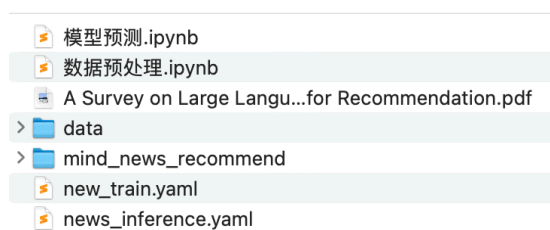


项目描述

本项目将大模型应用在推荐系统方向中，首先介绍了推荐系统的概念，然后讲述了大模型在推荐系统中的应用范式，最后以新闻推荐的 MIND 数据集为例，讲述了数据集预处理，基于 GLM4 的微调 and 训练后的模型预测。整个项目文件列表如下：



其中，mind_news_recommend 为训练好的 LoRA 参数文件，其余文件在如下文档中皆有介绍。

项目目标

- 熟悉推荐系统的概念
- 理解大模型在推荐系统中的应用范式
- 掌握大模型微调做新闻推荐

背景介绍

推荐系统的概念

推荐系统 (Recommendation system, RS) 是一种信息过滤系统，它旨在根据用户的历史行为、偏好、上下文等信息，为用户提供个性化的推荐内容。其核心目标是帮助用户在海量的信息中快速发现与自己兴趣相关、有价值的物品或服务，从而提高用户体验和满意度。推荐系统通常基于机器学习、数据挖掘和统计学等技术，通过分析用户数据来构建用户模型，并利用各种算法来预测用户可能感兴趣的内容。

推荐系统的典型应用场景如下：

新闻资讯：今日头条、腾讯新闻等新闻应用会根据用户的阅读偏好和浏览历史，为用户推送个性化的新闻内容，确保用户能够获取到自己感兴趣的资讯。

电子商务：在电商平台上，推荐系统可以根据用户的浏览历史、购买记录和搜索行为，为用户推荐相关的商品。例如，亚马逊、淘宝等电商巨头都广泛应用推荐系统来提高销售额和用户忠诚度。比如，如果用户经常购买运动装备，系统可能会推荐新的运动品牌或相关的健身器材。

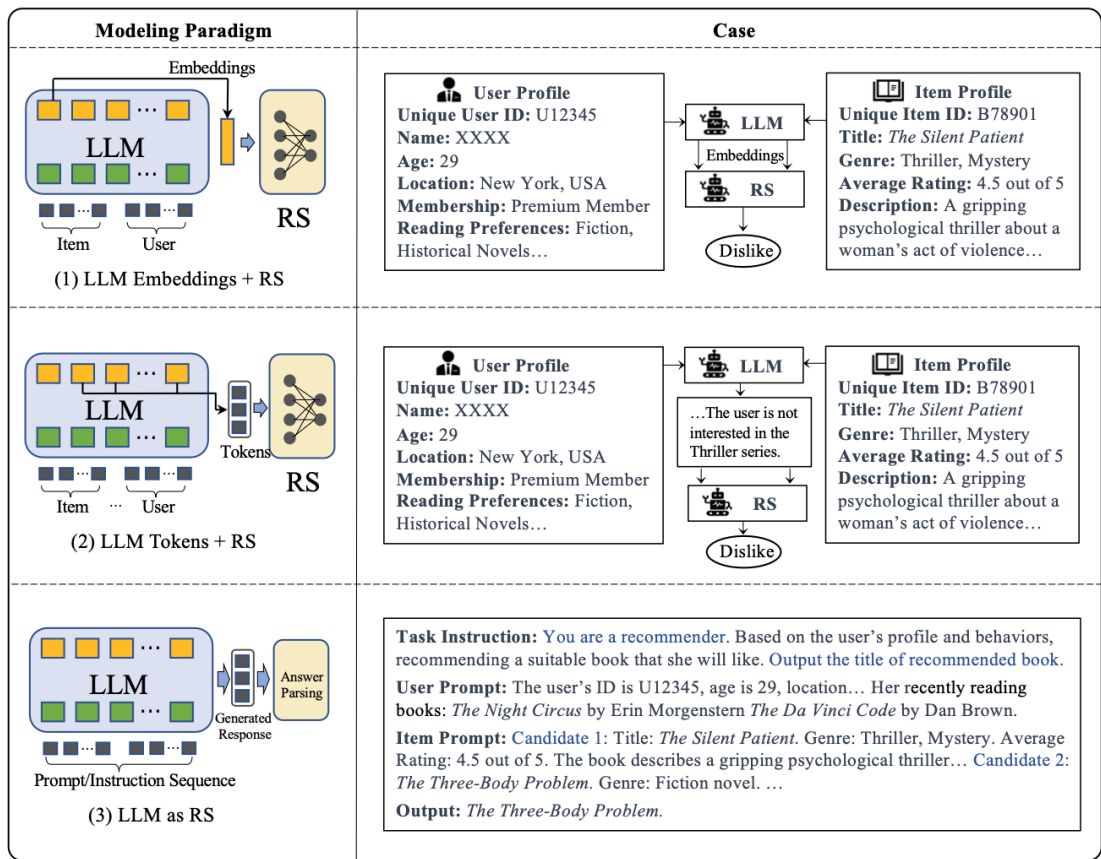
在线视频和音乐：像 Netflix、YouTube、QQ 音乐等平台利用推荐系统为用户推荐符合其口味的视频和音乐。如果用户喜欢观看科幻电影，系统会推荐更多同类型的影片。

社交媒体：社交媒体平台如 Facebook、微博等通过推荐系统为用户推荐可能感兴趣的好友、群组、话题或活动。

旅游出行：在旅游预订网站上，推荐系统可以根据用户的出行历史、目的地偏好和预算，为用户推荐合适的酒店、航班和旅游景点。

大语言模型在推荐系统中的应用

大语言模型 (Large language model, LLM) 包含了丰富的文本表示和广泛的外部知识，基于大语言模型的推荐系统可以捕捉上下文信息，更有效地理解用户查询、物品描述和其他文本数据，提高推荐的准确性和相关性。大语言模型还可以通过 zero-shot/few-shot 的能力解决推荐系统常见的数据稀疏问题。目前，基于大语言模型的推荐系统可以分为以下三种范式 [1]：



[1] LLM Embeddings + RS：将大语言模型作为特征提取器，输入用户

(User) 和商品 (Item) 的特征，利用大语言模型输出对应的

embeddings，传统的推荐系统模型利用大语言模型输出的 embeddings 完成推荐任务。

[2] LLM Tokens + RS：输入用户和商品的特征，大语言模型输出基于 token 的描述，这些描述蕴含了潜在的偏好，可以融入推荐系统的决策过程。

[3] LLM as RS：直接将大语言模型当作推荐系统。大语言模型的输入包括任务描述，用户和商品的特征，期望大模型基于指令输出合理的推荐结果。具体来说，可以编写 prompt 使用大语言模型直接做个性化推荐预测，比如下面一个 prompt [2]：

此人有一个喜欢的电影列表：{movies_liked}。 \

此人有一个不喜欢的电影列表：{movies_disliked}。 \

告诉我这个人是否喜欢每一部候选电影：{movies_candidates}。 \

返回布尔值列表并解释此人喜欢或不喜欢的原因。

返回一个 markdown 代码片段，其中包含格式如下的 JSON 对象列表：

```
{{  
  
  "title": string \ 候选电影中的电影名称  
  
  "like": Boolean \ true or false  
  
  "解释": 字符串 \ 解释为什么这个人喜欢或不喜欢候选电影  
  
}}
```

记住：候选电影中每个元素的每个布尔值和解释。

记住：解释必须与这个人喜欢和不喜欢的电影有关。

另外一种方式是采用微调大模型的方式进行预测，也是本项目的目标。

操作流程

数据说明

Microsoft News Dataset (MIND) 是一个用于新闻推荐研究的大规模数据集 [3]。它收集自 Microsoft News 网站的匿名行为日志。MIND 包含 100 万用户，大约 160k 篇英文新闻文章和超过 1500 万的曝光日志。每篇新闻文章都包含丰富的文本内容，包括标题、摘要、正文、类别和实体。每个曝光日志都包含该用户的点击事件、未点击事件和历史新闻点击行为。在这里，我们使用小数据量版本的 MIND 数据，即 MIND-small，包含了随机选取的 5 万个用户及其曝光日志。MIND-small 包括 4 个文件，即

- news.tsv (包括新闻的详细信息) ；
- behaviors.tsv (包括用户的点击历史和曝光日志行为) ；
- entity_embedding.vec (新闻中的实体向量表示) ；
- relation_embedding.vec (实体间关系向量表示)

news.tsv 示例数据如下图所示：

	ID	类别	子类别	标题	摘要	链接	标题实体	摘要实体
0	N55528	lifestyle	lifestyleroys	The Brands Queen Elizabeth, Prince Charles, an...	Shop the notebooks, jackets, and more that the...	https://assets.msn.com/labs/mind/AAGH0ET.html	[[{"Label": "Prince Philip, Duke of Edinburgh", "...	[]
1	N19639	health	weightloss	50 Worst Habits For Belly Fat	These seemingly harmless habits are holding yo...	https://assets.msn.com/labs/mind/AAB19MK.html	[[{"Label": "Adipose tissue", "Type": "C", "Wik...]	[[{"Label": "Adipose tissue", "Type": "C", "Wik...
2	N61837	news	newsworld	The Cost of Trump's Aid Freeze in the Trenches...	Lt. Ivan Molchanets peeked over a parapet of s...	https://assets.msn.com/labs/mind/AAJgNsz.html	[]	[[{"Label": "Ukraine", "Type": "G", "WikidataId...

其中，标题实体和摘要实体部分中的实体字段为：Label (Wikidata knwoledge 图中的实体名称) ;Type(实体在 Wikidata 中的类型) ;WikidataId (Wikidata 中的实体 ID) ；Confidence (实体链接的置信度) ；OccurrenceOffsets (字符级实体偏移量) ；SurfaceForms (原始文本中的原始

实体名称)。

behaviors.tsv 示例数据如下图所示：

曝光ID	用户ID	曝光时间	曝光前的新闻点击历史	曝光明细 (1表示点击；0表示非点击)
0	1	U13740 11/11/2019 9:05:58 AM	N55189 N42782 N34694 N45794 N18445 N63302 N104...	N55689-1 N35729-0
1	2	U91836 11/12/2019 6:11:30 PM	N31739 N6072 N63045 N23979 N35656 N43353 N8129...	N20678-0 N39317-0 N58114-0 N20495-0 N42977-0 N...
2	3	U73700 11/14/2019 7:01:48 AM	N10732 N25792 N7563 N21087 N41087 N5445 N60384...	N50014-0 N23877-0 N35389-0 N49712-0 N16844-0 N...

其中，曝光前的新闻点击历史表示在该曝光之前的新闻点击历史序列。曝光明细表示曝光给用户的新闻 ID 及其点击行为（1 表示点击；0 表示非点击），点击表示用户喜欢该新闻，也就是我们希望推荐给用户的新闻，注意曝光明细数据已经被随机打乱。

entity_embedding.vec 和 relation_embeddings.vec 文件包含通过 TransE 方法从 WikiData 中学习的实体和关系的 100 维嵌入，通常用于传统的推荐系统建模中，在此微调项目中不使用这两份数据。

数据预处理

基于上面的 MIND 数据集，我们采用 数据预处理.ipynb 文件进行数据预处理，首先处理 news.tsv 文件，获取 news ID 和标题的映射关系；然后处理 behaviors.tsv 文件，选择最近的 3 个曝光前的点击新闻标题，同时处理曝光明细，将最后一个曝光新闻作为预测目标，除最后一个外的曝光明细当作用户喜欢的（即为 1 点击的）和不喜欢的（即为 0 非点击的），同时我们定义了 system prompt。

对处理好的数据随机打乱后 8/2 划分，存储在 data/processed 文件下，训练集为 mind_train.json 文件（共 125572 个样本）；测试集为 mind_test.json 文件（共 31393 个样本）。

模型微调

在本项目中，我们使用优秀的 LLaMA-Factory 框架 [4] 基于 GLM-4-9B-Chat [5] 开源模型做新闻推荐的 LoRA 微调。

数据准备：将 mind_train.json 放在 LLaMA-Factory 的 data 目录下，然后修改 data 目录下的 dataset_info.json 文件，添加数据集表示为 mind，即

```
"mind": {  
  
    "file_name": "mind_train.json"  
  
}
```

模型微调：使用如下命令进行模型微调 `CUDA_VISIBLE_DEVICES=2 llamafactory-cli train ./new_train.yaml`，需要注意修改 yaml 文件内的大模型地址 `model_name_or_path`。

模型预测

当模型训练完毕后，我们使用 模型预测.ipynb 文件在 mind_test.json 文件上预测。具体来说，首先使用如下命令加载模型 `CUDA_VISIBLE_DEVICES=2 llamafactory-cli chat news_inference.yaml`，需要注意修改 yaml 文件内的大模型地址 `model_name_or_path`。由于数据量比较大，模型预测.ipynb 文件没有跑完全部测试集样本，在 3350 个样本上，得到的准确率为 97.8%。

参考文献

- [1] A Survey on Large Language Models for Recommendation, 2024
- [2] <https://pub.towardsai.net/using-llms-to-build-explainable-recommender-systems-e0b44365a46d>
- [3] <https://msnews.github.io/>
- [4] <https://github.com/hiyouga/LLaMA-Factory>
- [5] <https://github.com/THUDM/GLM-4>