BERT for Long Documents: A Case Study of Automated ICD Coding

Arash Afkanpour **Shabir Adeel*** **Hansenclever Bassani***

Arkady Epshteyn* **Hongbo Fan*** **Isaac Jones*** **Mahan Malihi***

Adrian Nauth* **Raj Sinha*** **Sanjana Woonna*** **Shiva Zamani***

Elli Kanal† **Mikhail Fomitchev†** **Donny Cheung†**

Google

arashaf@google.com

Abstract

Transformer models have achieved great success across many NLP problems. However, previous studies in automated ICD coding concluded that these models fail to outperform some of the earlier solutions such as CNN-based models. In this paper we challenge this conclusion. We present a simple and scalable method to process long text with the existing transformer models such as BERT. We show that this method significantly improves the previous results reported for transformer models in ICD coding, and is able to outperform one of the prominent CNN-based methods.

1 Introduction

The International Classification of Diseases (ICD) codes provide a standard way of keeping track of diagnoses and procedures during a patient visit. These codes are used worldwide for epidemiological studies, billing and reimbursement, and research in health care. The codes are maintained by the World Health Organization (WHO) and are revised and updated periodically. As of 2022 the ICD codes are in the 11th revision.

Assigning ICD codes to a clinical note, such as a discharge summary, is done by professional medical coders. Human coders require extensive training, and the process of coding is often time-consuming, costly, and error-prone. Due to these challenges there is an incentive to automate the coding process. Therefore in recent years this problem has gained interest among machine learning researchers in health care (See, Mullenbach et al. (2018); Li and Yu (2020); Zhang et al. (2020) and references therein). On the surface, the problem can be considered as a multi-label document classification problem. However, there are aspects of the problem that make it particularly challenging. The primary challenge is that there are tens of thousands of classes. For instance, billable ICD-10-CM codes consist of approximately 73,000 codes. In addition, the distribution of the codes is not uniform. Many of the codes are related to rare conditions and are mentioned infrequently in text, which makes it difficult to train a reliable classifier for them.

Transformer-based language models developed based on self attention (Vaswani et al., 2017) have become the state-of-the-art across many NLP problems by outperforming previous solutions that were mostly based on recurrent neural networks (RNN) and convolutional neural networks (CNN). So one would expect that they perform well in ICD coding too. However, examining the literature of ICD coding methods reveals that transformer-based solutions fail to outperform CNN-based models. Many studies have applied the BERT language model (Devlin et al., 2018) to this task, for example Pascual et al. (2021); Singh et al. (2020); Biseda et al. (2020); Amin et al. (2019). More recently, Ji et al. (2021) performed a comprehensive quantitative study to compare BERT and some of its variants pre-trained on medical text against CNN-based models such as Mullenbach et al. (2018) and Cao et al. (2020) to answer the question of whether the magic of BERT (as observed across many NLP problems) also applies to automated ICD coding. They concluded that BERT cannot outperform CNN-based models in the full ICD code case.

Unlike RNN or CNN models, which in theory can process sequences of arbitrary length, transformers' computational complexity scales quadratically with sequence length. This means that most of these models can handle limited size sequences. For instance, BERT models usually are pre-trained and fine-tuned on sequences with at most 512 tokens. Clinical notes normally contain long snippets of text beyond the sequence limit of transformers. We hypothesize that this constraint could explain

* Equal contribution
† Technical leadership

Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI), pages 100 - 107

the poor performance of transformers in this task, and will present empirical evidence for that.

We emphasize that we do not claim to achieve state-of-the-art performance in ICD coding, or that our design is the most efficient transformer architecture for processing long text. For a review of efficient transformers see Tay et al. (2020) and references therein. Our goal is to provide new empirical evidence that shows even the standard transformer models can outperform some of the previous prominent methods and are a viable solution for ICD coding.

2 Related work

Medori and Fairon (2010) applied a rule-based method to extract important snippets of text and encode them with ICD codes. Perotte et al. (2014) proposed SVM classification with bag-of-words features. They experimented with both flat SVM (i.e. one classifier per code) and a hierarchical classifier.

With the success of deep learning in NLP tasks, many researchers focused on using RNN and CNN models for ICD coding. CNN models provide a convenient way to learn a contextual representation of text in NLP problems (Chen, 2015). For example, Mullenbach et al. (2018) proposed the *CAML* model: a convolutional layer on word2vec embedding vectors to learn a contextual representation for each word. The word representations are combined into a class-specific document representation using the attention mechanism. They also suggested a method to leverage code descriptions via a regularization term. Li and Yu (2020) proposed Multi-filter Residual CNN (MultiResCNN) that uses convolutional layers with different kernel sizes to capture patterns with different lengths. Additionally, they used residual blocks on top of the convolutional layer. Similar to Mullenbach et al. (2018) they employed a per-class attention mechanism to make the document representation attend to different parts of the input for each code.

Recurrent neural networks (RNN) are also studied extensively for ICD coding. Shi et al. (2017) applied LSTM at character and word level to encode both the clinical note and the code description. Baumel et al. (2018) employs a two-layer bidirectional Gated Recurrent Unit (GRU) model, where the first layer encodes individual sentences, and the second layer encodes the document.

With the success of transformer architectures

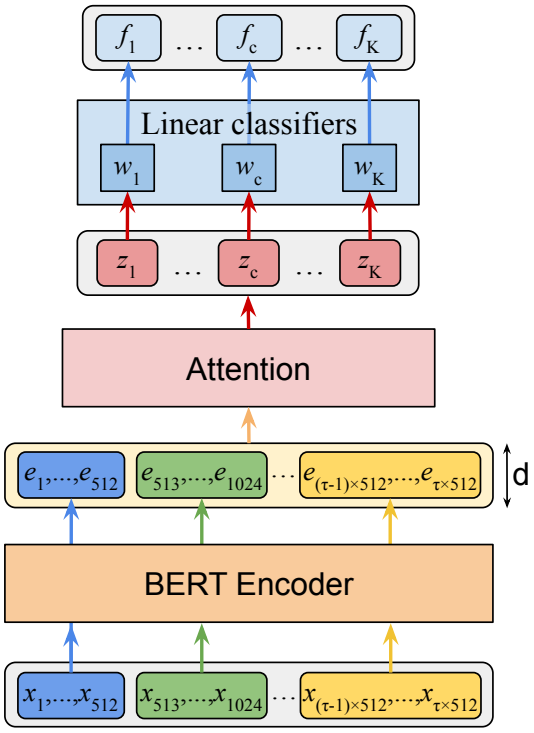

Figure 1: Model architecture proposed for handling long text inputs.

across many NLP tasks, researchers focused their attention to designing such models for ICD coding. BERT-XML (Zhang et al., 2020) with access to a large corpus of private data managed to pre-train the model with sequence length of 1024. Most of the work in this area, however, considered the standard BERT model and its variants pre-trained on medical text to encode the document (Pascual et al., 2021; Singh et al., 2020; Biseda et al., 2020; Amin et al., 2019). One observation with these models was that they were unable to outperform CNN-based models. Ji et al. (2021) performed a comprehensive study to answer a few research questions on the suitability of BERT models for ICD coding. They studied and compared different variants of BERT pre-training. They also proposed a hierarchical attention method so that long clinical notes can be processed with a BERT model with a limit of 512 tokens. Most importantly, they compared different BERT variants against traditional CNN-based models, and through extensive experiments showed that BERT-based models are not capable of outperforming CNN-based models in ICD coding. In the next sections we show that a simple method that enables processing of long text with transformers will attain results that contradict the findings of Ji et al. (2021).

3 Method

In this section we explain our method for building a model to predict medical codes. As illustrated in Figure 1, our model consists of an encoder that calculates token-level representation of the input text. This can be done in various ways, e.g. Mullenbach et al. (2018) used word2vec and a CNN layer to calculate word-level representations. We choose the BERT language model for this purpose. A class-specific representation of the document is then calculated using class-specific attention vectors, similar to Mullenbach et al. (2018). For d-dimensional token representations and K classes, this layer requires $d \times K$ parameters. Linear binary classifiers are built on top of the document representation to produce the probability that the document belongs to any of the K classes. This layer requires $(d+1) \times K$ parameters (one scalar for the offset).

Let $X = [x_1, \ldots, x_s]$ denote the tokenized input sequence with s tokens. Let $e_1(X), \ldots, e_s(X)$ denote the representation of tokens $1, \ldots, s$ obtained from an encoder. That is,

$$e_i(X) = \phi(x_i|X), \qquad i \in \{1, \ldots, s\},$$

where ϕ is an encoder, such as BERT, that returns a context-dependent representation for each token. For each class c, token-level representations are combined into a single vector that represents the entire document using the attention mechanism:

$$z_c(X) = \sum_{i=1}^{s} \alpha_{c,i}(X) e_i(X),$$

where

$$\alpha_{c,i}(X) = \frac{\exp\left(\langle e_i(X), q_c \rangle\right)}{\sum_{j=1}^{s} \exp\left(\langle e_j(X), q_c \rangle\right)}, \quad (1)$$
$$i \in \{1, \ldots, s\},$$

are the normalized attention coefficients and $\langle \cdot, \cdot \rangle$ denotes inner product, and q_c is the d-dimensional attention vector for class c. The predicted probability of the model for class c is calculated by

$$f_c(X) = \sigma\left(\langle z_c(X), w_c \rangle + b_c\right),$$

where w_c is the weight vector for class c, b_c is the scalar offset for class c, and σ is the sigmoid function.

3.1 Handling long text

Language models such as BERT can handle input text up to a certain length. For example, BERT can take input of at most 512 tokens. While it is possible to pre-train the model on longer sequences (mostly to learn useful positional embedding vectors), memory requirement grows quadratically with input size. So pre-training a BERT model on longer text is not scalable.

There are transformer-based models that can handle long sequences, such as BigBird (Zaheer et al., 2020), ETC (Ainslie et al., 2020), Longformer (Beltagy et al., 2020), and LongT5 (Guo et al., 2021). There are a few factors that limit their usability in the medical coding task. For example, these models are usually designed to train on TPU, so training on GPU is often a slow process, if feasible, especially for longer sequences. Also, pre-trained checkpoints of these models are limited, unlike the BERT models that have many pre-trained variants including those pre-trained on medical text.

In this paper, we propose a simple idea, which enables us to use a vanilla BERT model on long sequences. Inspired by the local attention feature of CNN models, we propose to split the input text into (optionally overlapping) segments of 512 tokens. These segments are passed sequentially to a BERT model, and the token representations are concatenated to form $[e_1(X), ..., e_{512}(X), e_{513}(X), ..., e_{1024}(X), ...]$. One may argue that a limitation of this approach is that the token representations are calculated with a 512-token attention span. However, we have observed that in practice this method performs well. In fact, we conjecture that in many cases short snippets of text (as evidence) are sufficient for assigning the correct ICD codes to the input document. Algorithm 1 shows the training procedure.

4 Evaluation

We evaluate the accuracy of the proposed method with several sequence lengths and compare it against the CAML method (Mullenbach et al., 2018), which is one of the prominent CNN-based methods for ICD coding.

4.1 Data sets

For this task we chose the publicly-available MIMIC-III (Johnson et al., 2016) and MIMIC-IV

Algorithm 1 Training on a single example.

1: **Input:** tokenized input text of length s: $X = [x_1, \ldots, x_s]$, sparse binary label vector $Y = [y_1, \ldots, y_K]$ for K classes, where $y_c = 1$ if the example belongs to class c, and 0 otherwise.
2: Pad input text $X = [x_1, \ldots, x_s]$ to length $\tau \times 512$ to obtain $X' = [x_1, \ldots, x_s, \ldots, x_{\tau \times 512}]$, where $\tau = \lceil s/512 \rceil$.
3: Split X' into segments of 512 tokens: $S_1 = [x_1, \ldots, x_{512}], S_2 = [x_{513}, \ldots, x_{1024}], \ldots, S_\tau$.
4: Pass S_i's, $i \in \{1, \ldots, \tau\}$ sequentially to the BERT module and obtain the corresponding token representations.
5: Concatenate token representations from all sequences to obtain $[e_1, \ldots, e_s, \ldots, e_{\tau \times 512}]$.
6: Calculate class-specific document representations by $z_c(X) = \sum_{i=1}^{s} \alpha_{c,i}(X) e_i(X)$, with $\alpha_{c,i}$ from Eq. 2.
7: Calculate model predictions for all classes: $f_c(X) = \sigma\left(\langle z_c(X), w_c \rangle + b_c\right),\ c \in 1, \ldots, K$.
8: Calculate and apply gradient updates for loss function $\sum_{c=1}^{K} \ell(y_c, f_c(X))$, where ℓ is binary cross-entropy.

(Johnson et al., 2020) data sets. MIMIC-III is a large de-identified data set of over 40,000 patients admitted to intensive care units at the Beth Israel Deaconess Medical Center. The data set contains structured and unstructured data, including lab measurements, vital signs, medications, clinical notes, etc. Following previous studies, we focus on predicting ICD codes for discharge summaries where each note corresponds to a hospital stay event. MIMIC-IV is an update to MIMIC-III, which incorporates contemporary data. It is sourced from two in-hospital database systems: a custom hospital wide EHR and an ICU specific clinical information system.

Each discharge summary in MIMIC-III is manually coded by human coders with one or more ICD-9 codes that specify diagnoses and procedures of that particular stay. The data set contains 8,921 unique ICD-9 codes, including 6,918 diagnosis and 2,003 procedure codes. There are patients with multiple admissions and therefore multiple discharge summaries. To be consistent with the previous studies and to ensure that all of the notes of a patient are assigned to one of train/validation/test sets we use the data split provided by Mullenbach et al. (2018). This results in 47,724 discharge summaries for training, 1,632 summaries and 3,372 summaries for validation and test sets respectively.

The discharge summaries in MIMIC-IV are additionally labeled with ICD-10 codes. At the time of writing this paper the MIMIC-Note module, which contains the discharge summaries, is not yet publicly available. In our experiments we only consider the ICD-10 diagnosis set, which contains 72,748 codes in the data set.

For tokenizing text we used the standard BERT vocabulary and tokenizer (Devlin et al., 2018). Figure 2 shows the cumulative distribution function of the number of tokens per note for MIMIC-III and MIMIC-IV.

4.2 Models

Our classification model uses a BERT language model with the method described in Section 3. We dub this model *LongBERT* below. The BERT checkpoint we use in the experiments is a model with 2 transformer blocks and 256-dimensional embedding vectors. The checkpoint can be downloaded from TensorFlow Hub.[1]

The baseline model (*BERT-baseline*) was trained and evaluated on the first 512 tokens of input text. To measure the impact of sequence length we trained and evaluated similar models on the first s tokens of each note, with $s \in \{1024, 2048, 4096, 8192\}$. All BERT parameters and the additional attention and classification parameters were fine-tuned during training. We used the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 2e-4. The batch size was set to 4 in all experiments, except for the models trained with the sequence length of 8192 which were trained with the batch size of 2 to avoid running out of memory. The models were trained for 1 million steps (each step is one batch). No hyper-parameter tuning was performed except for the number of training steps. The best model corresponds to the training step that achieves the highest validation micro F1 score.

[1] https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-2_H-256_A-4/2

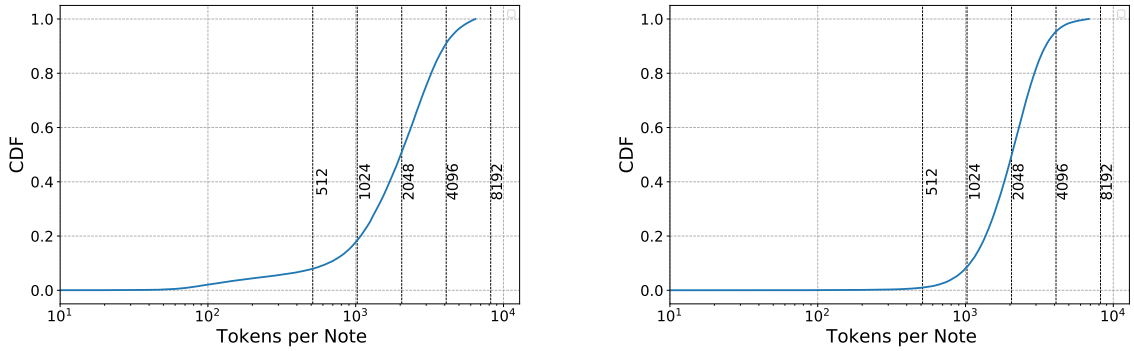

Figure 2: Cumulative distribution function (CDF) of the number of tokens per note for MIMIC-III (left) and MIMIC-IV (right) data sets.

We compare these models against a CAML model trained on sequences of 2500 words following Mullenbach et al. (2018). The hyperparameters were set according to the optimal values obtained in Mullenbach et al. (2018). Training was performed for 1 million steps, and the best model was selected according to validation micro F1 score.

Following previous work, in the MIMIC-III experiments, training and evaluation was performed on the full ICD-9 label set as well as the 50 most frequent codes. In the MIMIC-IV experiment, we consider only the ICD-10 diagnosis codes. Each ICD code has its own attention and classification weight vectors in the models. Table 1 breaks down the number of parameters of the models in the experiments.

4.3 Evaluation metrics

Our primary evaluation metric is micro-averaged F1 (micro F1 for short). Micro-averaged values are calculated by treating each code as a (binary) label for each note. That is, each (note, code) pair is counted as one instance for calculating the metrics. Let,

$$\text{micro precision} = \frac{\sum_{x,c} TP(x,c)}{\sum_{x,c} TP(x,c) + FP(x,c)},$$

$$\text{micro recall} = \frac{\sum_{x,c} TP(x,c)}{\sum_{x,c} TP(x,c) + FN(x,c)},$$

where $TP(x,c) = 1$ if class c is a true positive prediction for note x and 0 otherwise. $FP(x,c)$ (false positive) and $FN(x,c)$ (false negative) are defined analogously. Finally, micro F1 is the harmonic mean of micro precision and micro recall:

$$\text{micro F1} = 2\,\frac{\text{micro precision} \times \text{micro recall}}{\text{micro precision} + \text{micro recall}}.$$

The optimal threshold on model predictions, which is used to calculate $TP/FP/FN$ counts, is obtained by a grid search to maximize the validation set F1 score.

Additionally we report precision-recall AUC (PR-AUC), and ROC-AUC. In contrast to F1 score, these metrics are independent of a specific operating point and provide an aggregated view of model accuracy.

4.4 Results

Table 2 shows the results of the LongBERT and CAML models on the MIMIC-III full-code test set. Table 3 shows accuracy metrics obtained on the MIMIC-IV diagnosis code data set. Bold numbers represent the best value of each metric. A clear trend observed in both data sets is that as the sequence length of LongBERT increases, the accuracy of the model improves. These results demonstrate that the capability to process long text is critical in achieving high accuracy.

The LongBERT models with sequence lengths of 4096 and 8192 both outperform the CAML model. This finding contradicts the previous finding of Ji et al. (2021). While their hierarchical attention proposal and our method both handle long text by breaking it into segments of 512 tokens, one key difference is that they use the CLS token representation from each segment, whereas we use individual token representations. The best MIMIC-III full-code performance reported in Ji et al. (2021) was F1 = 0.47 with BioBERT full-text (Lee et al., 2020)

	MIMIC-III full	MIMIC-III top 50	MIMIC-IV diagnosis
Language model	9,591,040	9,591,040	9,591,040
Attention layer	2,283,776	12,800	18,623,488
Classification layer	2,292,697	12,850	18,696,236
Total	14,167,513	9,616,690	46,910,764

Table 1: Breakdown of the number of parameters of BERT-baseline and LongBERT with 2 transformer blocks and 256-dimensional embedding vectors. MIMIC-III full contains 8,921 classes, and MIMIC-IV diagnosis contains 72,748 classes.

checkpoint and hierarchical attention, while our small vanilla BERT model with sequence length of 8192 achieves F1 = 0.5680. These results show that with a proper modeling approach transformer-based models are indeed capable of outperforming CNN-based models in ICD coding.

MIMIC-III top 50. Following previous work, we also trained and evaluated the models on the MIMIC-III 50 most frequent codes. Table 4 shows the results. Similar to the full-code case we observe that processing longer segments results in higher accuracy.

In this case, however, there is no clear winner between LongBERT and CAML. While LongBERT achieves a higher micro F1 score, the CAML model has a higher PR-AUC. We conjecture that the smaller performance difference between the two models in this experiment compared to the full-code experiment is due to the amount of information in the data sets. By removing many of the labels in the top-50 experiment we essentially remove information. This information is more helpful to larger models (i.e. transformers) than smaller models, such as CAML. As a result, we observe a larger performance gap in the full-code experiment between LongBERT and CAML.

We also note that the accuracy numbers of the CAML model in this experiment are higher than those reported in Mullenbach et al. (2018). One difference here is that we do not discard notes that aren't assigned any of the top 50 codes as was done in the original paper. Such notes are used as negative examples for the top 50 codes. Therefore our data set contains more negative examples than the data set used in Mullenbach et al. (2018).

5 Discussion

Most of the existing BERT models pre-trained on generic or medical text can take input segments of up to 512 tokens. Clinical notes, however, are much longer than this limit. To deal with this limitation, much of the existing works in automated ICD coding that use BERT limit the input to the model by truncating the text or selecting specific spans of text. This results in loss of information and poor performance.

In this paper we proposed a simple method to apply BERT models to sequences longer than 512 tokens. Our method is simple and consists of two key components: (i) apply BERT sequentially to (optionally overlapping) segments of 512 tokens, and (ii) concatenate token-level representations from all segments, and combine them using a class-specific attention layer.

We demonstrated that processing long text sequences minimizes information loss and is critical for achieving high performance in automated ICD coding. We also showed that contrary to previous findings, this method with even a small vanilla BERT model outperforms CNN-based methods, and achieves competitive performance.

Future steps include evaluating medical variants of BERT, and exploring other transformer-based architectures that were designed to handle long sequences.

Limitations

While our method enables the processing of text longer than 512 tokens, one of the limitations of this approach is that context-dependent token representations are still calculated using a window of 512 tokens. Despite good performance of this method in practice, there could be cases where a context window of longer than 512 must be used to make accurate predictions.

Furthermore, while our method reduces computational complexity from quadratic (in sequence length) to linear, the memory requirement of the model could still be prohibitive in certain cases. For instance, for sequence length of 8192, and a

	Seq. length	Micro F1	Precision	Recall	PR-AUC	ROC-AUC
CAML	2500 (words)	0.5465	0.5973	0.5036	0.5361	**0.9831**
BioBERT full-text (Ji et al., 2021)	entire note	0.470	N/A	N/A	N/A	0.974
BERT-baseline	512	0.4149	0.4769	0.3672	0.3793	0.9745
LongBERT	1024	0.4697	0.5421	0.4144	0.4309	0.9766
LongBERT	2048	0.5036	0.5777	0.4463	0.4703	0.9794
LongBERT	4096	0.5514	0.6038	0.5074	0.5305	0.9820
LongBERT	8192	**0.5680**	**0.6148**	**0.5278**	**0.5402**	0.9827

Table 2: Accuracy metrics in the MIMIC-III full-code experiment.

	Seq. length	Micro F1	Precision	Recall	PR-AUC	ROC-AUC
CAML	2500 (words)	0.5439	0.5739	0.5169	0.5313	**0.9889**
BERT-baseline	512	0.4010	0.4298	0.3757	0.3580	0.9883
LongBERT	1024	0.4607	0.5094	0.4205	0.4254	0.9839
LongBERT	2048	0.4852	0.5268	0.4497	0.4559	0.9852
LongBERT	4096	0.5635	0.5925	0.5371	0.5450	0.9850
LongBERT	8192	**0.5703**	**0.6046**	**0.5397**	**0.5517**	0.9871

Table 3: Accuracy metrics in the MIMIC-IV diagnosis experiment.

	Seq. length	Micro F1	Precision	Recall	PR-AUC	ROC-AUC
CAML	2500 (words)	0.6390	**0.6506**	0.6278	**0.6410**	0.9102
BERT-baseline	512	0.5027	0.5367	0.4727	0.5117	0.8360
LongBERT	1024	0.5568	0.5923	0.5252	0.5406	0.8560
LongBERT	2048	0.5908	0.5987	0.5832	0.5604	0.8834
LongBERT	4096	0.6375	0.6157	0.6609	0.6229	0.9115
LongBERT	8192	**0.6522**	0.6417	**0.6629**	0.6303	**0.9181**

Table 4: Accuracy metrics in the MIMIC-III top-50 experiment.

small BERT checkpoint with only two transformer blocks we had to reduce batch size to 2 in order to train the models. Using a larger BERT checkpoint for long sequences requires more memory and multiple GPUs, which increases the cost of compute.

References

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. Etc: Encoding long and structured inputs in transformers. *arXiv preprint arXiv:2004.08483*.

Saadullah Amin, Günter Neumann, Katherine Dunfield, Anna Vechkaeva, Kathryn Annette Chapman, and Morgan Kelly Wixted. 2019. Mlt-dfki at clef ehealth 2019: Multi-label classification of icd-10 codes with bert. In *CLEF (Working Notes)*, pages 1–15.

Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noémie Elhadad. 2018. Multi-label classification of patient notes: case study on icd code assignment. In *Workshops at the thirty-second AAAI conference on artificial intelligence*.

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

Brent Biseda, Gaurav Desai, Haifeng Lin, and Anish Philip. 2020. Prediction of icd codes with clinical bert embeddings and text augmentation with

label balancing using mimic-iii. *arXiv preprint arXiv:2008.10492*.

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.

Yahui Chen. 2015. Convolutional neural network for sentence classification. Master's thesis, University of Waterloo.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2021. Longt5: Efficient text-to-text transformer for long sequences. *arXiv preprint arXiv:2112.07916*.

Shaoxiong Ji, Matti Hölttä, and Pekka Marttinen. 2021. Does the magic of bert apply to medical code assignment? a quantitative study. *Computers in Biology and Medicine*, 139:104998.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. 2020. Mimic-iv. *version 0.4). PhysioNet. https://doi.org/10.13026/a3wn-hq05*.

Alistair EW 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):1–9.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Fei Li and Hong Yu. 2020. Icd coding from clinical text using multi-filter residual convolutional neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34(05), pages 8180–8187.

Julia Medori and Cédrick Fairon. 2010. Machine learning and features selection for semi-automatic icd-9-cm encoding. In *Proceedings of the NAACL HLT 2010 Second Louhi Workshop on Text and Data Mining of Health Documents*, pages 84–89.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. *arXiv preprint arXiv:1802.05695*.

Damian Pascual, Sandro Luck, and Roger Wattenhofer. 2021. Towards bert-based automatic icd coding: Limitations and opportunities. *arXiv preprint arXiv:2104.06709*.

Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. 2014. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association*, 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*.

AK Singh, Mounika Guntu, Ananth Reddy Bhimireddy, Judy W Gichoya, and Saptarshi Purkayastha. 2020. Multi-label natural language processing to identify diagnosis and procedure codes from mimic-iii inpatient notes. *arXiv preprint arXiv:2003.07507*.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient transformers: A survey. *ACM Computing Surveys (CSUR)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297.

Zachariah Zhang, Jingshu Liu, and Narges Razavian. 2020. Bert-xml: Large scale automated icd coding using bert pretraining. *arXiv preprint arXiv:2006.03685*.