Journal of Universal Computer Science, vol. 29, no. 4 (2023), 349-373
submitted: 4/7/2022, accepted: 10/1/2023, appeared: 28/4/2023 CC BY-ND 4.0

Automatic assignment of diagnosis codes to free-form text medical note

Stefan Strydom
(Stellenbosch University, South Africa
https://orcid.org/0000-0003-4888-4397, stefan.strydom87@gmail.com)

Andrei Michael Dreyer
(Stellenbosch University, South Africa
https://orcid.org/0000-0001-5597-5153, andrei.dreyer1997@gmail.com)

Brink van der Merwe
(Stellenbosch University, South Africa
https://orcid.org/0000-0001-5010-9934, abvdm@cs.sun.ac.za)

Abstract: International Classification of Disease (ICD) coding plays a significant role in classifying morbidity and mortality rates. Currently, ICD codes are assigned to a patient's medical record by hand by medical practitioners or specialist clinical coders. This practice is prone to errors, and training skilled clinical coders requires time and human resources. Automatic prediction of ICD codes can help alleviate this burden. In this paper, we propose a transformer-based architecture with label-wise attention for predicting ICD codes on a medical dataset. The transformer model is first pre-trained from scratch on a medical dataset. Once this is done, the pre-trained model is used to generate representations of the tokens in the clinical documents, which are fed into the label-wise attention layer. Finally, the outputs from the label-wise attention layer are fed into a feed-forward neural network to predict appropriate ICD codes for the input document. We evaluate our model using hospital discharge summaries and their corresponding ICD-9 codes from the MIMIC-III dataset. Our experimental results show that our transformer model outperforms all previous models in terms of micro-F1 for the full label set from the MIMIC-III dataset. This is also the first successful application of a pre-trained transformer architecture to the auto-coding problem on the full MIMIC-III dataset.

Keywords: Prediction of ICD codes; transformer architecture
Categories: I.2.6, I.2.7, I.5.1
DOI: 10.3897/jucs.89923

1 Introduction[1]

The past two decades have seen an explosion in the uptake of electronic health records (EHRs). The United States Office of the National Coordinator for Health Information Technology (ONC) defines an EHR as a "*real-time, patient-centered record that makes information available instantly and securely to authorized users*". EHRs store information on all patient interactions with the health system, and include information such as patient demographics, medical history, family medical history, diagnoses, treatments, test results and medications prescribed. Much of this information is captured as free-form text. The

[1] Parts of this research is also discussed in the master's thesis of the first author [Strydom, 2021].

use of EHRs is expected to improve the efficiency of the healthcare systems by providing relevant data in a timely manner and to reduce waste by preventing duplication of tests already performed.

Recent studies, however, have identified numerous pitfalls and unintended consequences of the large-scale uptake of EHRs. Polling of US physicians shows that they have increased, rather than reduced, the burden on doctors. An online survey commissioned by Stanford Medicine and conducted by Harris Poll in March 2018 showed that, on average, physicians spend up to 60% of their time with patients completing their EHRs. While poll participants may have overestimated the time spent on EHRs, [Menachemi and Collum, 2011] found that at least an estimated 20% of productivity was lost initially when adopting the EHR system. Furthermore, studies have found an association between physician burnout and use of EHRs [Tajirian et al., 2020, Melnick et al., 2020]. In addition to design flaws which have increased the data capturing burden on physicians, another major drawback of EHRs has been the lack of standardisation between records [Ajami and Arab-Chadegani, 2013]. This means that patient records are not easily transferred between providers using different EHR systems.

Despite these practical issues, the large-scale uptake of EHRs has resulted in the capture of vast amounts of digital personal health information. Initially, the lack of standardisation of data being recorded adversely impacted the development of machine learning models based on EHRs. The "traditional" approach to predictive modelling in healthcare relied heavily on linear models such as least squares regression, logistic regression and support vector machines trained on hand-crafted feature sets. Hand-crafted features would normally take snapshots of the data (for example, a typical feature might be the "number of primary care visits during the past 12 months") and thus ignore much of the detailed information contained in a longitudinal patient record. In addition to the loss of information, constructing hand-crafted feature sets require a significant time investment and specialist clinical knowledge. End-to-end deep learning models have the potential to change this and extract significant value from EHRs.

Machine learning models trained on EHR data have been shown to outperform traditional risk classification methods for numerous tasks. While machine learning models have the ability (at least theoretically) to consider information from the entire EHR, traditional approaches rely on clinical algorithms calculated from a handful of patient and clinical characteristics that are readily available to clinicians at the point of care. One widely used traditional algorithm is the Framingham Risk Score which calculates the 10-year cardiovascular risk of individuals based on age, sex, smoking status, cholesterol and systolic blood pressure (captured at a single point in time). Recently, a number of studies have shown that models learned from longitudinal patient data, recorded in EHRs, outperform by significant margins Framingham and other traditional approaches such as SCORE and PROCAM when predicting cardiovascular events [Korsakov et al., 2019]. Machine learning models trained on EHRs have also been successfully applied to predict which patients are most likely to develop complications during or after hospital stays [Li et al., 2019, Wong et al., 2018a], to perform case-finding [Tedeschi et al., 2020], to predict patient outcomes [Wong et al., 2018b], and to predict patient mortality risk [Slattery et al., 2019].

1.1 Clinical coding

It is common practice in health systems to use standard coding systems to classify healthcare episodes. Accurate coding of healthcare episodes is used to (inter alia):

- Measure and understand disease burden for health system planning and resource allocation;
- Compare health outcomes between different geographies and different hospitals after allowing for differences in morbidity (“case-mix adjustment”);
- Reimburse healthcare practitioners according to the services provided or the complexity of patients treated.

Numerous types of coding systems exist, including systems to describe diagnoses, systems to describe procedures and laboratory tests, and systems to classify devices and drugs. The current standard practice is for trained clinical coders to assign codes from the health information recorded by the healthcare practitioner on the patient’s health record. This approach is costly and highly subjective. In the United States, the annual cost of coding errors and financial investment on training human coders, was already estimated at $25 billion more than a decade ago [Farkas and Szarvas, 2008].

Given the cost, clinical coding is often absent in lower-income countries. The knock-on effect of this is an inability to leverage the uses of clinical coding listed above, including the critical task of measuring and understanding the impact of disease burden on health system planning. When available, routinely collected diagnosis codes can be used to describe morbidity patterns and clinical needs of populations in a cost-effective and timely manner [Mannie and Kharrazi, 2020].

The current state of clinical coding practices presents an opportunity for machine learning to improve the accuracy and consistency of clinical coding, while reducing costs and freeing up clinically trained individuals working as coders to do clinical work. This work focuses on assigning International Classification of Diseases (ICD) codes to free-form text medical reports and does a systematic review of related research to determine which techniques have the biggest impact on classification accuracy.

1.2 Problem description

In practice, each healthcare episode is assigned one or more clinical codes to describe it in a standardised fashion. Codes selected from code sets (each consisting of tens or thousands of codes) are used to describe the diagnoses, therapies, procedures, pathology tests and medicines prescribed.

This research specifically focuses on the assignment of diagnosis codes from free-form text in hospital discharge summaries. The most widely used diagnoses coding set is the ICD, maintained by the World Health Organisation (WHO) [World Health Organization, 2004]. According to them, approximately 70% of the world’s total healthcare expenditures are allocated using the ICD system, either through reimbursement or budget allocation. In addition, the ICD system is used to report mortality data in over 100 countries.

The latest version of ICD is version 11, released by WHO in June 2018, adopted by the World Health Assembly in May 2019 and officially came into effect on 1 January 2022. The WHO has acknowledged that adoption by most countries will only occur at later dates. Version 10, which was adopted by the World Health Assembly in 1990 and started being used by member states in 1994, was only officially adopted by the United States in 2015.

The ICD system is a hierarchical structure. In version 10, all codes are divided into 21 chapters at the highest level of the hierarchy. The 21 chapters, broadly speaking, refer to body systems. For example, Chapter VI describes diseases of the nervous system; Chapter XI describes diseases of the digestive system. The hierarchical structure within each chapter is different, as codes are grouped according to features relevant to the type of conditions contained in the chapter, such as by mode of transmission and infecting organisms for infectious diseases, by behaviour and site for neoplasms, and so forth. For example, an acute myocardial infarction may be described by the following ICD-10 hierarchy: Diseases of the circulatory system → Ischaemic heart diseases → Acute myocardial infarction → Acute transmural myocardial infarction of inferior wall. There are, in total, over 70,000 codes in ICD-10. Version 9, which was notably still in use in the United States as late as 2015, had a similar hierarchical structure but fewer than 20,000 codes[2].

When a person seeks care from a healthcare provider, ICD codes are used to describe the diagnoses made by the provider. The codes could be assigned directly by the provider, but it is often retrospectively assigned by clinical coders based on free-form text notes made by the treating provider and other providers involved in the event. A principal diagnosis code is often assigned to describe the primary diagnosis or reason that a patient sought care. Additional secondary or associated diagnoses are usually codes indicating pre-existing comorbidities (i.e. the presence of two or more diseases or medical conditions) that the patient has.

In this research, we investigate a machine learning system that is able to automatically assign ICD codes to hospital events based on information contained in free-form text discharge summaries.

1.3 Data

We use the open-source MIMIC-II and III datasets, which contain de-identified medical records from intensive care unit (ICU) stays at the Beth Israel Medical Centre [Edward William Johnson et al., 2016]. A record contains detailed data on the patient stay, including clinical information such as laboratory tests, diagnosis codes, procedure codes and free-form text clinical notes. The aim of our work is to use the free-form text discharge summaries to automatically assign ICD-9 codes to each case.

The MIMIC-II dataset consists of 22,815 records with non-empty discharge summaries; the updated MIMIC-III dataset contains 52,722 records with non-empty discharge summaries. Each MIMIC record is assigned one or more ICD-9 codes. Each MIMIC-II note is assigned 9.5 ICD-9 codes on average; MIMIC-III notes are assigned 15.9 codes on average. The median number of words per document are 1,322 and 1,375 in the MIMIC-II and III datasets respectively[3]. Only 10% of MIMIC-II documents have fewer than 430 words and 10% of documents have more than 2,279 words. Similarly, 10% of MIMIC-III documents have fewer than 690 words and 10% are longer than 2,490 words. Figure 1 illustrates the distribution of MIMIC-II and III documents according to the number of words.

[2] Multiple versions and adaptations of the ICD-9 coding system exist and therefore references for the number of codes can vary significantly.

[3] Based on simple white-space tokenization. Other types of tokenizers are considered at other points in this work. While separating text by white-space in English texts results in a string of tokens that mainly consist of linguistic words, that does not have to be the case.

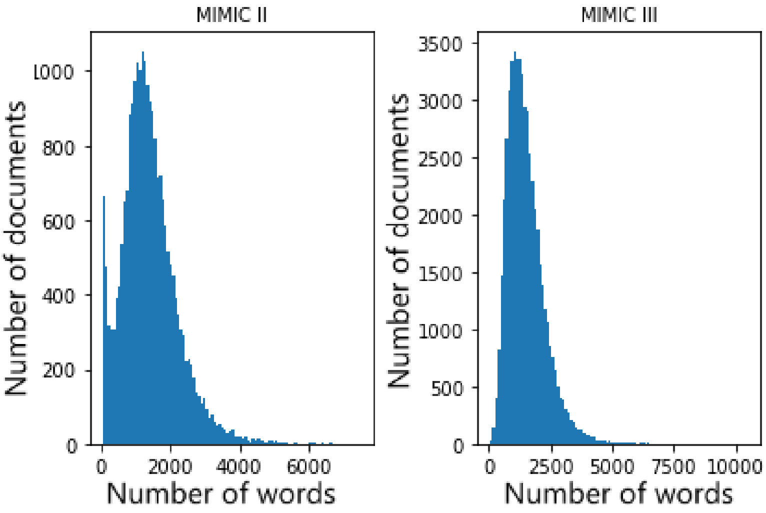

Figure 1: Distribution of the number of words per discharge summary.

1.4 Our contribution

We propose a small pre-trained transformer model based on XLNet, with a label-wise attention mechanism for the auto-coding task. We call our model Multilabel-XLNet, or M-XLNet. As far as we know, our model achieves state-of-the-art performance on the MIMIC-II and III auto-coding tasks using the full label set and is the first pre-trained transformer model that has been applied to the full label set on the MIMIC-II and MIMIC-III datasets.

1.5 Overview

The layout of the paper is as follows: Section 2 discusses work related to the auto-coding problem on the MIMIC-II and MIMIC-III datasets. Section 3 describes our Multilabel-XLNet architecture and outlines the training approach. In Section 4 we discuss the experimental setup. In Section 5 we present the results obtained from our experiments and compare the results to previous work. Section 6 provides our conclusions, and in Section 7 we list future work.

2 Related Work

A review of the literature identified a sizeable body of previous work on the same task and dataset.

2.1 Non-Transformer Models

[Perotte et al., 2013] utilised the binary relevance one-versus-all method with independent Support Vector Machines (SVMs). Two types of SVMs were implemented: a standard "flat" SVM and a "hierarchical" SVM. During pre-processing, the label space was expanded to include not only the gold standard ICD codes, but also codes higher and lower in the ICD hierarchy ("the expanded label space"). For the flat SVM, an independent binary classifier was trained for each label on all training documents. For the hierarchical SVM, the classifier for each code in the expanded label space was trained only on the subset of examples where the code's parent code was classified as positive. The authors hypothesize that by training the classifiers on more relevant data will result in better quality classifiers. The hierarchy is also enforced during evaluation, such that a code can only be predicted as positive if its ancestors were also predicted as positive. The hierarchical SVM outperformed the flat SVM in terms of micro-F1.

The authors discuss the importance of hierarchical evaluation metrics and propose several metrics. They show that based on the shared path, the hierarchical SVM is able to make predictions that are correct deeper into the ICD tree than the flat SVM, despite having lower precision than the flat SVM.

[Vani et al., 2017] tests a number of algorithms on both the MIMIC-II and III datasets and introduces a Gated Recurrent Unit (GRU)-based model called Grounded Recurrent Neural Networks (GRNN). All the algorithms were tested on the same expanded ICD set used by [Perotte et al., 2013]; however, results on the gold-standard label set are not reported.

The GRNN extends the standard GRU-based RNN by splitting the hidden state into *grounded* and *control* dimensions. The number of grounded dimensions equal the number of labels. Thus, the value of the grounded dimension i, at any point in time, reflects the current belief of the model that label i is present in a particular document. The value of the grounded dimension is updated at each time step by considering the current value of the grounded dimension, the current input and the current value of the control dimensions.

The logistic regression model significantly outperforms hierarchical SVM from [Perotte et al., 2013] on the MIMIC-II expanded label set (micro-F1: 0.523 vs 0.395). The continuous bag-of-words with attention and the best GRU model achieves similar performance to the logistic regression model (micro-F1: 0.520 and 0.512 respectively). The GRNN outperforms all baselines by significant margins in terms of micro-F1 on the expanded label set on both the MIMIC-II and III datasets. For the tests on MIMIC-III, Vani et al. also trained a GRU with a cell size of 846 which expanded the model capacity to the same number of parameters as the GRNN in order to test whether improved performance was simply due to increased model capacity, but found that the larger GRU performed worse than all other baseline models.

Rather than training their model on raw medical records, [Shi et al., 2017] extracts only the sections of records where diagnoses are listed using regular expression matching. They observe that diagnoses are typically listed in sections titled either 'discharge diagnosis' or 'final diagnosis'. The authors then use word- and character level Long Short-Term Memory (LSTM) networks to encode each of the diagnoses extracted from the patient record. For example, if they extracted m diagnoses from the medical record through pre-processing, the encoding step will result in m vectors, each representing a diagnosis. They use the same method to derive a vector representation for each assigned ICD code based on the ICD code's text description. Their model is limited to the top 50 most frequent ICD codes. An attention mechanism is then applied to match the hidden states of the diagnoses extracted from the patient record to the hidden states of the assigned ICD codes. They experiment with both a hard and soft attention mechanism. The model was tested on MIMIC-III restricted to the 50 most frequent ICD codes. They report micro-F1 of 0.480 for the hard-attention model and 0.532 for the soft-attention model.

[Baumel et al., 2017] introduces the Hierarchical Attention GRU (HA-GRU). The HA-GRU is an extension of the Hierarchical Attention Network (HAN) introduced by [Yang et al., 2016]. They report the F1 metric for the gold-standard labels on the MIMIC-II test set as well as for ICD-9 codes rolled up to the three-digit level (i.e. a lower level of specificity). Like HAN, HA-GRU consists of two GRU models. The first GRU runs over the tokens and encodes sentences such that the output is a vector representation of each sentence in the document. The second GRU runs over the encoded sentences to create a document representation. In HA-GRU, an attention mechanism is applied over the second GRU only. Their final classification step involves m binary softmax classifiers using the one-versus-all approach, as opposed to having a multi-label classification layer. On the MIMIC-II gold standard test set, HA-GRU achieves a micro-F1 measure of 0.366, which is significantly better than the hierarchical-SVM from [Perotte et al., 2013], but lags behind DR-CAML from [Mullenbach et al., 2018] (see below).

[Mullenbach et al., 2018] introduces the Convolutional Attention for Multi-Label classification (CAML) model. CAML extends the single-layer Convolutional Neural Network (CNN) for text classification by [Kim, 2014] by replacing the max-pooling layer with a per-label attention mechanism. CAML outperforms previous models by a significant margin for all tests performed. The authors also attempt to leverage the labels' text descriptions to solve the problem of dimensionality, where many ICD codes are rarely observed in the dataset. They train a secondary model to learn word embeddings for the ICD code descriptions, which are then used in a regularization function to drive the learned parameters of rarely-observed ICD codes closer to the parameters of more frequently-observed codes with similar text descriptions. The model, called Description Regularized CAML, or DR-CAML, achieves mixed results, outperforming CAML on some evaluation metrics and some datasets, but not on others.

[Li and Yu, 2020] proposes the Multi-Filter Residual Convolutional Neural Network (MultResCNN). Their model adopts the label-wise attention mechanism from CAML and proposes two further extensions. Firstly, their implementation allows for multiple filter sizes, as proposed by [Kim, 2014]. Secondly, they add a second convolutional layer to the model, adopting the residual framework proposed by [He et al., 2016]. The authors also adopt the pre-processing used by [Mullenbach et al., 2018], and the same evaluation metrics are reported, allowing for direct comparison. MultiResCNN outperforms CAML and DR-CAML by a significant margin in terms of micro-F1 on the MIMIC-II and III gold-standard label sets, as well as on the MIMIC-III top-50 label set.

[Vu et al., 2020] proposes the Label Attention Model (LAAT), which is a LSTM with label-wise attention mechanism similar to the one found in CAML and MultiResCNN [Vu et al., 2020]. They also propose a learning mechanism that takes the hierarchy of the labels into account. They call their hierarchical model JointLAAT. The label-attention mechanism proposed for CAML is a form of dot product attention, while the implementation used in LAAT is a modified general attention mechanism. Assume the feature vectors $\mathbf{c}_1, ..., \mathbf{c}_n$ is packed into a matrix $\mathbf{C} \in \Re^{n \times d}$ and consider a label embedding matrix $\mathbf{U} \in \Re^{L \times d_l}$. The dot product label attention used in CAML can then be calculated as

$$Attn_{CAML} = \text{softmax}(\mathbf{U}\mathbf{C}^T)\mathbf{C},$$

where the dimensions of $\mathbf{C}$ and $\mathbf{U}$ are equal, i.e. $d = d_l$. To extend this to the general attention form, let $\mathbf{W} \in \Re^{d_l \times d}$ be a weight matrix. Then:

$$Attn_{General} = \text{softmax}(\mathbf{U}\mathbf{W}\mathbf{C}^T)\mathbf{C}$$

The attention mechanisms used in LAAT modifies the general attention by including an activation function within the alignment function, thus:

$$Attn_{LAAT} = \text{softmax}(\mathbf{U}\tanh(\mathbf{W}\mathbf{C}^T))\mathbf{C}$$

The authors adopt the experimental setup by [Mullenbach et al., 2018] and [Li and Yu, 2020], allowing for direct comparison to CAML and MultiResCNN. They show that LAAT and JointLAAT outperform MultiResCNN on micro-F1 by significant margins on the MIMIC-II and III gold-standard label sets, as well as on the MIMIC-III top-50 label set.

[Teng et al., 2020] proposes a model that uses the textual descriptions associated with each label. They implement a multi-filter CNN with max-over-time pooled features, as proposed by [Kim, 2014], to embed the input text. In parallel to that, the authors calculate a vector embedding for each ICD code in the label space. The authors adopt the Structural Deep Network Embedding (SDNE) for medical knowledge graphs to form an embedding for each label. A label-wise attention mechanism, similar to that employed in the CAML, MultiResCNN and LAAT models are then used to match the document representation to the label representations. Unlike the other models where the label embeddings were randomly initialised, [Teng et al., 2020] leverage the textual descriptions of ICD codes.

The authors' second contribution is to apply an adversarial learning framework, specifically the fast gradient method (FGM) [Miyato et al., 2017], to the auto-coding problem. The rationale is that adversarial learning will dampen the effect of different writing styles while generating additional synthetic training examples. For a neural network $f(\mathbf{x}; \theta)$ and a loss function L, let $\mathbf{g}$ be the gradient with respect to the input embedding $\mathbf{x}$ calculated after a forward and backward pass through the network. The adversarial perturbation is then calculated as

$$\mathbf{r}_{adv} = \epsilon \frac{\mathbf{g}}{||\mathbf{g}||_2},$$

and added to the input, as follows:

$$\mathbf{x}_{adv} = \mathbf{x} + \mathbf{r}_{adv}$$

Another forward and backward pass is then performed with $\mathbf{x}_{adv}$ as input and all network parameters are updated according to the resulting gradient.

The model is only applied to MIMIC-III with the top-50 label set. On this label set, they report a higher micro-F1 score than CAML and MultiResCNN, but not LAAT and JointLAAT. Their ablation study shows that the addition of the label embeddings through a knowledge graph and adversarial learning are responsible for performance boosts independent of each other.

[Kim and Ganapathi, 2021] proposes a model using CNNs with self-attention and code-title based self-attention to create the Read, Attend, and Code (RAC) model. The architecture is based on two parts, a reader and a coder. The reader consists of a self-attention model, and the coder is a code-title guided attention module that is used to predict each clinical code's likelihood. The proposed model was applied on the full MIMIC-III dataset and achieved state-of-the-art results, outperforming all previous models including LAAT and JointLAAT on micro-F1 score.

[Dong et al., 2021] proposes a Hierarchical Label-wise Attention Network (HLAN) using both label-wise word-level attention mechanisms and label-wise sentence-level attention mechanisms. Each hidden layer in the HLAN network consists of two bidirectional-

			Micro-F1		
	Model Architecture	**Has Attention**	**MIMIC-II (Full)**	**MIMIC-III (top 50)**	**MIMIC-III (Full)**
Hierarchical-SVM	SVM	No	0.395	N/A	N/A
RegExp Matching + Soft Attn	LSTM-Based	Yes	N/A	0.532	N/A
DR-CAML	CNN + Attn	Yes	0.457	0.633	0.529
MultiResCNN	CNN + Attn	Yes	N/A	0.673	0.561
LAAT	LSTM + Attn	Yes	0.486	0.715	0.575
JointLAAT	LSTM + Attn	Yes	0.491	0.716	0.575
Read-Attend-Code (RAC)	CNN + Attn	Yes	N/A	N/A	0.586
Hierarchical Label-wise Attention (HLAN)	Bi-GRU + Attn	Yes	N/A	0.641	N/A
Multiple Synonyms Matching Network (MSMN)	LSTM + Attn	Yes	N/A	0.725	0.584

Table 1: Summary of results for non-transformer models.

GRU (bi-GRU) layers. The first bi-GRU layer takes in the word embeddings as well as the context matrix for the word-level attention mechanism, and produces an attention score for each of the labels. The second bi-GRU layer takes in sentence representations and the context matrix for the sentence-level attention mechanism and produces a sentence-level attention score for each label. These are then combined with a label-wise, dot product projection with logistic sigmoid activation to produce the predictions.

[Yuan et al., 2022] proposes the use of synonyms for the ICD codes, as they argue that code synonyms are able to provide more comprehensive knowledge instead of code hierarchies. The authors make use of UMLS [Bodenreider, 2004] to generate synonyms for each label and then use an LSTM to encode the labels. They introduce a special type of attention, multi-synonyms attention, similar to multi-headed attention [Vaswani et al., 2017]. Their model, called Multiple Synonyms Matching Network (MSMN), is applied to both the top-50 and full MIMIC-III dataset and achieves state-of-the-art results on the top-50 dataset, but is outperformed by the RAC model on the full dataset.

Research on automatic ICD coding has been conducted for quite some time and has resulted in a large body of previous work. For the non-transformer architectures, the biggest leaps in improvement came about when the attention mechanism was introduced and improved by various authors, using architectures such as CNNs and LSTMs. The best performing models are the MSMN and RAC models, for the MIMIC-III top-50 and full MIMIC-III label sets, respectively. The MSMN model makes use of synonyms for each label and embeds the labels using a LSTM, while the RAC model uses CNNs with self-attention and code-title based self-attention.

2.2 Transformer models

Research on transformer-based architectures for ICD auto-coding has recently exploded due to the invention of new attention mechanisms and transformer architectures, specifically designed for long input sequence lengths. Most of the research with transformer architectures focus on the MIMIC-III-50 label set, which is the top-50 most occurring labels found in the dataset. Research has looked into using different base-architectures such as BERT or Longformer, as well as new attention mechanisms. The best performing model on the MIMIC-III-50 is HiLAT [Liu et al., 2022] with a 0.735 micro-F1 score. While there are transformers that specifically cater towards the use of long input sequence lengths, HiLAT indicates that these architectures are still not powerful enough to replace full self-attention.

To the best of our knowledge, no state-of-the-art pre-trained transformer models have been successfully applied to the MIMIC ICD code assignment problem using the full label set. The authors in [Li and Yu, 2020] discuss how to incorporate BERT into their model as future work, and notes that it has not performed well in their preliminary experiments.

The authors [Huang et al., 2019] begin with the official XLNet and BERT checkpoints and further pre-train the models on an unlabeled corpus extracted from the MIMIC-III dataset. They refer to their resulting pre-trained models as ClincalXLNet and ClinicalBERT and test their models on the MIMIC-III dataset, attempting to predict mechanical ventilation longer than 7 days as well as 90-day mortality risk, given the clinical notes generated during the first 48 hours of the hospital stay. ClinicalXLNet and ClinicalBERT are tested against the following benchmarks: LSTM, LSTM with attention, Hierarchical Attention Network [Yang et al., 2016], Recurrent Convolutional Neural Network, BERT and XLNet. Their experiments show that ClinicalXLNet outperforms all other models in terms of area under the receiver-operating characteristic curve (ROC AUC). It is notable that XLNet and BERT, without further pre-training on clinical data prior to fine-tuning, do not consistently outperform the other baselines.

[Peng et al., 2019] also begin with the official BERT checkpoints and further pre-train the models on a corpus extracted from PubMed and MIMIC-III consisting of over 4B tokens. They show that their model sets new benchmarks on several clinical-domain natural language processing (NLP) tasks. Benchmarked against BERT, they also point out the importance of domain-specific pre-training.

[Biswas et al., 2021] proposes one of the first transformer architectures for the MIMIC-III auto-coding task. The model, named TransICD, makes use of an embedding layer before the transformer encoder layer and implements a code-specific attention model. The code-specific attention model further processes the hidden representations produced by the encoding layers. The authors apply a structured self-attention mechanism on the hidden representations. Let **H** be the hidden representation, then

$$\mathbf{a}_l = \text{softmax}(\tanh(\mathbf{HU})\mathbf{v}_l), \text{ and}$$

$$\mathbf{c}_l = \mathbf{H}^T\mathbf{a}_l,$$

where **U** and $\mathbf{v}_l$ are trainable parameters. The intuition behind this self-attention is that the vector $\mathbf{c}_l$ will encode sensitive information with respect to label l. The model proposed in this paper is only applied to MIMIC-III with the top-50 label set. While this was one of the first transformer architectures applied to the MIMIC-III coding problem, it did not achieve state-of-the-art results. In terms of micro-F1, the model achieved better results than CAML, but was outperformed by MultiResCNN, LAAT and JointLAAT.

[Pascual et al., 2021] proposes a BERT-based model that was pretrained on the PubMed dataset [Pyysalo et al., 2013] instead of the MIMIC-III dataset. To handle the sequence length limitations of BERT (512 tokens), the authors propose 5 different splitting strategies to generate the input from the discharge summaries:

- Front: first 512 tokens;
- Back: last 512 tokens;
- Mixed: first 256 tokens and last 256 tokens;
- All: split the whole discharge summary into consecutive chunks of 512 tokens;
- Paragraph: Choose the 200 most frequently named paragraphs, each with maximum length of 512 tokens.

The model in this paper was only applied to MIMIC-III with the top-50 label set, and only used the AUC evaluation metric. BERT-ICD (using the best-performing splitting strategy) is outperformed by most other models for which the same AUC metrics are reported.

[Zhou et al., 2021] proposes an Interactive Shared Representation Network with Self-Distillation (ISD) that uses multi-scale CNNs as encoders and a BERT-like decoder. The authors propose two new attention mechanisms, namely shared attention and interactive shared attention. The words in the clinical notes are first mapped into a low-dimensional embedding space and passed into CNNs with differing kernel sizes. The output from these convolutions are then concatenated into a matrix, which is the hidden representation matrix. The hidden representation matrix is then passed through the shared attention mechanism, which attempts to learn shared representations of low and high occurrence labels through attention. The shared attention mechanism, however, lacks interaction between coding relevant information. The authors propose interactive shared attention to fix this problem. The authors propose two tasks to ensure that the decoder can model the dynamic code co-occurrence pattern:

- Missing code completion: A code sequence is generated from a clinical note in the training set and a random code is masked. The decoder takes the code sequence as input and tries to predict the masked code.
- Incorrect code removal: A code sequence is constructed from a clinical note in the training set and a random incorrect code is added. The decoder takes the code sequence as input and tries to mask the incorrect code with a special mask token.

[Feucht et al., 2021] proposes the Description-based Label Attention Classifier (DLAC) that can be applied to different transformer-based architectures. The authors use BERT-base, Hierarchical BERT-base and Longformer-base. The overall architecture uses the transformer to create word embeddings for the discharge summaries, and a word2vec model [Mikolov et al., 2013] to create word embeddings for the ICD-9 code descriptions. These two word embedding matrices are fed into the DLAC layer. Let **E** be the word embedding matrix for the discharge summary and **D** be the description embeddings where **D** is set to be trainable. First, a dimension transformation is applied to the word embedding matrix with **U** matching the shape of the description matrix **D**. The label attention matrix is then calculated as:

$$\mathbf{A} = \text{softmax}(\mathbf{EU} \cdot \mathbf{D}^T), \tag{1}$$

					Micro-F1	
	Model Architecture	**Number of Heads**	**Number of Layers**	**Number of Hidden Units**	**MIMIC-III (Full)**	**MIMIC-III (top 50)**
TransICD	BERT-like Encoder	8	2	-	-	0.644
BERT-ICD*	BERT	8	12	768	-	-
ISD	BERT-like Decoder	-	-	-	0.559	0.717
DLAC	Longformer	12	12	768	-	0.62
HiLAT	XLNet	12	12	768	-	0.735

*Table 2: Summary of results for transformer models. *BERT-ICD was measured only with the AUC-ROC measurement.*

after which contextual embeddings for each label is calculated by aggregating information from **E**, as follows:

$$\mathbf{C} = \mathbf{E}^T\mathbf{A} \tag{2}$$

The architectures in the paper were only applied to MIMIC-III with the top-50 label set. The authors found that the Longformer with DLAC performed the best out of all their models; however, it only managed to perform better than CAML and was outperformed by both TransICD and ISD.

[Liu et al., 2022] proposes a Hierarchical Label-wise Attention Transformer (HiLAT) model that makes use of XLNet-base as the transformer model. HiLAT consists of 4 layers, namely the transformer layer, a token-level attention layer, a chunk-level attention layer, and a final layer consisting of multiple single feed-forward neural networks. The token-level attention layer and chunk-level attention layers both use the label attention mechanism proposed by [Vu et al., 2020]. The model was pretrained on the MIMIC-III dataset. Each discharge summary was chunked, then fed into the transformer and token-level attention layers. The output from this was then fed into the chunk-level attention, which finally fed into the feed-forward neural networks which were used to make the prediction. The architecture in this paper was only applied to MIMIC-III with the top-50 label set, but managed to achieve state-of-the-art results on the micro-F1 score, outperforming all previous architectures.

3 Our Approach

Despite their state-of-the-art performance on numerous NLP tasks, the application of transformer models to down-stream tasks with long input documents remains non-trivial. The main reason for this is that the self-attention mechanism scales quadratically with the sequence length [Vaswani et al., 2017].

Because they use an absolute positional encoding scheme, BERT and other models closely related to it, are trained with an absolute maximum sequence length. For the large pre-trained models, this sequence length is typically only 512 tokens. To put this in context, using the BERT word-piece tokenizer, the median document length in the MIMIC-II dataset (after pre-processing) is 1,779 tokens, and only 8.4% of documents are shorter than BERT's maximum length.

XLNet is also pre-trained with a maximum sequence length of 512 tokens, but because of its relative positional encoding scheme and use of a segment recurrence mechanism, it can be fine-tuned on documents of any length. However, the practical application of XLNet to long documents is still limited by the quadratic relationship between sequence length and complexity. Complicating matters further, is that the sentence-piece tokenizer used by the pre-trained XLNet leads to even longer input sequences. The median document length in the MIMIC-II dataset (after pre-processing) is 2,017 sentence-piece tokens. As a result, the naive implementation of state-of-the-art pre-trained transformer models to our particular problem does not produce promising results.

3.1 Our transformer

We propose Multilabel-XLNet (M-XLNet), a small XLNet-based model that is able to process input sequences of up to 3,072 tokens while still fitting into a single 32GB Nvidia V100 GPU with mini-batch sizes of at least four. We pre-trained our model on the permutation language modelling (PLM) task proposed by the XLNet authors. We then extend our model for multi-label classification by adding a label-wise attention layer and a fully-connected layer, and fine-tune it on the MIMIC auto-coding task. This section is organised as follows. We first describe the tokenizer employed in this model. We then outline the architecture of the XLNet encoder, followed by a discussion of the pre-training process. Finally, we detail our task-specific fine-tuning methods for multi-label classification.

3.2 M-XLNet tokenizer

Departing from the XLNet paper, we trained a BERT-like word-piece tokenizer instead of a sentence-piece tokenizer. The choice was for practicality: the word-piece tokenizer produces shorter input sequences than the sentence-piece tokenizer. Given the length of our input documents and the quadratic relationship between sequence length and complexity for the self-attention operation, this choice was important to effectively reduce the input sequence length.

The tokenizer adds a special classification token (denoted as `<cls>`) at the beginning of input sequences. The final hidden state for this token (i.e. the hidden representation at the final XLNet layer) is typically used as the aggregate representation of the input sequence, analogous to the final hidden state of RNNs, for classification tasks with transformers [Devlin et al., 2019, Yang et al., 2019].

3.3 M-XLNet encoder

Our encoder consists of an input layer and two to eight chained XLNet layers. Consider a sequence of input tokens $t_1, ..., t_T$. The vector representation for each token is calculated as the sum of its word embedding and its relative positional encoding. In the remainder of this section, we denote the input sequence as $\mathbf{x}_1, ..., \mathbf{x}_T$, where each $\mathbf{x_t}$ is the sum of the word and positional embeddings.

Following the outline of the original transformer by [Vaswani et al., 2017], each XLNet layer consists of an attention sub-layer and a fully-connected feed-forward neural network. XLNet replaces the standard transformer self-attention sub-layers with a two-stream self-attention sub-layer [Yang et al., 2019]. The XLNet layers produce a series of hidden states $\mathbf{h}_1, ..., \mathbf{h}_T$. The encoder architecture is illustrated in Figure 2.

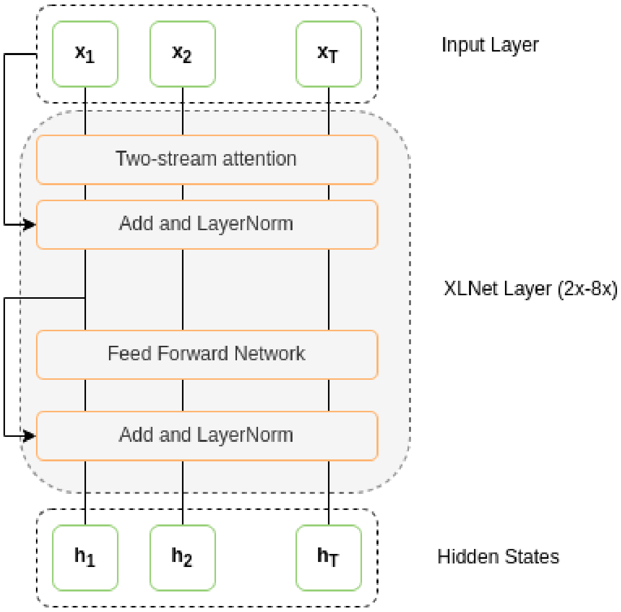

Figure 2: M-XLNet encoder architecture.

3.4 M-XLNet pre-training

We pre-train the M-XLNet encoder on the permutation language modelling (PLM) task, as outlined in [Yang et al., 2019], on all text notes in the MIMIC-III dataset. After data cleaning, our training corpus consisted of 508M word-piece tokens. We then segmented the corpus into 3072-token long sequences, such that the length of each training example was equal to our desired maximum model length.

Pre-training was performed in mini-batches of size four for 40,000 optimisation steps. We followed the same pre-training process as XLNet, using the Adam optimiser [Kingma and Ba, 2015] and linearly reducing the learning rate after each update step to reach zero after the final training step. Our initial learning rate was set to 3E-4, slightly higher than the initial learning rate of 4E-4 used by XLNet, to encourage faster convergence. Other hyperparameters were left to the default values in the Pytorch implementation ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and no weight decay).

3.5 Multi-label classification with M-XLNet

The conventional approach employed to perform classification with transformers, such as XLNet, is to use the final hidden state of a special classification token as the document representation [Devlin et al., 2019, Yang et al., 2019]. This document vector is then passed through a sigmoid output layer to produce the final classification output vector. Let $\mathbf{h}_{CLS} \in \Re^d$ be the final hidden representation of the document. Then:

$$\hat{\mathbf{y}} = \sigma(\mathbf{W}\mathbf{h}_{CLS} + \mathbf{b})$$

We propose a novel approach in which the label-wise attention mechanism is applied to the sequence of hidden states $\mathbf{h}_1, ..., \mathbf{h}_T$ produced by the XLNet encoder. The aim of the label-wise attention mechanism is to learn a probability distribution $\alpha^{(j)} \in \Re^{T}$ for each label l_j, such that $\alpha_i^{(j)}$ determines how much attention the model should place on the i^{th} feature when predicting the j^{th} label. Formally, each query vector represents a label, and the attention function calculates the alignment between each query and each feature vector.

Suppose the sequence of hidden states are packed into a matrix $C = [\mathbf{h}_1 ... \mathbf{h}_t] \in \Re^{T \times d}$. Let $\mathbf{Q} \in \Re^{L \times d}$, where L is the number of labels and d the size of the feature vectors. The values of the query vectors are randomly initialised and learned during training. Applying dot product attention, the weights are given by:

$$\alpha = \text{softmax}(\mathbf{Q}\mathbf{C}^{T})$$

The attention-weighted features can then be computed as:

$$\mathbf{M} = \alpha \mathbf{C} \in \Re^{L \times d}$$

Intuitively, the matrix **M** contains a feature vector for each label which holds all the information about that label found in the input document. The classification step then involves applying a binary sigmoid output layer to each feature vector. Thus, for the j^{th} label we have

$$\hat{y_j} = \sigma(\mathbf{m}_j \mathbf{w}_j^{T} + b_j),$$

where $\mathbf{m}_j$ is the j^{th} row in **M**. We implement this classification step by packing the weight vectors $\mathbf{w}_1, ..., \mathbf{w}_L$ into a matrix **W** and computing the element-wise product with **M** and summing across the hidden dimension

$$\hat{\mathbf{y}} = \sigma\big((\mathbf{W} \otimes \mathbf{M})\mathbf{1} + \mathbf{b}_w\big),$$

where **1** is a vector of ones and σ is the sigmoid function.

For this fine-tuning approach, we set the learning rate in our Adam optimizer to 2E-5 for the XLNet layers and 3E-4 for the attention and output layers. We find that setting the learning rate for the XLNet layers an order of magnitude smaller than the classification layers achieves better validation accuracy on the classification task than when the same learning rate is used for all layers. We speculate that using a larger learning rate results in the XLNet encoder "unlearning" some of the language representation learned during pre-training.

3.6 Evaluation Metrics

Consider the following definitions for a binary classification problem:

$$\widehat{P_i} = \begin{cases} 1 \text{ if example } i \text{ is predicted positive} \\ 0 \text{ if example } i \text{ is predicted negative} \end{cases}$$

$$\widehat{T_i} = \begin{cases} 1 \text{ if the ground truth for example } i \text{ is positive} \\ 0 \text{ if the ground truth for example } i \text{ is negative} \end{cases}$$

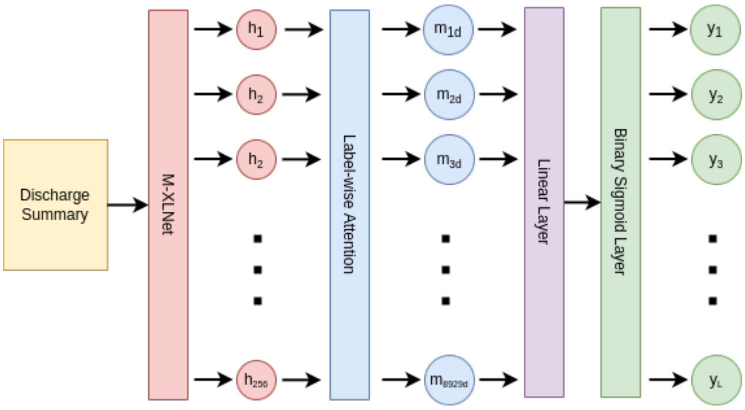

Figure 3: End-to-end architecture of our model.

Given this notation, the precision, recall and F1 metrics are defined as:

$$P = \frac{\sum_i \widehat{P}_i \text{ x } \widehat{T}_i}{\sum_i \widehat{P}_i}$$

$$R = \frac{\sum_i \widehat{P}_i \text{ x } \widehat{T}_i}{\sum_i \widehat{T}_i}$$

$$F1 = \frac{2 * P * R}{P + R}$$

In all classification tasks, there is a trade-off that occurs between the precision and the recall. The recall for any classification model that produces real valued output can be arbitrarily increased by changing the decision function that is applied to the real value output. However, this comes at the expense of reducing the precision of the model.

Recall, precision and F1 does not trivially extend to multi-label classification tasks. There are two main extensions to precision, recall and F1, which allow for the measurements to apply to multi-label classification tasks. These are the macro averaged F1 (macro-F1) and micro averaged F1 (micro-F1). The macro-F1 calculates the F1 score for each label and then computes the unweighted mean across all the labels. This implies that each label is assigned equal importance, regardless of how frequently it appears in the data. The micro-F1 was preferred in our research because it aggregates the contributions of all classes to compute the average metric, which places more weight on more frequent labels.

The micro-F1 is calculated by first summing the individual true positives, false positives and false negatives across all the classes for each sample. Extending the definition from above, the micro precision, recall and F1 are calculated as follows:

$$\begin{aligned}\widehat{P_i} &= \text{ set of predicted classes for sample } i\\ \widehat{T_i} &= \text{ set of ground-truth classes for sample } i\\ miP &= \frac{\sum_i |\widehat{P_i} \cap \widehat{T_i}|}{\sum_i |\widehat{P_i}|}\\ miR &= \frac{\sum_i |\widehat{P_i} \cap \widehat{T_i}|}{\sum_i |\widehat{T_i}|}\\ miF1 &= \frac{2 * miP * miR}{miP + miR}\end{aligned}$$

Extending this further to the hierarchical case, we want to penalize predictions that are closer to the true class less severely than the predictions where there is no overlap in the hierarchical path of the prediction and ground truth. We use the definition provided by [Silla and Freitas, 2011] to extend the given precision and recall for hierarchical classification tasks. The definitions of $\widehat{P_i}$ and $\widehat{T_i}$ are extended to take into account the ancestors of the predicted and ground truth classes, as follows:

$$\begin{aligned}h\widehat{P_i} &= \text{ set of predicted classes and their ancestors for sample i}\\ h\widehat{T_i} &= \text{ set of ground-truth classes and thier ancestors for sample i}\end{aligned}$$

$$\begin{aligned}hP &= \frac{\sum_i |h\widehat{P_i} \cap h\widehat{T_i}|}{\sum_i |h\widehat{P_i}|}\\ hR &= \frac{\sum_i |h\widehat{P_i} \cap h\widehat{T_i}|}{\sum_i |h\widehat{T_i}|}\end{aligned}$$

where hP is the hierarchical precision and hR the hierarchical recall. The definition for calculating the F1 score is extended as follows:

$$hF1 = \frac{2 * hP * hR}{hP + hR}$$

4 Experimental Setup

For MIMIC-II, the same procedure for pre-processing and data-splitting was followed as in [Perotte et al., 2013, Vani et al., 2017]. The discharge summaries and the related ICD-9 codes were extracted from the MIMIC-II dataset, and records without discharge summaries or empty discharge summaries were discarded. The de-identified discharge summaries contain tags such as <Hospital1> that are used in place of personal information. All of these tags were replaced by a single <anon> token during our preprocessing. All newline and special characters were removed from the discharge summaries, and all hyphenated words were split into separate tokens. Finally, all tokens were converted to lower case. Once the pre-processing was done, the data was split into a train-validation-test set with an 80%-10%-10% ratio. This leads to a final train-validation-test set with

	MIMIC-II			MIMIC-III		
	Precision	Recall	Micro-F1	Precision	Recall	Micro-F1
Flat metrics	0.5690	0.4443	0.4989	0.6338	0.5442	0.5856
Hierarchical metrics	0.7002	0.5400	0.6097	0.7434	0.6342	0.6845

Table 3: Results when using M-XLNet with label-wise attention on the MIMIC-II and III test sets.

18,250, 2,283 and 2,282 training, validation and testing samples respectively.

For MIMIC-III, the same procedure for pre-processing and data-splitting was followed as in [Mullenbach et al., 2018, Li and Yu, 2020]. All discharge summaries, their addenda (where available) and ICD-9 codes were extracted from the MIMIC-III dataset. Medical records without discharge summaries or empty discharge summaries were discarded. All punctuation tokens and numeric-only tokens were removed, and all tokens were converted to lower case. In addition to the ICD-9 diagnoses codes, the MIMIC-III dataset also includes ICD-9 procedure codes. The ICD-9 codes stored in the MIMIC-III dataset are stored without the standard dot notation. The ICD-9 codes were converted to the standard dot notation doing the following: for ICD-9 codes that started with an 'E' as the first character, the dot is replaced after the fourth character; for all other ICD-9 diagnoses codes, the dot is placed after the third character. For the ICD-9 procedure codes, the dot is placed after the second digit and for codes with fewer digits than the dot position, no dot is inserted. In total, this leads to 8,929 unique labels assigned to the MIMIC-III records where 6,918 labels are ICD-9 diagnoses codes and 2,011 are ICD-9 procedure codes. Once the pre-processing was done, the data was split into 47,723 training samples, 1,631 validation samples and 3,372 testing samples

5 Results

Our main results are presented in Table 3, Table 4 and Table 5. Our transformer implementation achieves state-of-the-art results on the MIMIC-II and MIMIC-III datasets. On the MIMIC-II test set, M-XLNet achieves a micro-F1 score of 0.4989, outperforming the previous state-of-the-art JointLAAT by 0.0079. On MIMIC-III, M-XLNet outperforms RAC by 0.008 in terms of micro-F1, setting a new benchmark at 0.5944. The results for the XLNet with 8 layers were verified by repeating the run 5 times, using a new random seed for each run. The mean and standard deviation are reported in Table 5.

To the best of our knowledge, this is the first successful implementation of a pre-trained transformer model on the MIMIC auto-coding problem using the full label set. Our transformer with 256 hidden units, 8 attention layers, and 4 attention heads per attention layer is small. By comparison, the pre-trained XLNet language model that achieved state-of-the-art performance on several NLP tasks consists of 24 attention layers, with 16 attention heads per attention layer, and has a hidden size of 1,024. It may therefore be possible to improve on these results by expanding the size of the transformer. Thus, despite its state-of-the-art performance on the MIMIC multi-label classification task, our model may currently best be viewed as a proof of concept.

The scope for improvements through expanding model capacity is likely to be constrained by the availability of relevant training data. We trained our XLNet language

	Flat Metrics			Hierarchical Metrics		
Layers	Precision	Recall	Micro-F1	Precision	Recall	Micro-F1
2	0.6338	0.5442	0.5856	0.7434	0.6342	0.6845
4	0.5974	0.5129	0.5508	0.6374	0.6756	0.6560
6	0.6548	0.5394	0.5916	0.7324	0.6885	0.7087
8	0.6419	0.5534	0.5944	0.7517	0.6742	0.7108

Table 4: M-XLNet with label-wise attention and varying number of layers on the MIMIC-III test set.

Model	**Micro-F1(50)**	**Micro-F1(Full)**
HLAN [Dong et al., 2021]	0.641	-
HAN [Dong et al., 2021]	0.594	0.407
Bi-GRU [Mullenbach et al., 2018]	-	0.417
CNN [Mullenbach et al., 2018]	-	0.419
SVM [Mullenbach et al., 2018]	-	0.441
DR-CAML [Mullenbach et al., 2018]	0.633	0.529
CAML [Mullenbach et al., 2018]	0.614	0.539
MultiResCNN [Li and Yu, 2020]	0.670	0.552
LAAT [Vu et al., 2020]	0.715	0.575
JointLAAT [Vu et al., 2020]	0.716	0.575
MSMN [Yuan et al., 2022]	0.725	0.584
RAC [Kim and Ganapathi, 2021]	-	0.586
HiLAT [Liu et al., 2022]	**0.735**	-
M-XLNet (Ours)	0.717	**0.5938** ± 0.0003
M-XLNet-HiLAT (Ours)	0.656	-

Table 5: MIMIC-III-50 and MIMIC-III-FULL Results. The (50) indicates results for the MIMIC-III-50 dataset, and the (Full) indicates results for the MIMIC-III-FULL dataset. M-XLNet-HiLAT is our XLNet model with HiLAT attention applied.

model on a corpus of 508M word-piece tokens; XLNet was pre-trained on 32.89B sentence-piece tokens. The results shown in Table 4 indicate that there is only a 1.50% and 3.84% improvement on the micro-F1 score between the 2 attention layer and 8 attention layer model for the flat metrics and hierarchical metrics respectively. One possible solution to the relative lack of relevant data is to supplement the pre-training corpus with other (non-EHR) bio-medical data. For example, [Peng et al., 2019] pre-trains BERT on a 4B token corpus extracted from PubMed and MIMIC-III, starting from the official BERT checkpoint. However, even though PubMed consists of bio-medical texts, it remains starkly different from clinical notes in terms of writing style and vocabulary. In terms of vocabulary, we noted that 35% of the tokens in the MIMIC discharge summaries are not present in a word2vec model pre-trained on PubMed by [Pyysalo et al., 2013]. Given BERT's maximum length constraint, we are unable to apply the model from [Peng et al., 2019] directly to our auto-coding problem.

Further, the results shown in Table 5 indicate that our model's performance seems to drop off when run on the MIMIC-III-50 dataset compared to other, current state-of-the-art models. The best performing model by quite some margin is the HiLAT model introduced by Li et al. [Liu et al., 2022]. However, we applied the HiLAT attention mechanism to a model of the same size as our M-XLNet model (indicated by **M-XLNet-HiLAT** in Table 5) to compare the different attention mechanisms. Our attention mechanism outperformed the small XLNet model with the HiLAT mechanism (M-XLNet-HiLAT). M-XLNet achieved a result of 0.717 Micro-F1 on the MIMIC-III-50 dataset, whereas the M-XLNet-HiLAT model only achieved a Micro-F1 of 0.656 on the MIMIC-III-50 dataset. This seems to indicate, that given enough computing power, a full-sized XLNet model with our attention mechanism would outperform the HiLAT model and could also indicate that XLNet is able to better model long dependencies when allowed to consume the entire input sequence on its own. Due to limited computational resources, we were unable to implement a small XLNet model with the HLAN attention mechanism, however, the results in Table 5 show that our M-XLNet model significantly outperformed the HLAN model on the MIMIC-III-50 dataset, achieving a Micro-F1 of 0.717, compared to 0.641 achieved by the HLAN model.

While the availability of training data will likely constrain the scope for performance gains through increasing model capacity, our pre-training results suggest that we have not fully exploited the available data yet. We observe that after a single training epoch consisting of 40,000 update steps, our XLNet model was still not over-fitting the training data. While this suggests that there is room to improve the language model even in the absence of additional training examples, it is less clear whether such improvements will necessarily result in better performance on our down-stream classification tasks. We leave further pre-training and fine-tuning of our XLNet model as future work.

We chose to keep the number of heads constant, and rather expanded on the number of layers. In [Michel et al., 2019] the authors showed that up to 60% of the heads can be pruned before considerable performance drop is observed in the BLEU task with an encoder-decoder attention mechanism, and up to 80-90% in encoder-encoder and decoder-decoder attention mechanisms.

5.1 Effect of the label-wise attention mechanism

Table 6 shows the results of our pre-trained XLNet model without the label-wise attention mechanism. In this setup, which we call the "standard approach", the entire input document is represented by the final hidden representation for the special classification `<cls>` token. Without the label-wise attention mechanism, the pre-trained XLNet model achieves similar results to RNNs and CNNs without label-wise attention mechanisms. Our XLNet model with label-wise attention outperforms the standard approach by 0.1439 in terms of flat micro-F1 and by 0.1541 in terms of hierarchical micro-F1 on MIMIC-II.

While the standard approach turned out to be successful on several benchmark text classification tasks, it is suboptimal on long input documents and a high-dimensional imbalanced label-space.

5.2 Critical considerations of automatic coding

While there are numerous benefits of applying machine learning to automate the clinical coding process, there are pitfalls. In this section, we will briefly discuss five potential downsides.

	Precision	Recall	Micro-F1
Flat metrics	0.6298	0.2472	0.3550
Hierarchical metrics	0.7705	0.3234	0.4556

Table 6: M-XLNet without label-attention results on the MIMIC-II test set.

Machine learning algorithms rely on the completeness and correctness of data to 'learn' properly. The current method of assigning ICD codes is to have trained clinical coders do it by hand. This approach is highly subjective, and different coders could reasonably assign a different code to a patient record for a different reason. [Kim and Ganapathi, 2021] attempts to establish a human benchmark by asking two qualified clinical coders to assign, by hand, ICD codes to a subset of the MIMIC-III test set. The authors evaluated the human coders' assignments to the benchmark MIMIC-III assignments, in the same way that their machine learning model is evaluated, and find that the model's code assignments are significantly more similar to the benchmark than the human coders' assignments. Rather than being being a benchmark of human performance, their study confirms a high degree of disagreement between trained clinical coders when given the same information. A potential benefit of machine learning models, trained on carefully curated training data, could be to improve the consistency of clinical coding, but further study is needed.

Another issue that is faced around data capturing is due to the nature of free-text notes. Doctors and other health professionals may not capture all of the required information in the notes, which would lead to the model trying to make a classification with sub-optimal or incomplete data and possibly not being able to consider all the dimensions required to factor in to a classification. If the doctor, for example, failed to capture the sex of the patient, the model might miss the classification of diagnosis that may be more prevalent in a specific sex. This leads to another issue which is implicit biases in clinical data. [Ghassemi and Nsoesie, 2022] reported on the issues that is faced in the medical world with these inherent biases and how machine learning models have automated these biases due to the data they use.

Another challenge is the interpretability of the predictions that machine learning models make. For the information provided by a deep learning model to be used and trusted in the health sector, health professionals must understand and be comfortable with the 'reasoning' behind why the model made the predictions that it did. Attention mechanisms has helped with interpretability in both NLP and computer vision. For example, attention weights in NLP can be used to identify the words and phrases in the input texts that most contributed to the model making the predictions that it did. However, explainability is an active research area in which more work is needed.

6 Conclusions

We presented a systematic analysis of widely-used and state-of-the-art approaches in natural language processing and text classification in general, and automated clinical coding in particular. We found that deep neural networks are unquestionably the current state-of-the-art.

The best performing models on the auto-coding task can all broadly be separated into two parts: a block of layers responsible for extracting a sequence of feature vectors from the input tokens, and a label-wise attention mechanism. The label-wise attention mechanism is the single most important feature of the deep neural networks, boosting the micro-F1 scores significantly for CNNs, RNNs and transformers, over standard pooling approaches.

Shortly before the onset of this research, BERT had just been published, and new transformer models were achieving state-of-the-art performance on NLP tasks on a regular basis, sometimes outperforming existing benchmarks by large margins. However, direct application of these state-of-the-art pre-trained transformer models on our auto-coding task proved disappointing. This can most likely be attributed to three discerning features of our problem: the size of our input documents, the specialised and non-standard vocabulary, and the high-dimensional label-space. Practical limitations regarding the pre-trained models' maximum sequence length aside, it is perhaps simply too much information to embed into a single document vector. It is only when we constructed a small XLNet model, able to process long input sequences in their entirety, that was pre-trained on domain-specific data and combined with a label-wise attention mechanism, that we started to witness the potential power of transformers.

Even though our XLNet model is small, and could possibly benefit from further pre-training, and was only fitted with a label-wise attention mechanism after the final transformer layer, we achieved state-of-the-art results on the MIMIC-II and III auto-coding problems. We were able to improve on previous benchmarks by comfortable margins in terms of micro-F1. We believe this implementation can act as a proof-of-concept and that further improvements can be achieved with relatively minor tweaks. We outline some of these ideas next under future work.

7 Future Work

Future work includes further development of our transformer model. We believe improved performance can be achieved by expanding the model size (increasing the hidden layer size, the number of layers and the number of attention heads per layer) and performing additional pre-training. At present, the label-wise attention mechanism is applied after the final transformer layer. We believe that adding label-wise attention layers next to earlier transformer self-attention layers could force the model to learn label-aware representations at earlier layers. We are also working on adaptations of the label-wise attention mechanism that directly embeds information about the labels. We have seen some success with this approach with other model architectures, but not when used in conjunction with our M-XLNet architecture. We would like to replace the single layer, fully-connected feed-forward network that is responsible for classification with an architecture that is more suited to handling hierarchical-classification tasks. We are also interested in fine-tuning our pre-trained transformer model on other medical-domain NLP tasks, particularly tasks that also involve long clinical texts. Finally, we would like to establish a human benchmark for the MIMIC-III auto-coding problem. Significant literature on the variability of clinical coding exists, but to the best of our knowledge, no human benchmark to assess whether state-of-the-art auto-coding systems perform near human level is available.

References

[Ajami and Arab-Chadegani, 2013] Ajami, S. and Arab-Chadegani, R. (2013). Barriers to implement electronic health records (EHRs). *Mater. Sociomed.*, 25(3):213–215.

[Baumel et al., 2017] Baumel, T., Nassour-Kassis, J., Cohen, R., Elhadad, M., and Elhadad, N. (2017). Multi-label classification of patient notes: Case study on ICD code assignment.

[Biswas et al., 2021] Biswas, B., Pham, T.-H., and Zhang, P. (2021). TransICD: transformer based code-wise attention model for explainable ICD coding.

[Bodenreider, 2004] Bodenreider, O. (2004). The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res*, 32(Database issue):D267–270.

[Devlin et al., 2019] Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pages 4171–4186.

[Dong et al., 2021] Dong, H., Suárez-Paniagua, V., Whiteley, W., and Wu, H. (2021). Explainable automated coding of clinical notes using hierarchical label-wise attention networks and label embedding initialisation. *Journal of Biomedical Informatics*, 116:103728.

[Edward William Johnson et al., 2016] Edward William Johnson, A., Joseph Pollard, T., Shen, L., Lehman, L.-w., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony G. Celi, L., and G. Mark, R. (2016). Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035.

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

[Feucht et al., 2021] Feucht, M., Wu, Z., Althammer, S., and Tresp, V. (2021). Description-based label attention classifier for explainable ICD-9 classification.

[Ghassemi and Nsoesie, 2022] Ghassemi, M. and Nsoesie, E. O. (2022). In medicine, how do we machine learn anything real? *Patterns*, 3(1):100392.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-December, pages 770–778. IEEE Computer Society.

[Huang et al., 2019] Huang, K., Singh, A., Chen, S., Moseley, E. T., Deng, C. Y., George, N., and Lindvall, C. (2019). Clinical XLNet: modeling sequential clinical notes and predicting prolonged mechanical ventilation.

[Kim and Ganapathi, 2021] Kim, B.-H. and Ganapathi, V. (2021). Read, attend, and code: Pushing the limits of medical codes prediction from clinical notes by machines. In *MLHC*.

[Kim, 2014] Kim, Y. (2014). Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

[Korsakov et al., 2019] Korsakov, I., Gusev, A., Kuznetsova, T., Gavrilov, D., and Novitskiy, R. (2019). Deep and machine learning models to improve risk prediction of cardiovascular disease using data extraction from electronic health records. *European Heart Journal*, 40.

[Li et al., 2019] Li, B., Oh, J., Young, V., Rao, K., and Wiens, J. (2019). Using machine learning and the electronic health record to predict complicated clostridium difficile infection. *Open Forum Infectious Diseases*, 6.

[Li and Yu, 2020] Li, F. and Yu, H. (2020). ICD coding from clinical text using multi-filter residual convolutional neural network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:8180–8187.

[Liu et al., 2022] Liu, L., Perez-Concha, O., Nguyen, A., Bennett, V., and Jorm, L. (2022). Hierarchical label-wise attention transformer model for explainable ICD coding.

[Mannie and Kharrazi, 2020] Mannie, C. and Kharrazi, H. (2020). Assessing the geographical distribution of comorbidity among commercially insured individuals in South Africa. *BMC Public Health*, 20(1).

[Melnick et al., 2020] Melnick, E. R., Dyrbye, L. N., Sinsky, C. A., Trockel, M., West, C. P., Nedelec, L., Tutty, M. A., and Shanafelt, T. (2020). The association between perceived electronic health record usability and professional burnout among us physicians. *Mayo Clinic Proceedings*, 95(3):476–487.

[Menachemi and Collum, 2011] Menachemi, N. and Collum, T. H. (2011). Benefits and drawbacks of electronic health record systems. *Risk Manag. Healthc. Policy*, 4:47–55.

[Michel et al., 2019] Michel, P., Levy, O., and Neubig, G. (2019). Are sixteen heads really better than one?

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*.

[Miyato et al., 2017] Miyato, T., Dai, A. M., and Goodfellow, I. (2017). Adversarial training methods for semi-supervised text classification. In *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*.

[Mullenbach et al., 2018] Mullenbach, J., Wiegreffe, S., Duke, J., Sun, J., and Eisenstein, J. (2018). Explainable prediction of medical codes from clinical text. In *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pages 1101–1111.

[Pascual et al., 2021] Pascual, D., Luck, S., and Wattenhofer, R. (2021). Towards BERT-based automatic ICD coding: Limitations and opportunities.

[Peng et al., 2019] Peng, Y., Yan, S., and Lu, Z. (2019). Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, pages 58–65.

[Perotte et al., 2013] Perotte, A., Pivovarov, R., Natarajan, K., Weiskopf, N., Wood, F., and Elhadad, N. (2013). Diagnosis code assignment: Models and evaluation metrics. *Journal of the American Medical Informatics Association : JAMIA*, 21.

[Pyysalo et al., 2013] Pyysalo, S., Ginter, F., Moen, H., Salakoski, T., and Ananiadou, S. (2013). Distributional Semantics Resources for Biomedical Text Processing. *Aistats*, 5:39–44.

[Shi et al., 2017] Shi, H., Xie, P., Hu, Z., Zhang, M., and Xing, E. P. (2017). Towards automated ICD coding using deep learning.

[Silla and Freitas, 2011] Silla, C. N. and Freitas, A. A. (2011). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1):31–72.

[Slattery et al., 2019] Slattery, S., Knight, D., Weese□Mayer, D., Grobman, W., Downey, D., and Murthy, K. (2019). Machine learning mortality□classification in clinical documentation with increased accuracy in visual□based analyses. *Acta Paediatrica*.

[Strydom, 2021] Strydom, S. (2021). Automatic assignment of diagnosis codes to free-form text medical notes. Master's thesis, Stellenbosch University.

[Tajirian et al., 2020] Tajirian, T., Stergiopoulos, V., Strudwick, G., Sequeira, L., Sanches, M., Kemp, J., Ramamoorthi, K., Zhang, T., and Jankowicz, D. (2020). The influence of electronic health record use on physician burnout: Cross-sectional survey. *J Med Internet Res*, 22(7):e19274.

[Tedeschi et al., 2020] Tedeschi, S., Cai, T., He, Z., Ahuja, Y., Hong, C., Yates, K., Dahal, K., Xu, C., Lyu, H., Yoshida, K., Solomon, D., Cai, T., and Liao, K. (2020). Classifying pseudogout using machine learning approaches with electronic health record data. *Arthritis Care & Research*.

[Teng et al., 2020] Teng, F., Yang, W., Chen, L., Huang, L., and Xu, Q. (2020). Explainable Prediction of Medical Codes With Knowledge Graphs. *Frontiers in Bioengineering and Biotechnology*, 8:867.

[Vani et al., 2017] Vani, A., Jernite, Y., and Sontag, D. (2017). Grounded recurrent neural networks.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 5999–6009.

[Vu et al., 2020] Vu, T., Nguyen, D. Q., and Nguyen, A. (2020). A Label Attention Model for ICD Coding from Clinical Text. *arXiv e-prints*, page arXiv:2007.06351.

[Wong et al., 2018a] Wong, A., Young, A., Liang, A., Gonzales, R., Douglas, V., and Hadley, D. (2018a). Development and validation of an electronic health record–based machine learning model to estimate delirium risk in newly hospitalized patients without known cognitive impairment. *JAMA Network Open*, 1:e181018.

[Wong et al., 2018b] Wong, J., Horwitz, M., Zhou, L., and Toh, S. (2018b). Using machine learning to identify health outcomes from electronic health record data. *Current Epidemiology Reports*, 5.

[World Health Organization, 2004] World Health Organization (2004). *ICD-10: International Statistical Classification of Diseases and Related Health Problems*. World Health Organization.

[Yang et al., 2019] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding.

[Yang et al., 2016] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

[Yuan et al., 2022] Yuan, Z., Tan, C., and Huang, S. (2022). Code synonyms do matter: Multiple synonyms matching network for automatic icd coding.

[Zhou et al., 2021] Zhou, T., Cao, P., Chen, Y., Liu, K., Zhao, J., Niu, K., Chong, W., and Liu, S. (2021). Automatic ICD coding via interactive shared representation networks with self-distillation mechanism. In *ACL*.