

# ARCH: Large-scale Knowledge Graph via Aggregated Narrative Codified Health Records Analysis

This document contains the supplementary material to the paper ‘‘ARCH: Large-scale Knowledge Graph via Aggregated Narrative Codified Health Records Analysis’’.

## S.1 Estimation of variance of PMI matrix

To estimate the standard deviation of  $\widetilde{\mathbf{V}}_i^\top \widetilde{\mathbf{V}}_j$ , denoting this estimator as  $\widehat{\sigma}_{ij}$ , we assume the following model:

$$X_{i,w}(t) \stackrel{i.i.d}{\sim} \text{Bernoulli}(1, p_w) \text{ for } 1 \leq w \leq d, 1 \leq t \leq T, 1 \leq i \leq n,$$

where  $X_{i,w}(t)$  is the indicator of the occurrence of concept  $w$  at time  $t$  for the  $i$ -th individual, and  $p_w$  is the marginal probability for the occurrence of concept  $w$ . This variance estimator is advantageous as it avoids the need to use patient-level data or apply the bootstrap algorithm, which can be computationally expensive and time-consuming. Furthermore, sharing patient-level data may not be feasible due to privacy and security concerns. Thus, our model is scalable, computationally economical, and avoids administrative challenges, making it easier to comply with privacy regulations.

Denote  $\mathbf{E} = \text{PPMI} - \text{PPMI}^*$ ,  $\mathbf{P} = \mathbb{U}_{n \times n}^* (\mathbb{U}_{n \times n}^*)^\top$ ,  $\widehat{\mathbf{P}} = \mathbb{U}^{(r)} (\mathbb{U}^{(r)})^\top$ ,  $\mathbf{M}$  as the difference of low-rank estimator of PPMI and true PPMI, where  $\text{PPMI}^*$  is true PPMI matrix and  $\mathbb{U}^*$  is the singular vectors of true PPMI matrix. Denote  $T$  as the average number of concepts per patient’s health record,  $q$  as the average number of concepts in one window size of one patient’s health record,  $p_i = \frac{\mathcal{C}(i, \cdot)}{\sum_{k=1}^n \mathcal{C}(k, \cdot)}$  as the marginal proportion for feature  $i$ . Then the estimated variance of the  $(i, j)$ th entry of the low-rank PPMI matrix can be computed with below equation:

$$\begin{aligned} \widehat{\text{Cov}}(\mathbf{M}_{ij}) &\approx \widehat{\text{Cov}}\left(\mathbf{e}_j^\top (\mathbf{P}\mathbf{E} + \mathbf{E}\mathbf{P})\mathbf{e}_i\right) \\ &= \left(\mathbf{P}\widehat{\text{Cov}}(\mathbf{E}_{\cdot i})\mathbf{P}\right)_{j,j} + \sum_{k=1}^n \mathbf{P}_{ki}^2 \left(\widehat{\text{Cov}}(\mathbf{E}_{\cdot k})\right)_{j,j} + \sum_{1 \leq k \neq l \leq n} \mathbf{P}_{ki}\mathbf{P}_{li} \left(\widehat{\text{Cov}}(\mathbf{E}_{\cdot k}, \mathbf{E}_{\cdot l})\right)_{j,j} \\ &\quad + \sum_{k=1}^n \left\{ \mathbf{P}_{ki} \left(\widehat{\text{Cov}}(\mathbf{E}_{\cdot k}, \mathbf{E}_{\cdot i})\mathbf{P}\right)_{j,j} + \mathbf{P}_{ki} \left(\mathbf{P}\widehat{\text{Cov}}(\mathbf{E}_{\cdot i}, \mathbf{E}_{\cdot k})\right)_{j,j} \right\}. \end{aligned} \tag{S.1}$$

Denote  $T_1 = Tq - \frac{q(q+1)}{2}$ , we have:

$$\begin{aligned} \widehat{\text{Cov}}(\mathbf{E}_{\cdot i}) &= \frac{1}{nT_1 p_i} \left( \mathbf{1}\mathbf{1}^\top \left(p_i - \frac{1}{2}\right) - \frac{1}{2} \mathbf{1}\mathbf{e}_i^\top - \mathbf{e}_i \mathbf{1}^\top \frac{1}{2} + \text{diag}(p_j^{-1}) \frac{1-p_i}{2} + \frac{1}{2p_i} \mathbf{e}_i \mathbf{e}_i^\top \right), \\ \widehat{\text{Cov}}(\mathbf{E}_{\cdot i}, \mathbf{E}_{\cdot j}) &= \frac{1}{nT_1} \left( \mathbf{1}\mathbf{1}^\top - \frac{1}{2p_i} \mathbf{1}\mathbf{e}_i^\top - \frac{1}{2p_j} \mathbf{e}_j \mathbf{1}^\top - \frac{1}{2} \text{diag}(p_j^{-1}) + \frac{1}{2p_i p_j} \mathbf{e}_j \mathbf{e}_i^\top \right). \end{aligned} \tag{S.2}$$

Once the entry of PPMI and the variance of the entry of PPMI are estimated, denoted by  $\text{PPMI}(i, j)$  and  $\widehat{\text{Cov}}(\text{PPMI}(i, j))$ , the  $p$ -value for testing whether  $\text{PPMI}^*(i, j)$  equals to zero from  $z$ -test is  $p_{ij} = 1 - \Phi(\text{PPMI}(i, j) / \widehat{\text{Cov}}(\text{PPMI}(i, j)))$ , where  $\Phi$  is the cumulative distribution function (CDF) of a standard normal distribution.

## 693 S.2 BH procedure under dependence

694 We have  $n$  hypotheses,  $H_{0,i}$  and  $p$ -values  $p_i$  for each. Assume that under  $H_{0,i} : p_i \sim U(0;1)$ . First  
 695 we order  $n$   $p$ -values  $p_{(1)} \leq p_{(2)} \leq \dots \leq p_{(n)}$  and let  $H_{(1)}, H_{(2)}, \dots, H_{(n)}$  be the corresponding  
 696 hypotheses. Then we compute  $k = \max_i \{i : p_{(i)} \leq \frac{i}{n(\ln(n) + 1)} \alpha\}$ , where  $\alpha$  is the target FDR.  
 697 Finally, we reject all  $H_{(i)}, i \leq k$ .

## 698 S.3 Parameter tuning for elastic net regularization

699 To obtain test residual, we obtained cooccurrence matrix from a different constitution and construct  
 700 its word representation  $\tilde{\mathbf{V}}^*$  in the same way above, and aligned  $\tilde{\mathbf{V}}$  and  $\tilde{\mathbf{V}}^*$  so that the  $i$ th row of  
 701  $\tilde{\mathbf{V}}$  represents the same concept as the  $i$ th row of  $\tilde{\mathbf{V}}^*$ . If the concept did not exist in the second  
 702 institution, we filled the corresponding row of  $\tilde{\mathbf{V}}^*$  with zeros. Then we computed

$$\begin{aligned} \lambda_j^A &= \arg \min_{\lambda} \left( \ln (\|\tilde{\mathbf{V}}_j - \tilde{\mathbf{V}}[, \mathbb{C}_j] \hat{\boldsymbol{\theta}}_j(\gamma, \lambda)\|) + \frac{N_j(\gamma, \lambda)}{p} \right), \\ \lambda_j^B &= \arg \min_{\lambda} \left( \ln (\|\tilde{\mathbf{V}}_j^* - \tilde{\mathbf{V}}^*[, \mathbb{C}_j] \hat{\boldsymbol{\theta}}_j(\gamma, \lambda)\|) + \ln (\|\tilde{\mathbf{V}}_j - \tilde{\mathbf{V}}[, \mathbb{C}_j] \hat{\boldsymbol{\theta}}_j(\gamma, \lambda)\|) + \frac{N_j(\gamma, \lambda)}{p} \right), \\ \lambda_j &= \begin{cases} \min\{\lambda_j^A, \lambda_j^B\} & , \text{ if } \mathbf{V}_j^{*\top} \mathbf{V}_j^* \neq 0 \\ \lambda_j^A & , \text{ if } \mathbf{V}_j^{*\top} \mathbf{V}_j^* = 0 \end{cases} \end{aligned}$$

703 where  $\{\lambda_j | j = 1, \dots, d\}$  are the final parameters we selected for elastic net regularization.

## 704 S.4 Data source

| CUI           |      |       |         |        |      |      |       |
|---------------|------|-------|---------|--------|------|------|-------|
| Semantic type | ACTI | CHEM  | DISO    | PHEN   | PHYS | PROC | Total |
| Number        | 353  | 12828 | 28282   | 1081   | 515  | 8364 | 51423 |
| Codes         |      |       |         |        |      |      |       |
| Class         | CCS  | Lab   | PheCode | RxNorm |      |      | Total |
| Number        | 224  | 6025  | 1776    | 1561   |      |      | 9586  |

Table 4: Number of features in each category.

## 705 S.5 Supervised learning for identifying drug side effects

The loss function for the supervised learning is defined below:

$$\begin{aligned} \mathcal{L}(M) &= \frac{1}{\alpha} \sum_i \log \left( 1 + \sum_{j \in \mathcal{P}_i} \exp(-\alpha(S_{ij} - \lambda)) \right) + \frac{1}{\beta} \sum_i \log \left( 1 + \sum_{j \in \mathcal{N}_i} \exp(\beta(S_{ij} - \lambda)) \right) , \\ \frac{\partial \mathcal{L}}{\partial M} &= - \sum_i \sum_{j \in \mathcal{P}_i} \frac{\mathbf{x}_i \mathbf{x}_j^\top \exp(-\alpha S_{ij})}{\exp(-\lambda \alpha) + \sum_{j \in \mathcal{P}_i} \exp(-\alpha S_{ij})} + \sum_i \sum_{j \in \mathcal{N}_i} \frac{\mathbf{x}_i \mathbf{x}_j^\top \exp(\beta S_{ij})}{\exp(\lambda \beta) + \sum_{j \in \mathcal{N}_i} \exp(\beta S_{ij})} , \end{aligned}$$

706 where  $S_{ij} = \mathbf{x}_i^\top M \mathbf{x}_j$ ,  $\mathbf{x}_i$  is the embedding vector of feature  $i$ , computed by  $\mathbf{x}_i = \mathbf{V}_i \hat{\sigma}_{ii}^{-1}$ ,  $\mathcal{P}_i$  is the  
707 set of side effects related with the  $i$ th drug in the training data set, and  $\mathcal{N}_i$  is the set of phenotypes  
708 unrelated with the  $i$ th drug in the training data set. Once  $M$  is obtained by minimizing the loss  
709 function above, we define  $S_{ij} = \mathbf{x}_i^\top M \mathbf{x}_j$  as the score of the pairs of drug  $i$  and side effects  $j$ .  
710 We then computed the AUC on the validation dataset to select the optimal parameters for each  
711 algorithm and demonstrate their performance on the test dataset. After optimizing the AUC for  
712 each algorithm on the validation dataset, we selected  $(\alpha, \beta, \lambda)$  to be  $(3, 3, 0)$  for ARCH(c) and  
713 SAPBERT,  $(3, 1, 0)$  for PubmedBERT and  $(1, 1, 0)$  for BioBERT.

## 714 S.6 Additional results

| pairs     | type         | group               | ARCH(c)      | ARCH(p)      | Pub   | Bio   | SAP          | num   |
|-----------|--------------|---------------------|--------------|--------------|-------|-------|--------------|-------|
| Code-Code | Similar      | PheCode Hierachy    | <b>0.970</b> | 0.901        | 0.612 | 0.566 | 0.764        | 4094  |
|           |              | Local Lab Mapping   | <b>0.834</b> | 0.797        | 0.652 | 0.640 | 0.788        | 1982  |
|           |              | summary             | <b>0.926</b> | 0.867        | 0.625 | 0.590 | 0.772        | 6076  |
|           | Related      | May Treat (Prevent) | <b>0.797</b> | 0.791        | 0.630 | 0.586 | 0.587        | 5129  |
|           |              | Classifies          | <b>0.906</b> | 0.860        | 0.667 | 0.631 | 0.784        | 4741  |
|           |              | ddx                 | <b>0.776</b> | 0.747        | 0.610 | 0.568 | 0.634        | 5938  |
|           |              | Causative           | <b>0.749</b> | 0.736        | 0.574 | 0.563 | 0.649        | 2873  |
| summary   | <b>0.810</b> | 0.786               | 0.624        | 0.588        | 0.662 | 18681 |              |       |
| CUI-Code  | Similar      | CUIPheCode          | <b>0.909</b> | 0.879        | 0.609 | 0.553 | 0.776        | 14096 |
|           |              | CUIRXNORM           | 0.993        | 0.980        | 0.993 | 0.993 | <b>0.997</b> | 1097  |
|           |              | CUILOINC            | <b>0.966</b> | 0.942        | 0.492 | 0.523 | 0.878        | 165   |
|           |              | CUICCS              | <b>0.982</b> | 0.958        | 0.875 | 0.790 | 0.972        | 63    |
|           |              | summary             | <b>0.916</b> | 0.887        | 0.636 | 0.585 | 0.794        | 15421 |
| CUI-CUI   | Similar      | Parent              | <b>0.864</b> | 0.860        | 0.679 | 0.608 | 0.819        | 39374 |
|           |              | Sibling             | 0.857        | <b>0.879</b> | 0.688 | 0.570 | 0.743        | 29752 |
|           |              | summary             | 0.861        | <b>0.869</b> | 0.683 | 0.592 | 0.786        | 69126 |
|           | Related      | May Treat (Prevent) | 0.799        | <b>0.834</b> | 0.675 | 0.557 | 0.547        | 10593 |
|           |              | Classifies          | <b>0.918</b> | 0.897        | 0.660 | 0.583 | 0.826        | 7666  |
|           |              | ddx                 | 0.803        | <b>0.845</b> | 0.670 | 0.560 | 0.613        | 6062  |
|           |              | Method_of           | <b>0.900</b> | 0.871        | 0.509 | 0.512 | 0.734        | 1702  |
|           |              | Causative           | <b>0.864</b> | 0.857        | 0.639 | 0.473 | 0.771        | 908   |
| summary   | 0.843        | <b>0.857</b>        | 0.658        | 0.559        | 0.661 | 26931 |              |       |

Table 5: AUCs of between-vector cosine similarity in detecting known similar pairs and related pairs with 1500-dimensional embedding from ARCH. Within each block, the last column shows the number of known-relation pairs within certain group of pairs.

