Supplementary Online Information

Leveraging Graph-Based Hierarchical Medical Entity Embedding for Healthcare Applications

Tong Wu¹, Yunlong Wang¹, Yue Wang¹, Emily Zhao¹, and Yilian Yuan¹

¹Advanced Analytics, IQVIA Inc., Plymouth Meeting, Pennsylvania, USA, 19462

Corresponding Author:

Yunlong Wang
Advanced Analytics, IQVIA Inc.
1 IMS Dr., Plymouth Meeting, PA, USA, 19462
E-mail: Yunlong.Wang@iqvia.com
Tel: +1 (610) 729-3910
A Details of ME2Vec

A.1 Service Embedding
The details of learning medical service embedding vectors are given in Algorithm 1. The combinations function lists all the unique pairs of medical services within the segment $J^{(i)}_{\text{seg}}$. To embed medical services, we first obtain the adjacency matrix $A_{svc}$ from patient journeys and use it to generate biased random walks, then optimize the embeddings of medical services by maximizing the probability of each service “seeing” its neighbors in the walks via stochastic gradient descent (SGD).

Algorithm 1 Medical service embedding

| Input: Patient journeys $\{J^{(i)}\}^P_{i=1}$, context window length $T$, dimension $p$, walks per node $r$, walk length $l$, context size $k$ |
| Output: Service embedding $S \in \mathbb{R}^{S \times p}$ |

1: $A_{svc} \leftarrow 0 \in \mathbb{R}^{S \times S}$
2: for $i = 1$ to $P$ do
3: for $j = 1$ to $\lceil \frac{|J^{(i)}|}{T} \rceil$ do
4: $J^{(i)}_{\text{seg}} \leftarrow J^{(i)}\left[(j-1)T:jT\right]$
5: for $s_x,s_y$ in combinations($J^{(i)}_{\text{seg}}$, 2) do
6: $A_{svc}[s_x,s_y] += 1$
7: $A_{svc}[s_y,s_x] += 1$
8: $E_{svc} \leftarrow A_{svc}, G_{svc} \leftarrow \{S,E_{svc}\}$
9: $walks \leftarrow \{}$
10: for $\text{iter} = 1$ to $r$ do
11: for all nodes $s \in S$ do
12: $walk \leftarrow \text{BiasedRandomWalk}(G_{svc}, s, l)$
13: Append $walk$ to $walks$
14: $S \leftarrow \text{SGD}(k, p, walks)$
15: return $S$

A.2 Doctor Embedding
In this work, we deployed two layers of graph attention networks to enhance the learning capability. The structure of the two-level doctor embedding model is shown in Figure 1.

Figure 1. Structural diagram of a 2-level attention-based doctor embedding model. The embedding vectors of the medical services conducted by doctor $d_j$ are pre-trained in the step of service embedding.
As the proposed auxiliary task is a supervised classification, we configure the final layer of the model as a softmax. In addition, the output embedding from each of the $K$ attention heads of the second GraphAttenNet are averaged instead of concatenated, followed by the final softmax:

$$d'_j = \sigma \left( \frac{1}{K} \sum_{k=1}^{K} \sum_{i \in N_d} \alpha_{ij}^k W^k s_i \right).$$  \hspace{1cm} (1)

We summarize the steps for doctor embedding in Algorithm 2, where GraphAttenNet-2L denotes the operations of two GraphAttenNet stacked together; CrossEnt denotes cross-entropy loss; $L_{gt}$ and $L_{pred}$ represent the ground-truth and the predicted doctor specialties, respectively.

**Algorithm 2 Doctor embedding**

**Input:** Patient journeys $\{J^{(i)}\}_{i=1}^P$, service embedding $S$, number of attention heads $K$, learning rate $\eta$

**Output:** Doctor embedding $D \in \mathbb{R}^{|D| \times p'}$, aggregation functions $\{W^k, a^k\}_{k=1}^K$

1: $L_{gt} \leftarrow D \leftarrow G_{doc} \leftarrow \{J^{(i)}\}_{i=1}^P$
2: for all node $d \in D$ do
3: \quad $d_{init} \leftarrow \frac{1}{|N_d|} \sum_{i \in N_d} s_i (s_i \in N_d)$
4: \quad $D' \leftarrow \text{GraphAttenNet-2L}(D_{init}, S)$
5: \quad $L_{pred} \leftarrow \text{softmax}(D')$
6: while CrossEnt($L_{pred}, L_{gt}$) is large do
7: \quad $D' \leftarrow \text{GraphAttenNet-2L}(D_{init}, S)$
8: \quad $L_{pred} \leftarrow \text{softmax}(D')$
9: \quad $\{W^k, a^k\} \leftarrow \{W^k, a^k\} - \eta \nabla (\text{CrossEnt}(L_{pred}, L_{gt}))$
10: $D \leftarrow \text{GraphAttenNet}(D_{init}, S)$
11: return $D$, $\{W^k, a^k\}_{k=1}^K$

**A.3 Patient Embedding**

We summarize the steps for patient embedding in Algorithm 3.

**Algorithm 3 Patient embedding**

**Input:** Patient journeys $\{J^{(i)}\}_{i=1}^P$, service embedding $S$, doctor embedding $D$, learning rate $\eta$

**Output:** Patient embedding $P \in \mathbb{R}^{P \times p'}$, linear transformation parameters $\{W_a, b_a\}$

1: $G_{pat} \leftarrow \{J^{(i)}\}_{i=1}^P$
2: $G_{pat} \leftarrow \text{Duplication_Annotation}(G_{pat}, S, D)$
3: for all node $p \in P$ do
4: \quad $p_2(h_{s_i,d_j}\|p_k) = \frac{\exp(h_{s_i,d_j} \cdot p_k)}{\sum_{l \in \{h\}} \exp(h_l \cdot p_k)}$
5: \quad $\hat{p}_2(h_{s_i,d_j}\|p_k) = \frac{\sum_{l \in \{p_k\}} w_{p_k \rightarrow h_{s_i,d_j}}}{\sum_{l \in \{p_k\}} w_{p_k \rightarrow h}}$
6: while $D_{KL}(\hat{p}_2 \| p_2)$ is large do
7: \quad Algorithmic steps in lines 3–5
8: \quad $\{P, W_a, b_a\} \leftarrow \{P, W_a, b_a\} - \eta \nabla (D_{KL}(\hat{p}_2 \| p_2))$
9: return $P$, $\{W_a, b_a\}$

Note that $D_{KL}(\hat{p}_2 \| p_2)$ denotes the Kullback–Leibler (KL) distance or divergence between $\hat{p}_2$ and $p_2$. 

2/3
B Details of Experiments

B.1 Criteria of Choosing Patients into Negative Cohort
Following criteria are used for generating the negative cohort: CLL risk factor, diagnosis, procedure, and prescription. A patient must meet at least three of the criteria before being included into the negative cohort. The risk factors include anemia, chills, fatigue, fever, night sweats and pain, Sjögren’s syndrome, weakness, and weight loss. The related diagnoses include recurrent infection, Epstein-Barr infection, Helicobacter pylori infection, HIV/AIDS, Human T-lymphotrophic virus Type-I, rheumatoid arthritis, hypogammaglobulinemia, psoriasis, Wiskott-Aldrich syndrome, and pneumonia. The related procedures include tissue culture and chromosome analysis, increased frequency of CBC/ or blood test, and flow cytometry. The related prescriptions include Dexamethasone, Neupogen, and Prednisolone.

B.2 Using ME2Vec as Pretrained Input Embeddings for Recurrent Models
As shown in Figure 2, the recurrent model we used for this experiment is a two-layer LSTM with 256-dimensional hidden units and a 128-dimensional input embedding layer. The hidden outputs of the LSTM enter a global max-pooling layer that for each of the 256 dimensions, keeps the maximum value from all the time steps. The outputs of the global max-pooling layer are further processed by an multilayer perceptron (MLP) ended with a sigmoid function to make the final prediction. The recurrent model is trained for 30 epochs using an Adam optimizer with a batch size of 64 and learning rate of 1e–4.

Figure 2. Structure of the recurrent model for predicting diagnosis results from patients’ service sequences.