BERT- BASED DISEASE PREDICTION FROM ELECTRONIC HEALTH RECORD

Akhil Rajan*
School of Computing
Newcastle University
Newcastle , UK
A.R.Rajan2@newcastle.ac.uk

Paolo Missier
Professor of Large Scale Info Management
School of Computing
Newcastle University
paolo.missier@newcastle.ac.uk

August 20, 2023

ABSTRACT

Recent advancements in healthcare, especially the use of electronic health records, have shown the potential to implement machine learning and deep learning-based methods in the healthcare field. These advancements can contribute to the improvement of treatments and early disease detection helping various patients with chronic disorders. Numerous studies have been done in the healthcare domain, such as BEHRT and Med-Bert, harnessing the power of AI-driven approaches. In this particular study, the prediction of diagnoses utilizing UK Biobank data was undertaken. This was achieved by developing four distinct methods with BERT serving as the base model. With the help of additional parameters such as age and gender, along with specialized techniques, the predictive capabilities of these methods were notably enhanced. The model with a precision score of 0.41 and an AUROC score of 0.89 has been chosen out of four models by comparing different factors. The resulting embedding diagram from this selected model is examined, revealing its capacity to capture diverse relationships among diagnoses within the respiratory, cardiovascular, and neural disorders, and other domains along with gender-based conditions. Additionally, the need for a modified masking strategy, and how it affects the training procedure is discussed and analysed. The selected model is deployed using Flask API for interacting with the model easily. This study shows the potential of advanced AI-driven methodologies in healthcare, offering valuable insights into disease prediction and medical treatments.

Keywords BERT · Disease Prediction · MLM

1 Introduction

In recent years, the healthcare domain has witnessed a remarkable surge in disease prediction capabilities, with the integration of Artificial Intelligence (AI) and machine learning algorithms [1, 2]. These advancements have not only revolutionized early diagnosis but have also resulted in substantial cost reductions for patients[3, 4]. At the core of these breakthroughs lies the growing accessibility and utilization of Electronic Health Records (EHR) data, with the UK taking the lead with an impressive adoption rate exceeding 94% [5, 6]. The vast repositories of EHR data, comprising essential patient information such as diagnoses, clinical remedies, and laboratory results, form the foundational basis for this research. Motivated by the transformative potential of AI in the clinical realm, this dissertation aims to unlock the power of Bidirectional Encoder Representation from Transformers (BERT) [7] for disease prediction[8, 9]. Initially, BERT is used as a language model catering to diverse natural language processing tasks, including text prediction [10], summarization [11], and question-answering [12], BERT has transcended the constraints of conventional models like Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks[13, 14]. Its transformer-based architecture [15] empowers BERT to effectively capture complex relationships within sequential data, making it a promising candidate for analyzing clinical information. Several successful applications of the BERT model in the clinical domain have been reported [16], particularly in disease diagnosis prediction and hospital length of stay

*

estimations. The inherent similarity between language data and EHR data, both amenable to representation as sequences, has significantly contributed to these achievements. Notably, previous studies such as BEHRT [17], G-BEHRT [18], and Med-BERT [19] have served as exemplary demonstrations of BERT's efficacy in clinical settings. Inspired by the successes witnessed in previous research, this work attempts to harness the predictive potential of the BERT model to predict disease diagnosis, utilizing data from the UK Biobank dataset [20]. The UK Biobank dataset represents an invaluable collection of over 150 thousand patient records, consisting of diverse demographics and diagnoses. Beyond the realm of disease prediction, this study will unravel intricate relationships between different diagnoses through the exploration of disease embeddings, unearthing hidden patterns and insights that traditional analyses may overlook. The primary objectives of this research are as follows:

1. To effectively leverage the BERT model for disease prediction utilizing EHR data from the UK Biobank dataset.
2. To address and overcome the challenges and limitations associated with the implementation of BERT in disease prediction within real-world clinical settings.
3. To gain novel insights from visualizing disease embeddings, unravelling intricate relationships between different diagnoses, and leveraging these insights to inform medical decision-making.

To address these research objectives, this dissertation will present a comprehensive literature review (Background), disease prediction methodologies, the application of AI in healthcare, and the pivotal role of BERT in clinical informatics.The literature review section is particularly suited for including more citations from research papers, especially in discussions revolving around disease prediction methods, AI applications in healthcare, and the specific achievements and contributions of BERT in the clinical domain. The methodology section will discuss pre-processing techniques, the architectural design, fine-tuning process of the BERT-based predictive model, and the analysis of appropriate evaluation metrics. The important findings and their interpretations will be meticulously presented and analysed in the results and analysis chapter. The conclusion will summarize the key contributions of this research, describing its inherent limitations, and offer valuable insights into potential directions for future investigations. By using relevant studies to strengthen the existing research base, this dissertation aims to make its findings more trustworthy and important. This also sets a clear background and reason for the main research questions.

2 Background

2.1 Word2vec

Machines, including deep learning models, rely solely on numerical information. Numbers are essential for these models to comprehend and process data. Similarly, for deep learning models, converting features into numerical values is crucial to quantify specific attributes. This quantification enables the models to predict corresponding numeric outputs, and it is effective for variables like temperature, pressure, and volume, which can be easily described using numbers. However, in the context of language and words, the process is more complex since directly providing words to machines is not possible. This is where vectorization becomes important. Vectorization helps machines grasp words by representing them as sets of characteristics, each associated with a numerical value. For instance, consider the word "apple." instead of just presenting the word itself, we can represent it using attributes such as "fruit," "edible," "colour," and "size," with each attribute assigned a specific number. This allows machines to learn about the word "apple" based on these numeric associations. During the prediction process, the model generates probabilities specific to different classes, thereby facilitating the identification of the predicted class based on the highest probability. This concept is explored in depth by Mikolov et al. in their work [21].

2.2 Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM)

Specialized deep learning models are designed for handling sequential data such as weather data and language data. Two such models are Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) [13] networks. These models operate by aggregating information from previous sequences and transmitting it to subsequent ones. This sequential accumulation of information facilitates the generation of comprehensive outcomes. Nonetheless, these models have limitations in capturing long-term dependencies, which results in a phenomenon known as the "vanishing gradient" problem [22]. This problem leads to a decrease in the impact of initial inputs as time progresses, affecting the model's effectiveness in remembering past information over extended periods. To address these short-term memory challenges, the LSTM model was introduced. LSTM is an enhanced version of RNN that employs intricate mechanisms. It minimizes the vanishing gradient problem and has the ability to capture and retain long-term dependencies, making it suitable for tasks involving long sequences and complex relationships. However, it could not solve the vanishing gradient problem and could only grasp the limited context.

2.3 Transformer Architecture and BERT model

The introduction of the transformer mechanism [17] has led to a significant revolution in the field of Natural Language Processing (NLP), making noteworthy contributions. Central to this network is the self-attention mechanism, which helps the model to grasp the connections and interrelations among input elements. As previously discussed, each word can be broken down into vectors, and the proximity of vector values determines their relationships. This is achieved through a process of computing the dot product between pairs of input elements. Consequently, the model comprehends the associations between words or entities during its training. For instance, consider the sentence "I am good" comprising three words. Once processed by a transformer's encoder , it generates three representations, namely z_1, z_2 and z_3 , as outlined in the equation 1 below. In this context, z_1 representing "I" captures 90% of its value from the vector representing "I," 7% from "am," and 3% from "good." This implies that the word "I" holds a stronger relation with "am" compared to "good."

$$\mathrm{z} = \begin{bmatrix} & I & am & good \\ z_1 & 0.9 & 0.07 & 0.03 \\ z_2 & 0.06 & 0.9 & 0.04 \\ z_3 & 0.1 & 0.1 & 0.8 \end{bmatrix} \tag{1}$$

The BERT model [7] employs a transformer encoder layer to create representations through the Masked Language Model (MLM) approach (refer to Appendix A for more details). BERT's distinctive advantage stems from the transformer mechanism, which enables it to predict masked words by taking into account of the surrounding words. This is based on the notion that the meaning of a word is often intricately tied to the words in its surroundings. BERT incorporates multiple stacked transformer layers to further enhance its prediction capabilities, thereby achieving improved performance. The model architecture can be seen in Figure 4.

2.4 BERT in Clinical Domain

The application of BERT within the clinical domain has been extensively investigated over the years, yielding noteworthy contributions exemplified by the efforts of Li et al.[17] (BEHRT) and Medbert by Rasmy et al [19]. Both endeavours have demonstrated promising outcomes within the domain. BEHRT was trained on a dataset encompassing 1.6 million patient records, enabling the prediction of 301 medical conditions using CPRD (Clinical Practice Research Datalink) [23] data. In contrast, Med-BERT's focus was on predicting an impressive array of 82,000 classes, achieved through training on the MIMIC III (ICU data) [24] dataset. In BEHRT, the evaluation of the model was conducted using a prediction score with a threshold of 0.5, and the results showed a remarkable 8% to 13% improvement over the existing state-of-the-art model. On the other hand, Med-BERT employed the AUROC score as a performance metric for its model. Comparing these two models has become challenging due to the divergence in evaluation matrices and dataset characteristics. In BEHRT, the model undergoes initial pre-training on the entire dataset, achieving a precision score of 0.6597. This pre-trained model is subsequently fine-tuned for disease prediction. However, it remains uncertain whether the same dataset is employed for model evaluation during the fine-tuning phase. Should this be the case, there exists a significant possibility that the model could become familiar with the evaluation (test) data as well, potentially explaining the precision score of 0.462. Furthermore, the Masked Language Model (MLM) task could serve as an alternative approach for predicting the next disease. The rationale behind employing a different method for this prediction, instead of simply masking the last token, is not clearly explained. Overall, The evaluation results of the model in BEHRT were not elucidated in a clear manner by the authors, a concern that is similarly highlighted within the Med-BERT paper.

However, it is more advisable to assess the model's performance based on the accuracy and F1 score of masked tokens, especially when dealing with a limited vocabulary size. It's crucial to recognize that certain situations allow for multiple tokens (diagnoses in this case) to seamlessly fit within a sequence. For instance, consider the phrase "I [MASK] dogs," where various words like "love," "like," or "hate" could viably replace the masked token. Therefore, relying solely on precision score, F1 score, and accuracy to evaluate BERT models might not provide a comprehensive representation of their effectiveness.

3 Methodology

This study involved the creation of various models tailored to the data using the MLM (Masked Language Model) approach to predict diseases. The shortcomings of each model are elaborated upon in the following sections.

3.1 Data Understanding

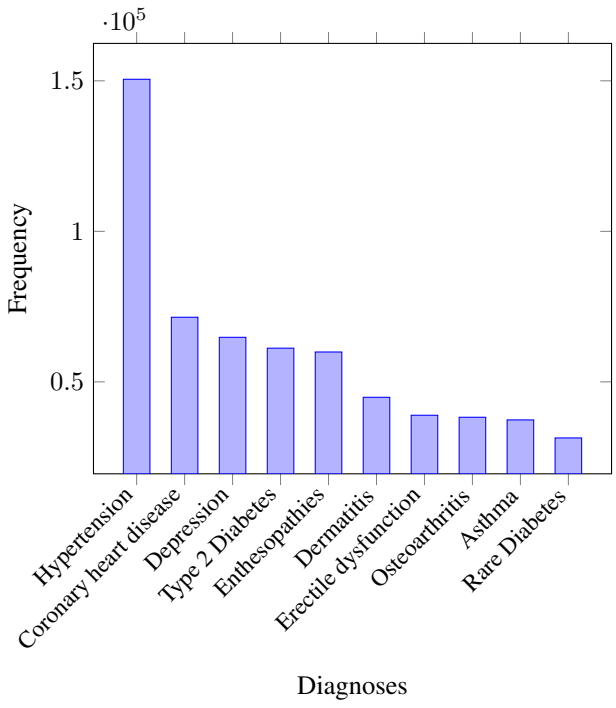
The data utilized in this research is taken from the UK Biobank, which collects data from various sources including NHS (National Health Service). The collected data undergoes pre-processing as part of the AI MULTIPLY [25] project, and it has information about 150,000 patients. This data represents longitudinal medical histories, including patients' diagnoses, prescriptions, and demographic information. For this study, the fields, namely patient ID, age, diagnosis, and gender are considered. Table 1 illustrates the structure of the data, where each row corresponds to a patient's diagnosis along with their demographic data at that particular time. It allows us to observe the conditions diagnosed on the same date as part of a particular visit.

Patient ID	Event Date	LTC	Age at Event	Sex
1024546	09-10-1984	Spondylosis	42	F
1024546	17-12-2007	Obesity	65	F
1024546	19-12-2007	Urinary Incontinence	65	F
1024546	19-12-2007	Female Genital Prolapse	65	F
1024546	28-02-2011	Type 2 Diabetes	69	F
1024546	30-03-2011	Unspecified Rare Diabetes	69	F
1024546	20-04-2011	Type 2 Diabetes	69	F
1024546	20-04-2011	Unspecified Rare Diabetes	69	F
1024546	13-06-2011	Unspecified Rare Diabetes	69	F

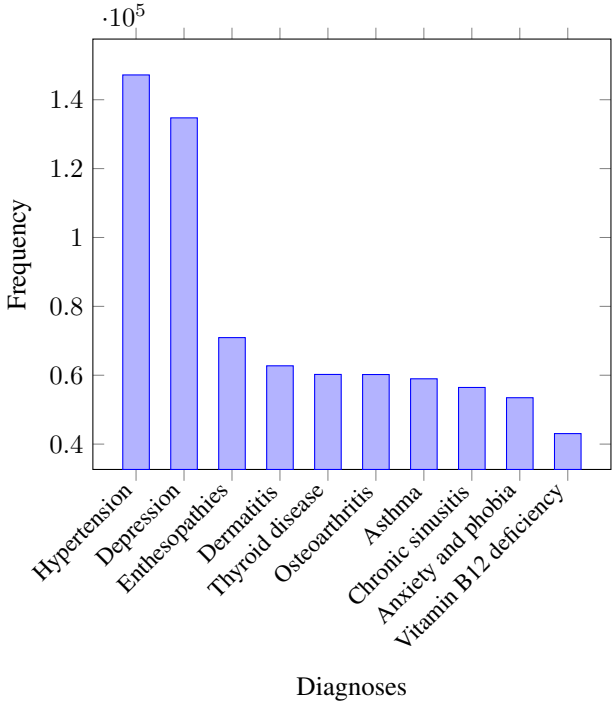
Table 1: structure of input data

3.2 Exploratory Data Analysis (EDA)

EDA shows that Hypertension stands out as the most commonly diagnosed medical condition among patients. It also highlights specific gender-related conditions, such as female infertility, genital prolapse, erectile dysfunction, and male infertility. These findings are visually depicted in Figures 1 and provide valuable insights into the distribution of medical conditions in the patient population. This also indicates that there is a class imbalance in the data source, which can potentially affect the model performance.



(a) Top 10 Diagnoses among Male Patients



(b) Top 10 Diagnoses among Female Patients

Figure 1: Top 10 Diagnoses among Patients

Additionally, the data shows that the frequency of diseases increases as age increases and the peak value is recorded at the age of 61 which is shown in Figure 2.

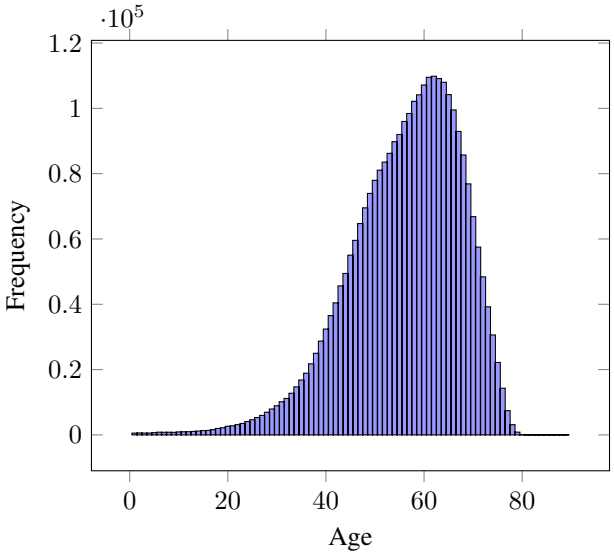


Figure 2: Visualization depicting the relationship between age and diagnosis frequency

3.3 Pre-Processing

1. The important columns were identified for the initial investigation, and the data was cleaned by removing duplicates and null values if existed.
2. The data is then transformed to the proper structure so that it aligns with the BERT's input.
3. The data in Table 1 has been transformed into the following format: [CLS] is added at the start of each string, which represents a single patient ID. [SEP] is inserted between each visit, where a visit is defined by the date of diagnosis for diseases/conditions. If multiple diseases are diagnosed on the same date, they are treated under a single visit. This approach ensures that the model understands the possibility of multiple diagnoses under a single visit.

   [CLS] spondylosis [SEP] obesity [SEP] urine_incont female_genital_prolapse [SEP] type_2_diabetes [SEP] unspecified_rare_diabetes [SEP] type_2_diabetes unspecified_rare_diabetes [SEP] unspecified_rare_diabetes [SEP] type_2_diabetes [SEP]
4. Patient IDs with a minimum of seven rows are chosen for processing, considering the 15% masking requirement. To ensure at least one masked token, seven tokens (excluding [CLS], [SEP], and [PAD]) are necessary, enhancing the effectiveness of Masked Language Modelling (MLM).

3.4 Model: 1

This model is constructed by leveraging the TensorFlow library and importing a pre-trained model from the Hugging Face repository. Tokenization is performed using Hugging Face's BertTokenizer; however, a custom tokenizing function is necessary due to the unsuitability of Hugging Face's built-in tokenizer for the current context. The custom tokenizing function splits the sequence based on whitespace as the delimiter, and then assigns a uniue class number to them. Subsequently, token IDs, token type IDs, and attention masks are generated, laying the groundwork for the Masked Language Model (MLM) (see Appendix A) implementation. In this regard, 15% of the input tokens are subjected to masking using the [MASK] token as mentioned in the original BERT paper [7], this value can be a parameter that impacts both the introduction of noise and the training capability of the model. In most cases it can lie anywhere between 15%-25%. The training process entails a comparison between the model's predicted tokens and the original tokens that were initially masked for the MLM training objective. The employed strategy in this method shows some limitations that need consideration. Firstly, the training process is inherently time-consuming, as the model was originally developed for a different domain, and it has a large size thus, full transfer learning cannot be achieved. Moreover, the model lacks the desired flexibility required for easy customization, adding challenges in finetuning. Lastly, the inability to incorporate

additional embeddings further restricts its adaptability and potential for capturing domain-specific requirements. These limitations make it less useful for the current study. The model achieved the following performance on the test data set with an accuracy of 0.36 and the below table 2 summarises the findings.

	Precision Score	Recall	F1 Score
Macro Average	0.26	0.22	0.23
Weighted Average	0.34	0.36	0.34

Table 2: Summary of Evaluation

3.5 Model: 2

The model's predictive capabilities can be improved by incorporating age and gender as contributing variables. This is achieved by introducing age and gender embeddings, which are added to the token and positional embeddings (See Appendix C for model architecture). To accommodate these new embeddings, two separate layers are introduced, one for age and another for gender. For age, there are 120 different embeddings, each representing a distinct age value. Similarly, there are two embeddings for gender, representing male and female. The embedding dimensions of all five layers (token, position, segment, age, and gender) are kept the same, enabling them to be combined and trained jointly. The process can be represented mathematically as follows, where T, P, S, A, and G denote the token vector, position vector, segment vector, age vector, and gender vector, respectively. The vector E represents the summation of these individual vectors:

$$\sum_{i=1}^{N} T_i = [t_{i1}, t_{i2}, \ldots, t_{in}] \, , \sum_{i=1}^{N} P_i = [p_{i1}, p_{i2}, \ldots, p_{in}] \tag{2}$$

$$\sum_{i=1}^{N} S_i = [s_{i1}, s_{i2}, \ldots, s_{in}] \, , \sum_{i=1}^{N} A_i = [a_{i1}, a_{i2}, \ldots, a_{in}] \tag{3}$$

$$\sum_{i=1}^{N} G_i = [g_{i1}, g_{i2}, \ldots, g_{in}] \tag{4}$$

In the equations above (Equations 2, 3, and 4), *N* stands for the sequence length, which is 985 in this scenario, and *n* represents the embedding dimension, set to 200. The variable *i* indicates the position of the token within the sequence.

$$\mathbf{E}_i = [T_i + P_i + S_i + A_i + G_i] \tag{5}$$

The equation 5 shows that the final embedding *E* is the sum of all the vectors representing token, position, segment, age, and gender. The resulting two-dimensional vector, with a shape of (sequence length, embedding dimension), is then fed into the model for training.

It is important to note that this model is experimental, and representing gender as a sequence for a particular patient might not be logically explainable, as it is better to consider gender as fixed parameter.
The model achieved the following performance on the test data set with an accuracy of 0.34 and the below table 3 summarises the findings.

	Precision Score	Recall	F1 Score
Macro Average	0.22	0.20	0.34
Weighted Average	0.33	0.34	0.33

Table 3: Summary of Evaluation

3.6 Model: 3

In this modeling approach, gender is treated as a constant factor instead of being part of the sequence (refer to Appendix D for the model's further details). It is incorporated into the model by appending it to the end of the vector created by

combining various vectors as shown in figure 3. Mathematically, this is illustrated as follows: The final embedding E is obtained by summing up the vectors representing tokens, positions, segments, and ages, as indicated in Equation 6, resulting in a shape of (sequence length, hidden size).

$$\mathbf{E}_i = [T_i + P_i + S_i + A_i] \tag{6}$$

To the resulting embedding, an additional value (v) is appended at the end, as depicted in Equation 7. Consequently, the embedding shape becomes (sequence length, hidden size + 1). This newly added value represents gender and can take two distinct values, each corresponding to a specific gender. In this context, e represents the embedding value for a particular dimension within the total number of hidden dimensions.

$$\mathbf{E}_i = [e_{i1}, e_{i2}, \ldots, e_{in}, v] \tag{7}$$

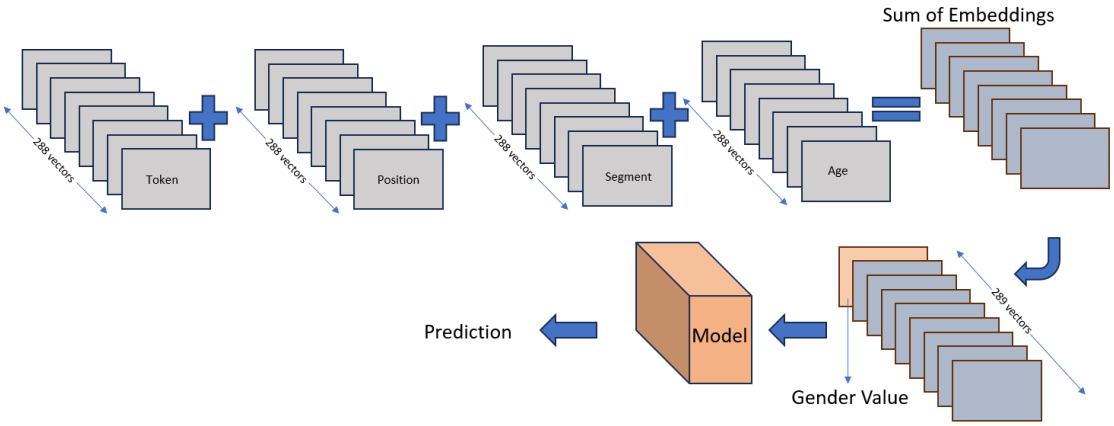


Figure 3: The flow of input data at a given position in a sequence through the model

In this way, the gender information is included, allowing the model to consider gender while making predictions by explicitly injecting the gender value as last vector in final embedding. This study is based on experimental methodology, providing valuable insights into the research question. However, it is essential to acknowledge that, like any experimental investigation, there might be potential sources of error and uncertainties. These findings provides exciting opportunities for further exploration and refinement, which could lead to even more robust findings in the future.
The model achieved the following performance on the test data set with an accuracy of 0.42 and Average AUROC is 0.89 and the below table 4 summarises the findings.

	Precision Score	**Recall**	**F1 Score**
Macro Average	0.32	0.23	0.24
Weighted Average	0.41	0.42	0.40

Table 4: Summary of Evaluation

3.7 Model: 4

The main idea involved in this method is representing gender as a separate token, rather than embedding it as a layer (See Appendix E for model architecture). By doing so, the transformer model can effectively capture the relationship between gender and each diagnosis during the self-attention process. For instance, when the input string contains gender-specific conditions, such as breast cancer for females, the corresponding gender token is incorporated alongside the input string. Consequently, the attention score between the gender token and the specific diagnosis becomes prominent, indicating their strong association during the training process. The entire process, spanning from pre-processing to the final outcomes, is visually depicted in the figure 4 provided below. This graphical representation elucidates the flow of information and the crucial role of the gender token in establishing meaningful relationships between gender and different medical diagnoses. From the analysis of the previous three models, it has become clear that a class

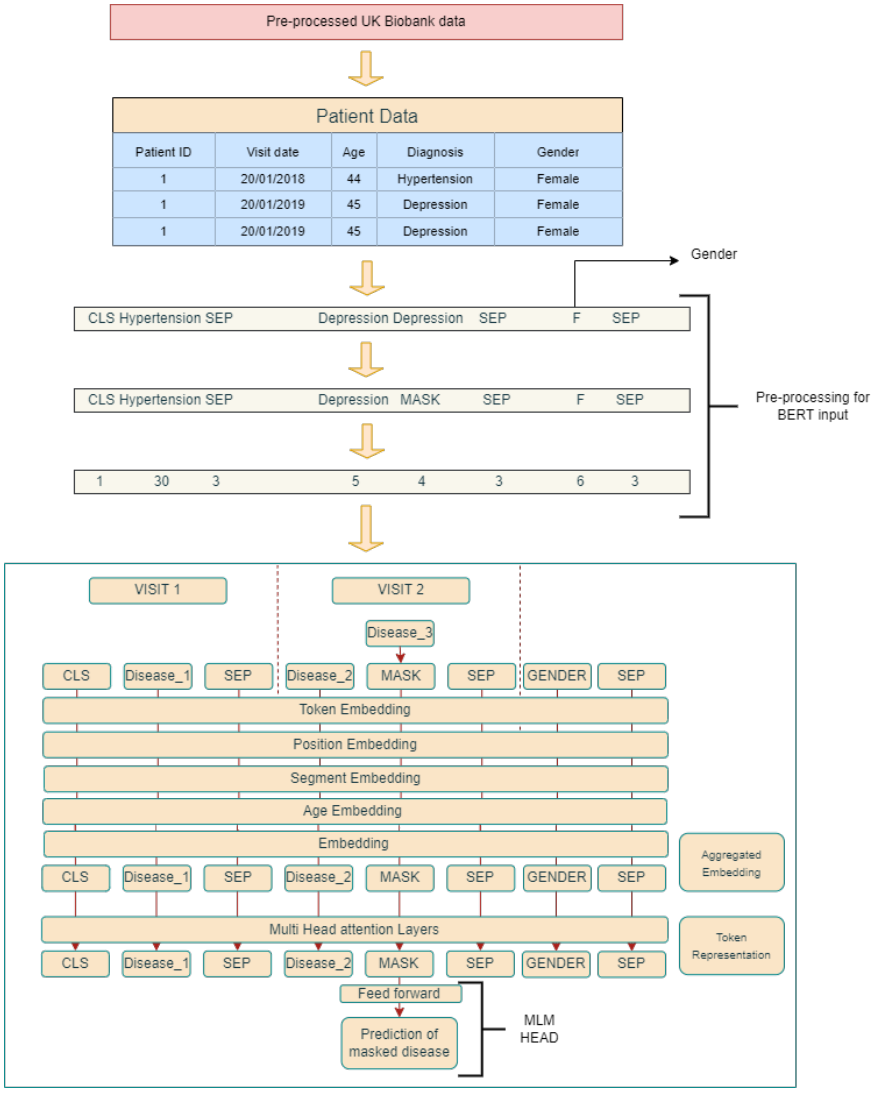


Figure 4: Overall Training Process

imbalance exists within the system, as depicted in Figure 1. Consequently, during the masking process, there is a higher likelihood of masking the most frequently occurring diagnoses, resulting in data bias and adversely impacting the model's performance. To address this issue, a more refined approach is proposed by adopting dynamic masking based on the frequency of 203 diagnoses in the dataset. This method involves assigning a masking score to each of these diagnoses through a specific function, where the function gives a higher value for the least frequently occurring diagnoses in the data and prioritizes masking for diseases having a high masked score. For instance, if the input data contains diseases like hypertension and asthma(low weightage), the algorithm will prioritize masking asthma to enable the model to gain a better understanding of this less frequent condition. This dynamic masking strategy grasps the potential to significantly enhance the model's performance by solving the effects of class imbalance and promoting a more balanced learning process.

$$\mathbf{f}(X = x) = 1 - \mathbf{P}(x) \tag{8}$$

The function *f(X=x)*, denote the masking score which is one subtracted from the probability of the particular diagnosis in the source data, where *x* can take values representing different diagnoses. In the previous scenario, certain diagnoses such as coronavirus, sickle cell anemia, and ADHD were excluded as masked tokens. This led to inadequate predictive capabilities of the model concerning these specific diagnoses. The graphical representation in Figure 5 provides a clear visualization of the improved distribution of masked tokens between the two scenarios. The newly employed approach successfully solved the abrupt spike in token distribution (indicated as red in figure 5) , resulting in a more balanced distribution overall. While this progress is notable, it is important to note that further optimization is still attainable.

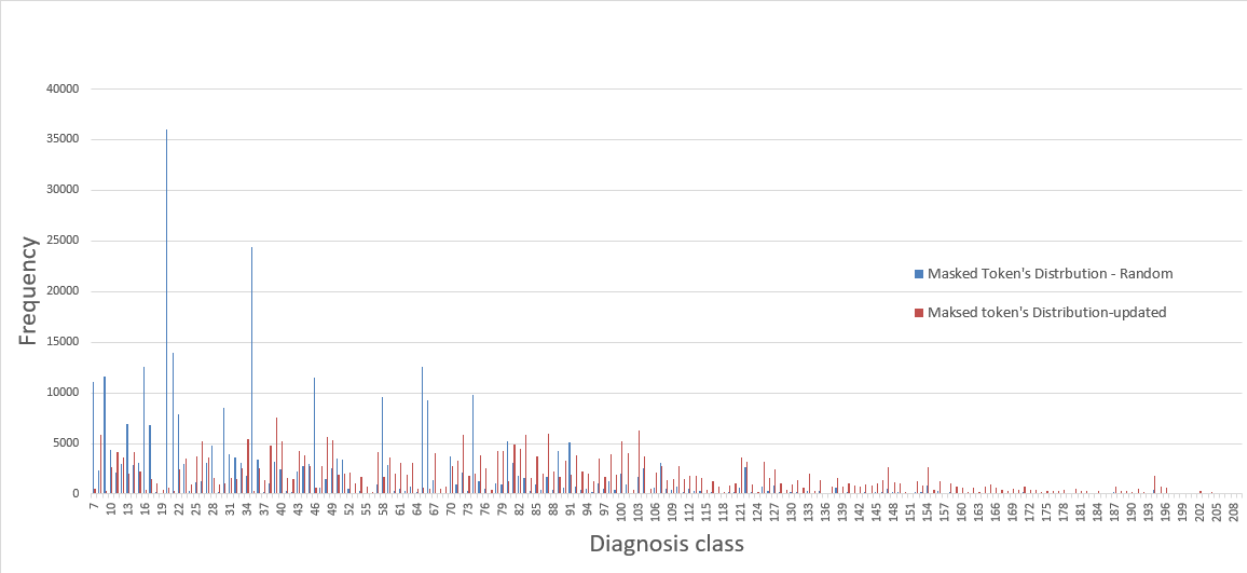


Figure 5: Distribution of Masked tokens for two methods

The model achieved the following performance on the test data set with an accuracy of 0.24 and average AUROC of 0.82 and the below table 5 summarises the findings.

	Precision Score	**Recall**	**F1 Score**
Macro Average	0.26	0.22	0.23
Weighted Average	0.25	0.24	0.24

Table 5: Summary of Evaluation

4 Results and Evaluation

Figure 6 illustrates that among the four models, model 3 exhibits superior performance in terms of precision, F1 score, and accuracy. The selected model showcases an Average AUROC score [26] of 0.89, which is comparable to current state-of-the-art models such as BEHRT (precision score = 0.464, AUROC = 0.954) [17] and Med-Bert (AUROC = 0.75) [19]. Using the AUC value as a standard for evaluating model performance is recommended, as emphasized in various prior studies [26, 27, 28]. It's noteworthy that evaluating models is a difficult process influenced by factors like data size, vocabulary size, and domain [29]. Contrary to expectations, the performance of model 4 proved to be far from satisfactory, even with the implementation of an enhanced masking algorithm. This discrepancy is likely due to the influence of the modified data distribution on model performance. For instance, consider the diagnosis "diabetic type 1" (63), the frequency of masking this diagnosis is lower compared to the frequency resulting from the new strategy, as illustrated in Figure 5. This discrepancy contributes to the higher F1 score achieved by model 4 in comparison to model 3. As such, for this specific condition, model 4 is a preferable choice. Similar situations can also be observed for other diagnoses, where in some cases, model 3 couldn't predict certain conditions such as spondylolisthesis and cystic renal. In essence, the model's ability to predict specific diagnoses varies, regardless of its final accuracy, precision, and AUROC scores. This variability is due to the effects of masking and how data is split between training and testing (see Appendix F for more details). To address this issue, the solution lies in adjusting the masking strategy. It would be wise to implement a dynamic masking strategy during training. This means that tokens are hidden or masked during different epochs within the same batch while training similar to image augmentation. This approach helps the model understand the diverse array of potential diagnoses, leading to an overall improvement in its performance. However, it's important to note that this approach might require more time due to the process of reducing the loss. Also, while splitting the data into test sets, it is important to ensure that all the diagnoses in the vocabulary will be masked in test set as well. Predicting tokens using MLM can be more complex compared to predicting just the last token in a sequence. This complexity can reflect in the evaluation results, where the performance of MLM on test data might not match the

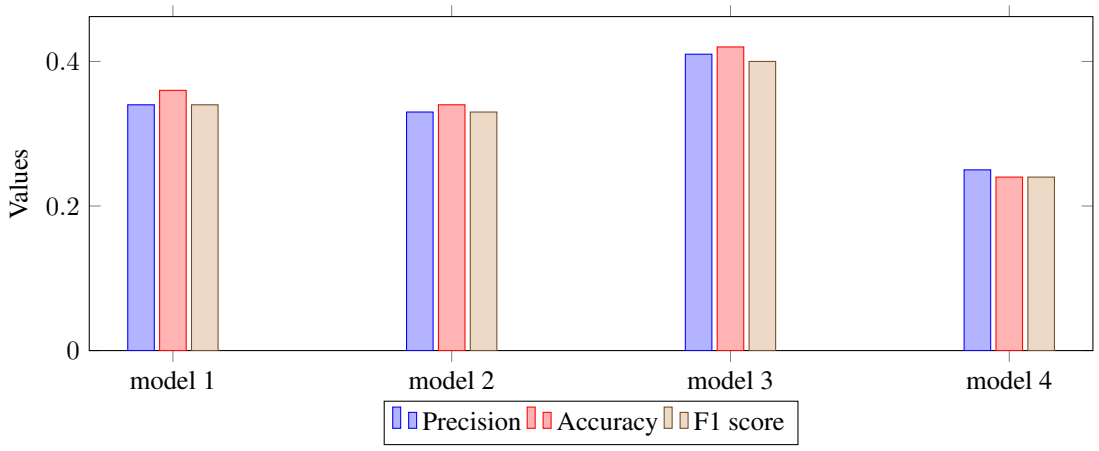
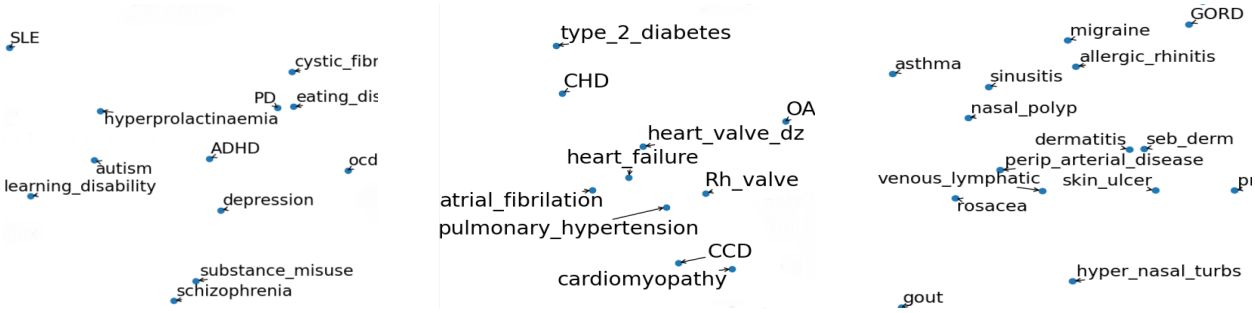


Figure 6: Comparison of all four models

accuracy achieved from next diagnosis prediction. The MLM model can predict the last diagnosis or any diagnosis in between the string. Nonetheless, retraining the model by masking the final token might lead to improved outcomes when predicting the subsequent diagnosis based on a sequence of medical histories. The figure 7 helps to see how the model grasps relations between diagnoses using 289 dimensions. To make this easier to visualise, 289 vectors are transformed into 2-dimensional vectors using a method called t-SNE (t-distributed Stochastic Neighbor Embedding) [30]. The images shows that certain diagnoses are grouped together, which means they are related (see Appendix G for more details). Moreover, it tells that diagnoses linked to gender are not found close to each other. However, by making it simpler, some details might be lost.



(a) Mental disorders (b) Cardiovascular disorder (c) Respiratory conditions

Figure 7: Diagrams showing similar diagnoses are clustered together

4.1 Environment

The training procedures were conducted on the AIM/BSU server, referred to as "Melville," which is provided by Newcastle University. This server is equipped with an NVIDIA Quadro RTX 5000 GPU, boasting an impressive 376 GB of memory. The training phase for each of these models extended beyond a span of 3 days. Additionally, Various combinations of batch sizes (4, 8, 16, 32) and epochs (30, 50, 100, 150) were employed. A consistent learning rate of 0.0001 was set, and the Adam optimizer was utilized across all the models with the help of TensorFlow API.

4.2 Deployment

The deployment is accomplished through a Flask API with a JavaScript-based front end. With the model trained on 203 diagnoses, users can conveniently choose the disease from a drop down menu. (see Appendix B)

5 Conclusion

The primary aims of this study involve utilizing UKBioBank's health data (EHR) to predict diagnoses. This is achieved by employing BERT as the base model and incorporating additional factors like age, gender, along with patients' medical history. Various strategies and techniques are employed for this purpose. Upon evaluating different models, a final model, that is model 3 is selected based on its precision score of 0.41, and AUROC of 0.89 which indicates a better result. This specific model that is model 3 was trained on data that includes gender details alongside a unique diagnosis token, while also incorporating age in a distinctive manner. The difference in the performance of model 4 and model 3 suggests that the evaluation of the model based on accuracy, precision and F1 score may not completely give a good picture as it depends on masking and train-test split. with the help of adaptive-dynamic masking strategies, the model 4 performance can be increased which can be extended to future work. In analyzing the model, hidden patterns and relationships within diagnoses become evident. These findings reveal the interconnections among different diagnoses, presenting valuable insights like how the diagnose related to particular domain are grouped together. This research has shown the potential of employing BERT, a language model, within the healthcare sector. This utilization holds the promise of advancing the health industry through early prediction of diagnoses. This, in turn, empowers medical professionals to offer proactive assistance, particularly in cases of chronic medical conditions. The methodologies discussed in this study have the potential to extend their benefits to new data through proper preprocessing. This adaptability could greatly benefit future researchers as well. However, the limitations, such as class imbalance and noisy data, exert a noticeable adverse impact on the training process. Another aspect of concern arises from the reduction of embeddings from 289 dimensions to just 2 dimensions, which may not provide a comprehensive understanding of the underlying data. Moreover, it's important to acknowledge that training a BERT-based model demands a substantial amount of time. The iterative nature of this process requires patience and multiple iterations to converge towards a satisfactory outcome.

Furthermore, the model developed here will find beneficial in fine-tuning tasks, especially if the domains remain similar. Enhancements to the models discussed are possible through extended training, more refined hyperparameter tuning, the enhancement of masking techniques and an ensemble approach. Incorporating additional parameters, such as prescription data (medication), could improve the model's capability in identifying diagnoses, particularly those arising from potential side effects of diseases.

This augmentation could also facilitate predictions beyond diagnoses, by the addition of factors like length of stay, lifestyle recommendations, post-period prognosis and mortality. Furthermore, this method can be applied in GPT [31] model which also has the potential to diagnose prediction. These advancements could create a better and smarter healthcare system.

References

[1] Fei Jiang, Yong Jiang, Hui Zhi, Yi Dong, Hao Li, Sufeng Ma, Yilong Wang, Qiang Dong, Haipeng Shen, and Yongjun Wang. Artificial intelligence in healthcare: past, present and future. *Stroke and Vascular Neurology*, 2(4):230–243, 2017.

[2] Kun-Hsing Yu, Andrew L Beam, and Isaac S Kohane. Artificial intelligence in healthcare. *Nat. Biomed. Eng.*, 2(10):719–731, October 2018.

[3] D. W Bates. Using information technology to reduce rates of medication errors in hospitals. *BMJ*, 320(7237):788–791, March 2000.

[4] G. Demiris, L. B. Afrin, S. Speedie, K. L. Courtney, M. Sondhi, V. Vimarlund, C. Lovis, W. Goossen, and C. Lynch. Patient-centered applications: Use of information technology to promote disease management and wellness. a white paper by the AMIA knowledge in motion working group. *Journal of the American Medical Informatics Association*, 15(1):8–13, January 2008.

[5] Department of Health and Human Services and Office of the National Coordinator for Health Information Technology. Electronic public health reporting ONC annual meeting. Online, 2018.

[6] S. Parasrampuria and J. Henry. Hospitals' use of electronic health records data, 2015-2017. Online, 2015.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.

[8] Fei Li, Yonghao Jin, Weisong Liu, Bhanu Pratap Singh Rawat, Pengshan Cai, and Hong Yu. Fine-tuning bidirectional encoder representations from transformers (BERT)–based models on large-scale electronic health record notes: An empirical study. *JMIR Medical Informatics*, 7(3):e14830, September 2019.

[9] Hoo-Chang Shin, Alvin Ihsani, Swetha Mandava, Sharath Turuvekere Sreenivas, Christopher Forster, Jiook Cha, and Alzheimer's Disease Neuroimaging Initiative. Ganbert: Generative adversarial networks with bidirectional encoder representations from transformers for mri to pet synthesis, 2020.

[10] Mohammed Boukabous and Mostafa Azizi. Crime prediction using a hybrid sentiment analysis approach based on the bidirectional encoder representations from transformers. *Indonesian Journal of Electrical Engineering and Computer Science*, 25(2):1131, February 2022.

[11] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders, 2019.

[12] Yun He, Ziwei Zhu, Yin Zhang, Qin Chen, and James Caverlee. Infusing disease knowledge into bert for health question answering, medical inference and disease name recognition, 2020.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[14] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471, October 2000.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[16] Madhumita Sushil, Simon Suster, and Walter Daelemans. Are we there yet? exploring clinical domain knowledge of BERT models. In *Proceedings of the 20th Workshop on Biomedical Language Processing*. Association for Computational Linguistics, 2021.

[17] Yikuan Li, Shishir Rao, José Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. BEHRT: Transformer for electronic health records. *Scientific Reports*, 10(1), April 2020.

[18] Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. Pre-training of graph augmented transformers for medication recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, August 2019.

[19] Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *npj Digital Medicine*, 4(1), May 2021.

[20] Kenneth Chi-Yin Wong, Yong Xiang, Liangying Yin, and Hon-Cheong So. Uncovering clinical risk factors and predicting severe COVID-19 cases using UK biobank data: Machine learning approach. *JMIR Public Health and Surveillance*, 7(9):e29544, September 2021.

[21] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[22] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, March 1994.

[23] Emily Herrett, Arlene M Gallagher, Krishnan Bhaskaran, Harriet Forbes, Rohini Mathur, Tjeerd van Staa, and Liam Smeeth. Data resource profile: Clinical practice research datalink (CPRD). *International Journal of Epidemiology*, 44(3):827–836, June 2015.

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

[25] Beatriz Poblador-Plou, Marjan van den Akker, Rein Vos, Amaia Calderón-Larrañaga, Job Metsemakers, and Alexandra Prados-Torres. Similar multimorbidity patterns in primary care patients from two european regions: Results of a factor analysis. *PLOS ONE*, 9(6):1–14, 06 2014.

[26] Jayawant N. Mandrekar. Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, 5(9):1315–1316, September 2010.

[27] Edward Choi, Mohammad Taha Bahadori, Joshua A. Kulas, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism, 2016.

[28] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M. Dai, Nissan Hajaj, Michaela Hardt, Peter J. Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, Patrik Sundberg, Hector Yee, Kun Zhang, Yi Zhang, Gerardo Flores, Gavin E. Duggan, Jamie Irvine, Quoc Le, Kurt Litsch, Alexander Mossin, Justin Tansuwan, De Wang, James Wexler, Jimbo Wilson, Dana Ludwig, Samuel L. Volchenboum, Katherine Chou, Michael Pearson, Srinivasan Madabushi, Nigam H. Shah, Atul J. Butte, Michael D. Howell, Claire Cui, Greg S. Corrado, and Jeffrey Dean. Scalable and accurate deep learning with electronic health records. *npj Digital Medicine*, 1(1), May 2018.

[29] Bin Wang, Angela Wang, Fenxiao Chen, Yuncheng Wang, and C.-C. Jay Kuo. Evaluating word embedding models: methods and experimental results. *APSIPA Transactions on Signal and Information Processing*, 8(1), 2019.

[30] Anna C. Belkina, Christopher O. Ciccolella, Rina Anno, Richard Halpert, Josef Spidlen, and Jennifer E. Snyder-Cappione. Automated optimized parameters for t-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nature Communications*, 10(1), November 2019.

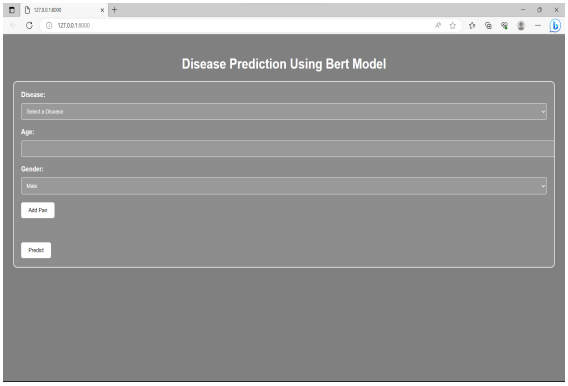
[31] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

A Masked Language Model (MLM)

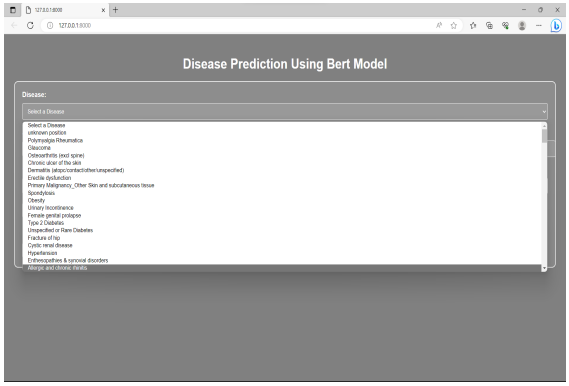
In typical deep learning or natural language processing (NLP) tasks, training is commonly conducted by predicting the final sequence of data given the preceding input data. However, in the context of Masked Language Modeling (MLM), a distinctive approach is employed, wherein certain tokens within the input data are deliberately masked, and the model is tasked with predicting these masked tokens. The position of the masked tokens can be anywhere within the input string, making the task more challenging and requiring the model to comprehend the context and dependencies of neighboring tokens. In MLM, approximately 15% of the input string is subjected to masking, which encourages the model to rely on contextual cues from the surrounding tokens to accurately predict the masked token. By considering tokens both to the left and right of the masked token, the model gains a more comprehensive understanding of the surrounding context, resulting in improved performance and enhanced representation capabilities. This approach has demonstrated notable advantages in various NLP applications, such as language understanding and generation tasks, as it encourages the model to learn more robust and contextually informed representations of language, leading to more accurate predictions and overall higher performance levels.

B Deployment Details

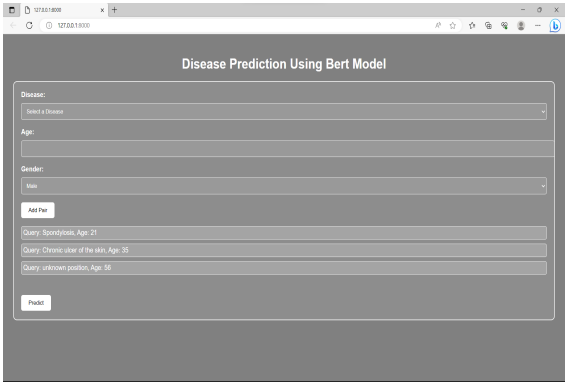
The following figures outline the sequence of deployment steps. Initially, the model is deployed on a local server. Upon access, users are directed to a webpage. On this webpage, users input their diagnosis history and age as paired data, which is temporarily stored in the frontend environment.Users can utilize a drop-down menu to select a diagnosis, considering that the model's training is specific to a predetermined set of diagnoses.Additionally, users provide their gender through radio buttons, as it remains fixed. These interactions are depicted in stages 2 and 3. Once users press the "Predict" button, the entered details are transmitted to the backend. Subsequently, the data undergoes pre-processing before being fed into the model. The output generated by the model is then shown, as illustrated in stage 4.



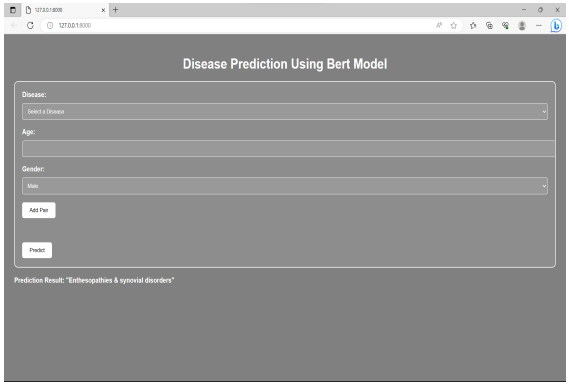
(a) Stage 1



(b) Stage 2



(c) Stage 3



(d) Stage 4

Figure 8: Image showing the steps involved in prediction using a web application

C Model 2 Summary

Layer (type)	Output Shape	Param #	Connected to
age_feature (InputLayer)	[(None, 985)]	0	[]
input_ids (InputLayer)	[(None, 985)]	0	[]
token_type_ids (InputLayer)	[(None, 985)]	0	[]
gender_ids (InputLayer)	[(None, 985)]	0	[]
age_embedding (Embedding)	(None, 985, 200)	24000	['age_feature[0][0]']
PositionalEmbedding/word (PositionalEmbedding)	(None, 985, 200)	42000	['input_ids[0][0]']
segment_embedding (Embeddi ng)	(None, 985, 200)	197000	['token_type_ids[0][0]']
gender_embedding (Embeddin g)	(None, 985, 200)	400	['gender_ids[0][0]']
add_layer (Add)	(None, 985, 200)	0	[]
input_1 (InputLayer)	[(None, 985, 1)]	0	[]
transformer_encoder_block (TransformerEncoderBlock)	(None, 985, 200)	480994	['add_layer[1][0]', 'input_1[0][0]']
transformer_encoder_block_ 1 (TransformerEncoderBlock)	(None, 985, 200)	480994	['transformer_encoder_block[1] [0]', 'input_1[0][0]']
transformer_encoder_block_ 2 (TransformerEncoderBlock)	(None, 985, 200)	480994	['transformer_encoder_block_1[1][0]', 'input_1[0][0]']
transformer_encoder_block_ 3 (TransformerEncoderBlock)	(None, 985, 200)	480994	['transformer_encoder_block_2[1][0]', 'input_1[0][0]']
transformer_encoder_block_ 4 (TransformerEncoderBlock)	(None, 985, 200)	480994	['transformer_encoder_block_3[1][0]', 'input_1[0][0]']
transformer_encoder_block_ 5 (TransformerEncoderBlock)	(None, 985, 200)	480994	['transformer_encoder_block_4[1][0]', 'input_1[0][0]']
masked_lm (MaskedLanguageM odel)	(None, 210)	42210	['transformer_encoder_block_5[1][0]', 'input_ids[0][0]']

Total params: 3191574 (12.17 MB) Trainable params: 3191574 (12.17 MB)
Non-trainable params: 0 (0.00 Byte)

D Model 3 Summary and Loss Plot

D.1 Model Details

Layer (type)	Output Shape	Param #	Connected to
age_feature (InputLayer)	[(None, 985)]	0	[]
input_ids (InputLayer)	[(None, 985)]	0	[]
token_type_ids (InputLayer)	[(None, 985)]	0	[]
age_embedding (Embedding)	(None, 985, 288)	34560	['age_feature[0][0]']
PositionalEmbedding/word (PositionalEmbedding)	(None, 985, 288)	60480	['input_ids[0][0]']
segment_embedding (Embeddi ng)	(None, 985, 288)	283680	['token_type_ids[0][0]']
gender_ids (InputLayer)	[(None, 985)]	0	[]
add_layer (Add)	(None, 985, 288)	0	['age_embedding[1][0]', 'PositionalEmbedding/word[1][0]', 'segment_embedding[1][0]']
gender_embedding (Embeddin g)	(None, 985, 1)	2	['gender_ids[0][0]']
con_layer (Concatenate)	(None, 985, 289)	0	['add_layer[1][0]', 'gender_embedding[1][0]']
input_4 (InputLayer)	[(None, 985, 1)]	0	[]
transformer_encoder_block_ 36 (TransformerEncoderBloc k)	(None, 985, 289)	1004850	['con_layer[1][0]', 'input_4[0][0]']
transformer_encoder_block_ 37 (TransformerEncoderBloc k)	(None, 985, 289)	1004850	['transformer_encoder_block_36 [1][0]', 'input_4[0][0]']
transformer_encoder_block_ 38 (TransformerEncoderBloc k)	(None, 985, 289)	1004850	['transformer_encoder_block_37 [1][0]', 'input_4[0][0]']
transformer_encoder_block_ 39 (TransformerEncoderBloc k)	(None, 985, 289)	1004850	['transformer_encoder_block_38 [1][0]', 'input_4[0][0]']
transformer_encoder_block_ 40 (TransformerEncoderBloc k)	(None, 985, 289)	1004850	['transformer_encoder_block_39 [1][0]', 'input_4[0][0]']
transformer_encoder_block_ 41 (TransformerEncoderBloc k)	(None, 985, 289)	1004850	['transformer_encoder_block_40 [1][0]', 'input_4[0][0]']

```
transformer_encoder_block_  (None, 985, 289)            1004850   ['transformer_encoder_block_41
42 (TransformerEncoderBloc                                        [1][0]',
k)                                                                 'input_4[0][0]']

transformer_encoder_block_  (None, 985, 289)            1004850   ['transformer_encoder_block_42
43 (TransformerEncoderBloc                                        [1][0]',
k)                                                                 'input_4[0][0]']

transformer_encoder_block_  (None, 985, 289)            1004850   ['transformer_encoder_block_43
44 (TransformerEncoderBloc                                        [1][0]',
k)                                                                 'input_4[0][0]']

transformer_encoder_block_  (None, 985, 289)            1004850   ['transformer_encoder_block_44
45 (TransformerEncoderBloc                                        [1][0]',
k)                                                                 'input_4[0][0]']

transformer_encoder_block_  (None, 985, 289)            1004850   ['transformer_encoder_block_45
46 (TransformerEncoderBloc                                        [1][0]',
k)                                                                 'input_4[0][0]']

transformer_encoder_block_  (None, 985, 289)            1004850   ['transformer_encoder_block_46
47 (TransformerEncoderBloc                                        [1][0]',
k)                                                                 'input_4[0][0]']

masked_lm (MaskedLanguageM  (None, 210)                 420210    ['transformer_encoder_block_47
odel)                                                             [1][0]',
                                                                   'input_ids[0][0]']

==================================================================================================
Total params: 12857132 (49.05 MB)
Trainable params: 12857132 (49.05 MB)
Non-trainable params: 0 (0.00 Byte)
```

D.2 Plot of Loss Over Epochs

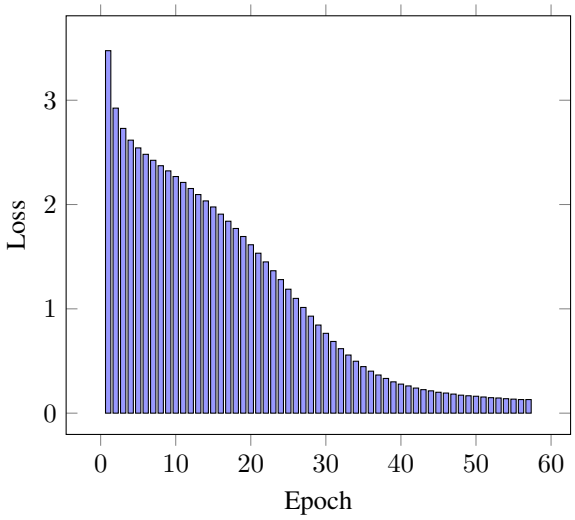


Figure 9: The figure shows that the loss decrease as the epoch increase for the model 3, which shows the model is able fit to the data.

E Model 4 Summary

Layer (type)	Output Shape	Param #	Connected to
age_feature (InputLayer)	[(None, 987)]	0	[]
input_ids (InputLayer)	[(None, 987)]	0	[]
token_type_ids (InputLayer)	[(None, 987)]	0	[]
age_embedding (Embedding)	(None, 987, 288)	34560	['age_feature[0][0]']
PositionalEmbedding/word (PositionalEmbedding)	(None, 987, 288)	60480	['input_ids[0][0]']
segment_embedding (Embedding)	(None, 987, 288)	284256	['token_type_ids[0][0]']
add_layer (Add)	(None, 987, 288)	0	['age_embedding[1][0]',
input_1 (InputLayer)	[(None, 987, 1)]	0	[]
transformer_encoder_block (TransformerEncoderBlock)	(None, 987, 288)	999072	['add_layer[1][0]', 'input_1[0][0]']
transformer_encoder_block_1 (TransformerEncoderBlock)	(None, 987, 288)	999072	['transformer_encoder_block[1][0]', 'input_1[0][0]']
transformer_encoder_block_2 (TransformerEncoderBlock)	(None, 987, 288)	999072	['transformer_encoder_block_1[1][0]', 'input_1[0][0]']
transformer_encoder_block_3 (TransformerEncoderBlock)	(None, 987, 288)	999072	['transformer_encoder_block_2[1][0]', 'input_1[0][0]']
transformer_encoder_block_4 (TransformerEncoderBlock)	(None, 987, 288)	999072	['transformer_encoder_block_3[1][0]', 'input_1[0][0]']
transformer_encoder_block_5 (TransformerEncoderBlock)	(None, 987, 288)	999072	[]
transformer_encoder_block_6 (TransformerEncoderBlock)	(None, 987, 288)	999072	['transformer_encoder_block_5[1][0]', 'input_1[0][0]']
transformer_encoder_block_7 (TransformerEncoderBlock)	(None, 987, 288)	999072	['transformer_encoder_block_6[1][0]', 'input_1[0][0]']
masked_lm (MaskedLanguageModel)	(None, 210)	419370	['transformer_encoder_block_7[1][0]',

Total params: 8791242 (33.54 MB)
Trainable params: 8791242 (33.54 MB)
Non-trainable params: 0 (0.00 Byte)

F Model 3 and 4 comparison

The table provided in Table 6 outlines the reasons behind the comparatively weaker performance of model 4 in comparison to model 3, considering traditional evaluation measures. If the value in the "difference in frequency" column is negative, it indicates a higher frequency of token masking for a particular diagnosis, under model 4 due to its modified masking approach. Notably, the table highlights elevated F1 scores for these diagnoses, suggesting better prediction capability. Also, model 4 successfully predicts some diagnoses that model 3 struggles to predict. The higher Overall F1 score for model 3 arises from assigning greater weight to its predictions, particularly driven by imbalanced classes. Addressing this imbalance is challenging due to the complexity of generating randomized medical data. However, refining masking strategies offers promise for performance enhancement.

Tokens	diagnosis	F1 score Model 3	F1 score Model 4	Difference in F1 scores	Frequency	New Frequency	Difference in Frequency
8	skin ulcer	0.6	0.67	-0.07	2314	5814	-3500
18	fracture hip	0.06	0.11	-0.05	157	1066	-909
19	cystic renal	0	0.25	-0.25	68	429	-361
20	hypertension	0.54	0.66	-0.12	36039	676	35363
27	CKD	0.16	0.16	0	3065	3581	-516
37	CCD	0	0.03	-0.03	204	1385	-1181
38	periph neuro	0.07	0.08	-0.01	1091	4748	-3657
39	diab eye	0.29	0.46	-0.17	3181	7612	-4431
53	folate deficiency	0	0.05	-0.05	138	1060	-922
55	Subarach	0.07	0.08	-0.01	114	749	-635
56	subdural haem	0	0.21	-0.21	38	244	-206
60	diab neuro	0	0.24	-0.24	278	1993	-1715
61	NAFLD NASH	0	0.09	-0.09	515	3116	-2601
62	spinal stenosis	0.03	0.12	-0.09	309	1936	-1627
63	type 1 diabetes	0.33	0.68	-0.35	755	3100	-2345
69	pri kidney	0.19	0.27	-0.08	121	696	-575
72	seb derm	0.06	0.11	-0.05	2121	5809	-3688
74	anxiety phobia	0.5	0.53	-0.03	9765	2002	7763
76	barretts	0.17	0.32	-0.15	536	2587	-2051
77	pri oesoph	0.24	0.26	-0.02	80	429	-349
78	perip arterial disease	0.18	0.35	-0.17	1050	4216	-3166
79	ulcer peptic	0.19	0.21	-0.02	965	4227	-3262
84	PD	0.12	0.22	-0.1	272	1589	-1317
90	dysmenorrhoea	0.05	0.16	-0.11	590	3259	-2669
92	neuro bladder	0.11	0.11	0	750	3875	-3125
95	autonomic neuro	0	0.09	-0.09	172	1299	-1127
102	asbestosis	0	0.04	-0.04	61	401	-340
105	learning disability	0.33	0.34	-0.01	105	506	-401
111	hyper nasal turbs	0	0.05	-0.05	216	1533	-1317
113	blindness	0	0.09	-0.09	304	1824	-1520
114	macula degen	0	0.12	-0.12	271	1599	-1328
115	hyperprolactinaemia	0.18	0.21	-0.03	85	402	-317
117	sjogren	0.06	0.22	-0.16	110	775	-665
118	resp failure	0	0	0	26	176	-150
119	sarcoid	0.2	0.23	-0.03	185	883	-698
120	pri bladder	0.09	0.13	-0.04	200	1107	-907
121	sleep apnoea	0.06	0.12	-0.06	668	3571	-2903
123	eating disorder	0.1	0.29	-0.19	189	984	-795
124	hiv	0.17	0.29	-0.12	38	253	-215
125	fibromyalgia	0.24	0.27	-0.03	700	3143	-2443
126	PSA	0.45	0.82	-0.37	334	1606	-1272
131	retinal vasc occl	0.04	0.13	-0.09	250	1415	-1165
132	PCOS	0.11	0.12	-0.01	84	597	-513
134	pri thyroid	0	0.25	-0.25	54	326	-272

135	sec metastasis	0.12	0.16	-0.04	323	1402	-1079
136	pri liver	0	0.08	-0.08	19	139	-120
139	spondylolisthesis	0	0.03	-0.03	108	777	-669
140	ptosis	0.15	0.15	0	180	1112	-932
141	congenital cardiac	0	0.09	-0.09	133	800	-667
144	vitiligo	0	0.18	-0.18	132	812	-680
146	TB	0	0.16	-0.16	206	1365	-1159
150	aplastic	0.13	0.32	-0.19	46	222	-176
151	MDS	0	0.06	-0.06	13	142	-129
152	hyperPTH	0.11	0.15	-0.04	205	1311	-1106
153	ank spond	0.35	0.43	-0.08	186	834	-648
155	Rh valve	0	0.5	-0.5	56	380	-324
157	sec polycythaemia	0	0.11	-0.11	17	154	-137
158	cirrhosis	0.14	0.31	-0.17	217	1010	-793
161	pulmonary hypertension	0	0.2	-0.2	19	159	-140
163	pri testicular	0	0.29	-0.29	36	252	-216
165	scoliosis	0	0.11	-0.11	116	961	-845
167	varices portal hypert	0	0.26	-0.26	64	448	-384
168	myasthenia	0	0.44	-0.44	78	366	-288
169	obstr reflux	0	0.06	-0.06	59	486	-427
171	SLE	0.22	0.26	-0.04	173	749	-576
172	chr cystitis	0	0.18	-0.18	72	455	-383
173	Intracereb haem	0	0.11	-0.11	65	413	-348
176	pri stomach	0.15	0.15	0	41	267	-226
177	adrenal insufficiency	0.24	0.47	-0.23	48	272	-224
179	sickle cell		0.9	-0.9	3	39	-36
180	autoimm liver	0.27	0.4	-0.13	114	540	-426
181	sys sclerosis	0.5	0.68	-0.18	53	276	-223
182	hocm	0	0.6	-0.6	40	265	-225
183	immunodef	0	0	0	16	106	-90
184	hodgkins	0.32	0.34	-0.02	46	283	-237
185	pri biliary	0	0	0	15	107	-92
186	intracranial htn	0	0	0	11	83	-72
189	spina bifida	0	0	0	64	295	-231
190	oth haem anaemia	0	0.13	-0.13	41	216	-175
191	collapsed vert	0	0.23	-0.23	90	534	-444
192	pri bone	0	0.07	-0.07	22	213	-191
193	MND	0.2	0.49	-0.29	27	148	-121
194	crohns	0.42	0.47	-0.05	443	1772	-1329
195	thrombophilia	0.08	0.14	-0.06	116	756	-640
196	pri uterine	0.07	0.12	-0.05	76	588	-512
197	cystic fibrosis		0	0	10	91	-81
198	autism	0	0.22	-0.22	17	97	-80
199	gastro angiodysplasia		0.06	-0.06	14	121	-107
200	cerebral palsy		0.25	-0.25	15	79	-64
201	pri mesothelioma	0	0.44	-0.44	22	129	-107
203	thala	0	0.07	-0.07	16	108	-92
204	sick sinus	0	0	0	28	156	-128
205	ADHD	0	0	0	15	72	-57
206	juv arth		0	0	4	37	-33
207	downs	0	0.4	-0.4	4	22	-18
208	entero arthro		0	0	5	29	-24
209	Sars Covid	0	0	0	0	1	-1

Table 6: Table showing comparison between model 3 and model 4

G Embedding Diagram

The figure 10 represent how the model understand the data, and by observing the figure, various insights can be revealed such as how the diagnose related to each other. The spatial distribution across a 289-dimensional space significantly influence subsequent diagnosis predictions. While it can indeed reveal significant insights, comprehending the pattern needs a certain degree of medical expertise.

Figure 10: Embedding Diagram generated from model 3