Conditional Generation Net for Medication Recommendation

Rui Wu†
rhyswu@outlook.com
Southeast University
China

Zhaopeng Qiu
zhaopengqiu@tencent.com
Tencent
China

Jiacheng Jiang
acejjiang@tencent.com
Tencent
China

Guilin Qi
gqi@seu.edu.cn
Southeast University
China

Xian Wu†
kevinxwu@tencent.com
Tencent
China

ABSTRACT
Medication recommendation targets to provide a proper set of medicines according to patients’ diagnoses, which is a critical task in clinics. Currently, the recommendation is manually conducted by doctors. However, for complicated cases, like patients with multiple diseases at the same time, it’s difficult to propose a considerate recommendation even for experienced doctors. This urges the emergence of automatic medication recommendation which can help treat the diagnosed diseases without causing harmful drug-drug interactions. Due to the clinical value, medication recommendation has attracted growing research interests. Existing works mainly formulate medication recommendation as a multi-label classification task to predict the set of medicines. In this paper, we propose the Conditional Generation Net (COGNet) which introduces a novel copy-or-predict mechanism to generate the set of medicines. Given a patient, the proposed model first retrieves his or her historical diagnoses and medication recommendations and mines their relationship with current diagnoses. Then in predicting each medicine, the proposed model decides whether to copy a medicine from previous recommendations or to predict a new one. This process is quite similar to the decision process of human doctors. We validate the proposed model on the public MIMIC data set, and the experimental results show that the proposed model can outperform state-of-the-art approaches.

CCS CONCEPTS
• Information systems → Data mining; • Applied computing → Health informatics.

KEYWORDS
medication recommendation, electronic health record, generation

†This work was done when Rui Wu was an intern at Tencent.
†Xian Wu is the Corresponding Author.

ACM Reference Format:
Rui Wu, Zhaopeng Qiu, Jiacheng Jiang, Guilin Qi, and Xian Wu. 2022. Conditional Generation Net for Medication Recommendation. In Proceedings of the ACM Web Conference 2022 (WWW ’22), April 25–29, 2022, Virtual Event, Lyon, France. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3485447.3511936

1 INTRODUCTION

Figure 1: An example of electronic health record.

Figure 2: The histogram of Jaccard between current medications and historical medications (left) and the histogram of the proportion of current medications that occur in past visits (right).

Medication recommendation aims to provide a set of medicines to treat the set of diagnosed diseases of a patient. Take the patient in Figure 1 for example, this patient visits the hospital for three times. In each visit, this patient is diagnosed with a set of diseases and is prescribed a set of medicines to treat these diseases. Take the patient in Figure 1 for example, this patient visits the hospital for three times. In each visit, this patient is diagnosed with a set of diseases and is prescribed a set of medicines to treat these diseases. Currently, medication recommendation is mainly conducted by doctors according to their expertise and experiences. However, many patients are diagnosed with multiple diseases at one time. To conduct a considerate medication recommendation, on one hand, the doctor...
needs to select proper medicines for each disease; on the other hand, the doctor needs to avoid harmful drug-drug interactions (DDI) among selected medicines. Therefore, for complicated cases, medication recommendation is time-consuming for experienced doctors and error-prone for inexperienced doctors. To address this problem, the automatic medication recommendation that can assist doctors in decision making is urged.

Due to the clinical value, medication recommendation has attracted growing research interests. A series of deep learning based medication recommendation methods [27, 28] have been proposed, which can be divided into two categories: Instance-based models [7, 18, 29] only use patient’s current diagnoses and procedures to conduct recommendations while ignoring the longitudinal patient history. In this manner, instance-based based models fail to consider the historical disease development process. To address this issue, Longitudinal models [14, 19, 27, 28] are designed to take use of the longitudinal patient history and capture the temporal dependencies. Existing longitudinal models usually consist of two stages, first aggregating the known information into a patient-level representation, and then conducting medication recommendation based on it.

One problem of existing works is that they do not explicitly model the relationship between medication recommendations for the same patient. However, in clinical practice, the recommendations for the same patients are closely related. For example, for patients with chronic diseases, they may keep using the same medicine all their lives. As shown in Figure 1, we conduct a statistical analysis on the MIMIC-III dataset. For each visit, we calculate the proportion of medications that have appeared in history and the Jaccard between current medications and past medications. We can see that in the most visits, a large portion of prescribed medicines have been recommended before. Inspired by this, we rethink about taking use of historical information from a medication-level perspective. The challenge here is how to accurately determine whether a historical medication is still relevant at present.

In this paper, we propose an encoder-decoder based generation network to produce the appropriate medications in a sequential manner, named Conditional Generation Net (COGNet). The proposed COGNet consists of the basic model and the copy module. The basic model conducts recommendation only based on patients’ health conditions in current visit; the copy module introduces the information of historical visits in modeling. Different from the basic model, in generating each medicine, the copy module decides whether to copy a medicine from historical recommendations or to predict a new one. The experiments on a public dataset demonstrate the effectiveness of the proposed model. We summarize our major contributions as follows:

- We conduct comprehensive experiments on a public dataset MIMIC-III to demonstrate the effectiveness of the proposed COGNet.

2 PROBLEM FORMULATION

2.1 Electrical Health Records (EHR)

The basic unit of EHR is patient and each patient consists of several visits. Let \( \mathcal{R} = \{\mathcal{V}^{(i)}\}_{i=1}^{N} \) denote a longitudinal EHR of \( N \) patients. Each patient has visited the hospital as least once and the \( i \)-th patient can be represented as a sequence of multivariate observations \( \mathcal{V}^{(i)} = [\mathcal{V}^{(i)}_1, \mathcal{V}^{(i)}_2, \ldots, \mathcal{V}^{(i)}_T] \). Here \( T \) is the number of visits of the \( i \)-th patient. To simplify the notation, we ignore the index \( i \) and describe our methods with a single patient, then a patient is represented as \( \mathcal{V} = [\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_T] \). Let \( \mathcal{D} = \{d_1, d_2, \ldots, \} \) denotes the set of diagnoses, \( \mathcal{P} = \{p_1, p_2, \ldots, \} \) denotes the set of procedures and \( \mathcal{M} = \{m_1, m_2, \ldots, m_{|\mathcal{M}|}\} \) denotes the set of medications. \( |\mathcal{D}|, |\mathcal{P}| \) and \( |\mathcal{M}| \) indicate the number of all possible diagnoses, procedures and medications, respectively. Then, each visit of the patient can be represented by \( \mathcal{V}_t = (\mathcal{D}_t, \mathcal{P}_t, \mathcal{M}_t) \), where \( \mathcal{D}_t \subseteq \mathcal{D}, \mathcal{P}_t \subseteq \mathcal{P} \) and \( \mathcal{M}_t \subseteq \mathcal{M} \).

2.2 EHR-DDI Graph

\( G_e = (\mathcal{M}, \mathcal{E}_e) \) and \( G_d = (\mathcal{M}, \mathcal{E}_d) \) denote the EHR graph and DDI graph respectively, where \( \mathcal{E}_e \) is all the possible medication combinations in \( \mathcal{R} \) and \( \mathcal{E}_d \) is the known DDIs. Formally, we use the adjacency matrix \( A_e, A_d \in \mathbb{R}^{|\mathcal{M}| \times |\mathcal{M}|} \) to illustrate the construction of the graphs. \( A_e[i, j] = 1 \) means the \( i \)-th and \( j \)-th medications have appeared in the same visit. For \( A_d \), only the pair-wise DDIs are considered. \( A_d[i, j] = 1 \) means the \( i \)-th and \( j \)-th medications are mutually interacted. \( G_e \) and \( G_d \) are the same for all patients.

2.3 Medication Recommendation Problem

Given a patient’s current diagnoses \( \mathcal{D}_t \), procedures \( \mathcal{P}_t \), historical visit information \( [\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_{t-1}] \), and the EHR and DDI graphs \( G_e \) and \( G_d \), the goal is to train a model (i.e., COGNet) which can recommend the proper medication combination \( \mathcal{M}_t \) for this patient.

Table 1: Notations used in COGNet

| Notation | Description |
|----------|-------------|
| \( \mathcal{R} \) | Electronic Health Records |
| \( \mathcal{V} \) | Record for a single patient |
| \( \mathcal{D}, \mathcal{P}, \mathcal{M} \) | Diagnoses, Procedure and Medication set |
| \( G_e \) | EHR or DDI Graph |
| \( \mathcal{E}_e, A_e \) | Edge set and adjacency matrix of graph \( G_e \) |
| \( E_e, E_d \in \mathbb{R}^{|\mathcal{M}| \times |\mathcal{M}|} \) | Origin embedding tables |
| \( E_d \in \mathbb{R}^{|\mathcal{M}| \times |\mathcal{M}|} \) | Graph embedding for medications |

3 FRAMEWORK

Figure 3 illustrates the architecture of COGNet. We employ an encoder-decoder based generative network to generate the appropriate medication combination based on the patient’s current and
Figure 3: An overview of our proposed COGNet. The basic model recommend the medication only based on the patient’s health condition in current visit. The other part, named copy module, considers the information of patient's historical visits.

3.1 Input Representations
We design three embedding tables, $E_d \in \mathbb{R}^{|D| \times s}$, $E_p \in \mathbb{R}^{|P| \times s}$, and $E_m \in \mathbb{R}^{|M| \times s}$, where each row is an embedding vector for a distinct diagnosis code, procedure code or medication code. Here $s$ denotes the dimension of the embedding space. For each diagnosis set $D_t$ ($t \in [1, t]$), we first convert its every element $d \in D_t$ to a $s$-dimensional vector $d$ via the embedding matrix $E_d$. Then, we can obtain the representation of the diagnosis set $D_t \in \mathbb{R}^{|D_t| \times s}$. For each procedure set $P_t$ ($t \in [1, t]$) and medication set $M_j$ ($j \in [1, t-1]$), we can also obtain their representations $P_t \in \mathbb{R}^{|P_t| \times s}$ and $M_j \in \mathbb{R}^{|M_j| \times s}$ via the embedding matrices $E_p$ and $E_m$, respectively.

3.2 Basic Model
In the basic model, we recommend the medication combination only based on the patient’s health condition in current visit.
basic model is an encoder-decoder generation model. It consists of four modules: diagnosis encoder, procedure encoder, medication graph encoder, and medication combination decoder.

3.2.1 Diagnosis Encoder. The diagnosis encoder aims to represent the patient’s health conditions based on the diagnosis codes. It is a Transformer-based [22] network and has the following two major sub-layers.

Multi-Head Self-Attention. This sub-layer aims to capture relations among all diagnoses in the same visit. Given three input matrices $Q \in \mathbb{R}^{LQ \times s}$, $K \in \mathbb{R}^{Lk \times s}$ and $V \in \mathbb{R}^{Lv \times s}$ where $L_k = L_v$, the attention function is defined as:

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d}})V$$  \hspace{1cm} (1)

Multi-head attention layer $\text{MH}(:,:, :) = \text{softmax}^\top \text{O}^3 \text{W}^O$ will further project the input to multiple representation subspaces and capture the interaction information from multiple views [17].

$$\text{MH}(Q, K, V) = [\text{head}_1; \ldots; \text{head}_h]W^O$$ \hspace{1cm} (2)

$W^Q_i, W^K_i, W^V_i \in \mathbb{R}^{s \times s/h}$ and $W^O \in \mathbb{R}^{s \times s}$ are the parameters to learn. $h$ is the number of heads.

Position-wise Feed-Forward. This sub-layer consists of two linear projections with a ReLU activation in between.

$$\text{FFN}(H) = \text{ReLU}(HW^F + b^F)W^F + b^F$$ \hspace{1cm} (3)

where $W^F_1 \in \mathbb{R}^{s \times s}$, $W^F_2 \in \mathbb{R}^{s \times s}$, $b^F \in \mathbb{R}^s$ and $b^F \in \mathbb{R}^s$ are trainable parameters.

The diagnosis encoder then employs the residual connection and layer normalization function $\text{LayerNorm}(\cdot)$ defined in [3] around the above two sub-layers to extract the diagnosis representation:

$$D'_t = \text{Enc}_d(D_t) = \text{LayerNorm}(H + \text{FFN}(H))$$ \hspace{1cm} (4)

where $H = \text{LayerNorm}(D_t + \text{MH}(D_t, D_t, D_t))$

$D'_t \in \mathbb{R}^{|D_t| \times s}$ is the diagnosis representation of the $t$-th visit encoded by this encoder.

3.2.2 Procedure Encoder. The procedure encoder has the same network structure as the diagnosis encoder but their parameters are different. Through the same encoding process, we can also obtain the procedure representation $P'_t$ via the procedure encoder:

$$P'_t = \text{Enc}_p(P_t)$$ \hspace{1cm} (5)

3.2.3 Medication Graph Encoder. This encoder aims to model two kinds of drug relations:

- Medication EHR co-occurrence relationship: Some medications are often prescribed together for better efficacy. For example, ranitidine and sucralfate are the popular drug combination used to treat acute gastritis. Hence, modeling these co-occurrence relations can help the decoder to recommend the drug according to the partially generated medications.

- DDI: As mentioned above, some medications have the DDI and cannot be used together. When recommending the drug, the decoder should avoid that it is conflict with the past recommended drugs. Hence, modeling this relation can help to recommend a safe and effective medication combination.

Inspired by [19], we use the Graph Convolutional Network [12] to model two kinds of relations based on the medication EHR and DDI graphs.

Given the input medication features $X \in \mathbb{R}^{|M| \times s}$ and the medication graph adjacency matrix $A \in \mathbb{R}^{|M| \times |M|}$, the GCN($\cdot$, :) will obtain the new medication representations as follows:

$$\text{GCN}(X, A) = \sigma((A^{1/2}\hat{A}^{1/2}X)$$ \hspace{1cm} (6)

$\hat{A} = A + I$, $I$ is the identity matrix and $\hat{O}$ is the diagonal node degree matrix of $\hat{A}$ (i.e., $O_{ii} = \sum_j A_{ij}$).

First, we use a two-layer GCN to model the medication EHR co-occurrence relations based on the EHR graph adjacency matrix $A_e$.

$$G_e = \text{GCN}((\text{RelU})(\text{GCN}(E_m, A_e))W^g_e, A_e)$$ \hspace{1cm} (7)

where $W^g_e$ is the learnable parameter.

Then, we use another two-layer GCN to model the DDI relations based on the DDI graph $A_d$.

$$G_d = \text{GCN}((\text{RelU})(\text{GCN}(E_m, A_d))W^g_d, A_d)$$ \hspace{1cm} (8)

where $W^g_d$ is the learnable parameter.

Finally, we fuse two medication relation representations $G_e$ and $G_d$ to obtain the relation-aware medication representations.

$$E_g = G_e - \lambda G_d$$ \hspace{1cm} (9)

where $\lambda$ is a learnable parameter.

3.2.4 Medication Combination Decoder. The decoder will recommend the medication one by one for the $t$-th visit. For example, at the $i$-th decoding step, the decoder will recommend the next medication $M_{t,i}$ based on the partially generated medication combination $\{M_{t,0}, \ldots, M_{t,i-1}\}$ and the patient health conditions in the $t$-th visit (i.e., $D'_t$ and $P'_t$). The decoding process is as follows.

First, we convert all generated medications to vectors via the original embedding matrix $E_m$ and relation representation matrix $E_g$ and can obtain two representations of the partially generated medication combination $\hat{M}^m_t$ and $\hat{M}^g_t$. We fuse two representations to obtain the relation-aware medication combination representation.

$$\hat{M}_t = \hat{M}^m_t + \hat{M}^g_t$$ \hspace{1cm} (10)

Then, we use the multi-head self-attention mechanism (defined in Eq.(2)) to capture the interactions among the recommended medications.

$$\hat{M}_t = \text{LayerNorm}(\hat{M}_t + \text{MH}(\hat{M}_t, \hat{M}_t, \hat{M}_t))$$ \hspace{1cm} (11)

Intuitively, the medication recommendation task aims to generate the recommendation of drugs that can cover all diseases of the patient. Thus, the patient’s disease and procedure representations $D'_t$ and $P'_t$ play an auxiliary role during the decoding process. To this end, we align the patient’s health conditions to adaptively model the uncovered diseases to guide the next medication recommendation.

$$\hat{M}_t = \text{LayerNorm}(\hat{M}_t + \text{MH}(\hat{M}_t, D'_t, D'_t) + \text{MH}(\hat{M}_t, P'_t, P'_t))$$ \hspace{1cm} (12)

Finally, we use the last row of $\hat{M}_t$ (i.e., $\hat{M}_{t,i-1}$) to predict the $i$-th medication via an MLP layer.

$$P_{r, i} = \text{Softmax}(\hat{M}_{t,i-1}W^g_r + b^g_r)$$ \hspace{1cm} (13)
where $W_g \in \mathbb{R}^{x|M|$} and $b_g \in \mathbb{R}^{1|M|$} are learnable parameters. $Pr_g$ denotes the probabilities of all medications in the vocabulary in which the medication with the maximum probability is the predicted medication at step $i$.

### 3.3 Copy Module

In Basic Model section, we introduce the encoder-decoder based model to generate the medication recommendation results based on the patient’s current health conditions. It doesn’t consider the information of patient’s historical visits. In this section, we design a novel copy module to extend the basic model, which first compares the health conditions of current and historical visits and then copies the reusable medications to prescribe for current visit according to the condition changes.

Since the patient may have multiple visits, we use the hierarchical selection mechanism to conduct the copy process at each decoding step. First, we use the visit-level selection to pick a similar visit by comparing their health conditions. Then, at the specific decoding step, we use the medication-level selection to pick a particular medication from the prescriptions of the selected visit and add it to the recommendation results.

Finally, we repeat the above process to copy the reusable medications iteratively to form the complete recommendation medication list. Note that the selection process is “soft” like the attention mechanism [26] and assigns different probabilities for all visits/medications to highlight the choice. We will take the $i$-th medication recommendation of the $t$-th visit (i.e., $M_{t,i}$) as an example to illustrate the copy process.

#### 3.3.1 Visit-level Selection

We first uses two gated aggregation layers to encode the visit-level health conditions of all visits by aggregating their diagnosis and procedure representations, respectively:

$$v_f^d = \text{Softmax}(\text{tanh}(D_f^t W_d^1 + b_d^1)W_d^2 + b_d^2)^T D_f^t$$

$$v_f^p = \text{Softmax}(\text{tanh}(P_f^t W_p^1 + b_p^1)W_p^2 + b_p^2)^T P_f^t$$

where $W_d^1 \in \mathbb{R}^{x \times k}$, $b_d^1 \in \mathbb{R}^k$, $W_d^2 \in \mathbb{R}^{k \times 1}$ and $b_d^2 \in \mathbb{R}$ are trainable parameters.

Then we calculate the visit-level selection score of the past $j$-th visit ($1 \leq j \leq t-1$) by measuring the similarity between it and the current $t$-th visit.

$$c_j = \text{Softmax}(\frac{v_f^d \cdot v_f^j + v_p^d \cdot v_p^j}{\sqrt{2}})$$

$c_j \in \mathbb{R}$ denotes the selection score of the $j$-th visit.

#### 3.3.2 Medication-level Selection

We use the hidden state $M_{t-1}^f$ formulated in Eq.(12), which comprehensively encodes the information of diagnoses, procedures and recommended medications, to determine which historical medications $m \in \bigcup_{j=1}^{t-1} M_t$ are reusable in current situation.

We firstly obtain the medication representations of all past visits through the similar process in Section 3.2.1:

$$M_j^m = \text{Enc}_m(M_j)$$

where $M_j \in \mathbb{R}^{|M_j| \times x}$.

Then, we use the hidden state $M_{t-1}^f$ as the query vector to calculate the selection score along the medication dimension. The selection score of $k$-th medication in $j$-th visit $M_{j,k}$ ($1 \leq j \leq t-1$ and $1 \leq k \leq |M_t|$) is

$$q_{j,k} = \frac{\exp(\hat{q}_{j,k})}{\sum_{j=1}^{t-1} \sum_{k=1}^{|M_j|} \exp(\hat{q}_{j,k})}$$

$$\hat{q}_{j,k} = \frac{(M_{t-1,j}^f W_c) \cdot M_{j,k}}{\sqrt{2}}$$

where $W_c \in \mathbb{R}^{x \times 1}$ is a learnable parameter. $q_{j,k} \in \mathbb{R}$ denotes the medication-level selection score.

#### 3.3.3 Copy Mechanism

We use the combination of visit-level and medication-level scores to determine the copy probability of each medication in past visits. Moreover, since a medication $m_i \in M$ may be used in multiple prescriptions, we gather its final copy probability as follows:

$$p^c_c = \frac{\hat{p}^c_c(\hat{c}, c)}{\sum_{i=1}^{|M|} \hat{p}^c_c(\hat{c}, c)}$$

$$\text{where } \hat{p}^c_c(i, j) = \sum_{j=1}^{t-1} \sum_{k=1}^{|M_t|} q_{j,k} \ast c \ast 1_{\{M_{j,k} = m_i \}}$$

Finally, we combine the generation probabilities and copy probabilities to conduct the prediction.

$$Pr = w_g \ast Pr_g + (1 - w_g) \ast Pr_c$$

$$w_g = \text{Sigmoid}(M_{t-1}^f W_f + b_f)$$

where $W_g \in \mathbb{R}^{x \times 1}$ and $b_g \in \mathbb{R}$ are learnable parameters. $Pr$ denotes the probabilities of all medications in the medication set $M$ in which the medication with the maximum probability is the predicted $i$-th medication of the $t$-th visit.

### 3.4 Training

We train the model by minimizing regular cross-entropy loss:

$$\mathcal{L}(\theta) = - \sum_{R} \sum_{t} \sum_{l} \log Pr(M_{t,i} | V_{c,t}, D_t, P_t, M_{t-1}; \theta)$$

where $R$ is the training EHR set. $M_{t,i}$ is the $i$-th medication of the medication set of the $t$-th visit. $Pr(M_{t,i} | V_{c,t}, D_t, P_t, M_{t-1}; \theta)$ is the predicted probability of the $M_{t,i}$ and can be calculated by the Eq.(22). $\theta$ denotes all trainable parameters in COGNet.

During the training phase, we use the teacher-forcing to train the model. Specifically, when predicting the $i$-th medication, the model takes the real $(i-1)$-th medication rather than the predicted $(i-1)$-th medication as the input of the decoder. At the first step, the input of the decoder is set to a special token (START).

### 3.5 Inference

During the inference phase, inspired by the medication recommendation method [29] and some NLP generation methods [15, 16, 30], we use the beam search trick to improve the performance. Different from the greedy search which only selects one best candidate as the partially generated medication combination for each decoding...
Table 2: Performance Comparison on MIMIC-III. Best results are highlighted in bold.

| Model         | Jaccard | F1    | PRAUC | DDI    | Avg. # of Drugs |
|---------------|---------|-------|-------|--------|-----------------|
| LR           | 0.4865 ± 0.0021 | 0.6434 ± 0.0019 | 0.7509 ± 0.0018 | 0.0829 ± 0.0009 | 16.1773 ± 0.0942 |
| ECC          | 0.4996 ± 0.0049 | 0.6569 ± 0.0044 | 0.6844 ± 0.0038 | 0.0846 ± 0.0018 | 18.0722 ± 0.1914 |
| RETAIN       | 0.4887 ± 0.0028 | 0.6481 ± 0.0027 | 0.7556 ± 0.0033 | 0.0835 ± 0.0020 | 20.4051 ± 0.2832 |
| LEAP         | 0.4521 ± 0.0024 | 0.6138 ± 0.0026 | 0.6549 ± 0.0033 | 0.0731 ± 0.0008 | 18.7138 ± 0.0666 |
| DMNC         | 0.4864 ± 0.0025 | 0.6529 ± 0.0030 | 0.7580 ± 0.0039 | 0.0842 ± 0.0011 | 20.0000 ± 0.0000 |
| GAMENet      | 0.5067 ± 0.0025 | 0.6626 ± 0.0025 | 0.7631 ± 0.0030 | 0.0864 ± 0.0006 | 27.2145 ± 0.1141 |
| MICRON       | 0.5100 ± 0.0033 | 0.6654 ± 0.0031 | 0.7687 ± 0.0026 | 0.0641 ± 0.0007 | 17.9267 ± 0.2172 |
| SafeDrug     | 0.5213 ± 0.0030 | 0.6768 ± 0.0027 | 0.7647 ± 0.0025 | 0.0589 ± 0.0005 | 19.9178 ± 0.1604 |
| COGNet       | 0.5336 ± 0.0011 | 0.6869 ± 0.0010 | 0.7739 ± 0.0009 | 0.0852 ± 0.0005 | 28.0903 ± 0.0950 |

Table 3: Statistics of processed MIMIC-III.

| Item                              | Number       |
|-----------------------------------|--------------|
| # of visits / # of patients       | 14,995 / 6,350 |
| diag. / prod. / med. space size   | 1,958 / 1430 / 131  |
| avg. / max # of visits            | 2.37 / 29    |
| avg. / max # of diagnoses per visit | 10.51 / 128  |
| avg. / max # of procedures per visit | 3.84 / 50   |
| avg. / max # of medications per visit | 11.44 / 65  |
| total # of DDI pairs              | 448          |

step, the beam search algorithm selects multiple alternatives at each decoding step based on conditional probability.

4 EXPERIMENTS

In this section, we first introduce the experimental setups. Then we conduct some experiments to demonstrate the effectiveness of our COGNet model.

4.1 Dataset

We use the Medical Information Mart for Intensive Care (MIMIC-III)[10] dataset released on PhysioNet. It contains a total of 46520 patients and 58976 hospital admissions from 2001 to 2012. For a fair comparison, we use the data processing script released by [28] and take the processed data as our benchmark. Details of the processing can be found in Appendix. Table 3 shows some statistics on the processed data.

4.2 Baselines and Metrics

We evaluate the performance of our approach by comparing it with several baseline methods.

- LR, standard Logistic Regression;
- ECC [18], Ensembled Classifier Chain, which uses a multi-hot vector to encode the diagnosis and procedure sets and leverage multiple SVM classifiers to make multi-label prediction;
- RETAIN [6], which uses the attention and gate mechanism to improve the prediction interpretability;
- LEAP [29], which uses the LSTM based generation model to conduct medication recommendation based on the diagnose information;
- DMNC [14], which proposes a new memory augmented neural network model to improve the patient encoder;
- GAMENet [19], which further combines the memory network and graph neural network to recommend the medication combinations;
- MICRON [27], which first uses an recurrent residual learning model to predict the medication changes and then conducts medication recommendation based on the medication changes and the medication combination of the last visit;
- SafeDrug [28], which combines the drug molecular graph and DDI graph to predict the safe medication combination.

Following the previous medication recommendation work [19, 27, 28], we use Jaccard Similarity Score (Jaccard), Average F1 (F1), Precision Recall AUC (PRAUC), and DDI Rate as our evaluation metrics. Each metric is averaged over all patients. The metric definitions can be found in Appendix.

4.3 Result Analysis

4.3.1 Overall Comparison. Table 2 shows the results of all methods. Overall, our proposed model COGNet outperforms all baselines with the higher Jaccard, F1 and PRAUC. The performances of LR, ECC and LEAP are poor as they are instance-based models that only consider the diagnoses and procedures in the current visit. RETAIN, DMNC, GAMENet, SafeDrug and MICRON perform relatively better because they preserve longitudinal patient information in different ways. RETAIN and DMNC only encode the patients’ historical information, while GAMENet introduces additional graph information and SafeDrug incorporates the drug molecule structures in medication recommendation, resulting in a further performance improvement. MICRON also notices that some medications in the current visit keeps consistent with the last visit and uses the recurrent residual method to inherit the unchanged part. However, it fails to consider the correlations among the recommended medications and the recurrent like method is hard to model the long range visit information. Hence, COGNet performs better than MICRON. SafeDrug achieves a lower DDI rate by introducing the additional
To verify the effectiveness of each module of COGNet, we design the following ablation models:

### 4.4 Ablation Study

#### 4.4.1 Effect of number of visits.

To further explore whether our COGNet can better capture historical medication information, we investigate the impact of the number of visits on the performance of different models. Since most patients in MIMIC visit the hospital less than five times, we take the first five visits for each patient in the test set to conduct the analysis. As a comparison, we also take the two strongest baselines SafeDrug and MICRON, which also incorporate historical information, to conduct the analysis. Figure 4 shows the results. We can see that COGNet achieves relatively better performance with more visits, while the performance of SafeDrug almost stays flat and MICRON shows a decreasing trend. The reason may be that COGNet uses the attention based hierarchical selection mechanism, which can more effectively incorporate the information of past visits than RNN like mechanism used in SafeDrug. Moreover, MICRON iteratively updates the past medication combination to form the new medication set, which will be affected by the error accumulation problem.

#### 4.4.2 Ablation Study

To verify the effectiveness of each module of COGNet, we design the following ablation models:

- **COGNet w/o Copy**: We remove the copy module, which means changing the Eq.(22) to $P_r = P_{r_{gy}}$.
- **COGNet w/o $c_j$**: We maintain the copy module but remove the visit-level selection by changing the $q_{j,k}$ in Eq.(21) to $q_{j,k}$.
- **COGNet w/o $G$**: We remove the EHR and DDI graphs in decoder, which means changing the Eq.(10) to $M_t = M_{t_{ey}}$.
- **COGNet w/o $D$**: We remove the diagnoses information in each visit.
- **COGNet w/o $P$**: We remove the procedures information in each visit.
- **COGNet w/o BS**: We use the greedy search strategy in the inference phase rather than beam search.

Table 4 shows the results for the different variants of COGNet. As expected, the results of COGNet w/o Copy indicate that the copy mechanism brings a significant improvement to the basic model. COGNet can improve the medication recommendation by replicating historical drugs. COGNet w/o $c_j$ illustrates the effectiveness of visit-level selection scores.

Both COGNet w/o $D$ and COGNet w/o $P$ yield poor results among all ablation models, which suggest that diagnosis and procedure information play a great role in medication recommendation. The results of COGNet w/o $G$ and COGNet w/o BS indicate that graphs and beam search also have contributions to the final result. Overall, the complete COGNet outperforms all ablation models, which means each component of our model is integral.

#### 4.5 Case Study

We present an example patient in MIMIC-III to illustrate how COGNet conducts the copy mechanism to improve medication recommendation. This patient visited the hospital twice. At the first time, the patient was mainly diagnosed with pneumonia, sepsis, etc. Later, the patient returned for a second visit. In addition to the previously diagnosed diseases, the patient also had some new diseases, like gastroesophageal laceration and gastric...
5 RELATED WORK

5.1 Medication Recommendation

Due to the clinical value, medication recommendation has received increasing attention in recent years. According to the used information, existing approaches can be broadly categorized into rule-based, instance-based, and longitudinal methods.

5.1.1 Rule-based Methods. Rule-based methods [1, 5, 8, 13] rely on the human-designed recommendation protocols. For example, Gunlicks-Stoessel et al. [8] attempts to recommend proper treatments for adolescent depression based on rules. However, these methods require a lot of effort from clinicians and lack generalization.

5.1.2 Instance-based Methods. Instance-based methods [7, 29] only take the information of current visit as input. For example, Zhang et al. [29] firstly encode the patient’s current diagnosis and then use a recurrent decoder to generate the medication recommendations based on the encoded information. However, they ignore the historical visit information of the patient.

5.1.3 Longitudinal Methods. These approaches [4, 19, 24, 25, 28] use the historical information of patients and explore the sequential dependency between visits. Most of them basically model the longitudinal patient information by RNNs. Le et al. [14] and Shang et al. [19] combine memory networks with RNNs to enhance the memory ability. Yang et al. [28] further incorporate the drugs’ molecule information to improve the medication representation learning. Yang et al. [27] explicitly model the health condition changes of the patient to enhance the correlations between continuous visits by a recurrent residual learning approach. However, these methods do not explicitly consider the relationship between the medication recommendations of the same patient and the RNN like methods are hard to handle the long range visit dependency.

5.2 Graph Neural Network

Recently, graph neural networks (GNN) [2, 9, 12, 21, 23, 31] have received wide attention in many fields. The convolutional GNN can learn powerful node representations by aggregating the neighbors’ features over the graph. Some works [18, 28] have attempted to leverage the GNN to improve the medication recommendation. For example, Yang et al. [28] take the drugs’ molecule structures as graphs and use GNN to learn the better medication representations to improve medication recommendation. Shang et al. [19] use the GNN to encode the EHR and DDI graphs to introduce the medication correlation information. In this paper, inspired by [19], we use the GNN to encode the medication co-occurrence and DDI relationships to improve the recommendation.

6 CONCLUSION

In this paper, we proposed a novel medication recommendation model, COGNet, to better leverage historical information from a medication-level perspective. COGNet works under an encoder-decoder based framework and introduces a copy-or-predict mechanism to accurately determine whether a historical medication is still relevant at present. Experiment results on the publicly available MIMIC-III dataset demonstrate that COGNet outperforms existing medication recommendation methods. We also investigate the impact of number of visits on the performance, which shows that COGNet can effectively incorporate the information of multiple past visits. Further ablation study results also suggest the effectiveness of each module of COGNet.

ACKNOWLEDGMENTS

This work was supported by National Key R&D Program of China, No. 2018YFC0117000. This work was also partially supported by Natural Science Foundation of China grant, No. U21A20488.
A ADDITIONAL EXPERIMENTAL SETUPS

A.1 Dataset Processing
In this section, we elaborate the operation of data processing. The original data is from "ADMISSIONS.csv", "DIAGNOSES_ICD.csv", "PROCEDURES_ICD.csv" and "PRESCRIPTIONS.csv" files from the 1.40 version MIMIC-III. These tables are merged through admission id and subject id ("Admission" has the same meaning as "Visit" in this paper). We utilized all the patients with at least 2 visits. We prepared the available medications by retaining the top 300 medications in terms of number of occurrences. This is to improve the training speed for easier analysis. We extracted Top-40 severity DDI types from TWOSIDES [20], which are reported by ATC Third Level codes. In order to be able to compute DDI score, we transform the NDC drug codes to same ATC level codes. After the above operation was completed, we divided the data into training, validation and test by the ratio of \( \frac{2}{3} : \frac{1}{3} : \frac{1}{3} \). We counted the frequency of all drug occurrences on the training set, and then resort the medications in original data is from "ADMISSIONS.csv", "DIAGNOSES_ICD.csv", "PROCEDURES_ICD.csv" and "PRESCRIPTIONS.csv" files from the 1.40 version MIMIC-III. These tables are merged through admission id and subject id ("Admission" has the same meaning as "Visit" in this paper). We utilized all the patients with at least 2 visits. We prepared the available medications by retaining the top 300 medications in terms of number of occurrences. This is to improve the training speed for easier analysis. We extracted Top-40 severity DDI types from TWOSIDES [20], which are reported by ATC Third Level codes. In order to be able to compute DDI score, we transform the NDC drug codes to same ATC level codes. After the above operation was completed, we divided the data into training, validation and test by the ratio of \( \frac{2}{3} : \frac{1}{3} : \frac{1}{3} \). We counted the frequency of all drug occurrences on the training set, and then resort the medications in original data is from "ADMISSIONS.csv", "DIAGNOSES_ICD.csv", "PROCEDURES_ICD.csv" and "PRESCRIPTIONS.csv" files from the 1.40 version MIMIC-III. These tables are merged through admission id and subject id ("Admission" has the same meaning as "Visit" in this paper). We utilized all the patients with at least 2 visits. We prepared the available medications by retaining the top 300 medications in terms of number of occurrences. This is to improve the training speed for easier analysis. We extracted Top-40 severity DDI types from TWOSIDES [20], which are reported by ATC Third Level codes. In order to be able to compute DDI score, we transform the NDC drug codes to same ATC level codes. After the above operation was completed, we divided the data into training, validation and test by the ratio of \( \frac{2}{3} : \frac{1}{3} : \frac{1}{3} \). We counted the frequency of all drug occurrences on the training set, and then resort the medications in

A.2 Implementation Details
Our method is implemented by PyTorch 1.9.0 based on python 3.9.6, tested on an Intel Xeon 8255C machine with 315G RAM and 8 NVIDIA Tesla V100 GPUs. We choose the optimal hyperparameters based on the validation set, where the dimension size \( s = 64 \), number of beam search states is \( 4 \) and maximum generation length is \( 45 \). Models are trained on Adam optimizer [11] with learning rate \( 1 \times 10^{-4} \) and batch size 16 for 50 epochs. We fixed the random seed as 1203 for PyTorch to ensure the reproducibility of the models. During the test process, for a fair comparison, we apply bootstrapping sampling instead of cross-validation according to [28]. Precisely, we random sample 80% data from test set for a round of evaluation and the results of 10 rounds are used to calculate the mean and standard deviation, which are finally reported. As the sampling process is random, the final experimental results may vary slightly.

A.3 Metrics
In this section, we present the definition of each metric used in the experiment section.

- **Jaccard** for a patient is calculated as below:

\[
\text{Jaccard} = \frac{1}{T} \sum \frac{|M_i \cap \hat{M}_i|}{|M_i \cup \hat{M}_i|} \tag{25}
\]

where \( M_i \) is the ground-truth medication combination and \( \hat{M}_i \) is the predicted result.

- **F1** is the harmonic mean of precision and recall. For a patient, it is calculated as follows:

\[
\text{Precision}_i = \frac{|M_i \cap \hat{M}_i|}{|M_i|} \tag{26}
\]

\[
\text{Recall}_i = \frac{|M_i \cap \hat{M}_i|}{|\hat{M}_i|} \tag{27}
\]

\[
F1 = \frac{1}{T} \sum \frac{2 \times \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{28}
\]

\[
\text{PRAUC} = \frac{1}{T} \sum \sum \text{Precision}(k) \Delta \text{Recall}(k)_i \tag{29}
\]

\[
\Delta \text{Recall}(k)_i = \text{Recall}(k)_i - \text{Recall}(k - 1)_i
\]

where \( k \) is the rank in the sequence of drugs. \( \text{Precision}(k)_i \) is the precision at cut-off \( k \) in ordered retrieval list, and \( \Delta \text{Recall}(k)_i \) is the change in recall when deriving the \( k \)-th drug.

- **DDI** measure the interaction between the recommended medications, which is calculated by:

\[
\text{DDI} = \frac{1}{T} \sum \frac{\sum_{j=1}^{|M_i|} \sum_{k=j+1}^{|M_i|} 1(A_d(\hat{M}_i^{(j)}, \hat{M}_i^{(k)}) = 1)}{\sum_{j=1}^{|M_i|} \sum_{k=j+1}^{|M_i|} 1} \tag{30}
\]

where \( A_d \) is the adjacency matrix of DDI graph defined in section 4. \( \hat{M}_i^{(j)} \) denoted the \( j \)-th recommended medication and \( 1(\cdot) \) is a function that return 1 when expression in \( \cdot \) is true, otherwise 0.

B ADDITIONAL EXPERIMENTS

B.1 Performance with Different Label Order
Since new medications are recommended under the conditions of already recommended medications, the relative order of the medications may have an impact on the final results. To explore which kind of sequencing would be more appropriate of medication
recommendation, we conduct four different heuristics to order the data:

- **Rare first** ranks medications by their frequencies in the training data, placing the those that appear less frequently first.
- **Frequent first** put frequent medications before more rare medications.
- **Early first** sorts medications by chronological order, with those used earlier placed first.
- **Late first** arranges the medications in reverse chronological order.

Fig 6 shows the performance of COGNet with different ordering heuristics over training epochs. Rare first outperform the alternative labeling strategies, because it allows the model to focus more on unusual medications, thereby alleviating the data imbalance problem. Frequent first converges faster but performs poorly. The main reason is that the recommended medications are predominated by non-informative popular medications. The final results of Early first and Late first are weak, which indicates that chronological order is not a good choice in this task. In addition, we tried to randomly disrupt all the drugs, which is far worse than the above heuristics, so we do not show it.