User Heterogeneous Health Information Fusion for Medication Recommendation

Anonymous Author(s)

ABSTRACT

Medication recommendation is an important application of AI healthcare, which provides personalized advice for doctors based on patients' health conditions. Existing studies have shown the effectiveness of medication recommendation algorithms, while how to combine the heterogeneous features of patients to conduct recommendations is still an unsolved challenge: 1) little effort is made by previous studies in jointly modeling heterogeneous features of patients; hence only limited types of information could be utilized; 2) previous studies exclusively use patients' diagnoses and procedures, ignoring some personalized features, e.g., static personal information and laboratory test results. In this work, we propose MedicalBERT, a pre-training and fine-tuning based method to fuse heterogeneous features for better patient representations for medication recommendation. Not only patient diagnoses and procedures, but also static and laboratory data are used in our framework. First, we design an elaborate strategy to convert heterogeneous medical records into sequences for pre-training, which can be easily extended to any other type of feature. Then, we propose several schemes for efficient pre-training. Experimental results show that MedicalBERT outperforms state-of-the-art baselines significantly even with only a linear prediction layer. Besides, the ablation study demonstrates that the newly introduced static and laboratory data do contribute to better patient representations.

KEYWORDS

Medication Recommendation, Heterogeneous Health Information, Pre-training.

ACM Reference Format:

Anonymous Author(s). 2022. User Heterogeneous Health Information Fusion for Medication Recommendation. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Healthcare is one of the most promising areas for deep learning applications. With massive patients' electronic health records (EHR) data, deep learning has achieved initial success in assisting clinical decision-making [1, 6, 17, 30, 34]. Medication recommendation algorithms, aiming to help doctors (rather than patients) improve the efficiency and accuracy of prescribing, have drawn growing research interest these years [5, 21, 32, 37].

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

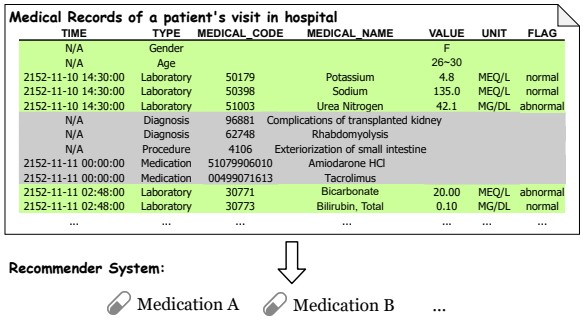


Figure 1: An example of complex heterogeneous EHR data for recommending medications to doctors. Grey records are commonly used in previous works. Green records are introduced new types of patient features. The content of the records is modified for privacy protection.

Unlike recommendation studies in other scenarios (e.g., movie or e-commerce) [23, 24] where sequential user interaction history is enough to model their needs, medication recommendation is more challenging due to the importance of precisely modeling patients and the complex structure of patients' heterogeneous EHR data. As shown in Figure 1, EHR data typically consist of records of different types of medical entities of a patient, and the medication recommendation task is to predict the medication combination given the heterogeneous EHR data. So how to make full use of the heterogeneous EHR data is vital for medication recommendation.

However, most existing works focus on modeling sequential features only using patients' diagnosis and procedure history in EHR data [5, 16, 21, 31]. Despite various efforts on modeling patients for medication recommendation and their improvements, current methods still suffer from the following limitations:

- **Inability to flexibly model heterogeneous medical data.** Previous works often design one specific module for each type of EHR data to learn their representations and then introduce complex interaction modules [12, 16] to get the final patient representation. In spite of satisfactory performance, these models are limited to particular parts of EHR data, unable to be expanded easily to tap into other side information.
- **Lack of utilization of static and laboratory data.** Diagnoses and procedures of patients are the only two types of features utilized in most existing medication recommendation studies. Limited by their choice of features, these methods are more like modeling the correlation between diseases and medications, so they may fail to achieve personalized recommendations. Hence, we argue that static personal information (such as age, gender, etc.) and laboratory test results of patients should be taken into consideration to achieve better performances.

In this work, we propose to apply pre-training approaches to learn representations of medical entities with fused data types, where static personal information and laboratory test results are also included. Pre-training techniques have achieved great success in natural language processing (NLP) [3, 9]. Especially, multilingual pre-trained models [7, 38] have demonstrated their power to combine information from different sources for unified modeling. Inspired by the success of pre-training approaches in NLP, we experiment with applying pre-training techniques to medical data to learn patient representations uniformly. It provides a simple but effective way of incorporating new types of EHR data into medical representation learning, i.e., static and laboratory data. Note that different from previous recommendation methods that adopt transformer [12] in modeling user sequential history, patients' heterogeneous features are carefully designed to be utilized here.

In preparation for pre-training, we devise an elaborate strategy to convert structured heterogeneous EHR data into sequences. In particular, we design one specialized approach for each type of EHR data to convert them into tokens. Then we propose multiple strategies to concatenate tokens into sequences. It is worth noting that we can easily incorporate new types of EHR data into our framework by only devising a strategy for converting them to tokens. We follow the mask-predict scheme [9] in our pre-trained models for pre-training. After pre-training, we fine-tune the pre-trained model on the medication recommendation task. Extensive experiments and analyses are conducted to verify the effectiveness of our proposed method.

The main contributions of this study are summarized as follows:

- We devise MedicalBERT, a new pre-training framework to integrate heterogeneous patient information, which can generate better patient representations with various types of features for medication recommendations.
- We introduce static features and laboratory data, which are often overlooked in previous works, into heterogeneous information for medication recommendations. Schemes for pre-training are carefully designed to improve the patient representation results for recommendations.
- Experiments on a real-world public dataset show that the proposed framework significantly outperforms state-of-the-art methods with different prediction structures. Further analyses also demonstrate the success of introducing pre-training methods and taking heterogeneous information into consideration.

2 RELATED WORKS

2.1 Medication Recommendation

Existing medication recommendation studies can be roughly categorized into two types [21, 32], instance-based and longitudinal-based methods.

Instance-based methods [10, 26, 37] focus on features from the current visit of the patient. Zhang et al. [37] formulate the medication recommendation task as a multi-label multi-instance learning problem and propose an attention-based sequence-to-sequence model. Gong et al. [10] utilize graph embedding techniques and convert the medication recommendation task into a link-prediction problem on medical knowledge graphs.

On the other hand, longitudinal-based methods [2, 5, 16, 21, 31, 32] leverage the temporal dependencies of the patients' historical visits. Choi et al. [5] propose an interpretable two-level neural attention model to predict medical events. Le et al. [16] present a memory augmented neural network to handle asynchronous two-view sequential inputs in the medication recommendation task. An et al. [2] devise a dual adaptive LSTM along with an attentive meta-learning network to fuse heterogeneous EHR data to predict treatment medicines. Some recent longitudinal-based studies also propose to introduce extra drug-drug interaction knowledge [21, 32] into the recommendation system and achieve state-of-the-art performances. Moreover, several methods also adopt new neural models in this task, e.g., reinforcement learning [27], BERT [20], and GAN [28].

However, all these studies rely on patients' diagnosis and procedure features, where static and laboratory information is either neglected or leveraged insufficiently. We want to argue that these personal features are crucial for the effectiveness of patient profiling and the downstream medication recommendation task. Besides, previous studies often adopt LSTM and attention methods to utilize sequential medical features, which may fail in modeling heterogeneous features. In this study, we propose a new pre-training framework to cope with this challenge.

2.2 Pre-training for Recommendation

Pre-training techniques have been verified to be helpful in various natural language processing tasks [3, 8, 9, 38], so some recent works also try to apply them in recommendation scenarios.

On the one hand, pre-training based models are applied in sequential recommendation methods to model user interaction sequences. For instance, Sun et al. [24] devise a deep bidirectional model BERT4Rec and Chen et al. [4] pre-train the BERT as the session encoder to capture the bidirectional correlations within sessions. Qiu et al. [19] propose U-BERT to leverage content-rich information such as reviews in pre-training. On the other hand, pre-training is adopted to represent textual features, especially in news recommendations. Zhang et al. [36] propose UNBERT to enhance the textual representation of news and capture user-news matching signals at word-level and news-level. Wu et al. [29] leverage pre-trained language models (PLM) to model news and show the effectiveness of PLM. Several recent works also attempt to adapt PLM to achieve effective knowledge transfer [35].

Despite the improvements made by these studies, we find that they model only traditional sequential features in recommendation using pre-training, e.g., interaction history and textual content. In this study, we propose to model the heterogeneous features of patients in medication recommendations.

Note that previous recommendation studies use other strategies to fuse heterogeneous information, e.g., co-attention mechanism [13], collaborative filtering [22], and graph neural networks [33]. This is the first time that pre-training has been adopted to integrate heterogeneous information for the recommendation. Besides, due to the success that pre-training methods achieved in other tasks with heterogeneous information, e.g., multi-modal [23], cross-lingual tasks [8, 14], we believe that pre-training will contribute a lot for heterogeneous information fusion in recommendations.

3 PRELIMINARY

3.1 Task Definition

The goal of the medication recommendation task is to predict a combination of medications given heterogeneous patient features. Let $\mathbf{X}_i{}^{(t)}$ denotes the complete EHR of patient i's t-th visit in the hospital and $\mathbf{X}_\mathbf{i}'{}^{(t)}$ means $\mathbf{X}_i{}^{(t)}$ without medication records. Our task can be defined formally:

Given the t-th visit of the patient i with her/his EHR data $\mathbf{X}_\mathbf{i}'{}^{(t)}$ and patient history $\mathbf{X}_i{}^{(1)}, \cdots, \mathbf{X}_i{}^{(t-1)}$, we want to generate a medication recommendation result $\hat{\mathbf{y}}_\mathbf{i}^\mathbf{t} \in \{0,1\}^{|M|}$ for her/his doctors, where M is the candidate medication set in the task and $|\cdot|$ denotes the size of the set.

Note that the recommendation results are shown to doctors but not patients for the consideration of safety.

3.2 Structure of EHR Data

EHR contain content-rich records produced during each patient's hospital visit, which are composed of different types of medical codes. We can represent EHR of patient i as $\mathcal{X}_\mathbf{i} = \{\mathbf{X}_i^{(1)}, \cdots, \mathbf{X}_i^{(v_i)}\}$, where v_i is the number of visits of the i-th patient. Each $\mathbf{X}_i{}^{(t)}$ contains heterogeneous medical records within the visit.

In this work, user heterogeneous EHR of a visit can be written as $\mathbf{X}_i^{(t)} = \{\mathbf{s}_i^{(t)}, \mathbf{d}_i^{(t)}, \mathbf{p}_i^{(t)}, \mathbf{m}_i^{(t)}, \mathbf{l}_i^{(t)}\}$, where $\mathbf{s}_i^{(t)}, \mathbf{d}_i^{(t)}, \mathbf{p}_i^{(t)}, \mathbf{m}_i^{(t)}, \mathbf{l}_i^{(t)}$ are the static personal information, diagnoses, procedures, medications, and laboratory test results of t-th visit of the patient i, respectively. An example of EHR is shown in Figure 1. The record of static information contains the invariable information of the patient, such as age and gender; The records of diagnosis and procedure contain the corresponding medical codes; The record of medication contains its prescription time and the medical code of the drug. The record of lab test results typically contains the test time, the code of the lab item, the test value, the unit of the value, and a string indicating whether the value is in the normal range.

3.3 Pre-training Techniques

Pre-training techniques have achieved great success in modeling fused information in multiple languages [8, 38]. We believe that pre-training techniques are also suitable for fusing heterogeneous medical information. On the one hand, pre-training techniques eliminate the need to design separate modules and supervision signals for each type of information. They enable unified modeling of any number of different information types. On the other hand, the multi-head attention used in pre-training techniques can sufficiently model the dependencies among different types of information. There are two main paradigms for pre-training, namely auto-regressive training and mask prediction. Since bidirectional modeling is more conducive to the interaction of heterogeneous information, we choose the mask prediction method in our framework. For clarity, we briefly describe it here.

BERT [9] is the representative model of mask prediction, which is composed of multiple Transformer [25] layers. The pre-training of BERT consists of two self-supervised training tasks:

- **Masked Language Modeling (MLM)**: MLM first randomly selects tokens in a sequence to be masked, and then in the training phase, the model is required to restore the masked word given its context. After training with MLM, the model can learn powerful representations of tokens.
- **Next Sentence Prediction (NSP)**: NSP is a binary task aiming to predict whether two sentences are adjacent. It is designed to help the model capture the relation of sequences, as many downstream tasks require an understanding of the sequence relations.

Hence, a typical input of the mask prediction tasks is as follows:

Input = `[CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]`
Label = `IsNext`

Where [CLS] is a special token used for classification at the beginning of each sequence; [SEP] is used to denote sentence boundaries; The two tasks of the pre-training are to predict the words replaced by [MASK] and predict the label.

Pre-training techniques for mask prediction have matured, but applying them to EHR still faces some challenges, e.g., how to define the input format and the supervision signals to fit the characteristics of medical data, how to utilize them in recommendations, etc.

4 FRAMEWORK

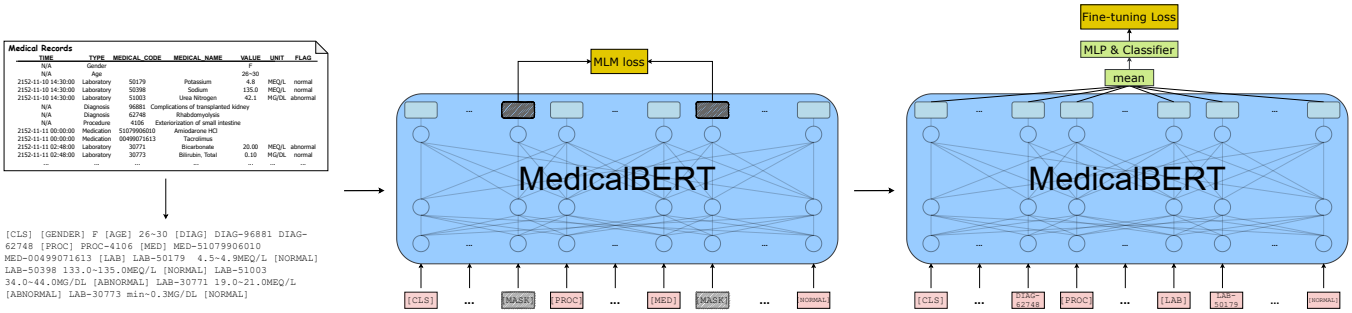
In this section, we introduce the details of the proposed MedicalBERT for modeling heterogeneous information for medication recommendations. As shown in Figure 2, the framework can be divided into three modules: (1) **Data preprocessing module**: Convert medical records to sequences; (2) **Pre-training module**: Pre-train patient representations with self-supervised tasks; (3) **Fine-tuning module**: Adopt the pre-trained user representations on the medication recommendation task. We will introduce them one by one.

4.1 Data Preprocessing Module

Hospitals are accumulating vast amounts of EHR data every day. In this work, We adopt a public EHR dataset MIMIC-III [15] for pre-training. However, as introduced in Section 3.2, the EHR data from MIMIC-III are heterogeneous. Since contemporary pre-training techniques are designed for sequential data, converting EHR data to sequences is an inevitable step. And it is also a very critical step in our framework, because many factors in generating the sequence, such as the length of the sequence, the correlation and arrangement of the tokens, could seriously affect the effect of pre-training. We consider several types of EHR here, including static, diagnosis, procedure, medication, and laboratory records. A two-steps conversion is carefully designed to transform these heterogeneous features into sequences for the mask-predict pre-training:

I. Convert heterogeneous records to tokens

Each type of EHR is converted into tokens through a carefully designed strategy. For medical records of static information, medications, diagnoses, and procedures, each record is converted to one token. We omit the timestamp since the absolute timestamp is not valuable and the time order can be preserved by order of tokens. For static information, we extract 11 kinds of records from the MIMIC-III dataset, namely gender, age, ethnicity, marital status, religion, language, insurance type, admission location, admission type, discharge location, and death status. Note that discharge location and death status are not used in the fine-tuning stage. We



Stage 1: Data Preprocessing Stage 2: Pre-training Stage 3: Fine-tuning

Figure 2: Framework Overview. We first convert medical records into sequences of tokens, randomly mask some tokens, then feed the result sequence to a BERT encoder for pre-training. Finally, we fine-tune the pre-trained model for medication recommendations.

concatenate an abbreviation of the type and the medical code for records of medications, diagnoses, and procedures to form the final token to avoid conflict. For example,

Medical Records

TIME	TYPE	MEDICAL_CODE	MEDICAL_NAME	VALUE
N/A	Gender			F
N/A	Age			26~30
N/A	Diagnosis	96881	Complications of transplanted kidney	
N/A	Diagnosis	62748	Rhabdomyolysis	
N/A	Procedure	4106	Exteriorization of small intestine	
2152-11-11 00:00:00	Medication	51079906010	Amiodarone HCl	
2152-11-11 00:00:00	Medication	00499071613	Tacrolimus	
...	...	...	...	...

```
F  26~30  DIAG-96881  DIAG-62748  PROC-4106
MED-51079906010  MED-00499071613
```

For medical records of laboratory test results, each record is converted to three tokens. The first token represents the laboratory test item; the second token represents the value of the laboratory test; the third token indicates whether the test value is within normal range; typical examples are [NORMAL] or [ABNORMAL]. For enumerable features, such as the laboratory test items and flags denoting normal or abnormal here, we directly convert them into tokens. For non-enumerable features, we perform segmental discretization and convert them into tokens. For example, for test values here, we first collect all the values in all visits of patients for the particular kind of laboratory item. Then we sort the collected values and divide them into ten ranges, with each range containing approximately the same amount of values. We construct one token for each range. The values within one range are converted to a unified token. The value range token is composed of a minimum and a maximum of the range as well as the unit of the laboratory item. Note that we perform unit conversions to ensure that all values are in the same unit. For example,

Medical Records

TIME	TYPE	MEDICAL_CODE	MEDICAL_NAME	VALUE	UNIT	FLAG
2152-11-10 14:30:00	Laboratory	50179	Potassium	4.8	MEQ/L	normal
2152-11-10 14:30:00	Laboratory	50398	Sodium	135.0	MEQ/L	normal
2152-11-10 14:30:00	Laboratory	51003	Urea Nitrogen	42.1	MG/DL	abnormal
...	...	...	...	...	...	...

```
LAB-50179  4.5~4.9MEQ/L  [NORMAL]  LAB-50398
133.0~135.0MEQ/L  [NORMAL]  LAB-51003  34.0~44.0MG/DL
[ABNORMAL]
```

II. Organize tokens into sequences

Intuitively, records within one visit of a patient are related closely. We consider putting tokens from the same visit $\mathbf{X}_i^{(t)}$ into a single sequence and omit the historical visits $\mathbf{X}_i{}^{(1)}, \cdots, \mathbf{X}_i{}^{(t-1)}$ to avoid extra-long sequences. The key question is how to arrange the order of the tokens in pre-training. We add special type tokens denoting the following token types to make the sequence easier to understand, and propose two strategies here. For clarity, suppose that we have medical records of one patient given in Figure 1. The two strategies of generating sequences are as follows.

Chronological. Though we omit the information of the time when converting records to tokens, we can retain most of it by arranging the tokens in order of time. Records without time are placed at the beginning of the sequence. In detail, the sequence is arranged as follows:

```
[CLS] [GENDER] F [AGE] 26~30 [DIAG] DIAG-96881
DIAG-62748 [PROC] PROC-4106 [LAB] LAB-50179
4.5~4.9MEQ/L [NORMAL] LAB-50398 133.0~135.0MEQ/L
[NORMAL] LAB-51003 34.0~44.0MG/DL [ABNORMAL] [MED]
MED-51079906010 MED-00499071613 [LAB] LAB-30771
19.0~21.0MEQ/L [ABNORMAL] LAB-30773 min~0.3MG/DL
[NORMAL]
```

Aggregate. Tokens of the same type appearing in multiple places in the sequence can make modeling difficult. Regardless of the order of event time, we aggregate tokens of the same type to appear together in the final sequence. We maintain the order of time within the subsequence of a specific type. In this case, the sequence is organized as follows:

```
[CLS] [GENDER] F [AGE] 26~30 [DIAG] DIAG-96881
DIAG-62748 [PROC] PROC-4106 [MED] MED-51079906010
MED-00499071613 [LAB] LAB-50179  4.5~4.9MEQ/L
[NORMAL] LAB-50398 133.0~135.0MEQ/L [NORMAL]
LAB-51003  34.0~44.0MG/DL [ABNORMAL] LAB-30771
19.0~21.0MEQ/L [ABNORMAL] LAB-30773 min~0.3MG/DL
[NORMAL]
```

Another factor to consider is the choice of data. In practice, the converted sequence of a visit can still be sometimes very long, hence we propose only to use data from the first day of the visit to

avoid extra-long sequences. The number of items within one visit is significantly fewer when we only use first-day data (shown in Table 1), resulting in sequences of appropriate length in pre-training.

All strategies above can generate proper input data for pre-training, and in Section 5.2.2 we will show their experimental results.

4.2 Pre-training Module

After data preprocessing, we adopt the architecture of BERT [9] to construct our pre-training model for EHR. The process of utilizing medical records for pre-training is depicted in the middle picture of Figure 2. Specially, we omit the NSP task originally in BERT and only apply the MLM task. The main reason is that the EHR data contains massive data of single-visit patients, which the sequence-level task could not utilize. In addition, we choose to ignore patients' previous visits here to generate more pre-training data.

To conduct mask-prediction, a random sample of the input sequence tokens is selected and replaced by the special token [MASK]. Like BERT, we select 15% of the input sequence tokens for replacement, of which 80% are replaced by [MASK], 10% are left unchanged, and others are replaced by a random token within the same type. We do not choose special tokens ([CLS] token and tokens indicating types) to be masked. The objective of the MLM is a cross-entropy loss on predicting masked tokens.

However, the tokens generated from heterogeneous information are not the same as those in NLP. The masking strategies on different types of tokens also have a great impact on the pre-training performance. We devise two masking strategies here and evaluate them in Section 5.2.2:

Mask All. All tokens except for special tokens are taken into consideration to replace with [MASK]. This strategy assumes all tokens are equal and learns their representations equally.

Mask Part. In this strategy, we do not replace the laboratory test tokens and static information tokens with [MASK]. Intuitively, these tokens are relatively objective, and it is more valuable to predict the type of token that requires professionals to decide originally, such as diagnosis, procedure, and medication. On the other hand, the huge-amount laboratory test tokens tend to occupy most of the sequence, and replacing them with [MASK] may impair learning for other types of tokens.

4.3 Fine-tuning Module

We fine-tune our pre-trained model on the medication recommendation task as a downstream task to evaluate its performance. We convert the medical records to sequences as the input of the fine-tuning task using the same strategy as in the pre-training phase. Note that we also do not include historical visits of patients here. To obtain a fixed-dimensional representation of a patient's visit using the pre-trained model, we take the mean representation of the final hidden state of the model. We denote this vector as $\mathbf{c} \in \mathbb{R}^H$, where H is the size of the hidden state. Theoretically, complex decoders could be used to predict the medication combinations given $\mathbf{c}$, but in this work, we adopt a simple MLP and a linear classifier as the decoder to verify the effectiveness of our framework. The parameters of the MLP and the classifier are fine-tuned jointly with all parameters of the pre-trained model. The prediction

$$\hat{\mathbf{y}} = \mathbf{W} \cdot \mathrm{MLP}(\mathbf{c}). \tag{1}$$

Where $\mathbf{W}$ is the parameter matrix of the linear classifier. The prediction vector is of the same dimension as the number of drugs ($\hat{\mathbf{y}} \in \mathbb{R}^{|M|}$), where M is the medication set of the task. Hence, each dimension of this vector represents a score on a drug.

Loss Function. As a multi-label prediction task, two commonly used multi-label loss functions are combined to form our loss function on the fine-tuning task, namely, the binary cross-entropy loss $\mathcal{L}_{bce}$ and the multi-label margin loss $\mathcal{L}_{multi}$.

$$\begin{aligned} \mathcal{L}_{bce} &= -\sum_{i}^{|M|} y_i \log \sigma\left(\hat{y}_i\right) + \left(1 - y_i\right) \log\left(1 - \sigma\left(\hat{y}_i\right)\right) \\ \mathcal{L}_{multi} &= \sum_{i,j:y_i=1,y_j=0} \frac{\max(0, 1 - \left(\sigma\left(\hat{y}_i\right) - \sigma(\hat{y_j})\right)}{|M|} \end{aligned} \tag{2}$$

Where σ is the sigmoid function, $\hat{y}_i$ is the prediction vector, and the y_i is the ground truth vector.

The position of L_{multi} is to widen the margin between the predicted value of the true label and the predicted value of the wrong label as much as possible so that we can easily predict labels by setting a fixed threshold. We use a weighted sum of the two losses as the final loss to optimize.

$$\mathcal{L} = \alpha_{bce}\mathcal{L}_{bce} + \alpha_{multi}\mathcal{L}_{multi} \tag{3}$$

Where α_{bce} and α_{multi} are hyperparameters. In this work, we assign α_{bce} to 0.9 and α_{multi} to 0.02.

Inference recommendation results. In the inference phase, we fix the threshold values as 0.5 to predict labels in the label set:

$$\hat{Y}_i = \begin{cases} 1 & \text{if } \sigma(\hat{y_i}) > 0.5 \\ 0 & \text{if } \sigma(\hat{y_i}) \leq 0.5 \end{cases} \tag{4}$$

5 EXPERIMENTS

5.1 Experiments Setup

5.1.1 ***Data.*** We conduct experiments on the MIMIC-III dataset [15] like many previous studies [21, 32]. The MIMIC-III dataset contains data of patients staying in ICU in the Beth Israel Deaconess Medical Center from 2001 to 2012. It is composed of multiple types of EHR data. In this work, we adopt data types of static information, diagnoses, procedures, medications, and laboratory test results. We extract static information from PATIENTS and ADMISSIONS table, diagnoses from DIAGNOSES_ICD table, procedures from PROCEDURES_ICD table, medications from PRESCRIPTIONS table, and laboratory test results from LABEVENTS table of MIMIC-III dataset[1].

We pre-train the model with data of all patients. The statistics of them are shown in Table 1 with two selection strategies, where 'First Day' denotes only considering data on the first day of the visit, and 'All data' denotes using all available data in the visit.

To be consistent with previous works, we fine-tune with only multi-visit patient data. We follow the data-processing methods of

[1]Note that due to the data use agreement, we cannot open this dataset directly, but people can apply for it in https://physionet.org/content/mimiciii/1.4/

Table 1: Statistics of Pre-training Data. The diag., proc., and med. represent diagnosis items, procedure items, and medication items, respectively.

Items	Size	
	All Data	First Day
# of patients / # of visits	46520 / 58960	46520 / 58960
# of static items / # of lab items	199 / 173	199 / 173
# of diag. / # of proc.	6946 / 2009	6946 / 2009
# of med.	4203	3386
avg. / max # of visits	1.27 / 42	1.27 / 42
avg. / max # of diag. per visit	11.05 / 39	11.05 / 39
avg. / max # of proc. per visit	4.38 / 35	4.38 / 35
avg. / max # of med. per visit	50.67 / 898	19.48 / 156
avg. / max # of static items per visit	10.39 / 11	10.39 / 11
avg. / max # of lab items per visit	345.03 / 12337	48.90 / 544

Table 2: Statistics of Fine-tuning Data. The diag., proc., and med. represent diagnosis items, procedure items, and medication items, respectively.

Items	Size
# of patients / # of visits	15032 / 6430
# of diag. / # of proc. / # of med.	1958 / 1430 / 131
# of static items / # of lab items	145 / 86
avg. / max # of visits	2.37 / 29
avg. / max # of diag. per visit	13.63 / 39
avg. / max # of proc. per visit	4.54 / 32
avg. / max # of med. per visit	19.20 / 53
avg. / max # of static items per visit	9.00 / 9
avg. / max # of lab items per visit	14.03 / 40

SafeDrug [32]. We only keep the medications during the first 24 hours of a visit and convert the medication coding from NDC to ATC in the fine-tuning task. Note that NDC is maintained in the pre-training task to keep as much information as possible. Since we consider static and lab data in pre-training, these new types of data are also added into the fine-tuning tasks. The statistics of our processed data for the fine-tuning task are listed in Table 2.

5.1.2 ***Baselines.*** Following algorithms are adopted as baselines.

- **Nearest** recommends the same medications as the patient's last visit.
- **Logistic Regression (LR)** is logistic regression with L2 regularization, in which input data are represented as multi-hot vectors and the binary relevance is adopted [18].
- **DMNC [16]** predicts medication combinations through dual-channel memory augmented neural network based on differentiable neural computers (DNC) [11].
- **LEAP [37]** adopts a sequence-to-sequence paradigm to recommend medications, which turns the medication recommendation task into a generative task.
- **RETAIN [5]** is an RNN-based two-level neural attention model. It relies on the attention mechanism to detect and focus on influential variables in medical records.
- **G-BERT [20]** integrates representations generated with GNNs into two BERT-based encoders for pre-training to model diagnoses and medications features, respectively.
- **GAMENet [21]** predicts medication combinations through memory neural networks and graph convolutional networks based on history EHR data and Drug-Drug Interaction (DDI) data.
- **SafeDrug [32]** utilizes molecular-level medication information to enhance model knowledge of medications in the medication recommendation task.

5.1.3 ***Metrics.*** Following evaluation metrics are adopted.

- **The Jaccard Coefficient** (higher is better) is defined as the size of the intersection set divided by the size of the union set of ground truth medications $Y_i^{(j)}$ and the predicted medications $\hat{Y}_i^{(j)}$.

$$\text{Jaccard} = \frac{1}{\sum_i^N V_i} \sum_i^N \sum_j^{V_i} \left| Y_i^{(j)} \cap \hat{Y}_i^{(j)} \right| / \left| Y_i^{(j)} \cup \hat{Y}_i^{(j)} \right| \tag{5}$$

  Where N is the number of patients in the test set, V_i is the number of visits of i-th patient.

- **Precision-Recall Area Under Curve (PRAUC)** (higher is better) computes the area under the precision-recall curve.

$$\begin{aligned} Precision_i^{(j)} &= \left| Y_i^{(j)} \cap \hat{Y}_i^{(j)} \right| / \left| \hat{Y}_i^{(j)} \right| \\ Recall_i^{(j)} &= \left| Y_i^{(j)} \cap \hat{Y}_i^{(j)} \right| / \left| Y_i^{(j)} \right| \\ \text{PRAUC} &= \frac{1}{\sum_i^N v_i} \sum_i^N \sum_j^{v_i} \sum_{k=1}^{|M|} Precision(k)_i^{(j)} \Delta Recall(k)_i^{(j)}, \\ \Delta Recall(k)_i^{(j)} &= (Recall(k)_i^{(j)} - Recall(k-1)_i^{(j)}) \end{aligned} \tag{6}$$

  Where $Precision(k)_i^{(j)}$ and $Recall(k)_i^{(j)}$ represent the precision and recall in the k-th medication in the whole set M.

- **The F1 score** (higher is better) is the harmonic mean of precision and recall.

$$\text{F1} = \frac{1}{\sum_i^N v_i} \sum_i^N \sum_j^{v_i} \frac{2 \cdot Precision_i^{(j)} \cdot Recall_i^{(j)}}{Precision_i^{(j)} + Recall_i^{(j)}} \tag{7}$$

5.1.4 ***Evaluation Strategies.*** [2] We process the dataset following the same setting as SafeDrug [32] and split the processed dataset into the training set, validation set, and test set in a 2/3: 1/6: 1/6 ratio. We set the number of layers of the model to 12, the hidden size to 768, the filter size to 3072, and the number of attention heads to 16. We set the max length of sequences to 512 and apply a learnable positional embedding. We use the Adam optimizer in both the pre-training phase and fine-tuning phase. The learning rate is set to 5e-5 in the pre-training stage and 1e-5 in the fine-tuning stage. We set the dropout rate to 0.1 in both stages. In the pre-training stage, we train the model on 4 NVIDIA GeForce RTX 2080 Ti GPU for 200k steps and set the batch size to 8. In the fine-tuning stage, we fine-tune the model for 5 epochs on one GPU on the medication recommendation task. A 4-layer MLP with a hidden size of 3072 is concatenated to the pre-trained model to predict medications. The batch size in fine-tuning is set to 1.

[2] Our codes will be released on Github after acceptance for reproducibility.

Table 3: Performance Comparison of Different Methods on Medication Recommendation Task on *MIMIC-III*. '+': significantly better than the strongest baseline ($p < 0.01$). '++': significantly better than the strongest baseline ($p < 0.001$). We denote the strongest baseline with an underline.

Model	Jaccard	PRAUC	F1-score
Nearest	0.4682	0.4800	0.6263
LR	0.4903	0.7577	0.6494
DMNC	0.4554	0.6596	0.6166
LEAP	0.4484	0.6427	0.6115
RETAIN	0.5038	<u>0.7795</u>	0.6615
G-BERT	0.5003	0.7629	0.6583
GAMENet	0.5175	0.7700	0.6733
SafeDrug	<u>0.5183</u>	0.7742	<u>0.6747</u>
MedicalBERT$_{Linear}$	0.5411^{++}	0.7887	0.6945^{++}
MedicalBERT$_{MLP}$	**0.5457**$^{++}$	**0.7919**$^{+}$	**0.6977**$^{++}$

5.2 Experimental Results

5.2.1 ***Performance on Medication Recommendation.*** In Table 3, we report the medication recommendation performance of MedicalBERT compared with baseline methods, and significance test results with a two-tailed t-test are reported. MedicalBERT$_{Linear}$ denotes that the predictor in the fine-tuning stage is a linear classifier, and MedicalBERT$_{MLP}$ means the predictor is an MLP.

Firstly, both of our proposed methods, MedicalBERT$_{Linear}$ and MedicalBERT$_{MLP}$, outperform all baseline methods significantly, and MedicalBERT$_{MLP}$ achieves over 5.29% improvements than the best score of previous models in the Jaccard similarity score. These results indicate that our pre-training and fine-tuning framework is suitable for fusing heterogeneous EHR data in building powerful representations for patients to conduct medication recommendations. Secondly, for the baseline methods, GAMENet and SafeDrug perform better than others. One reason is that the patient's historical visits are used in prediction. In contrast, our model does not utilize this information. Hence, the improvement is entirely from the fusion modeling of multiple data sources within the same visit. Thirdly, for the predictors in the fine-tuning stage, the MLP makes an improvement of 0.0046 compared with linear classifiers in the Jaccard similarity score, implying that MLPs can better utilize the pre-trained representations for recommending drugs. We use MLP as the default predictor in the following experiments.

5.2.2 ***Analyses on Model Variants.*** We evaluate three types of key variants of our framework in the pre-training phase.

Data used in pre-training

- **All** selects all medical records in a visit to convert into tokens for pre-training.
- **First Day** only considers medical records on the first day of the visit to avoid extra-long sequences.

Order of different types of tokens

- **Chronological** arranges tokens in the order of their corresponding record time.
- **Aggregate** assures that tokens of the same type appear in the same subsequence.

Mask strategies

- **Mask All** randomly replaces all tokens except for special tokens in the sequence with [MASK].
- **Mask Part** only replaces diagnosis tokens, procedure tokens, and medication tokens with [MASK] randomly.

We conduct two analyses about the performance of the above settings. One is the token recovery accuracy in pre-training, and the other is the performance of the fine-tuning task.

For a fair comparison, we only compare the recovery accuracy of the MLM task on three types of tokens, namely diagnosis, procedure, and medication tokens. We evaluate the recovery accuracy directly on the training set. Figure 3 shows the average recovery accuracy of these types of tokens during pre-training. For all types of tokens, the *First Day* strategy and the *Mask Part* strategy can achieve a significantly higher recovery accuracy, indicating that these strategies are beneficial for the model to build robust representations for patients. For the *Aggregate* strategy, the recovery accuracy is higher than the *Chronological* strategy combined with the *First Day* strategy and the *Mask Part* strategy, implying that it is also a helpful setting for pre-training.

We also compare the performance of these variants on the medication recommendation task. The results are shown in Table 4. It demonstrates that the *First Day*, *Aggregate*, and *Mask Part* strategies are combined to achieve the best performance. The results

Table 4: Performance Comparison of Model Variants on Medication Recommendation Task.

Data	Order	Mask	Evaluation		
			Jaccard	PRAUC	F1-score
All	Chro.	All	0.5388	0.7868	0.6916
All	Agg.	All	0.5399	0.7874	0.6930
All	Agg.	Part	0.5433	0.7903	0.6960
First Day	Chro.	All	0.5399	0.7886	0.6929
First Day	Agg.	All	0.5389	0.7878	0.6922
First Day	Agg.	Part	**0.5457**	**0.7919**	**0.6977**

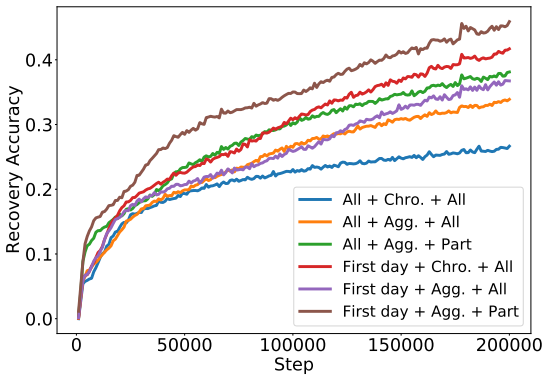


Figure 3: Average Recovery Accuracy of diagnosis, procedure, and medication tokens through pre-training.

Table 5: Ablation Study on Data Source in Pre-training.

Pre-training Data			Evaluation		
Standard	Static	Lab	Jaccard	PRAUC	F1-score
			0.5242	0.7719	0.6798
✓			0.5409	0.7892	0.6939
✓	✓		0.5434	0.7892	0.6960
✓		✓	0.5437	0.7902	0.6964
✓	✓	✓	**0.5457**	**0.7919**	**0.6977**

also show that the *Mask Part* strategy consistently outperforms the *Mask All* strategy in the above settings.

To sum up, the strategies of *First Day*, *Aggregate*, and *Mask Part* are combined to achieve the best performance in both pre-training and fine-tuning phases.

5.2.3 ***Ablation Study on Data Sources.*** We conduct ablation studies on both pre-training and fine-tuning data to verify the effectiveness of our model in fusing heterogeneous features. The commonly used features in previous studies [21, 32] are denoted as 'Standard', including diagnosis, procedures, and medications.

For the pre-training stage, we remove one or more types of tokens in the training data and evaluate the performance on the medication recommendation task. We report the best result when varying the input of the fine-tuning task. As shown in Table 5, removing lab data and static data in pre-training both hurt the performance of the model, indicating that our framework is capable of utilizing the two types of features for medication recommendation. The first row of Table 5 does not involve any data type, which is equivalent to skipping the pre-training stage and directly training on the medication recommendation task from scratch. The result of this setting is significantly reduced, indicating that pre-training does build effective representations for medical codes from heterogeneous information.

For the fine-tuning stage, the input of the medication recommendation task is typically diagnoses and procedures (denoted as 'Standard'), as in GAMENet [21] and SafeDrug [32]. We experiment with expanding the inputs with static and lab data and observe the corresponding effects. We use the pre-trained model that takes all data types (row 5 in Table 5, including static data and lab data) for fine-tuning. The results are shown in Table 6. From the results, we can imply that static data in the input data of the medication recommendation task can help further improve the performance. While the introduction of lab data does not improve the performance of the fine-tuning task. We think a possible reason is that the lab data might interfere with the modeling of some important information, such as diagnosis tokens, in the fine-tuning stage.

5.2.4 ***Observations in Disease-level Evaluation.*** Although the introduction of static and lab data has further improved the performance of medication recommendations, there still remains a question: which disease benefits more from these new heterogeneous features? Therefore, we conduct a disease-level evaluation experiment to observe the impact of the static and lab data on the performances of different diseases.

Table 6: Ablation Study on Input Data Source in Fine-tuning.

Fine-tuning Input			Evaluation		
Standard	Static	Lab	Jaccard	PRAUC	F1-score
✓			0.5424	0.7916	0.6950
✓	✓		**0.5457**	**0.7919**	**0.6977**
✓	✓	✓	0.5444	0.7913	0.6965

Table 7: Top 5 Diagnoses benefit from the addition of static and lab data.

Diagnosis	Jaccard		ΔJaccard
	Standard	All data	
Other specified surgical operations and procedures causing abnormal patient reaction, or later complication, without mention of misadventure at time of operation	0.5390	**0.5688**	+0.0299
Other iatrogenic hypotension	0.5260	**0.5527**	+0.0267
Sepsis	0.5191	**0.5451**	+0.0260
Tobacco use disorder	0.5351	**0.5561**	+0.0210
Subendocardial infarction, initial episode of care	0.6001	**0.6183**	+0.0182

We take the pre-trained models with and without static and lab data for comparison, namely the model in the second row (denoted as 'Standard') and the fifth row (denoted as 'All Data') in Table 5. We choose diseases that occur at least 100 times in the test set for evaluation. We construct one specific test set for each disease, with every piece of data containing the disease. We compare the performance of the two pre-trained models on the same disease and show the five diseases with the most benefit in Table 7.

We see from the table that the performances of medication recommendations for the diseases listed in the table do improve significantly due to the introduction of static and lab data in the recommender system. We can draw inspiration that these diseases may indeed require more consideration of the patient's personal information and laboratory test results when prescribing. It provides us with an automated inference method that can suggest which patients need more attention to their static and lab test results in their treatment. It may help medical professionals focus more on patients who need more care.

6 CONCLUSION

In this work, we propose to apply pre-training techniques to introduce new types of medical data and fuse heterogeneous medical data for modeling, and a new framework named MedicalBERT is devised. With an elaborately designed strategy to convert medical records into sequences, MedicalBERT can effectively combine and model different types of medical data, e.g., newly introduced patient static and lab data. Significant improvements are achieved on the medication recommendation task in a real-world dataset. In the

future, we plan to generalize our pre-trained model and explore its effects on more medical tasks. We also consider generalizing our framework to other recommendation tasks that require heterogeneous information modeling.

REFERENCES

[1] Daniel Almirall, Scott N. Compton, Meredith Gunlicks-Stoessel, Naihua Duan, and Susan A. Murphy. 2012. Designing a pilot sequential multiple assignment randomized trial for developing an adaptive treatment strategy. *Statistics in Medicine* 31, 17 (7 2012), 1887–1902. https://doi.org/10.1002/SIM.4512

[2] Yang An, Liang Zhang, Haoyu Yang, Leilei Sun, Bo Jin, Chuanren Liu, Ruiyun Yu, and Xiaopeng Wei. 2021. Prediction of Treatment Medicines with Dual Adaptive Sequential Networks. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[3] 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. 2020. Language Models are Few-Shot Learners. *arXiv* (5 2020). http://arxiv.org/abs/2005.14165

[4] Xusong Chen, Dong Liu, Chenyi Lei, Rui Li, Zheng-Jun Zha, and Zhiwei Xiong. 2019. Bert4sessrec: Content-based video relevance prediction with bidirectional encoder representations from transformer. In *Proceedings of the 27th ACM International Conference on Multimedia*. 2597–2601.

[5] Edward Choi, Mohammad Taha Bahadori, Joshua A Kulas, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *arXiv preprint arXiv:1608.05745* (2016).

[6] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. 2016. Doctor AI: Predicting Clinical Events via Recurrent Neural Networks. , 301–318 pages. https://proceedings.mlr.press/v56/Choi16.html

[7] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised Cross-lingual Representation Learning at Scale. (11 2019). https://arxiv.org/abs/1911.02116v2

[8] Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems* 32 (2019), 7059–7069.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).

[10] Fan Gong, Meng Wang, Haofen Wang, Sen Wang, and Mengyue Liu. 2021. Smr: Medical knowledge graph embedding for safe medicine recommendation. *Big Data Research* 23 (2021), 100174.

[11] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature 2016 538:7626* 538, 7626 (10 2016), 471–476. https://doi.org/10.1038/nature20101

[12] Yong He, Cheng Wang, Nan Li, and Zhenyu Zeng. 2020. Attention and Memory-Augmented Networks for Dual-View Sequential Learning. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 20 (2020), 10. https://doi.org/10.1145/3394486

[13] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (7 2018), 1531–1540. https://doi.org/10.1145/3219819.3219965

[14] Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. *arXiv preprint arXiv:1909.00964* (2019).

[15] 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. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data 2016 3:1* 3, 1 (5 2016), 1–9. https://doi.org/10.1038/sdata.2016.35

[16] Hung Le, Truyen Tran, and Svetha Venkatesh. 2018. Dual memory neural computer for asynchronous two-view sequential learning. In *Proceedings of the 24th ACM SIGKDD*. 1637–1645.

[17] Zachary C. Lipton, David C. Kale, Charles Elkan, and Randall Wetzel. 2015. Learning to Diagnose with LSTM Recurrent Neural Networks. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings* (11 2015). https://arxiv.org/abs/1511.03677v7

[18] Oscar Luaces, Jorge Díez, José Barranquero, Juan José del Coz, and Antonio Bahamonde. 2012. Binary relevance efficacy for multilabel classification. *Progress in Artificial Intelligence* 1, 4 (12 2012), 303–313. https://doi.org/10.1007/S13748-012-0030-X/FIGURES/3

[19] Zhaopeng Qiu, Xian Wu, Jingyue Gao, and Wei Fan. 2021. U-BERT: Pre-training User Representations for Improved Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4320–4327.

[20] Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. 2019. Pre-training of graph augmented transformers for medication recommendation. *arXiv preprint arXiv:1906.00346* (2019).

[21] Junyuan Shang, Cao Xiao, Tengfei Ma, Hongyan Li, and Jimeng Sun. 2019. Gamenet: Graph augmented memory networks for recommending medication combination. In *proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1126–1133.

[22] Chuan Shi, Xiaotian Han, Li Song, Xiao Wang, Senzhang Wang, Junping Du, and Philip S. Yu. 2021. Deep Collaborative Filtering with Multi-Aspect Information in Heterogeneous Networks. *IEEE Transactions on Knowledge and Data Engineering* 33, 4 (4 2021), 1413–1425. https://doi.org/10.1109/TKDE.2019.2941938

[23] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7464–7473.

[24] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

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

[26] Lu Wang, Xiaofeng He, Wei Zhang, and Hongyuan Zha. 2018. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (7 2018), 2447–2456. https://doi.org/10.1145/3219819.3219961

[27] Shanshan Wang, Zhaochun Ren, Pengjie Ren, Jun Ma, Zhumin Chen, and Maarten De Rijke. 2019. Order-free medicine combination prediction with graph convolutional reinforcement learning. *International Conference on Information and Knowledge Management, Proceedings* 10 (11 2019), 1623–1632. https://doi.org/10.1145/3357384.3357965

[28] Yanda Wang, Weitong Chen, Dechang PI, Lin Yue, Sen Wang, and Miao Xu. 2021. Self-Supervised Adversarial Distribution Regularization for Medication Recommendation. 3 (8 2021), 3134–3140. https://doi.org/10.24963/IJCAI.2021/431

[29] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering News Recommendation with Pre-trained Language Models. *arXiv preprint arXiv:2104.07413* (2021).

[30] Cao Xiao, Edward Choi, and Jimeng Sun. 2018. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association* 25, 10 (10 2018), 1419–1428. https://doi.org/10.1093/JAMIA/OCY068

[31] Chaoqi Yang, Cao Xiao, Lucas Glass, and Jimeng Sun. 2021. Change Matters: Medication Change Prediction with Recurrent Residual Networks. (5 2021), 3728–3734. https://arxiv.org/abs/2105.01876v1

[32] Chaoqi Yang, Cao Xiao, Fenglong Ma, Lucas Glass, and Jimeng Sun. 2021. SafeDrug: Dual Molecular Graph Encoders for Safe Drug Recommendations. *arXiv preprint arXiv:2105.02711* (2021).

[33] Jun Yang, Weizhi Ma, Min Zhang, Xin Zhou, Yiqun Liu, and Shaoping Ma. 2021. LegalGNN: Legal Information Enhanced Graph Neural Network for Recommendation. *ACM Transactions on Information Systems (TOIS)* 40, 2 (2021), 1–29.

[34] Lin Yue, Dongyuan Tian, Weitong Chen, Xuming Han, and Minghao Yin. 2020. Deep learning for heterogeneous medical data analysis. *World Wide Web* 23, 5 (9 2020), 2715–2737. https://doi.org/10.1007/S11280-019-00764-Z/TABLES/4

[35] Zheni Zeng, Chaojun Xiao, Yuan Yao, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2021. Knowledge transfer via pre-training for recommendation: A review and prospect. *Frontiers in big Data* 4 (2021).

[36] Qi Zhang, Jingjie Li, Qinglin Jia, Chuyuan Wang, Jieming Zhu, Zhaowei Wang, and Xiuqiang He. 2021. UNBERT: User-News Matching BERT for News Recommendation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*. 3356–3362.

[37] Yutao Zhang, Robert Chen, Jie Tang, Walter F Stewart, and Jimeng Sun. 2017. LEAP: learning to prescribe effective and safe treatment combinations for multimorbidity. In *proceedings of the 23rd ACM SIGKDD international conference on knowledge Discovery and data Mining*. 1315–1324.

[38] Zhengyan Zhang, Yuxian Gu, Xu Han, Shengqi Chen, Chaojun Xiao, Zhenbo Sun Yuan Yao, Fanchao Qi, Jian Guan, Pei Ke, Yanzheng Cai, Guoyang Zeng, Zhixing Tan Zhiyuan Liu, Minlie Huang, Wentao Han, Yang Liu, Xiaoyan Zhu, and Maosong Sun. 2021. CPM-2: Large-scale cost-effective pre-trained language models. *AI Open* 2 (1 2021), 216–224. https://doi.org/10.1016/J.AIOPEN.2021.12.003