深圳大学考试答题纸

(以论文、报告等形式考核专用)  
二○二三～二○二四学年度第一学期

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程编号 | 2703093 | | 课程名称 | | 数据挖掘 | | 主讲教师 | | 陈小军 | 评分 |  |
| 学 号 | 2300271074 | | 姓名 | 李海龙 | | 专业年级 | | 计算机科学与技术2023级 | | | |
|  | | | | | | | | | | | |
| 教师评语： | | | | | | | | | | | |
| 项目名称： | | 电商知识图谱链接预测 | | | | | | | |  | |

摘 要：本赛题以电商行业为背景，进行知识图谱链接预测，数据集为OpenBG500，包含了500种不同的关系和百万级别规模的训练数据，要求为测试集中每条记录预测10个尾实体。实验部分对数据集进行处理，并用pytorch实现2013年提出的TranE模型来生成预测，最终的提交结果在长期赛排名第9位。

[简单列出已经实现的功能]

深圳大学课程项目报告

|  |  |
| --- | --- |
| 课程名称： | 数据挖掘 |
| 项目名称： | 电商知识图谱链接预测 |
| 学院： | 计算机与软件学院 |
| 专业： | 计算机科学与技术 |
| 姓名学号： | 李海龙 2300271074 |
| 任课教师： | 陈小军 |
| 提交时间： | 2024年1月8日 |

教务处制

目录

一、项目介绍

* 1. 赛题及背景介绍
  2. 项目摘要
  3. 项目要求

二、数据处理

2.1 数据集介绍

2.2 数据集读取并加载

三、模型搭建

3.1 TransE模型介绍

* 1. 代码实现

四、实验结果

4.1 参数设置

* 1. 运行结果及排名

五、总结

六、参考文献

1. **项目介绍**
   1. **赛题及背景介绍**

­此课程项目选取的赛题是天池学习赛中的电商知识图谱连接预测，赛题以电商行业为背景，进行知识图谱链接预测。

知识图谱是一种以图形化的方式呈现和展示知识的方式。它通过将知识体系中的概念、实体以及它们之间的关系进行可视化，帮助人们更好地理解和掌握知识[1]。知识图谱的构建需要经过多个步骤，包括知识获取、知识表示、知识推理和知识可视化等。在知识图谱中，概念和实体被表示为节点，而它们之间的关系则被表示为边。这种图形化的表示方式使得知识图谱具有很强的直观性和可理解性。通过知识图谱，人们可以快速地了解概念和实体之间的关系，从而更好地理解和掌握知识。知识图谱在多个领域都有应用，如教育、医疗、商业等。

知识图谱是AI时代一项非常重要的技术，然而知识图谱普遍存在不完备的问题，知识图谱链接预测任务主要基于实体和关系的表示对缺失三元组进行预测。本任务旨在提升电商场景下知识图谱嵌入效果，满足商品推荐等应用对推理商品潜在关联性的需求。

* 1. **项目摘要**

知识图谱一般通过三元组（h，r，t）的形式组织数据，其中h被称为头实体，t为尾实体，r为连接头、尾实体的关系。由于知识图谱构建中部分知识的缺失及知识动态变化等原因，现有的知识图谱通常是不完备的，知识图谱中总是存在关系r下头实体h或者尾实体t缺失的情况。基于知识图谱的链接预测任务，就是已知头实体（或尾实体）和关系的情况下，预测缺失的尾实体（或头实体）的任务。我们在当前任务下所提供的知识图谱的头实体h通常为商品，尾实体t通常为商品所对应相关属性信息，如颜色、适用人群、细分市场等，关系r为具体的属性类型。因为商品属性关系中多对一的情况十分普遍，所以在做关系推理和链接预测任务时只考虑预测尾实体。

赛题数据集使用OpenBG500，这是一个电子商务领域的知识图谱，包含500个关系，从阿里巴巴藏经阁开放商业知识图谱AliOpenKG中筛选采样得到，数据格式均为tsv格式。

* 1. **项目要求**

此赛题的测试集给出一个三元组的头实体h和关系r，要求选手预测对应的尾实体t，如图1所示，要求给出头实体化妆棉在关系为适用群体的情况下对应的尾实体，输出可为“女性”或其同义词。

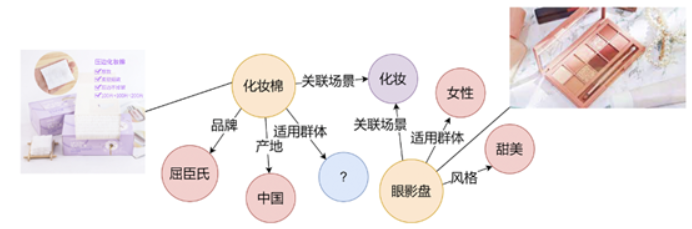


图1 链接预测示例

本任务采用MRR、HIT@10、HIT@3、HIT@1作为评测指标。其中主评测指标为尾实体预测的MRR，即Mean Reciprocal Rank，该评测指标是链接预测正确实体排名的倒数平均，对于一个query：（头实体，关系，？），若第一个正确答案排在第n位，则该次query的分数为1/n，由于只计算概率Top10的实体，所以此处n最大为10，如果Top10实体中不包含正确答案，则此次query的得分为0，最终对多个query的分数求平均；其他3个指标HIT@1、HIT@3、HIT@10则描述了链接预测中得分最高的top K（K=1，3，10）个实体中包含正确实体的概率。

选手需要为测试集中每条记录预测10个尾实体，验证集不可以混入训练集用于模型训练。

1. **数据处理**

**2.1 数据集介绍**

OpenBG500数据集是一个大规模的知识图谱数据集，包含了500种不同的关系，并且具有百万级别规模的图谱数据。该数据集主要用于链接预测任务，特别是电子商务领域的知识图谱构建。

OpenBG500数据集的结构化存储使用了tsv格式，方便读取和处理。具体来说，它包括了以下几部分：

1）OpenBG500\_train.tsv：训练集，包含了训练所需的数据。

2）OpenBG500\_dev.tsv：验证集，用于验证模型的性能。

3）OpenBG500\_test.tsv：测试集，用于最终测试模型的性能。

4）OpenBG500\_entity2text.tsv：实体描述文件，包含了知识图谱中实体的中文描述信息。

5）OpenBG500\_relation2text.tsv：关系描述文件，包含了知识图谱中关系的中文描述信息。

6）OpenBG500\_example\_pred.tsv：示例三元组文件，包含了用于预测的示例三元组数据。

OpenBG500数据集的规模和多样性使其成为电子商务领域知识图谱构建的理想选择，特别是对于需要进行大规模链接预测的任务。通过使用这个数据集，研究人员和工程师可以开发出更准确、更有效的知识图谱链接预测算法，从而更好地服务于电子商务领域的实际应用需求。

**2.2 数据集读取并加载**

数据预处理是指在进行正式的数据分析或建模之前，对原始数据进行必要的处理，以提高数据质量并使其符合特定算法的要求。数据预处理的常见方法包括数据清洗、数据集成、数据变换和数据规约。而特征工程是指从原始数据中提取有用特征的过程，这些特征可用于建立机器学习模型。它可能包括从原始数据中选择有用的特征，转换特征以便更容易处理，或者构造新的特征，以提高模型的准确性。

由于OpenBG500数据集通过id对应实体以及关系，主体部分只包括id三元组，数据比较干净，所以没有进行数据预处理和特征工程的必要，因此下面直接对tsv文件进行读取，并加载构建可用于模型训练的数据集。

首先，导入加载数据所需要用到的库文件。

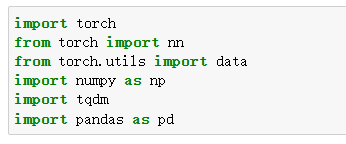


图2 库文件导入

使用pandas的read\_csv来加载数据集，将数据转换为dataframe格式，查看一下数据包含的具体内容，由于给出的tsv文件分隔符为“\t”且不包含表头，因此read\_csv函数的参数sep设置为”\t”，header设置为None，用name给出的列表设置列名。

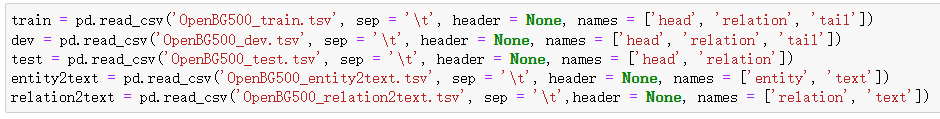


图3 tsv文件读取

将实体对应文本entity2text按第一列entity升序输出，如图4所示，可以看到，OpenBG500数据集的实体总数目为249746，包含魔方周边、森林、短衣短裤等实体。



图4 entity2text示例

对实体的文本描述长度进行统计，并绘制直方图，如图5所示，可以看到，大多实体的文本长度较小，但也有小部分实体有着极长的文本描述。

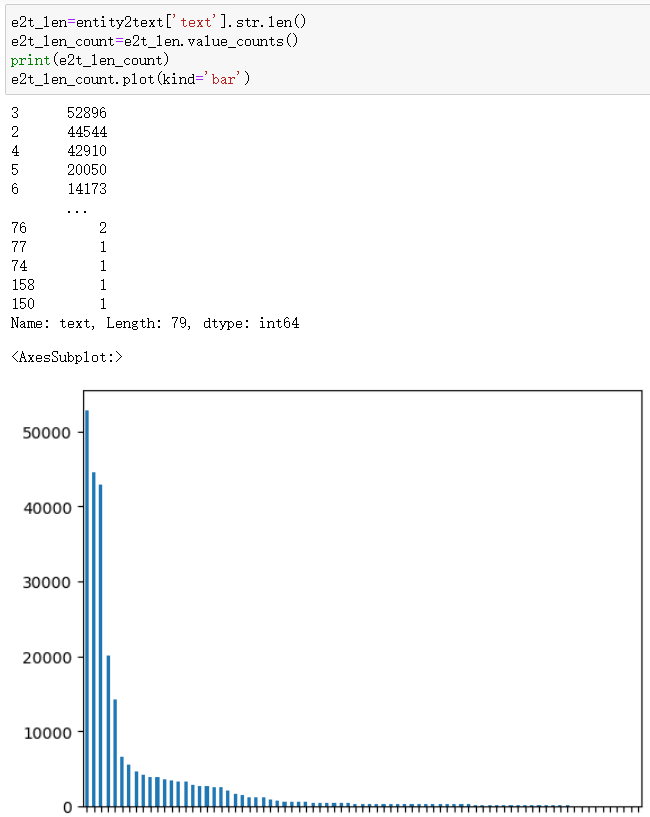


图5 实体文本描述长度统计

将关系对应文本relation2text按第一列relation升序输出，如图6所示，可以看到，OpenBG500数据集的关系总数目为500，包含刷头材质、倍率类型、魔方种类以及生产厂家等关系。



图6 relation2text示例

同样对关系的文本描述长度进行统计，并绘制直方图，如图7所示，可以看到，关系的文本长度都大于0，且都偏向于简短。

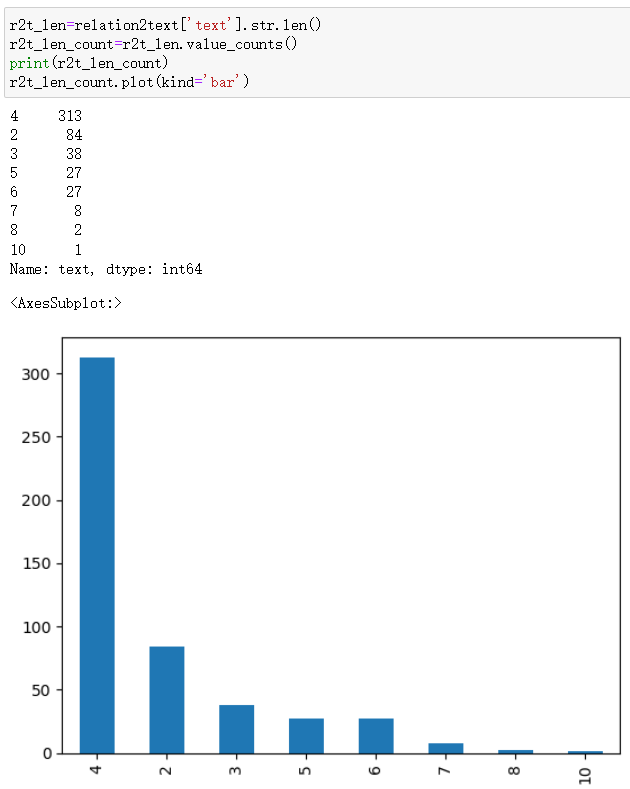


图7 关系文本描述长度统计

将训练集train分别按第一列head和第三列tail升序输出，如图8所示，可以看到，训练集中包含的三元组总数目为1242550，覆盖到了所有实体。

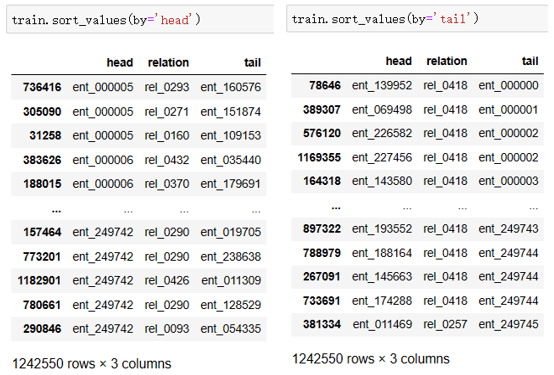


图8 训练集train示例

将验证集dev以及测试集test按第一列entity升序输出，如图9所示，可以看到，验证集和测试集的大小都为5000，测试集缺少尾实体，需要训练模型给出预测。

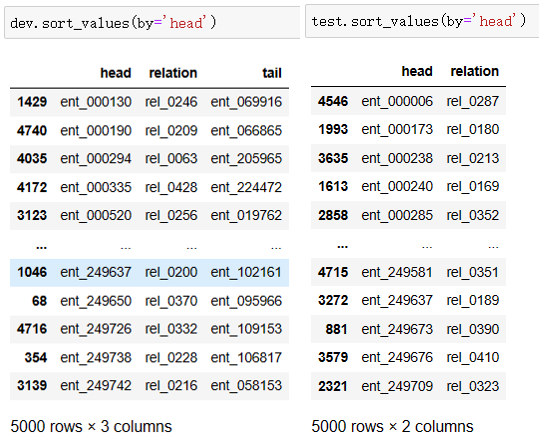


图9 验证集dev、测试集test示例

通过上述对tsv文件的读取可知，OpenBG500数据集共包含249746个实体以及500种关系，给出的训练集大小为1242550，验证集和测试集大小都为5000，由这些信息下面给出TripleDataset类和TestDataset类的实现，并使用torch.utils.data中的Dataloader来加载用于模型训练的可迭代对象。

图10所示为TripleDataset类和TestDataset类的相关代码，两者的数据成员均为ent2id、rel2id以及data，其中ent2id记录实体id同下标的对应关系，rel2id记录关系id同下标的对应关系，data记录训练集、验证集或者测试集中包含的所有id三元组，方法\_\_len\_\_获取加载的数据集大小，\_\_getitem\_\_获取对应下标index的三元组，由于测试集缺少尾实体，所以TestDataset类的\_\_getitem\_\_返回的是二元组。

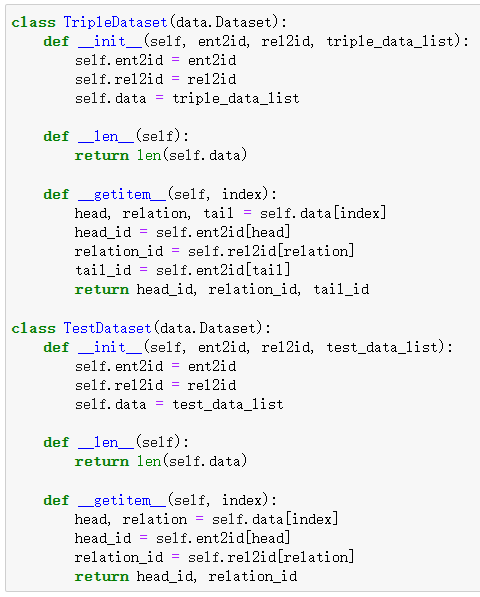


图10 TripleDataset类以及TestDataset类

逐行读取给出的tsv文件中的内容，如图11所示，由于实体对应文本和关系对应文本都不是按照实体id、关系id有序排列的，所以用ent2id和id2ent来记录实体id和其下标的对应关系，用rel2id来记录关系id和其下标的对应关系。

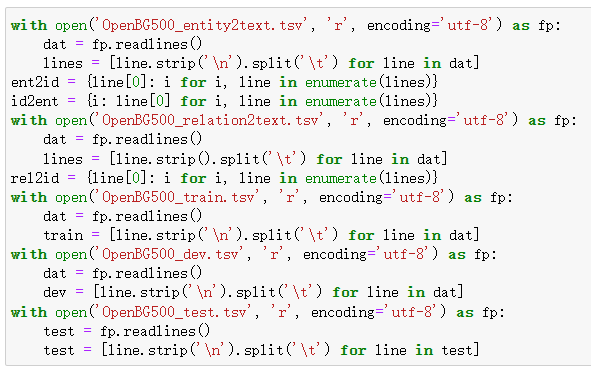


图11 tsv文件逐行读取

根据以上代码，得到训练集、验证集对应的TripleDataset对象以及测试集对应的TestDataset对象，然后用data的Dataloader加载各个数据集，设置各自的批次大小，即batch\_size，其中训练集的shuffle设置为True，用于每一轮开始时随机打乱数据。

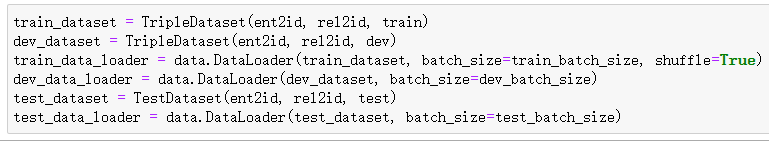


图12 数据集加载

1. **模型搭建**

**3.1 TransE模型介绍**

TransE[2]模型的文章标题为《Translating Embeddings for Modeling Multi-relational Data》，发表在2013年的NeurIPS上，这是一种将实体和关系嵌入到低维向量空间中进行计算的建模方法，该方法较为简单、参数较少，是基于平移距离建模和知识图谱表示学习的开山之作，其主要思想为将实体与关系映射到同一低维向量空间中，并约束头实体向量与关系向量的和尽可能接近尾实体向量，在两个知识库上的实验证明，TransE在链接预测任务中取得了较理想的效果，优于当时的最新方法。

TransE模型解决的主要问题在于如何建模多关系型数据。多关系数据是指有向图，节点对应于形式为三元组（头实体，关系，尾实体），表示为（h，r，t）

），每个三元组都表明实体的头和尾的名称之间存在关系。大型知识库，例如Freebase和Google Knowledge Graph，其中的知识大多是通过三元组的形式进行表达的。知识库中的每个实体代表世界的抽象概念或具体的实体，而关系是表示其中两个事实实体的谓词。建模关系型数据的困难之处在于无论是实体还是关系都有许多不同的类别并会同时出现，所以对多关系数据建模要用更通用的方法，需要适当的方法来提取出模式。当时处理多关系数据大多数方法采用的是学习潜在特征的方法。然而大多数模型均较为复杂，会使得正则化项难以设置导致过拟合，过多的局部极小值点带来的非凸优化问题会导致欠拟合。

给定包含多个三元组的训练集，该训练集的每个三元组由两个实体（实体的集合）和关系（关系的集合）组成。TransE模型受word2vec思想的启发，利用词向量的平移不变性，对于每个实体和每个关系学习一个低维嵌入向量，对于一个三元组而言，头实体向量和关系向量之和与尾实体向量越接近，那么说明该三元组越接近一个正确的三元组，差距越大，那么说明这个三元组越不合理，目的是使正确的三元组的嵌入向量满足，否则应当远离。

TransE是一个基于“能量”的模型，每一个三元组的能量等于，其中表示差异，可以将其取为L1或者L2范数，其损失函数如下

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

其中表示为正时取、为负或等于0时取0；是一个大于0的超参数，作用相当于是一个正确三元组与错误三元组之间的差距修正，越大，则两个三元组之间被修正的差距就越大；，其中以及w为错误的三元组，即相当于负样本，是随机替换三元组中的头实体或尾实体得到的，并且头实体和尾实体分别替换，不会同时替换，从而得到打破的三元组。可以看出，TransE模型是针对给定三元组进行计算“能量”达到优化目的，其中负样本是通过替换头实体或者尾实体自行构造的，优化目标就是使得正负例样本距离最大化，通过最小化正样本的“能量”，最大化负样本的“能量”，达到优化嵌入表示的目的。

TransE的算法流程如图13所示，可划分为嵌入向量初始化（步骤1~3）、批训练数据集构建（步骤6~11）和嵌入向量更新（步骤12、5）三部分。其中嵌入向量初始化采用k维随机均匀分布对每个实体和每个关系进行初始化；批训练数据集构建则从训练集合中随机选出正样本（h，r，t），然后基于正样本，在不同时替换的前提下，随机替换三元组中的头实体或尾实体得到的，获得负样本，合并在一起构成批训练数据集；嵌入向量更新则采用随机梯度下降法，对批训练数据集中正、负样本的向量表示进行更新。

TransE依赖的参数较少，因为它对于每个实体和每个关系仅学习一个低维嵌入向量，计算的复杂度明显降低，并且在大规模稀疏知识图谱上也同样具有较好的性能与可扩展性，但是TransE模型不能用在处理复杂关系上，比如一对多、多对一、多对多等 3 种关系类型。

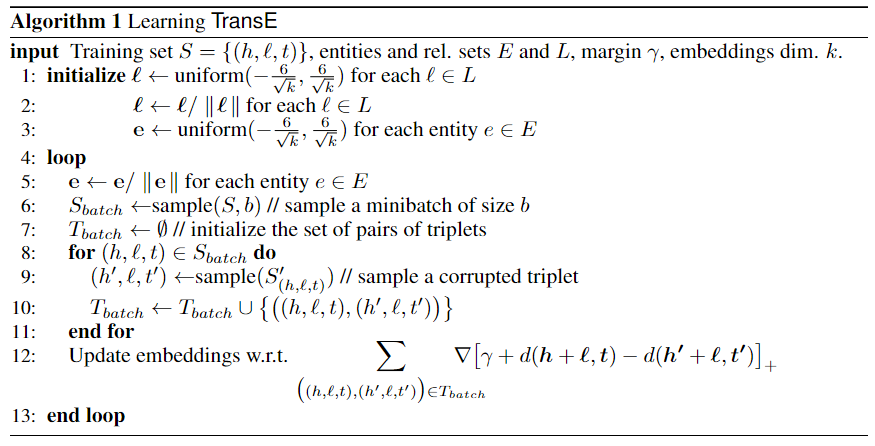
******

图13 TransE算法流程

**3.2 代码实现**

依据上述给出算法流程，编写TransE类，包含5个数据成员，如图14所示，norm表示采用什么范数，论文中作者采用的是L1范数，但是实际效果并不算好，因此还采用L2、L3范数进行训练，dim为嵌入向量的维数，默认为100，entity\_num为实体总数目，entities\_emb、relation\_emb初始分别为实体、关系通过成员函数\_init\_emb获取的随机嵌入向量。

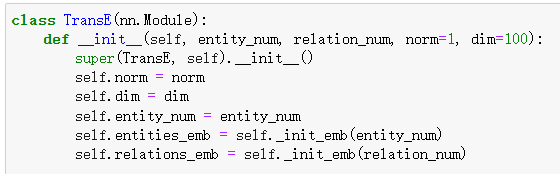


图14 TransE类\_\_init\_\_()

图15所示为TransE类的部分方法，其中\_init\_emb通过torch.nn的Embedding得到个数为num\_embeddings，维数为self.dim的随机初始化的向量；forward调用\_distance分别计算并返回正、负三元组样本的“能量”，用于前向传播；\_distance获取三元组中头实体、关系以及尾实体的向量，并计算。

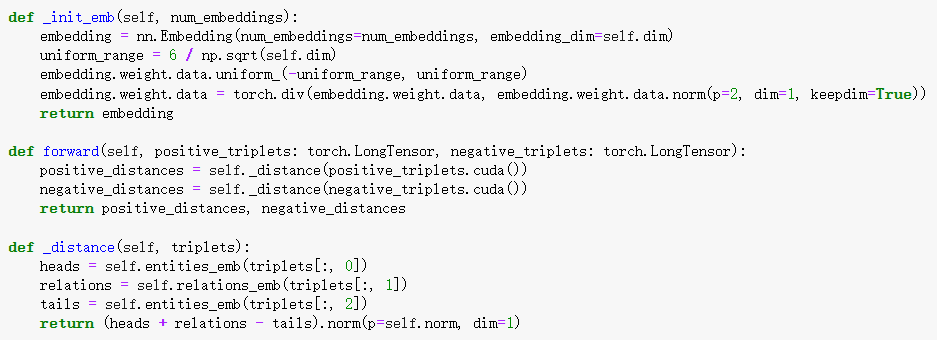


图15 TransE类嵌入向量初始化及能量计算

此外，如图16所示，TransE类还有link\_predict用于生成测试集的10个尾实体预测结果，evaluate用于在验证集上评估模型的效果，给出相应的MRR、HIT@1、HIT@3以及HIT@10。

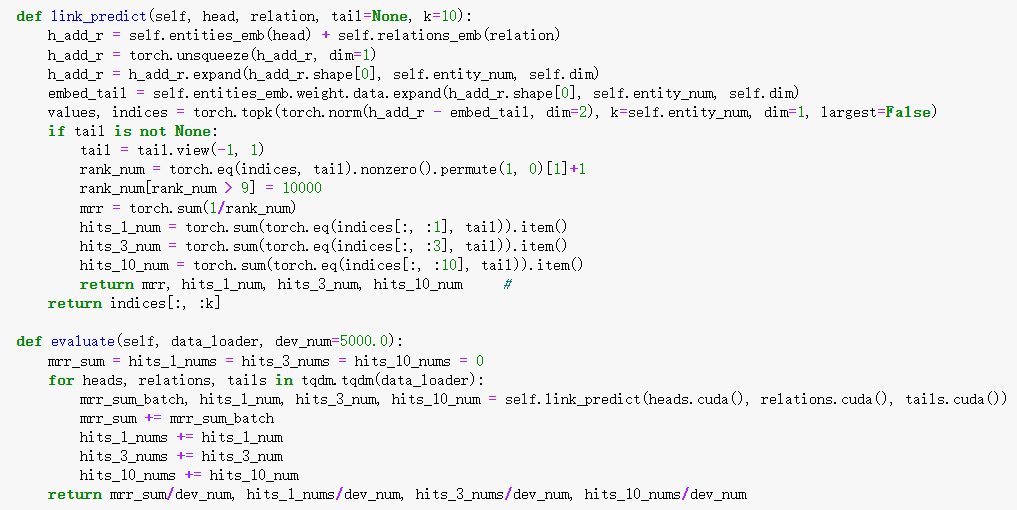


图16 TransE类评估及生成预测

1. **实验结果**

**4.1 参数设置**

表1所示为实验的相关参数描述。

|  |  |
| --- | --- |
| 参数名称 | 作用 |
| train\_batch\_size | 训练集的批次大小 |
| epochs | 运行轮数 |
| margin | MarginRankingLoss采用的偏移值 |
| print\_frequency | 打印步长 |
| validation | 是否验证 |
| dev\_interval | 每多少轮进行一次验证 |
| best\_mrr | 记录验证集上MRR的最佳结果 |
| learning\_rate | Adam优化器的学习率 |
| distance\_norm | 范数，可取L1、L2、L3 |
| embedding\_dim | 词向量维数 |

各参数的初始设置如图17所示。

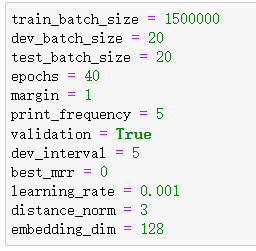


图17 各参数设置

**4.2 运行结果及排名**

按照给出的参数设置，运行完整代码，第40个epoch运行结束时在验证集上得到的结果如图18所示。

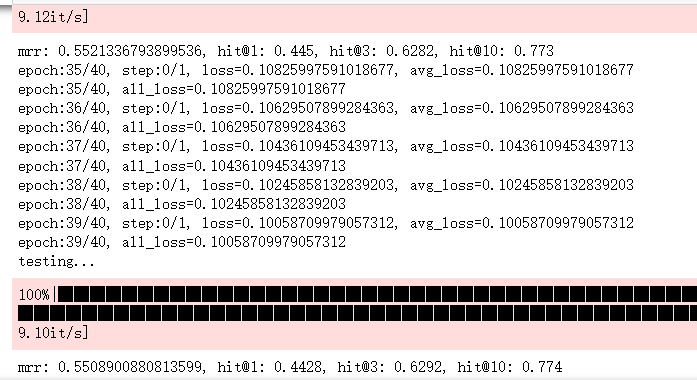


图18 验证集评测结果

为测试集生成尾实体预测并提交，网站评测的最佳MRR、HIT@1、HIT@3、HIT@10如图19所示，在长期赛中排名第9位。

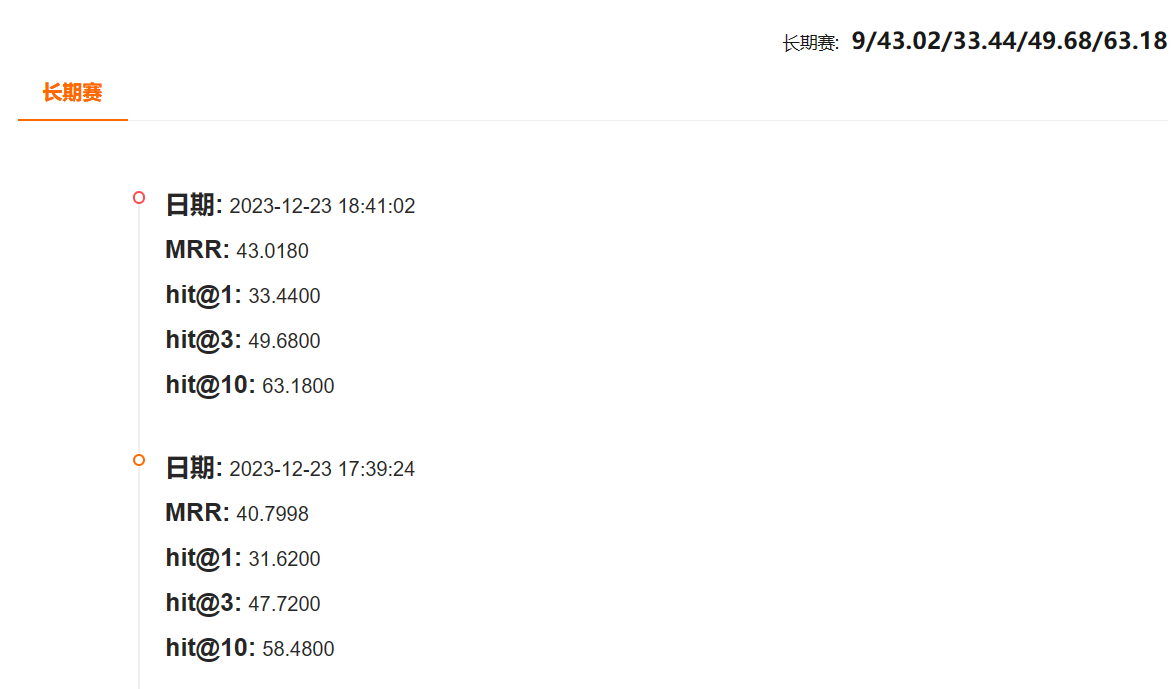


图19 长期赛排名

1. **总结**

选取的赛题是以电商行业为背景的知识图谱链接预测，数据集规模不大，使用TransE模型好在易于理解、实现简单且参数较少，我自己在实验过程中对知识图谱以及词向量的相关内容有了更近一步的了解，也稍微熟练了对Pytorch的使用，得到了一定收获。

关于不足的方面，TransE毕竟不属于神经网络模型，因而欠缺对句子含义的理解，得分并不算优秀，而且TransE作为知识表示模型最有影响力的模型之一，之后有许多工作都是在TransE模型的基础上展开，比如TransH[3]、TransR[4]等等，因此在这个赛题上我还有着很大的改进空间。

1. **参考文献**

[1] 刘知远, 韩旭, 孙茂松. 知识图谱与深度学习 [M] 北京：青海大学出版社, 2022.

[2] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[J]. Advances in neural information processing systems, 2013, 26.

[3] Wang Z, Zhang J, Feng J, et al. Knowledge graph embedding by translating on hyperplanes[C]//Proceedings of the AAAI conference on artificial intelligence. 2014, 28(1).

[4] Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion[C]//Proceedings of the AAAI conference on artificial intelligence. 2015, 29(1).