNN) to generate sentence-level representations, and recent efforts on improving ICD coding performances have been focusing on either constructing fancier architectures based on these representations, or utilizing information within ICD codes to enhance model performances. In this paper, we try to improve model performances by changing the architecture of encoding sentence-level representations with Transformers Language Models (LMs) such as BERT and RoBERTa. However, most Transformers LMs accepts sequences with no more than 512 tokens. We propose the **B**i-directional **En**semble **T**ransf**o**rmers (BEnTo) which makes it possible for us to utilize Transformers LM for ICD coding. Our method outperforms previous work that relys on CNN and RNN in AUC scores, but show low F1 scores.

1 Introduction

The International Classification of Diseases (ICD) is a healthcare classification system supported by the World Health Organization. It serves as a unique and standardized classification system indicating diseases, symptoms, signs, etc.. ICD codes have been used in a variety of ways, including billing and predicting clinical events (Denny et al., 2010; Ranganath et al., 2015; Choi et al., 2016; Avati et al., 2018). Since manual ICD coding has been demonstrated as expensive, time-consuming, and error-prone, research communities have been studying automatic ICD coding. The task can be viewed as a multi-label classification problem given clinical notes. This task is difficult because of two factors. First, we have a high-dimensional label space, with over 15,000 ICD codes in ICD-9 taxonomy, and over 140,000 codes combined in the newer ICD-10-CM and ICD-10-PCS taxonomies (Gr, 1988), which is strongly imbalanced. Second, the clinical notes used to predict ICD codes includes misspellings and hardly recognizable abbreviations, and are usually long documents with irrelevant information.

Among all methods of automatic ICD coding, methods with deep learning architectures have been the most successful (Perotte et al., 2014; Mullenbach et al., 2018; Li and Yu, 2020; Cao et al., 2020; Vu et al., 2020). However, all these models haven't utilized pretrained Transformers Language Models (LMs) (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020; Beltagy et al., 2020), which have achieved the state-of-the-art performances in a number of Natural Language Understanding (NLU) tasks. This is because sequence lengths of clinical notes usually exceed the maximum sequence length that a standard Transformers LM accepts.

In this paper, we propose an ensemble model used for ICD Coding. We only add a fully-connected layer of each model we use. By ensembling all predictions we have, we have obtained results close to or better than Mullenbach et al. (2018) in AUC scores. However, since the models have low recall, our results are still far behind Mullenbach et al. (2018) in F1 scores.

2 Related Work

Automatic ICD Coding. Automatic ICD coding is challenging and important in the medical informatics community, and has been studied extensively with traditional machine learning methods (Perotte et al., 2014; Kavuluru et al., 2015) and neural network methods (Shi et al., 2017; Xie and Xing, 2018). Specific to ICD coding with discharge summary, Perotte et al. (2014) propose a hierarchical SVM. More recently, Mullenbach et al. (2018) propose a method using a CNN followed by an attention layer. Li and Yu (2020) extend that single-filter CNN to multiple filters with residual connection.

998.32: Disruption of external operation wound ... wound infection, and **wound breakdown** ...
428.0: Congestive heart failure ... DIAGNOSES: 1. **Acute congestive heart failure** 2. Diabetes mellitus 3. Pulmonary edema ...
202.8: Other malignant lymphomas ... a 55 year-old female with **non Hodgkin's lymphoma** and acquired C1 esterase inhibitor deficiency ...
770.6: Transitory tachypnea of newborn ... Chest x-ray was consistent with **transient tachypnea of the newborn** ...
424.1: Aortic valve disorders ... mild **aortic stenosis with an aortic valve area** of 1.9 cm squared and 2+ **aortic insufficiency** ...

Table 1: Results on MIMIC-III, 50 labels from Li and Yu (2020).

Cao et al. (2020) propose a method that utilizes both the method proposed by Mullenbach et al. (2018) and graphical structure within ICD codes. Vu et al. (2020) propose a method that combines a Bi-LSTM, a label-wise attention mechanism, and a hierachical classifier that both predicts the chapter of an ICD code and the code itself. See **??** for examples.

Long Text Classification with Transformers LMs. Standard Transformers LMs accepts up to 512 tokens of the sequence, which fails to meet lengths of long documents with more than 512 tokens. To tackle the problem of long-text classification with standard transformers, Pappagari et al. (2019) separate texts into non-overlapping snippets, pass them respectively to Transformers LM, and aggregate the [CLS] representations by a recurrent layer or a transformer layer before feeding the final sentence-level representation to a fully-connected layer. In terms of discharge summaries, Mulyar et al. (2019) propose a similar method in aggregating [CLS] representations. Huang et al. (2020) utilizes rule-based method to extract useful snippets before passing each of them into a Transformers LM, and aggregate the snippet-level representations through attention layers.

3 Method

In this section, we introduce our **Bi**-directional **En**semble **T**ransf**o**rmers (BEnTo), which consists of a two types of model, a snippet-based type and an ngram-based type, to capture short-term and long-term dependencies respectively. We first describe each type of model in details, then talk about ensembling model predictions.

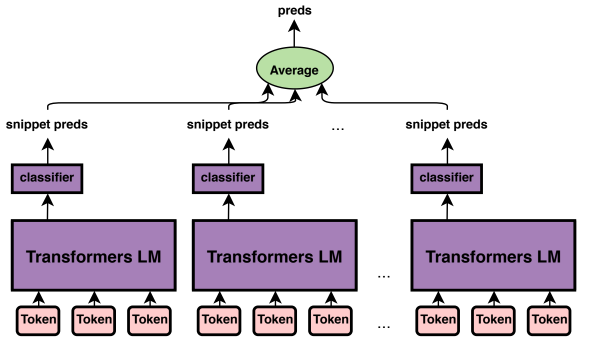

Figure 1: The architecture of snippet-based model. Note that during training, the loss function is computed through snippet-level predictions. We only average predictions for evaluation.

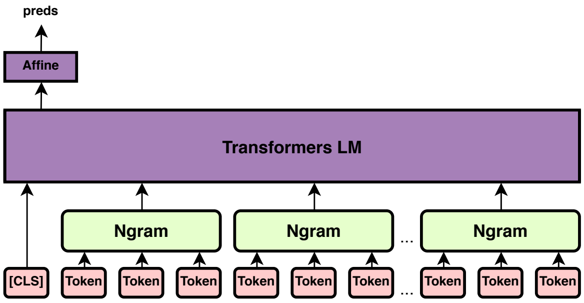

Figure 2: The architecture of ngram-based model.

3.1 Snippet-Based Model

Let $X = (x_1, x_2, ..., x_T)$ to be a discharge summary with length, and the number of all ICD codes is L. We first extract N snippets of M ($M < 512$) tokens sequentially to cover the whole sequence of discharge summary. Note that we always obtain the last M' tokens from the last snippet.

We pass each model to a standard Transformers LM, where we obtained hidden states of snippet i as $H_i^{snippet} = (h_1^i, h_2^i, ..., h_M^i)$. We simply add a fully-connected layer with a sigmoid fuction after the [CLS] hidden state h_1^i to obtain predictions $\hat{y_i}^{snippet} \in \mathcal{R}^L$.

During training, we use the corresponding ICD codes of the entire discharge summary sequence as the labels for every snippet extracted from the sequence. In testing time, we simply average the sigmoid of logits of all snippets, i.e.

$$\hat{y}^{snippet} = \frac{1}{N} \sum_{i=1}^{N} \hat{y_i}^{snippet}. \tag{1}$$

The architecture is shown in 2.

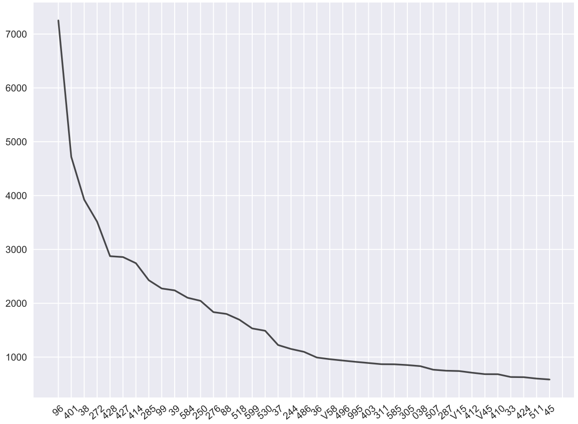

Figure 3: Frequencies of the most frequent 50 ICD Codes.

3.2 Ngram-based Model

We first pass the entire sequence to get word embeddings of the Transformers LM $W = (w_1, w_2, ..., w_T)$. We then compute the Ngram embeddings by adding up word embeddings within the Ngram of size G. Note that we keep embedding of [CLS], and we only consider non-overlapping Ngrams, i.e. the ith Ngrams's embedding e_i is computed as

$$e_i = \sum_{j=G*(i-1)+2}^{G*(i)+1} w_j. \tag{2}$$

We then pass the $K + 1$ the Ngram embeddings $E = (w_1, e_1, ..., e_K)$ to the Transformers LM and get the contextual embeddings $H^{ngram} = (h_1, h_2, ..., h_{K+1})$. We simply add a fully-connected layer with a sigmoid function after the [CLS] hidden state h_1 to obtain predictions $\hat{y}^{ngram} \in \mathcal{R}^L$. The architecture is shown in 1.

3.3 Ensembling Model Predictions

Assume we have C model predictions, one way of ensembling is to take the element-wise average of the predictions, i.e.

$$\hat{y} = \frac{1}{C}\sum_{c=1}^{C} \hat{y}_c. \tag{3}$$

After finding that individual models have low recalls, we also consider taking the element-size maximum values of predictions.

4 Experiments

4.1 Data

In this paper, we use the third version of **M**edical **I**nformation **M**art for **I**ntensive **C**are (MIMIC-III) (Johnson et al., 2016).

Discharge Summary of the MIMIC-III. Following previous work, we use discharge summaries, which condensed all information of a stay into a single document. We have In MIMIC-III, some admissions have multiple stays in discharge summary, for which we concatenate them following Mullenbach et al. (2018). There are 8,921 ICD codes in total. For this paper we only consider top-50 codes by frequency.

Data Preprocessing. We follow Mullenbach et al. (2018) in preprocessing the data. We use ICD-9 codes. We keep discharge summaries that contain at least one of the top-50 codes. As a result, we have 8,067, 1574, 1730 discharge summaries in the training, validation, and test set respectively. By simply splitting toklens with spaces, the average text length is 1,797.19, and the maximum text length is 8,417. The frequencies of top-50 ICD codes is shown in 3.

4.2 Hyperparameters

Transformers LM. We first test the ngram-based model with `bert-base-uncased` and `Bio_Discharge_Summary_BERT`. We find little difference between the performances of two models, then consider only using `bert-base-uncased` for ensembling.

Other hyperparameters. We perform hyperparameter searching of ngram-based models with Ngram size in $\{28, 32, 64, 128\}$. Ngram-based models are trained with 32 samples per batch. For snippet-based models, we set snippet size to be 510, and the number of overlapping tokens 128. Snippet-based models are trained with batch size 16. We use all models with Adam optimizer with learning rate 2e-5 and epsilon 1e-8, and save two checkpoints, one with the highest micro AUC score, and one with highest F1 score.

4.3 Baselines

We only compare performances of our models with models provided by Mullenbach et al. (2018) or earlier work. More recent methods proposed by Li and Yu (2020), Cao et al. (2020) and Vu et al. (2020) lead to much better performances in ICD coding.

	AUC		F1		
Model	Macro	Micro	Macro	Micro	P@5
C-MemNN (Prakash et al., 2017)	0.833	–	–	–	0.42
C-LSTM-Att (Shi et al., 2017)	–	0.900	–	0.532	–
Logistic Regression (Mullenbach et al., 2018)	0.829	0.864	0.477	0.533	0.546
CNN (Mullenbach et al., 2018)	0.876	0.907	**0.576***	0.625	**0.620**
Bi-GRU (Mullenbach et al., 2018)	0.828	0.868	0.484	0.549	0.591
CAML (Mullenbach et al., 2018)	0.875	0.909	0.532	0.614	0.609
DR-CAML (Mullenbach et al., 2018)	0.884	**0.916***	**0.576***	**0.633**	0.618
BEnTo-snippet-auc-avg	**0.890***	0.910	0.326	0.439	0.576
BEnTo-snippet-f1-avg	**0.889**	0.907	0.365	0.467	0.579
BEnTo-ngram-auc-avg	0.848	0.886	0.417	0.510	0.543
BEnTo-ngram-f1-avg	0.850	0.884	0.489	0.562	0.558
BEnTo-all-f1-avg	0.888	0.910	0.478	0.560	0.589
BEnTo-all-auc-avg	0.884	0.909	0.387	0.497	0.577
BEnTo-all-all-avg	**0.890**	**0.913**	0.444	0.532	0.589

Table 2: Results on MIMIC-III, 50 labels. All names of model with ensemble predictions follow {BEnTo}_{model_type}_{criterion}_{ensembling_method}.

However, they propose fancier architectures, sometimes with additional information like chapters of ICD codes and graph structure within ICD codes. We are open to try these fancier architectures in future work.

C-MemNN The **C**ondensed **Mem**ory **N**eural **N**etwork is proposed by Prakash et al. (2017). They utilizes iterative condensed memory representations in the model.

C-LSTM-Att The **C**haracter-aware **LSTM**-based **Att**ention model is proposed by Shi et al. (2017), which utilizes character-aware LSTMs to generate subsection-wise representations of discharge summaries, and apply attention to match the representations and top-50 codes.

CAML & DR-CAML The **C**onvolutional **A**ttention network for **M**ulti-**L**abel classification was proposed by Mullenbach et al. (2018). CAML utilizes a convolutional layer and an attention layer before using label-aware representations to perform multi-label classification (McCallum, 1999). The **D**escription **R**egularized CAML is extended CAML with descriptions of ICD codes to regularize model training.

Logistic Regression, CNN, Bi-GRU We use results provided by Mullenbach et al. (2018).

5 Results

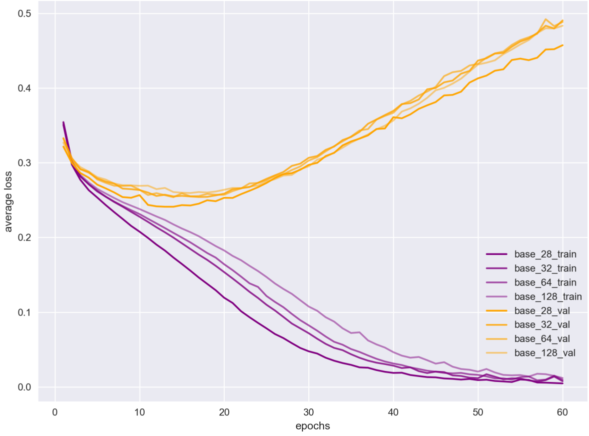

Figure 4: The learning curve of ngram-based model.

In this section, we compared our model with baseline models mentioned above. We first tune Ngram size to find the size that yields the best result. See 4 for the learning curve. We find that ngram-based models with smaller Ngram size always perform better. This is quite intuitive as Ngram embeddings preserves more token-level information. We only consider ngram-based model with Ngram size 28 in ensembling.

We find that Ngram-base models overfit at an early stage. Moreover. we continue to train overfitted models, model predictions decrease in precision and increase in recall, leading to lower AUC scores but higher F1 scores in validation. This happens to both ngram-based and snippet-based model. Therefore, for each model we save one checkpoint with the highest micro AUC score and one with the

Criterion	Ngram-based	Snippet-based	Cross-model-class
Micro AUC	0.1502	0.1214	**0.1953**
Micro F1	0.2128	0.1395	**0.3020**

Table 3: Within-model-class and across-model-class variations scores for best micro AUC and best micro F1 checkpoints respectively.

	AUC		F1		
Model	Macro	Micro	Macro	Micro	P@5
snippet-auc-avg	0.890	0.910	0.326	0.439	0.576
snippet-auc-max	0.881	0.901	0.440	0.528	0.559
snippet-f1-avg	0.874	0.895	0.481	0.554	0.556
snippet-f1-max	0.889	0.907	0.365	0.467	0.579
ngram-auc-avg	0.848	0.886	0.417	0.510	0.543
ngram-auc-max	0.843	0.881	0.472	0.558	0.535
ngram-f1-avg	0.850	0.884	0.489	0.562	0.558
ngram-f1-max	0.846	0.880	0.499	0.540	0.546
all-f1-avg	0.888	0.910	0.478	0.560	0.589
all-f1-max	0.871	0.897	0.509	0.545	0.552
all-auc-avg	0.884	0.909	0.387	0.497	0.577
all-auc-max	0.870	0.896	0.495	0.563	0.551
all-all-avg	0.890	0.913	0.444	0.532	0.589
all-all-max	0.870	0.897	0.502	0.536	0.552

Table 4: Results with ensembling methods (maximal value, average) on MIMIC-III, 50 labels. All names of model with ensemble predictions follow {model_type}_{criterion}_{ensembling_method}.

highest F1 score.

We train each model 5 times with 5 random seeds. We then ensemble model predictions by simply averaging them. Since all ensemble predictions have low recall, we also try element-wise maximum in the later section.

Results of our method is shown in 2. We find that even though ensemble predictions that only consider ngram-based models have lower AUC scores than CAML, most of the ensemble predictions outperform CAML, and are comparable to DR-CAML in AUC scores. Note that DR-CAML utilizes ICD codes descriptions to regularize model training.

We also find that ensembling results from ngram-based models and snippet-based models preserve the highest performances whether by AUC, F1 or precision at top-5 codes.

However, ensemble predictions have lower F1 scores compared with either CNN, CAML, or DR-CAML since predictions have low recalls. Moreover, in predicting top-5 codes our ensemble results also fail to catch up with results of these methods.

6 Model Analysis

6.1 Are Models Really Different?

In the previous section we have shown that ensembling predictions of ngram-based and snippet-based models lead to better performances. We are then interested to see whether predications across models are really different.

We begin with defining variation score within a model class. Given M models within the model class and N samples, the within-model-class variation score is

$$\frac{1}{NM(M-1)}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k\neq m}^{M}||f^m(x^n)-f^k(x^n)||_1 \tag{4}$$

where f^m, f^k are models , and $||f^m(x^n) - f^k(x^n)||_1$ is the L1 distance between $f^m(x^n)$ and $f^k(x^n)$.

We also define the cross-model-class variation score. Consider two model class both with M models and N samples, the cross-model-class variation score is

$$\frac{1}{NM^2}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{M}||f^m(x^n) - g^k(x^n)||_1. \quad (5)$$

The variation scores are shown in 3. We notice that:

- checkpoints of best micro AUC have lower variation scores compared with checkpoints of best F1.
- Snippet-based models have lower within-model-class than ngram-based models.
- Cross-model-class variations scores are higher than within-model-class variation scores.

Therefore, we can say with high confidence that ngram-based models and snippet-based models are generating different predictions. However, whether ngram-based models capture long-term dependencies and snippet-based models capture short-term dependencies remains a question to be answered.

6.2 Ablation Study

As discussed in Section 5, our ensemble predictions have low F1 scores. One possible explanation could be the wrong choice of method to ensemble predictions. To better understand the effect of ensemble methods, we compare results of ensemble predictions generated by averaging and taking maximal values.

Intuitively, ensembling by taking the maximal predictions sacrifices model precision for recall. As shown in 4, ensembling by taking maximal predictions leads to higher F1 scores, and lower AUC scores and precision of top-5 codes. Moreover, the increased F1 scores still fails to catch up CAML, suggesting that the issue of low recall predictions has to do with the model architecture.

7 Discussion and Future Work

Classification with Imbalanced Label Distribution As discussed in Section 5 and 6.2, our method generates ICD codes with high precision but low recall. In other words, our method gives predictions accurate results when a label is present, but it is too "cautious" too tag ICD codes. Moreover, given from precision at 5, for the most frequent ICD codes our method fail to generate precise predictions. It is natural then to point out the problem of imbalanced label distribution. Vu et al. (2020) try to solve this problem by generating label-wise features with a hierarchical classification architecture. Another possible solution is to weight labels in the loss function (Kaku et al., 2019), but one needs to be extremely careful to design the weights.

Interpretability As we only add a classifier head on the contextual [CLS] embeddings, our method does not have interpretability. Previous works tackles interpretability with from various perspectives (Lei et al., 2016; Li et al., 2016; Ribeiro et al., 2016). Mullenbach et al. (2018) simply picks out the phrase with the highest attention weights given. This method can be applicated to our ngram-based model, yet not applicable to the snippet-based. Moreover, as we average predictions for snippet-based models and for ensembling all model predictions, building an explainable classifier becomes a challenge.

Medical Domain-Specific Transformers LMs Medical texts are composed of various types of uncommon tokens compared with pretraining texts for a standard Transformers LM (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020; Raffel et al., 2020). Gururangan et al. (2020) point out that continuing pretraining on domain-specific text does help a downstream task. We have tried Clinical BERT(Alsentzer et al., 2019) for the ngram-based model but finds similar performances compared with `bert-base-uncased`. We think it worths a try for Clinical BERT in snippet-based model, as ngram-based models lose token-level information.

8 Conclusion

We present BEnTo, a method based on simple architectures but can utilize Transformers LMs which only accpets shorter sequences. We ensemble model predictions of 20 checkpoints belonging to either an ngram-based model class or a snippet-based model class. We perform experiments on MIMIC-III with top-50 codes, and find that predictions are often with high AUC scores but low F1 scores. Moreover, we show that predictions by either model class is different from the other. Though our method are still below state-of-the-art performances, it be not only a strong baseline for utilizing Transformers LMs for ICD coding, but also a new method for long-text classification.

References

Emily Alsentzer, J. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A.

McDermott. 2019. Publicly available clinical bert embeddings. *ArXiv*, abs/1904.03323.

A. Avati, Kenneth Jung, S. Harman, L. Downing, A. Ng, and N. Shah. 2018. Improving palliative care with deep learning. *BMC Medical Informatics and Decision Making*, 18.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv*, abs/2004.05150.

Pengfei Cao, Yubo Chen, K. Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020. Hypercore: Hyperbolic and co-graph representation for automatic icd coding. In *ACL*.

Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. 2016. Doctor ai: Predicting clinical events via recurrent neural networks. In *Proceedings of the 1st Machine Learning for Healthcare Conference*, volume 56 of *Proceedings of Machine Learning Research*, pages 301–318, Northeastern University, Boston, MA, USA. PMLR.

J. Denny, M. Ritchie, M. Basford, J. Pulley, L. Bastarache, K. Brown-Gentry, Deede Wang, D. Masys, D. Roden, and D. Crawford. 2010. Phewas: demonstrating the feasibility of a phenome-wide scan to discover gene–disease associations. *Bioinformatics*, 26:1205 – 1210.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Brämer Gr. 1988. International statistical classification of diseases and related health problems. tenth revision.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *ACL*.

Kexin Huang, Sankeerth S. Garapati, and A. Rich. 2020. An interpretable end-to-end fine-tuning approach for long clinical text. *ArXiv*, abs/2011.06504.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, L. Lehman, M. Feng, M. Ghassemi, Benjamin Moody, Peter Szolovits, L. Celi, and R. Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3.

Aakash Kaku, Chaitra V. Hegde, Jeffrey Huang, S. Chung, X. Wang, M. Young, Alireza Radmanesh, Y. Lui, and N. Razavian. 2019. Darts: Denseunet-based automatic rapid tool for brain segmentation. *ArXiv*, abs/1911.05567.

Ramakanth Kavuluru, Anthony Rios, and Y. Lu. 2015. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artificial intelligence in medicine*, 65 2:155–66.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942.

Tao Lei, R. Barzilay, and T. Jaakkola. 2016. Rationalizing neural predictions. In *EMNLP*.

F. Li and H. Yu. 2020. Icd coding from clinical text using multi-filter residual convolutional neural network. *ArXiv*, abs/1912.00862.

J. Li, Xinlei Chen, E. Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in nlp. *ArXiv*, abs/1506.01066.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Andrew Kachites McCallum. 1999. Multi-label text classification with a mixture model trained by em. In *AAAI 99 Workshop on Text Learning*.

J. Mullenbach, Sarah Wiegreffe, J. Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. *ArXiv*, abs/1802.05695.

Andriy Mulyar, Elliot Schumacher, M. Rouhizadeh, and Mark Dredze. 2019. Phenotyping of clinical notes with improved document classification models using contextualized neural language models. *ArXiv*, abs/1910.13664.

Raghavendra Pappagari, Piotr Żelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. 2019. Hierarchical transformers for long document classification. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844.

A. Perotte, R. Pivovarov, K. Natarajan, N. Weiskopf, F. Wood, and Noémie Elhadad. 2014. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association : JAMIA*, 21:231 – 237.

Aaditya Prakash, Siyuan Zhao, Sadid A. Hasan, V. Datla, Kathy Lee, Ashequl Qadir, J. Liu, and Oladimeji Farri. 2017. Condensed memory networks for clinical diagnostic inferencing. *ArXiv*, abs/1612.01848.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, M. Matena, Yanqi Zhou, W. Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

R. Ranganath, A. Perotte, Noémie Elhadad, and D. Blei. 2015. The survival filter: Joint survival analysis with a latent time series. In *UAI*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Haoran Shi, Pengtao Xie, Zhiting Hu, M. Zhang, and E. Xing. 2017. Towards automated icd coding using deep learning. *ArXiv*, abs/1711.04075.

Thanh Vu, Dat Quoc Nguyen, and A. Nguyen. 2020. A label attention model for icd coding from clinical text. In *IJCAI*.

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, Melbourne, Australia. Association for Computational Linguistics.