Collaborative Filtering on Keywords Recommendation for Clinical Trial Records

Xiao Zhou[1]

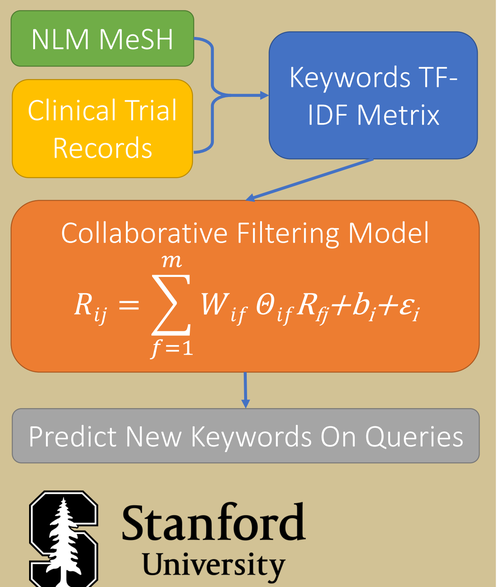
[1]*Standford Center for Continuous Development*

Stanford
CS229 Project

Abstract

Being able to discover similar clinical studies is critical for therapeutic product development. However, due to the complexity of medical research and terminology, a query may not contain all the necessary keywords, affecting the final search results. Here, a keyword-recommending algorithm using collaborative filtering is developed and examined. The recommended keywords can be used in the downstream document matching algorithm, providing a potentially more accurate way of retrieving similar documents.

Overview



Stanford University

Medical Keywords Extraction

A great challenge is to extract the keywords from medical documents. The challenge lies in (1) extract useful medical terms from a sequence of words (2) medical terms often have several synonyms or abbreviations. To tackle this issue, the Medical Subject Headings (MeSH) (meshb.nlm.nih.gov) were used. The MeSH thesaurus is a controlled and hierarchically-organized vocabulary for medical subjects. The MeSH terms in the thesaurus were transformed into a Trie data structure which was then used to scan text for extracting keywords.

TF-IDF Utility Matrix

The TF-IDF of MeSH terms in each clinical trial record (hereby referred to as ”document”) is calculated based on the following formula:

$$U_{ij} = \underbrace{\log_{10}(1 + \text{count of MeSH } i \text{ in document } j)}_{TF} \cdot \underbrace{\log_{10}\left(\frac{\text{total documents}}{\text{count of documents having MeSH } i}\right)}_{IDF}$$

and is organized into a utility matrix U, a matrix has m MeSH terms and n documents :

$$\begin{array}{c} \\ 1 \\ \vdots \\ i \\ \vdots \\ m \\ \\ \text{total n} \end{array} \begin{array}{c} \\ \square \\ \vdots \\ \square \\ \vdots \\ \square \\ \\ \end{array} \begin{array}{c} \\ \square \\ \vdots \\ \square \\ \vdots \\ \square \\ \\ \end{array} \begin{array}{c} \\ \cdots \\ \ddots \\ \cdots \\ \ddots \\ \cdots \\ \\ \end{array} \begin{array}{c} \\ \square \\ \square \\ \square \\ \square \\ \square \\ \\ \end{array}$$

Notice that U_{ij} is nonnegative.

Model Development

The hypothesis is that for MeSH terms i in document j, its TF-IDF can be estimated by a linear combination of the TF-IDF, of i’s neighboring MeSH terms in document j. That is

$$R_{ij} = \underbrace{\sum_{f=1}^{m} W_{if}\Theta_{if}R_{fj} + \vec{b}_i}_{\hat{R}_{ij}} + \vec{\epsilon}_i \qquad W_{if} = e^{\tau[1-cos(R_{i\cdot},R_{f\cdot})]^k}$$

W is a neighbor weighting matrix. Since TF-IDF cannot be negative, a ReLu is added to activate $\hat{R}_{ij}$, and the Frobenius norm is used to regulate $\boldsymbol{\Theta}$. The loss function then can be written as

$$J(\Theta, \vec{b}) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(\text{ReLu}(\sum_{f=1}^{m} W_{if}\Theta_{if}R_{fj} + \vec{b}_i) - R_{ij}\right)^2 + \frac{1}{2}\lambda\sum_{i=1}^{m}\sum_{j=1}^{m}\Theta_{ij}^2$$

To deal with the large data size, mini-batch gradient descent is used, and the algorithm is shown below:

$\Theta \leftarrow 0$
$epochCount \leftarrow 0$
for $epochCount < maxNumberOfEpoch$ **do**
  $//maxNumberOfEpoch$ needs to be determined by experiments
  for B, *a batch of column vectors from* R **do**
    $\hat{B} \leftarrow (W \odot \Theta)B + \vec{b}$
    $\Theta \leftarrow \Theta - \alpha\frac{1}{|B|}W \odot \left([(\hat{B} - B) \odot \text{ReLu}'(\hat{B})]B^T\right) + \lambda\Theta$
    $\vec{b} \leftarrow \vec{b} - \alpha\frac{1}{|B|}[(\hat{B} - B) \odot \text{ReLu}'(\hat{B})]\mathbf{1}$
  end
  Report the loss $J(\Theta, \vec{b}) = \frac{1}{2n}\mathbf{1}^T\{[\text{ReLu}(\hat{R}) - R]^2\}\mathbf{1}$
  $epochCount \leftarrow epochCount + 1$
end

The whole data utility matrix *U* is divided randomly column-wise to the training matrix R, the development matrix X_{dev}, and the test matrix X_{test} in the ratio of 8:1:1. For a given document vector r, the prediction is:

$$\hat{\vec{r}} = W \odot \Theta\vec{r} + \vec{b}$$

Model Training and Test

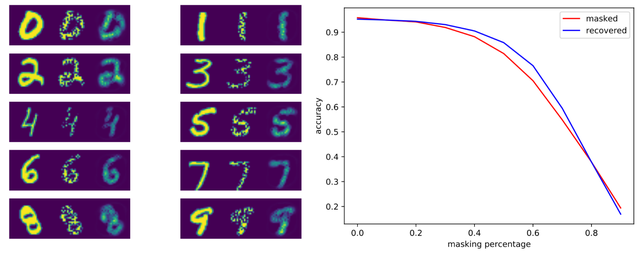
For a known document vector, 50% of its > 0 values will be randomly masked to 0. Then the model will predict on the masked document vector, and the performance is measured by MSE (mean squared errors), MSER (sum of squared errors divided by the number of non-zero terms), and APTK (percentage of correctly predicted top K keywords, K=3) as well as APRK as comparison (percentage of correct prediction by randomly picked 3 keywords).

The model was built and tested on a data set with 337371 documents with 15071 MeSH terms. The final performance is listed below:

MSE	MSER	APTK	APRK
0.001047	0.001608	0.5031	0.001532

Other Usage: Data Recovery

The model predicts information based on “non-empty” information of a record vector, which can be used as a data recovery method. To test this, MNIST Handwritten Digit Classification Data Set is masked and recovered by the model:



The model recovered the trace and also increased the accuracy of a neuron network trained on the original data.