Doctor2Vec: Dynamic Doctor Representation Learning for Clinical Trial Recruitment

Siddharth Biswal[1,2], Cao Xiao[1], Lucas M. Glass[1], Elizabeth Milkovits [1], and Jimeng Sun[2]

[1] Analytic Center of Excellence, IQVIA, Cambridge, USA

[2] Computational Science and Engineering, Georgia Institute of Technology, Atlanta, USA

Abstract

Massive electronic health records (EHRs) enable the success of learning accurate patient representations to support various predictive health applications. In contrast, doctor representation was not well studied despite that doctors play pivotal roles in healthcare. How to construct the right doctor representations? How to use doctor representation to solve important health analytic problems? In this work, we study the problem on *clinical trial recruitment*, which is about identifying the right doctors to help conduct the trials based on the trial description and patient EHR data of those doctors. We propose Doctor2Vec which simultaneously learns 1) doctor representations from EHR data and 2) trial representations from the description and categorical information about the trials. In particular, Doctor2Vec utilizes a dynamic memory network where the doctor's experience with patients are stored in the memory bank and the network will dynamically assign weights based on the trial representation via an attention mechanism. Validated on large real-world trials and EHR data including 2,609 trials, 25K doctors and 430K patients, Doctor2Vec demonstrated improved performance over the best baseline by up to 8.7% in PR-AUC. We also demonstrated that the Doctor2Vec embedding can be transferred to benefit data insufficiency settings including trial recruitment in less populated/newly explored country with 13.7% improvement or for rare diseases with 8.1% improvement in PR-AUC.

Introduction

The rapid growth of electronic health record (EHR) data and other health data enables the training of complex deep learning models to learn patient representations for disease diagnosis (Choi et al. 2018; Choi et al. 2016; Xiao, Choi, and Sun 2018), risk prediction (Xiao et al. 2018), patient subtyping (Baytas et al. 2017; Che et al. 2017), and medication recommendation (Shang et al. 2019b; Shang et al. 2019a). However, almost all existing works focus on modeling patients. Deep neural networks for doctor representation learning are lacking.

Doctors play pivotal roles in connecting patients and treatments, including recruiting patients into clinical trials for drug development and treating and caring for their patients. Thus an effective doctor representation will better support a wider range of health analytic tasks. For example, iden-

tifying the right doctors to conduct the trials *site selection* so as to improve the chance of completion of the trials (Hurtado-Chong et al. 2017) and doctor recommendation for patients (Xu et al. 2018).

In this work, we focus on studying the *clinical trial recruitment* problem using doctor representation learning. Current standard practice calculates the median enrollment rate [1] for a therapeutic area as the predicted enrollment success rate for all participating doctors, which is often inaccurate. In addition, some develop a multi-step manual matching process for site selection which is labor-intensive (Hurtado-Chong et al. 2017; Potter et al. 2011). Recently, deep neural networks were applied on site selection tasks via static medical concept embedding using only frequent medical codes and simple term matching to trials (Gligorijevic et al. 2019). Despite the success, two challenges remain open.

1. Existing works do not capture the time-evolving patterns of doctors experience and expertise encoded in EHR data of patients that the doctor have seen;

2. Existing works learn a static doctor representation. However, in practice given a trial for a particular disease, the doctor's experience of relevant diseases are more important. Hence the doctor representation should change based on the corresponding trial representation.

To fill the gap, we propose Doctor2Vec which simultaneously learns 1) doctor representations from longitudinal patient EHR data and 2) trial embedding from the multimodal trial description. In particular, Doctor2Vec leverages a dynamic memory network where the representations of patients seen by the doctor are stored as memory while trial embedding serves as queries for retrieving from the memory. Doctor2Vec has the following contributions.

1. **Patient embedding as a memory for dynamic doctor representation learning**. We represent doctors' evolving experience based on the representations from the doctors' patients. The patient representations are stored as a memory for dynamic doctor representation extraction.

2. **Trial embedding as a query for improved doctor selection**. We learn hierarchical clinical trial embedding where

---

[1] Enrollment rate of a doctor is the number of patients enrolled by a doctor to the trial.

the unstructured trial descriptions were embedded using BERT (Devlin et al. 2018). The trial embedding serves as queries of the memory network and will attend over patient representation and dynamically assign weights based on the relevance of doctor experience and trial representation to obtain the final context vector for an optimized doctor representation for a specific trial.

We evaluated Doctor2Vec using large scale real-world EHR and trial data for predicting trial enrollment rates of doctors. Doctor2Vec demonstrated improved performance in site selection task over the best baselines by up to 8.7% in PR-AUC. We also demonstrated that the Doctor2Vec embedding can be transferred to benefit data insufficiency settings including trial recruitment in less populated/newly explored countries or for rare diseases. Experimental results show for the country transfer, Doctor2Vec achieved 13.7% relative improvement in PR-AUC over the best baseline. While for embedding transfer to rare disease trials, Doctor2Vec achieved 8.1% relative improvement in PR-AUC over the best baseline.

Related Works

Deep Patient Representation Learning   The collection of massive EHR data has motivated the use of deep learning for accurate patient representation learning and disease or risk prediction (Xiao, Choi, and Sun 2018; Fu et al. 2019; Baytas et al. 2017; Choi et al. 2018). In this work, we learn hierarchical patient representation in a similar way as (Choi et al. 2018). But our focus is to construct doctor representation based on the embedding of their patients.

Machine Learning Based Clinical Trial Recruitment Previously clinical trial enrollment either relies on simple statistics (e.g., medium enrollment) or manual matching (Hurtado-Chong et al. 2017). With the collection of clinical trial data, there has been some effort on developing machine learning-based models for trial site selection. For example, (van den Bor et al. 2017) applied LR with L1 regularization to determine a subset that is optimal for predicting site enrollment success. More recently, (Gligorijevic et al. 2019) learns static medical concept embedding and matches them to features derived from trial terms for site selection. However, no existing works learn trial embedding from multi-modal trial data and automatically match them to most relevant doctors.

Memory Augmented Neural Networks   (MANN) have shown initial success in NLP research areas such as question answering (Weston, Chopra, and Bordes 2015; Sukhbaatar et al. 2015; Miller et al. 2016; Kumar et al. 2016). Memory Networks (Weston, Chopra, and Bordes 2015) and Differentiable Neural Computers (DNC) (Graves et al. 2016) proposed to use external memory components to assist the deep neural networks in remembering and storing things. After that, various MANN based models have been proposed such as (Sukhbaatar et al. 2015; Kumar et al. 2016; Miller et al. 2016). In healthcare, memory networks can be valuable due to their capacities in memorizing medical knowledge and patient history. DMNC (Le, Tran, and

Venkatesh 2018) proposed a MANN model for medication combination recommendation task using EHR data alone. In (Shang et al. 2019b), the authors use a memory component to fuse multi-model graphs as a memory bank to facilitate medication recommendation.

Method

Problem Formulation

Definition 1 (Doctor Records) For each doctor, the clinical experience can be represented as a sequence of patients that the doctor has seen, denoted as $\mathbf{C(m)} = \{\mathbf{P_1^{(m)}}, \mathbf{P_2^{(m)}}, \cdot, \mathbf{P_k^{(m)}}\}$ where m indicates the m-th doctor. Here each patient can also be represented as a sequence of multivariate observations $\mathbf{P(k)} = \{\mathbf{v_1^{(k)}}, \mathbf{v_2^{(k)}}, \cdot, \mathbf{v_T^{(k)}}\}$ where k indicates the k-th patient and subscript $1, 2, \ldots, T$ indicates different visits for the k-th patient. Each visit $\mathbf{v}_t^{(k)}$ is the combination of diagnosis codes $\mathbf{c_d}$, medication codes $\mathbf{c_m}$, and procedure codes $\mathbf{c_p}$. Medical codes $\mathbf{c_d}, \mathbf{c_p}, \mathbf{c_m} \in 0, 1^{|C|}$ are represented as multi-hot vectors and $|C|$ represent the total size of the code sets. We also use the demographic information available about doctors and patients available. These static information are denoted as $\mathbf{Doc_{static}}$ which are used with other features.

Definition 2 (Clinical Trial Data) Clinical trial data comprises of two data modalities: the trial descriptions in unstructured text and the categorical features such as trial phase, primary indication, primary outcome, secondary outcome, and study type. We denote each clinical trial as a combination of text data and categorical data.

$$\mathbf{Q(cat)(l)} = \{\mathbf{f_1^{(1)}}, \mathbf{f_2^{(1)}}, \cdot, \mathbf{f_v^{(1)}}\}$$
$$\mathbf{Q(l)} = [\mathbf{Q(cat)(l)}; \mathbf{Q(text)(l)}]$$

where $\mathbf{l} \in \{1, 2, \cdot, \cdot, L\}$ is the index of clinical trials, $\mathbf{f_i}$ is the representation for the categorical trial features, and $\mathbf{v}$ is the number of categorical features in a trial.

Table 1: Notations used in Doctor2Vec.

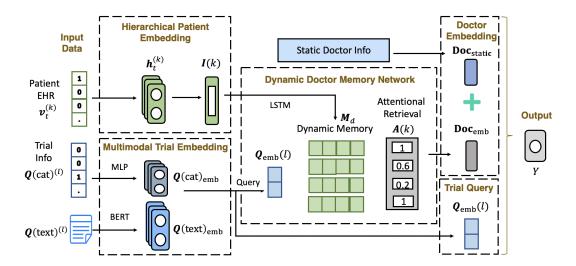| Symbol	Definition and description
$\mathcal{D}$	Set of unique diagnosis codes
$\mathcal{P}$	Set of unique procedure codes
$\mathcal{M}$	Set of unique medication codes
$\mathbf{C(m)}$	Notation for a doctor
$\mathbf{P_k^{(m)}}$	Notation for a patient for m-th doctor
$\mathbf{v_t^{(k)}}$	the t-th visit of Patient k
$\mathbf{Q(l)}$	The l-th clinical trial
$\mathbf{Q(cat)^{(l)}}$	Categorical features of the l-th clinical trial
$\mathbf{Q(text)^{(l)}}$	Text features of the l-th clinical trial
$\mathbf{I_f(k)}$	Final patient representation
$\mathbf{Doc_{emb}}$	Vector representation of a Doctor
$\mathbf{Doc_{static}}$	Static features of a Doctor

Figure 1: The Doctor2Vec Framework. (1) Hierarchical patient embedding: We obtain patient embeddings $\mathbf{h_t(k)}$ from patient visits using a Bi-LSTM with attention module. Unstructured text and categorical data are modeled using MLP and BERT respectively. (2) Multimodal clinical trial information embedding: The obtained clinical trial embeddings $\mathbf{Q_{emb}(l)}$ are fused together to form the query vector for the memory network.(3) Memory network module: This query vector to used to attend over the memory bank is obtained from combination of patient embeddings. The attentional vector is used to obtain the final doctor embedding . We combine the doctor embedding $\mathbf{Doc_{emb}}$ with clinical trial embedding $\mathbf{Q_{emb}(l)}$ and static information about the doctors $\mathbf{Doc_{static}}$ to predict the enrollment rate of the clinical trial (output).

Doctor2Vec Framework

As illustrated in Fig. 1, Doctor2Vec includes the following components: *hierarchical patient embedding*, *multimodal clinical trial information embedding*, a *memory network module* for dynamical doctor information retrieval, and a memory output module for generating the current time doctor representation. This doctor representation is used with clinical trial representation to predict the *enrollment rate* of the clinical trial. Next, we will first introduce these modules and then provide details of training and inference.

Hierarchical Patient Embedding   As discussed before, a doctor has seen a set of patients during his/her medical practice. We model that the doctor's experience and expertise by the function over the embeddings of their patients' EHR data. If a patient has seen two doctors, the corresponding portion of the EHR data will be modeled as two separate patients for both doctors, respectively.

To learn patient representation, motivated by (Choi et al. 2018), we leverage the inherent multilevel structure of EHR data to learn patient embedding hierarchically. The hierarchical structure includes the patients on the top, followed by visits the patient experiences over time, then at the leaf level the set of diagnosis, procedure, and medication codes recorded for each visit. Here given clinical visits of a patient, denoted as $\mathbf{v_t^{(k)}}$, we firstly pass these visits through a multi-layer perception (MLP) to get visit embedding $\mathbf{h_t^{(k)}}$ as follows.

$$\mathbf{h_t^{(k)}} = \mathbf{W_{emb}v_t^{(k)}} \tag{1}$$

Without ambiguity, we will ignore the patient index (k) for brevity. Next to learn a patient embedding based on a sequence of visit embeddings, we pass visit embedding $\mathbf{h_k}$ to a bi-directional long short-term memory (bi-LSTM) networks and then add an attention layer on top of the bi-LSTM to attend on important visits.

$$\mathbf{g_1, g_2, ...g_t} = \text{bi-LSTM}(\mathbf{h_1, h_2, ..., h_t}) \tag{2}$$

$$e_t = \mathbf{w_\alpha^T g_t} + b_\alpha \tag{3}$$

$$\alpha_1, \alpha_2, ..\alpha_t = \text{softmax}(\text{e}_1, \text{e}_2, ..\text{e}_t) \tag{4}$$

Then we obtain final patient context vector $\mathbf{I(k)}$ by summing over attended visit representations as given by Eq. 5.

$$\mathbf{I(k)} = \sum \alpha_t \cdot \mathbf{h_t} \tag{5}$$

Here $\mathbf{I(1)} \ldots \mathbf{I(k)}$ are patient representations that will be fed into dynamic doctor memory bank as memory vectors.

Multimodal Trial Information Embedding   Clinical Trials are conducted to evaluate a specific drug or procedure. We use public and private information about clinical trials. Here we obtained clinical trial descriptions from publicly available clinical trial database clinicaltrials.gov. The collected trial description comprises multiple data modalities, including unstructured text and categorical features. In this module, we employ a multi-modal method to embed them in a shared space.

1. **Unstructured Text**. Each trial has a text description for inclusion and exclusion criteria which describe the requirements and restrictions for recruiting to the trial. For unstructured text, we applied the BERT (Devlin et al. 2018) model. BERT builds on the Transformer (Vaswani et al. 2017) architecture and improves the pre-training using a masked language model for bidirectional representation. Essentially BERT model is a stack of Transformer blocks. Each transformer block is a combination

of self-attention block and feedforward layer. **Pretraining**: We use the same pre-training techniques as in (Devlin et al. 2018) (1) Masked Language Model: This modeling task consists of masking 15% of the input tokens and using the model to predict the masked tokens. This is trained with cross entropy loss. (2) Next sentence prediction task: In this task two sentences are fed to BERT. The model outputs a binary prediction of whether these two sentences are in consecutive order. The final pre-training objective function based on the two tasks is the sum of the log-likelihood of the masked tokens and the log-likelihood of the binary variable indicating whether two sentences are consecutive. We pretrained a BERT model using MIMIC text data (Johnson et al. 2017) to extract embeddings for each word and combine the word embeddings to obtain the final text embeddings. In order to pre-train BERT on text corpus, we first obtain preprocessed data as required by BERT which includes tokenization, indexing and masking. The procedure is formulated as below. Denote unstructured text associated with each trial as $\mathbf{Q}(\mathbf{text})^{(l)}$, the embedding $\mathbf{Q}(\mathbf{text})_{\mathbf{emb}}$ is given by Eq. 6. We average over embeddings obtained from each word to compute the final embedding for the entire document.

$$\mathbf{Q}(\mathbf{text})_{\mathbf{emb}} = \mathrm{BERT}(\mathbf{Q}(\mathbf{text})^{(l)}) \qquad (6)$$

2. **Categorical Features**. The categorical features of each clinical trial include the geographic location of the trial, hospital system, primary and secondary therapeutic area, pharmaceutical company information, phase of the trial, condition or disease, objectives of the trial, intervention model, etc. More details of the categorical features in the Appendix. The dimensions of these categorical variables ranged from 18 to 1456. For these features, we first encode them using one-hot vectors and then pass the one-hot vectors through multi-layer perception (MLP) layer. This can be expressed as below. Denote the categorical features as $\mathbf{Q}(\mathbf{cat})^{(l)}$, the categorical feature embedding is obtained as in Eq. 7.

$$\mathbf{Q}(\mathbf{cat})_{\mathbf{emb}} = \mathbf{W_c}\mathbf{Q}(\mathbf{cat})^{(l)} + \mathbf{b_c} \qquad (7)$$

After obtaining embeddings from both types of clinical trial data, we fuse the embeddings from categorical data $\mathbf{Q}(\mathbf{cat})_{\mathbf{emb}}$ and text data $\mathbf{Q}(\mathbf{text})_{\mathbf{emb}}$ to obtain the final embedding $\mathbf{Q}_{\mathbf{emb}}(l)$. We fuse these two embeddings weighted multiplicative fashion as in Eq. 8.

$$\begin{aligned}\mathbf{Q}_{\mathbf{emb}}(l) = (\mathbf{W_{ci}}\mathbf{Q}(\mathbf{cat})_{\mathbf{emb}} + \mathbf{b_{ci}})\odot \\ (\mathbf{W_{ti}}\mathbf{Q}(\mathbf{text})_{\mathbf{emb}} + \mathbf{b_{ti}}) \qquad (8)\end{aligned}$$

where $\odot$ is element-wise multiplication. The clinical trial embeddings $\mathbf{Q}_{\mathbf{emb}}(l)$ will be fed into dynamic doctor memory network as the query to extract related patient representation memory.

Dynamic Doctor Memory Networks Since each doctor sees a diverse set of patients, doctor representation should be dynamically constructed for a given trial as opposed to a static embedding vector staying the same for all trials. The way we achieved that is by a dynamic memory network where patients are stored as memory vectors of the doctor. Then using a trial embedding as a query, we fetch the relevant patient vectors from the memory bank and dynamically assemble a doctor representation for this trial.

Inspired by (Weston, Chopra, and Bordes 2015), four memory components $\mathbf{I}$, $\mathbf{G}$, $\mathbf{O}$, $\mathbf{R}$ are proposed which mimics the architecture of modern computer in storing and processing information.

1. **Input Memory Representation**. This layer converts the patient representations to the input representation. We pass all the patient representations through a dense layer to obtain the input representations.

$$\mathbf{I_f}(\mathbf{k}) = \mathbf{W_i}\mathbf{I}(\mathbf{k}) + \mathbf{b_i} \qquad (9)$$

2. **Generalization**. Typically generalization can be referred to as the process of updating memory representation for the memory bank. In our case, we use the patient representations to initialize the memory representation $\mathbf{M_d}$ which is the combination of all the patient representations. We then apply an LSTM layer to update the memory via multiple iterations.

$$\mathbf{M_d} = \mathrm{LSTM}(\mathbf{I_f(1)}, \cdots, \mathbf{I_f(k)}) \qquad (10)$$

3. **Output**. In this step, the final output memory representation is generated. We calculate the relevance between trial embedding $\mathbf{Q}_{\mathbf{emb}}(\mathbf{l})$ and doctor embedding $\mathbf{M_d}$ to obtain $A(k)$ as the attention vector over patient representations.

$$A(k) = \mathrm{softmax}[\mathbf{Q}_{\mathbf{emb}}(\mathbf{l})^{\mathbf{T}}\mathbf{M_d}] \qquad (11)$$

4. **Response**. In this step, we obtain the final $\mathbf{Doc}_{\mathbf{emb}}$ using the patient embeddings and attention weights over the patients.

$$\mathbf{Doc}_{\mathbf{emb}} = \sum A(k)\mathbf{I_f(k)} \qquad (12)$$

We use the doctor representation which is composed of patients and the clinical trial representation to obtain a final context vector. Besides dynamic doctor embedding $\mathbf{Doc}_{\mathbf{emb}}$, we also include static information about doctors in the final embedding such as their educational history, length of practice, length of practice into the feature vector. The resulting final embedding vectors are then fed into a fully connected layer and passed through a softmax to obtain class labels.

$$\mathbf{Y} = \mathrm{Softmax}([\mathbf{Doc}_{\mathbf{emb}}; \mathbf{Q}_{\mathbf{emb}}(\mathbf{l}); \mathbf{Doc}_{\mathbf{static}}]) \qquad (13)$$

where the input to Softmax are concatenation of dynamic doctor embedding $\mathbf{Doc}_{\mathbf{emb}}$, trial query embedding $\mathbf{Q}_{\mathbf{emb}}(\mathbf{l})$ and static doctor embedding $\mathbf{Doc}_{\mathbf{static}}$.

The enrollment rate category is obtained by binning the continuous enrollment rate. We divide the continuous enrollment scores into five discrete classes ranging at $0 \sim 0.2$, $0.2 \sim 0.4$, $0.4 \sim 0.6$. $0.6 \sim 0.8$, $0.8 \sim 1.0$. The 5 enrollment categories are used labels for classification.

Training and Inference

During training, We train our models by minimizing the cross entropy loss to optimize $\mathbf{W_{emb}}, \mathbf{W_c}, \mathbf{W_i}$, weight matrices of Bi-LSTM. We denote the network parameters by θ_c which is updated by optimizing for the loss function.

$$\mathcal{L} = -\frac{1}{N} \sum (\mathbf{y_i} log(\hat{\mathbf{y_i}}) + (1 - \mathbf{y_i})^T log(1 - \hat{\mathbf{y_i}})) \quad (14)$$

In the inference phase, we use calibrated threshold value for obtaining predicted labels from the predicted probability values where we obtain the probability values from the final layer of the network as mentioned in Eq. 13. Our Doctor2Vec model is summarized in Algorithm 1.

Algorithm 1: Model Training for Doctor2Vec

Input: Training dataset, input $(\mathbf{C}, \mathbf{Q})$ and target $\mathbf{Y}$; epochs N_{epoch}
Output: Trained model for enrollment rate prediction with parameter $\theta_{\mathbf{c}}$

1 Initialization;
2 **for** $i = 1, \ldots, N_{epoch}$ **do**
3   **foreach** *mini-batch in the training set* **do**
4   Obtain $\mathbf{h_k}(\mathbf{m})$ using MLP in Eq. 1 ;
5   Compute $\mathbf{I}(\mathbf{k})$ using BiLSTM and Attn. Eq. 5;
6   Compute $\mathbf{Q_{emb}}$ using combination of MLP and BERT by trial embedding;
7        $\mathbf{I_f}(\mathbf{k})$ is obtained from $\mathbf{I}(\mathbf{k})$;
8        $\mathbf{A}(\mathbf{k})$ is generated from inner product of $\mathbf{Q_{emb}}$ and $\mathbf{I_f}(\mathbf{k})$
9   Compute Doctor representation $\mathbf{Doc_{emb}}$ in Eq. 12;
10   Combine $\mathbf{Doc_{emb}}, \mathbf{Q_{emb}}(\mathbf{l})\mathbf{Doc_{static}}$ for the final prediction Eq. 13;
11   Calculate prediction loss $\mathcal{L}$ using Eq. 14;;
12   Update parameters according to the gradient of $\mathcal{L}$;
13   **end**
14 **end**

Experiment

We designed experiments to answer the following questions.
Q1: Does Doctor2Vec have better performance in predicting clinical trial enrollment to support site selection?
Q2: Can Doctor2Vec embedding perform in transfer learning setting for trials across countries or across diseases?

Implementation We implemented Doctor2Vec [2] with Py-Torch 1.0 (Paszke et al. 2017). For training the model, we used Adam (Kingma and Ba 2014a) with the mini-batch of 128 samples. The training was performed on a machine equipped with an Ubuntu 16.04 with 128GB memory and Nvidia Tesla P100 GPU.

---

[2]Code: https://github.com/sidsearch/Doctor2vec

Data Source We obtained patient and trial information from the following three data sources.

1. We extracted trial data from IQVIA's real-world patient and clinical trial database, which can be accessed by request [3]. It contains 2609 clinical trials formed during 2014 and 2019. This dataset includes 25894 doctors across 28 countries. It includes both unstructured eligibility criteria and categorical features including the geographic location of the trial, hospital system, primary, secondary therapeutic areas, drug names, etc. The data also includes outcome measures such as the trial enrollment rate. In ground truth, the distribution of the enrollment categories are 12%, 33%, 37%, 12%, 6% respectively for $0 \sim 0.2$, $0.2 \sim 0.4$, $0.4 \sim 0.6$. $0.6 \sim 0.8$, $0.8 \sim 1.0$ bins of enrollment score.

2. We also obtained real world patient claims dataset from Database 1. This dataset contains a longitudinal treatment history from 430,239 patients over 7 years. In addition to medical codes about diagnosis, procedure, medication, it also includes information about doctors such as specialty, education, hospital location, geographical location.

3. We also extract clinical trial descriptions from publicly available clinical trial database clinicaltrials.gov. We match the trial information with our Database 1 on NCT ID which is a universal clinical trial ID.

Enrollment Rate Enrollment rate for each investigator is defined as

$$\text{Enrollment Rate} = \frac{\text{\# sub. randomized} - \text{\# sub. discontinued}}{\text{enrollment window}}$$

After obtaining the enrollment numbers, we perform a min-max normalization step to obtain normalized enrollment rate which is between 0 and 1. In this normalization step, we only consider the investigators associated with each clinical trial.

Table 2: Data Statistics

# of clinical trials	2,609
# of doctors	25,894
# of doctor-trial pair(samples)	102,487
# of patients	430,239
Avg # of Dx codes per visit	4.23
Max # of Dx codes per visit	56
Avg # of Procedure codes per visit	1.23
Max # of Procedure codes per visit	18
Avg # of Med codes per visit	9.36

Baselines We consider the following baselines.

1. Median Enrollment (Median). Current industry standard that considers the median enrollment rate for each therapeutic area as estimated rate for all trials in that area.

2. Logistic Regression (LR). We combine the medication,diagnosis and procedure codes along with the clinical trial information to create feature vectors, and then apply LR to predict the enrollment rate category.

---

[3]https://www.iqvia.com/insights/the-iqvia-institute

3. Random Forest (RF) (Breiman 2001). We combine the medication, diagnosis and procedure codes along with the clinical trial information to create feature vectors and then pass it to RF to predict the enrollment rate category.

4. AdaBoost (Schapire 1999). We combine the medication,diagnosis and procedure codes along with the clinical trial information to create feature vectors and then apply AdaBoost classifier to predict the enrollment rate categories.

5. Multi-layer Perceptron (MLP). We use MLP to process doctor features. In this case, we obtain the doctor features by converting all the visit vectors associated with a doctor to a count vector of different diagnosis, medication, procedure codes. We convert categorical information of clinical trials to multi-hot vectors and obtain TF-IDF features from text information of clinical trials.

6. Long Short-Term Memory Networks (LSTM) (Hochreiter and Schmidhuber 1997). We process all the temporally ordered visit vectors associated with a doctor using an LSTM. The embedding obtained from LSTM is concatenated with embedding obtained from categorical and text information of clinical trials to predict enrollment rate.

7. DeepMatch (Gligorijevic et al. 2019) In this model, the features for the doctors are obtained by collecting the top 50 most frequent medical codes and passed through an MLP layer to obtain an embedding vector. This embedding is concatenated with embedding obtained from categorical and text information of clinical trials via MLP and TF-IDF to finally predict enrollment rate.

Evaluation Metrics To evaluate the performance of enrollment prediction , We used PR-AUC as the metric for the classification task, and the coefficient of determination (R^2) score for the regression task. Details of the metrics are provided in appendix.

Experiment Setup and Evaluation Strategies We split our data into train, test, validation split with 70:20:10 ratio. We also ensured that the clinical trails are unique and no overlap in train, test, validation split. We used Adam (Kingma and Ba 2014b) optimizer at learning rate 0.001 with learning rate decay. We fix the best model on evaluation set within 200 epochs and report the performance in test set. Details about reproducibility including hyperparameters are provided in Appendix.

Q1: Doctor2Vec achieved the best predictive performance in clinical trial enrollment prediction

We conducted experiments for both classification (e.g., predict enrollment rate category) and regression (e.g., predict actual rate) tasks. Results are provided in Table 7. From the results, we observe that Doctor2Vec achieved the best performance in both settings.

For category classification, Doctor2Vec has 8.7% relative improvement in PR-AUC over the best baseline LSTM. Among the baselines, the Median method performs the worst, indicating the population level information is not accurate enough for each individual trial. Tree based models such as RF and Adaboost performs better than Median

Table 3: Doctor2Vec achieves the best performance on both metrics in predicting actual enrollment rate (regression task) and rate categories (classification task) compared to state-of-the-art baselines. Results of ten independent runs.

	PR-AUC	R^2 Score
Median	0.571 ± 0.014	0.54 ± 0.072
LR	0.672 ± 0.041	0.314 ± 0.082
RF	0.731 ± 0.034	0.618 ± 0.034
AdaBoost	0.747 ± 0.002	0.684 ± 0.146
MLP	0.761 ± 0.019	0.762 ± 0.049
LSTM	0.792 ± 0.034	0.780 ± 0.621
DeepMatch	0.735 ± 0.068	0.821 ± 0.073
Doctor2Vec	$\mathbf{0.861 \pm 0.021}$	$\mathbf{0.841 \pm 0.072}$

enrollment and LR, which can be attributed to the large number of features they leverage as well as their ability of distilling complex features. MLP performs better than tree based models due to having adequate number of layers for better capturing information and ability control overfitting. The LSTM network further improves over MLP since it is able to extract the temporal information present from the visits of patients. Compared with these approaches, the DeepMatch models achieved much lower PR-AUC since the model leverages the 50 most frequent codes for medical concept embedding, thus missing many important information of the doctors.

In actual rate prediction task, Doctor2Vec gains 2.4% relative improvement in R^2 over best baseline DeepMatch. As for the baselines, the LR model performs the worst, indicating linear models cannot capture the complex and temporal information in the data. Median enrollment is worse than most baselines but better than LR since for some more common diseases median enrollment can be a good predictor. Again, MLP and LSTM work better than tree-based models due to they can better capture complex features. DeepMatch in the regression settings tends to perform better than MLP and LSTM, which can be attributed to the majority of actual scores being in the range of [0.4-0.65] leading to improved performance.

Q2: Doctor2Vec can perform well in trial recruitment prediction even across countries and across diseases

One major challenge for clinical trial recruitment is when conducting trials in a less populated country or a country that is newly explored, or building a trial for a rare disease, the recruitment rate is often hard to estimate since there is not enough historical data to refer to. In this section we design two experiments to explore whether the embedding learned by Doctor2Vec will be useful in order to benefit the aforementioned data insufficiency settings.

1. Trained on United States data and transfer to a less populated/newly explored country;

2. Trained on common diseases and transfer to rare/low prevalence diseases.

For the first experiment, we trained Doctor2Vec on 1443 clinical trials in the United states during the time 2014-2019 and test on 47 clinical trials in South Africa during the time 2014-2019. We perform the same model transfer for all baselines. Results are provided in Table 4.

For the second experiment, we test the model on 38 clinical trials for drugs about idiopathic pulmonary fibrosis (IPF, a rare lung disease) and inflammatory bowel disease(IBD, a low prevalence chronic inflammatory bowel disease). The model was trained on 2569 clinical trials from the rest of the available diseases. We perform the same model transfer for all baselines. Results are provided in Table 5.

Table 4: Doctor2Vec achieves the best performance when we transfer the model trained on US data to predict trial enrollment in South Africa.

	PR-AUC	R^2 Score
Median	0.524 ± 0.032	0.420 ± 0.039
LR	0.601 ± 0.023	0.279 ± 0.014
RF	0.661 ± 0.038	0.552 ± 0.048
AdaBoost	0.672 ± 0.01	0.581 ± 0.039
LSTM	0.758 ± 0.013	0.721 ± 0.025
DeepMatch	0.703 ± 0.087	0.756 ± 0.031
Doctor2Vec	$\mathbf{0.862 \pm 0.003}$	$\mathbf{0.817 \pm 0.025}$

Table 5: Doctor2Vec achieves the best performance when we transfer the model trained on common disease trials to rare and low prevalence disease trials.

	PR-AUC	R^2 Score
Median	0.413 ± 0.013	0.387 ± 0.001
LR	0.521 ± 0.021	0.225 ± 0.028
RF	0.610 ± 0.019	0.517 ± 0.032
AdaBoost	0.623 ± 0.002	0.548 ± 0.046
LSTM	0.725 ± 0.002	0.623 ± 0.038
DeepMatch	0.638 ± 0.021	0.678 ± 0.049
Doctor2Vec	$\mathbf{0.784 \pm 0.032}$	$\mathbf{0.716 \pm 0.014}$

For both settings, Doctor2Vec performs much better than state-of-the-art baselines. For the country transfer, Doctor2Vec achieved 13.7% relative improvement in PR-AUC over best baseline LSTM and 8.1% relative improvement in R^2 over best baseline DeepMatch. While for embedding transfer to rare disease trials, Doctor2Vec achieved 8.1% relative improvement in PR-AUC over the best baseline LSTM and 5.6% relative improvement in R^2 over best baseline DeepMatch.

For country transfer, we also examine the R^2 scores. Based on the R^2 values, the DeepMatch model and LSTM model accounts for 69.2% and 67.3% of the variance in the data, respectively. While our Doctor2Vec accounts for 83.6% of the variance. This shows our model and the prediction fit more to the real observation.

Case Study

We present case studies to demonstrate the effectiveness of the proposed Doctor2Vec model.

Phase I trial for Gemcitabine plus Cisplatin This phase I trial is a combination cancer therapy. A doctor in the US who has worked in internal medicine during the past 3 years has run the trial. The actual enrollment rate is 0.72. The rate estimation provided by the best baseline LSTM is 0.57. While the estimated rate from Doctor2Vec is 0.69, which is much closer to the ground truth. The reason for Doctor2Vec to perform more accurately is the internal medicine doctor has a broader coverage of diseases. Baseline models consider all these diseases that the doctor treated when measuring the match between the doctor and the trial. While Doctor2Vec was able to focus more on the patients who had cancer diagnosis instead of all patients which leads to improved prediction.

Phase II trial for Alzheimer's Disase This phase II trial is about an amyloid drug for treating Alzheimer's patients. A doctor in the US who has treated cancer during the past 4 years runs the trial and has a trial enrollment rate at 0.62. The estimated rate from the best baseline LSTM model is 0.45. Doctor2Vec predicts the enrollment rate will be 0.58, which is much closer to the ground truth. For this case, Doctor2Vec is more accurate because Doctor2Vec is able to learn better doctor representations for this doctor by focusing on the neurological disease patients compared to other disease type patients seen by the doctor.

Conclusion

In this work, we proposed Doctor2Vec, a doctor representation learning based on both patient representations from longitudinal patient EHR data and trial embedding from the multimodal trial description. Doctor2Vec leverages a dynamic memory network where the representations of patients seen by the doctor are stored as memory while trial embedding serves as queries for retrieving the memory. Evaluated on real world patient and trial data, we demonstrated via trial enrollment prediction tasks that Doctor2Vec can learn accurate doctor embeddings and greatly outperform state-of-the-art baselines. We also show by additional experiments that the Doctor2Vec embedding can also be transferred to benefit the data insufficient setting (e.g., model transfer to less populated/newly explored country or from common disease to rare disease) that is highly valuable yet extremely challenging for clinical trials.

Acknowledgement

References

[Baytas et al. 2017] Baytas, I. M.; Xiao, C.; Zhang, X.; Wang, F.; Jain, A. K.; and Zhou, J. 2017. Patient subtyping via time-aware lstm networks. In *KDD*.

[Breiman 2001] Breiman, L. 2001. Random forests. *Mach. Learn.* 45(1):5–32.

[Che et al. 2017] Che, C.; Xiao, C.; Liang, J.; Jin, B.; Zho, J.; and Wang, F. 2017. An rnn architecture with dynamic temporal matching for personalized predictions of parkinson's disease. In *SDM*.

[Choi et al. 2016] Choi, E.; Bahadori, M. T.; Sun, J.; Kulas, J.; Schuetz, A.; and Stewart, W. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*.

[Choi et al. 2018] Choi, E.; Xiao, C.; Stewart, W.; and Sun, J. 2018. Mime: Multilevel medical embedding of electronic health records for predictive healthcare. In *Advances in Neural Information Processing Systems*, 4547–4557.

[Devlin et al. 2018] Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Fu et al. 2019] Fu, T.; Gao, T.; Xiao, C.; Ma, T.; and Sun, J. 2019. Pearl: Prototype learning via rule learning. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '19, 223–232. New York, NY, USA: ACM.

[Gligorijevic et al. 2019] Gligorijevic, J.; Gligorijevic, D.; Pavlovski, M.; Milkovits, E.; Glass, L.; Grier, K.; Vankireddy, P.; and Obradovic, Z. 2019. Optimizing clinical trials recruitment via deep learning. *Journal of the American Medical Informatics Association*.

[Graves et al. 2016] Graves, A.; Wayne, G.; Reynolds, M.; Harley, T.; Danihelka, I.; Grabska-Barwińska, A.; Colmenarejo, S. G.; Grefenstette, E.; Ramalho, T.; Agapiou, J.; et al. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature* 538(7626):471.

[Hochreiter and Schmidhuber 1997] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9.

[Hurtado-Chong et al. 2017] Hurtado-Chong, A.; Joeris, A.; Hess, D.; and Blauth, M. 2017. Improving site selection in clinical studies: a standardised, objective, multistep method and first experience results. *BMJ open* 7(7):e014796.

[Johnson et al. 2017] Johnson, A. E.; Stone, D. J.; Celi, L. A.; and Pollard, T. J. 2017. The mimic code repository: enabling reproducibility in critical care research. *JAMIA*.

[Kingma and Ba 2014a] Kingma, D. P., and Ba, J. 2014a. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

[Kingma and Ba 2014b] Kingma, D. P., and Ba, J. 2014b. Adam: A method for stochastic optimization. *ICLR*.

[Kumar et al. 2016] Kumar, A.; Irsoy, O.; Ondruska, P.; Iyyer, M.; Bradbury, J.; Gulrajani, I.; Zhong, V.; Paulus, R.; and Socher, R. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, 1378–1387.

[Le, Tran, and Venkatesh 2018] Le, H.; Tran, T.; and Venkatesh, S. 2018. Dual memory neural computer for asynchronous two-view sequential learning. In *Proceedings of the 24rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1637–1645. ACM.

[Miller et al. 2016] Miller, A.; Fisch, A.; Dodge, J.; Karimi, A.-H.; Bordes, A.; and Weston, J. 2016. Key-value memory networks for directly reading documents. In *Empirical Methods in Natural Language Processing*, 1400–1409.

[Paszke et al. 2017] Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch.

[Potter et al. 2011] Potter, J. S.; Donovan, D. M.; Weiss, R. D.; Gardin, J.; Lindblad, R.; Wakim, P.; and Dodd, D. 2011. Site selection in community-based clinical trials for substance use disorders: Strategies for effective site selection. *The American journal of drug and alcohol abuse* 37(5):400–407.

[Schapire 1999] Schapire, R. E. 1999. A brief introduction to boosting. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'99, 1401–1406. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

[Shang et al. 2019a] Shang, J.; Ma, T.; Xiao, C.; and Sun, J. 2019a. Pre-training of graph augmented transformers for medication recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, 5953–5959. AAAI Press.

[Shang et al. 2019b] Shang, J.; Xiao, C.; Ma, T.; Li, H.; and Sun, J. 2019b. Gamenet: Graph augmented memory networks for recommending medication combination. *AAAI*.

[Sukhbaatar et al. 2015] Sukhbaatar, S.; Weston, J.; Fergus, R.; et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, 2440–2448.

[van den Bor et al. 2017] van den Bor, R. M.; Grobbee, D. E.; Oosterman, B. J.; Vaessen, P. W.; and Roes, K. C. 2017. Predicting enrollment performance of investigational centers in phase iii multi-center clinical trials. *Contemporary clinical trials communications* 7:208–216.

[Vaswani et al. 2017] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

[Weston, Chopra, and Bordes 2015] Weston, J.; Chopra, S.; and Bordes, A. 2015. Memory networks. In *ICLR*.

[Xiao et al. 2018] Xiao, C.; Ma, T.; Dieng, A.; Blei, D. M.; and Wang, F. 2018. Readmission prediction via deep contextual embedding of clinical concepts. *PLOS ONE* 13(4):1–15.

[Xiao, Choi, and Sun 2018] Xiao, C.; Choi, E.; and Sun, J. 2018. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics*.

[Xu et al. 2018] Xu, X.; Fu, Y.; Xiong, H.; Jin, B.; Li, X.; Hu, S.; and Yin, M. 2018. Dr. right!: Embedding-based adaptively-weighted mixture multi-classification model for finding right doctors with healthcare experience data. In *ICDM*.

Appendix

Data Preprocessing

- **Investigator Data**: We group different treatments done by investigators over time based on time. This creates a temporal view of the treatments performed by the investigator over time. The temporal view of the investigator consists of all the diagnoses, procedures, medication codes prescribed to patients. We combine the diagnosis, procedure, medication codes into a single treatment. These codes are further converted to multi-hot representations which are used for the input for the model. Other information such as physician's location, specialty area, professional certifications, etc is also used for the input for the model. We provide the details of these features in the supplementary section.

- **Clinical Trials Data**: Clinical trials data were preprocessed to have combine information about the trial from both private and public sources. Our private dataset contains information about the success of the clinical trials. It also contains information about the pairing of investigators with the trials information. Similar to the investigator data, there are lot of variables which are categorical in nature and some variables are textual in nature. We convert the categorical variables into multi-hot representation after removing null values. We combine the public information about the trials with the private information using the a global ID. This helps us get more context about the clinical trials. The textual information is converted continuous value embeddings using BERT(Bidirectional Encoder Representations from Transformers). We combine all these information to provide as the input for the clinical trial embedding module. We will provide the details about the different variables in the supplementary section of this manuscript.

Notations

- Definition 1(Patients): We denote the patients as

$$P(n) = x_1^{(n)}, x_2^{(n)}, ...x_T^{(n)} \qquad (15)$$

where $n \in 1,2,....N$ patients in the cohort and T is the number of visits by individual patients.

Each visit x_1^N is the combination of diagnosis, medication, procedure codes c_d, c_p, c_m.

- Definition 2(Providers):

- Definition 3(Clinical Trials):

Enrollment Rate Definition

Enrollment rate is defined as

$$\text{Enrollment Rate} = \frac{\text{subjects randomized} - \text{subjects discontinued}}{\text{enrollment window}} \qquad (16)$$

Baselines

Evaluation Metrics

1. Precision: Here precision measures the fraction of relevant doctors among the recommended doctors.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

2. Recall: Here recall measures the fraction of relevant doctors that have been recommended over the total amount of qualified doctors.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

3. PR-AUC: Area Under the Precision-Recall Curve.

$$\text{PR-AUC} = \sum_{k=1}^{n} \text{Prec}(k)\Delta\text{Rec}(k),$$

where k is the k-th precision and recall operating point $(\text{Prec}(k), \text{Rec}(k))$.

Doctor2Vec Model Hyper parameter details

In this section, we provide details about model hyperparameters:

Bi-LSTM:

- Activation function:tanh
- bias usage: True
- Number of hidden units: 124
- kernel Regularizer: L2 regularizer

MLP for categorical variables of clinical data:

- Number of layers: 4
- Dimension of layers: [128,256,128,64]

MLP in memory network:

- Number of layers: 3
- Dimension of layers: [128,128,64]

BERT background and formulation

Here we provide more details about the BERT formulation used in our method. We also provide details about pretraining performed. BERT model is based on a multi-layer Transformer encoder (Vaswani et al. 2017) and it is usually BERT is pre-trained using two unsupervised tasks (1) Masked Language Model and (2) Next Sentence Prediction.

Input formatting: Since BERT is trained on specific type of symbols, we need to format our input data in that format to be used with the pretrained model. We use special tokens to mark the beginning ([CLS]) and separation/end of sentences ([SEP]). This also requires token IDs from BERTs tokenizer and mask IDs to indicate which elements in the sequence are tokens and which are padding elements. BERT uses WordPiece tokenization method which splits words into multiple splits.

Transformer as BERT building block: The transformer model was proposed in Attention is all you need paper (Vaswani et al. 2017).

BERT model definition: BERT is basically a trained Transformer Encoder stack. BERT model has large number of encoder layers or Transformer blocks. Each transformer block contains a self attention block with feed forward blocks. Each encoder block has a residual connection around it. It is also followed by a layer-normalization step.

Pretraining for BERT: Pretraining of BERT is performed by two tasks (1) Masked LM (2) Next sentence prediction

Word embeddings from BERT: Pre-trained BERT can be used to create contextualized word embeddings. It has been shown that BERT based word embeddings are highly successful in downstream tasks. We extract the vectors from the last four layers of the BERT model. These vectors are of 768 dimension. Instead of concatenating these vectors, we averaged these vectors to obtain the final word embeddings which are of 768 dimension.

Baseline Hyperparameter and Implementation Details

The parameters are initialized as per the original paper. The model dimensionality is set to 768. We use the Adam optimizer with a learning rate of 0.0002 . The maximum sequence length supported by the model is set to 512, and the model is first trained using shorter sequences. The model is trained using a maximum sequence length of 128 for 75,000 iterations on the masked language modeling and next sentence prediction tasks, with a batch size 32.The model is trained on longer sequences of maximum length 512 for an additional 75,000 steps with a batch size of 4.

- **Logistic Regression Hyperparameters**
  - Penalty function: L2
  - tolerance(stopping criteria): 1e-5
  - Inverse regularization strength: 1.2
  - Bias(fit intercept): True
  - class weight: Balanced
  - Solver: liblinear
  - maximum iteration: 3000

- **Random Forest Hyperparameters**
  - number of estimators: 12
  - criterion: Gini impurity
  - Maximum Depth: 16
  - Minimum samples split: 5
  - Minimum samples leaf: 3
  - Minimum weight fraction leaf: 0
  - Maximum Features: automatic selection
  - Maximum leaf nodes: 4

- **Adaboost Hyperparamters**
  - Booster: gbtree
  - learning rate: 0.2
  - Max Depth: 8
  - Minimum Child Weight: 1
  - Maximum Delta Step:0.4

  - lambda(L2 regularization): 0.3

- **MLP Hyperparameters**
  - Activation Function: ReLU
  - Number of Layers: 6
  - Size of Layers: [512, 512, 512, 256, 128, 64]
  - Optimizer: Adam
  - Learning Rate: 0.001

- **RNN Hyperparameters**
  - Activation Function: tanh
  - Number of layers: 2
  - Size of Layers: [128, 128]
  - Optimizer: Adam
  - Learning Rate: 0.001

- **DeepMatch Hyperparameters**
  - Medical concept embedding dense layer dimension: 200
  - Medical concept embedding activation: ReLU
  - Clinical trial embedding layer dimension: 300
  - Optimizer: Adam
  - Learning Rate: 0.001

BERT improvement compared to word2vec

Table 6: Text embedding method comparison

	PR-AUC	F1-score
Word2vec	0.801 ± 0.018	0.812 ± 0.024
char-CNN	0.819 ± 0.024	0.823 ± 0.038
BERT	0.861 ± 0.021	0.841 ± 0.032

MSE results for overall enrollment prediction

We have reported the mean squared error rates for the regression task here. Due to space constraint, this results is not presented in table 3.

Table 7: Doctor2Vec achieves the best performance in terms of MSE metrics for the regression task

	MSE
Median	1.220 ± 0.041
LR	28.210 ± 1.341
RF	0.981 ± 0.032
AdaBoost	0.736 ± 0.015
MLP	0.663 ± 0.148
LSTM	0.462 ± 0.047
DeepMatch	0.381 ± 0.023
Doctor2Vec	$\mathbf{0.221 \pm 0.038}$