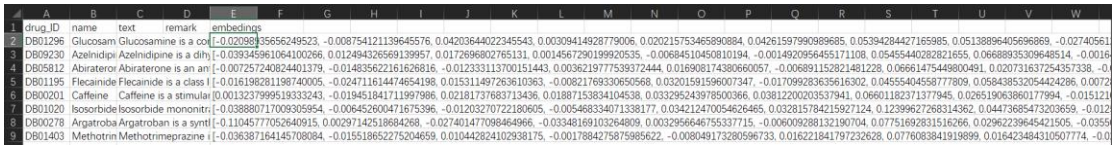
数据

进展

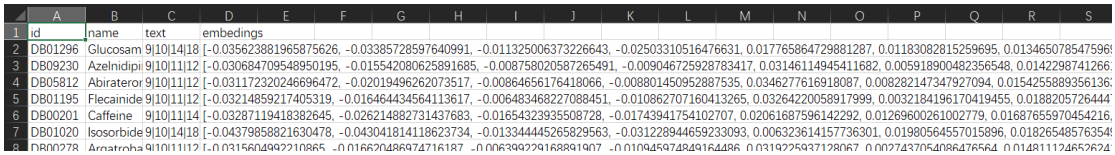
将四张关于药物的英文描述转化为向量，向量维度为 384。

dataset1-embed



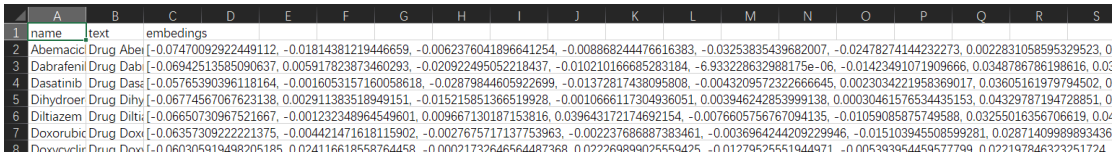
drug_ID	name	text	remark	embedings
DB01296	Glucosam	Glucosamine is a co		[-0.020981 3656249523, -0.008754121139645576, 0.04203644022345543, 0.0030941928779006, 0.020215753465890884, 0.04261597990989685, 0.05394284427165985, 0.05138896405696869, -0.027405611
DB09230	Azelnidipi	Azelnidipine is a dihy		[-0.03934596106400266, 0.012494326569139957, 0.0172696802765131, 0.0014567290199920535, -0.00684510450810194, -0.0014920956455171108, 0.05455440282821655, 0.06688935309648514, -0.0016
DB05812	Abirateror	Abiraterone is an ani		[-0.007257240824401379, -0.014835622161626816, -0.012333113700151443, 0.0036219777539372444, 0.016908174380660057, -0.006891152821481228, 0.06661475449800491, 0.020731637254357338, -0.(
DB01195	Flecainide	Flecainide is a class I		[-0.016198281198740005, -0.024711614474654198, 0.015311497263610363, -0.008217693306505681, 0.032015915960073 47, -0.017099283635616302, 0.04555404558777809, 0.05843853205442486, 0.0072
DB00201	Caffeine	Caffeine is a stimulan		[0.0013237999519333243, -0.019451841711997986, 0.021817376837134 36, 0.018871538341045 38, 0.033295243978500366, 0.038122002035379 41, 0.066011823713779 45, 0.026519063860177994, -0.015121(
DB01020	Isosorbide	Isosorbide mononitra		[-0.038880717009305954, -0.006452600471675396, -0.012032707221806 05, -0.005468334071381 77, 0.034212470054626465, 0.032815784215927124, 0.12399627268314362, 0.044736854732036 59, -0.012
DB00278	Argatroba	Argatroban is a synth		[-0.11045777052640915, 0.0029714251868426 8, -0.027401477098464966, -0.033481691032648 09, 0.0032956467553377 15, -0.006009288132190704, 0.07751692831516266, 0.029622396454521505, -0.0355(
DB01403	Methotrin	Methotrimeprazine i		[-0.036387164145708084, -0.015518652275204659, 0.010442824102938175, -0.001788427587598 5622, -0.008049173280596733, 0.016221841797232628, 0.07760838419189 99, 0.016423484310507774, -0.0(

dataset2-embed



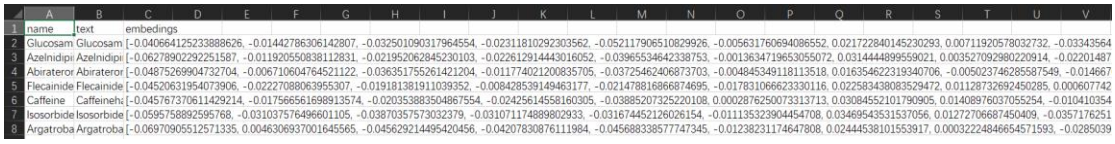
id	name	text	embedings
DB01296	Glucosam	9|10|14|18	[-0.035623819655875626, -0.03385728597640991, -0.011325006373226643, -0.02503310516476631, 0.01776586472988128 7, 0.011830828152595695, 0.013465078547596
DB09230	Azelnidipi	9|10|11|12	[-0.030684709548950195, -0.015542080625891685, -0.008758020587265491, -0.009046725928783417, 0.031461149454116 82, 0.005918900482356548, 0.014229874126
DB05812	Abirateror	9|10|11|12	[-0.031172320246696472, -0.020194962620735157, -0.008646561764180 66, -0.008801450952887535, 0.034627761691808 7, 0.008282147347927094, 0.015425588935613
DB01195	Flecainide	9|10|11|12	[-0.032148592174053 19, -0.016464434564113617, -0.006483468227088451, -0.010862707160413265, 0.032642200589179 99, 0.003218419617041945 5, 0.018820572644
DB00201	Caffeine	9|10|11|14	[-0.032871194183826 45, -0.026214882731437683, -0.016543239355087 28, -0.017439417541027 07, 0.020616875961422 92, 0.012696002610027 79, 0.016876559704542 16,
DB01020	Isosorbide	9|10|14|18	[-0.043798588216304 78, -0.043041814118623734, -0.013344445265829563, -0.031228944659233093, 0.006323614157736301, 0.019805645570158 96, 0.018265485763545
DB00278	Argatroba	9|10|11|12	[-0.031560499221 0865, -0.016620486974716187, -0.006399229166891907, -0.010945974849164486, 0.031922593712806 7, 0.002743705408647656 4, 0.014811124652624

dataset3-embed



name	text	embedings
Abemacic	Drug Abe	[-0.07470092922449112, -0.01814381219446659, -0.006237604189664125 4, -0.008868244476616383, -0.03253835439682007, -0.024782741442322 73, 0.002283105859532952 3, 0
Dabrafeni	Drug Dab	[-0.06942513585090637, 0.005917823873460293, -0.020922495052218437, -0.01021016668528318 4, -6.933228632988175e-06, -0.014234910719096 66, 0.034878678619 8616, 0.0
Dasatinib	Drug Das	[-0.05765390396118164, -0.001605315716005861 8, -0.028798446059226 99, -0.013728174380958 08, -0.004320957232266 6645, 0.002303422195836 9017, 0.036051619797945 02, 0
Dihydroer	Drug Dihy	[-0.06774567067623138, 0.002911383518949151, -0.015215851366519928, -0.001066611730493605 1, 0.003946242853999138, 0.000304615765344351 53, 0.043297871947288 51, 0
Diltiazem	Drug Dilti	[-0.06650730967521667, -0.001232348964549601, 0.009667130187153816, 0.039643172174692154, -0.007660575676709413 5, -0.010590858757495 88, 0.032550163567066 19, 0.04
Doxorubic	Drug Doxo	[-0.06357309222221375, -0.004421471618115902, -0.002767571713775396 3, -0.002237686887383461, -0.003696424420922994 6, -0.015103945508599281, 0.028714099898934 36
Doxycyclir	Drug Doxy	[-0.060305919498205185, 0.024116618558764458, -0.000217326465648738 8, 0.022269899025559425, -0.012795255519449 71, -0.005393954459577799, 0.022197846323251724

dataset4-embed



name	text	embedings
Glucosam	Glucosam	[-0.040664125233888626, -0.01442786306142807, -0.032501090317964554, -0.023118102923035 62, -0.052117906510829926, -0.005631760694086552, 0.021722840145230293, 0.007119205780327 32, -0.033435 64
Azelnidipi	Azelnidipi	[-0.06278902292251587, -0.011920550838112831, -0.021952062845230103, -0.022612914443016052, -0.039655346423387 53, -0.001363471965305572, 0.031444489955902 1, 0.003527092980220914, -0.022014 87
Abirateror	Abirateror	[-0.048752699047327 04, -0.006710604764521122, -0.036351755261421204, -0.011774021200835705, -0.037254624068737 03, -0.004845349118113518, 0.016354622319340706, -0.005023746285587549, -0.014667
Flecainide	Flecainide	[-0.045206319540739 06, -0.022270880639553 07, -0.019191381911039352, -0.008428539149463177, -0.021478816866874695, -0.017831066623330116, 0.022583438083529472, 0.011287326924502 85, 0.000607742
Caffeine	Caffeineh	[-0.045767370611429214, -0.017566561699891357 4, -0.020353883504867554, -0.024256145581603 05, -0.038852073252201 08, 0.000287625007331371 3, 0.030845521017909 05, 0.014089760370552 54, -0.010410354
Isosorbide	Isosorbide	[-0.059575889259576 8, -0.031037576496601105, -0.038703575730323 79, -0.031071174889802933, -0.031674452126026154, -0.011135323904454708, 0.034695435135705 6, 0.012727066874504 09, -0.035717625 1
Argatroba	Argatroba	[-0.069709055125713 35, 0.004630693700164556 5, -0.045629214495420456, -0.042078308761119 84, -0.045688338577747345, -0.012382311746478 08, 0.024445381015539 17, 0.003222484665457159 3, -0.0285039

问题

1、dataset3-embed（药物 ddi 信息）和 dataset4-embed（药物 MACCS 键和物理特性信息）还没找到合适的英文描述，还需要找找，或由 LLM 生成。暂时先用着。

代码

进度

```python
class GNN3(nn.Module):  3 用法
    def __init__(self,dataset,tail_len,relation_len,args,dict1,drug_name,**kwargs):
        super(GNN3,self).__init__(**kwargs)
        self.text_embed_3 = self.load_pretrain_embeds(dict1, embed_path: '../data-text-embed/dataset3-embed.csv')
        self.kg, self.dict1 = dataset["dataset3"], dict1
        self.drug_name, self.args = drug_name, args
        self.drug_embed = nn.Embedding(num_embeddings=572, embedding_dim=args.embedding_num)
        self.rela_embed = nn.Embedding(num_embeddings=67, embedding_dim=args.embedding_num)
        self.ent_embed = nn.Embedding(num_embeddings=572, embedding_dim=args.embedding_num)
        self.W1 = nn.Parameter(torch.randn(size=(572,args.embedding_num,args.embedding_num)))
        self.b1 = nn.Parameter(torch.randn(size=(args.neighbor_sample_size,args.embedding_num)))
        self.W2 = nn.Parameter(torch.randn(size=(572, args.embedding_num, args.embedding_num)))
        self.b2 = nn.Parameter(torch.randn(size=(args.neighbor_sample_size, args.embedding_num)))
        self.Linear1 = nn.Sequential(nn.Linear(args.embedding_num * 2, args.embedding_num),
                                     nn.ReLU(),
                                     nn.BatchNorm1d(args.embedding_num))
        self.relu = nn.ReLU()
        self.soft=nn.Softmax(dim=1)
        for m in self.modules():
            if isinstance(m, nn.Linear):
                nn.init.xavier_uniform_(m.weight)
                nn.init.zeros_(m.bias)
            elif isinstance(m, nn.BatchNorm1d):
                nn.init.constant_(m.weight, val: 1)
```

在每个GNN中输入文本向量，并用dict1与GNN输出的向量对齐

```python
182  class GNN3(nn.Module):  3 用法
208      def forward(self,arguments):
209          kg, dict1, args, drug_name = self.kg, self.dict1, self.args, self.drug_name
210          gnn2_embedding,gnn1_embedding,idx=arguments
211          text_embed_3 = self.text_embed_3
212          adj_tail,adj_relation=self.arrge(kg,dict1,args.neighbor_sample_size)
213
214          drug_name=torch.LongTensor(drug_name)
215          adj_tail=torch.LongTensor(adj_tail)
216          adj_relation=torch.LongTensor(adj_relation)
217
218          drug_embedding = self.drug_embed(drug_name)
219          rela_embedding = self.rela_embed(adj_relation)
220          ent_embedding = self.ent_embed(adj_tail)
221          drug_rel = drug_embedding.reshape((572, 1, args.embedding_num)) * rela_embedding
222          drug_rel_weigh = drug_rel.matmul(self.W1) + self.b1
223          drug_rel_weigh = self.relu(drug_rel_weigh)
224          drug_rel_weigh = drug_rel_weigh.matmul(self.W2) + self.b2
225          drug_rel_score = torch.sum(drug_rel_weigh, axis=-1, keepdims=True)
226          drug_rel_score = self.soft(drug_rel_score)
227          weighted_ent = drug_rel_score.reshape((572, 1, args.neighbor_sample_size)).matmul(ent_embedding)
228          drug_e = torch.cat( tensors: [weighted_ent.reshape(572, args.embedding_num), drug_embedding.reshape((572, arg
229          drug_f = self.Linear1(drug_e)
230          drug_f_text_embed = torch.cat( tensors: [drug_f, text_embed_3], dim=1)
231          return  drug_f_text_embed,gnn2_embedding,gnn1_embedding,idx
```

拼接，drug_f的形状为（572，128）
text_embed_3的形状为（572，384）

```python
class FusionLayer(nn.Module):  2 用法
    def __init__(self, args):
        super().__init__()
        self.fullConnectionLayer = nn.Sequential(          修改融合层的维度
            # (args.embedding_num + 384)
            nn.Linear((args.embedding_num + 384) * 4 * 2, (args.embedding_num + 384) * 4),
            nn.ReLU(),
            nn.BatchNorm1d((args.embedding_num + 384) * 4),
            nn.Dropout(args.dropout),
            nn.Linear((args.embedding_num + 384) * 4, (args.embedding_num + 384) * 2),
            nn.ReLU(),
            nn.BatchNorm1d((args.embedding_num + 384) * 2),
            nn.Dropout(args.dropout),
            nn.Linear((args.embedding_num + 384) * 2,  out_features: 65))
        for m in self.modules():
            if isinstance(m, nn.Linear):
                nn.init.xavier_uniform_(m.weight)
                nn.init.zeros_(m.bias)
            elif isinstance(m, nn.BatchNorm1d):
                nn.init.constant_(m.weight,  val: 1)
                nn.init.constant_(m.bias,  val: 0)
```

问题

1、向量拼接已完成，还缺少向量融合

```python
class FusionLayer(nn.Module):  2 用法
    def forward(self, arguments):
        idx = idx.numpy().tolist()
        drugA = []
        drugB = []
        for i in idx:
            drugA.append(i[0])
            drugB.append(i[1])


        Embedding = torch.cat( tensors: [gnn1_embedding[drugA],gnn2_embedding[drugA],gnn3_embedding[drugA],gnn4_embed
                                gnn1_embedding[drugB],gnn2_embedding[drugB],gnn3_embedding[drugB],gnn4_embedding[dr
                                                在该部分添加LLM融合向量部分
        return self.fullConnectionLayer(Embedding)
```

打算使用 Llama-3.2-1B 来完成融合

流程

向量拼接部分已完成，后续再冻结 LLM 参数来融合向量。