

清华大学

综合论文训练

题目：基于大语言模型的推荐系统技术研究

系别：电子工程系

专业：信息与通信工程

姓名：程逸鸣

指导教师：袁坚教授

2024年6月17日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签 名：_____ 导师签名：_____ 日 期：_____

中文摘要

推荐系统技术根据用户过往交互行为提取喜好特征为其推荐物品，被广泛应用于用户生产内容平台、互联网购物平台及城市生活综合平台等多种场景，以期更精准高效地为用户提供推荐结果，从而提升用户体验、减小用户交互负担并综合提升社会经济效益。在现有的技术中，推荐系统技术主要有内容推荐、行为推荐以及混合推荐几种最主要的技术路径，并且以协同过滤为代表的行为推荐技术由于庞大的用户物品交互数据更为成熟。但是这种推荐技术更加关注行为而不是用户和物品本身的特征，缺乏对主体的理解。

随着自然语言处理技术的提升、文本向量化技术的成熟以及大语言模型技术的发展，基于语义理解内容本身的推荐系统技术也开始日益涌现。为了更好提升推荐系统推荐质量以及内容语义理解对推荐系统理解能力的提升，本工作提出了一种基于大语言模型技术的推荐系统框架，该框架能够实现物品内容的语义理解，并融合用户交互行为的长短期兴趣数据进行推荐：第一，为了能够实现用户长短期兴趣推荐，本工作提出了一种协同过滤模型的对注意力范围的量化优化方法，并基于大语言模型设计智能体用于识别用户偏好变化并调用相应模型从而实现长短期兴趣的自适应推荐。第二，为了让推荐系统对内容具备理解能力，利用自然语言处理技术对内容进行向量化，并利用大语言模型对内容进行理解，生成相似检索词，从而增强现有检索系统在内容相似物品上的检索能力。第三，对上述两种经过大语言模型优化的内容推荐召回的物品进行修正对齐，让最终的推荐系统能够混识别用户行为及内容，并提出一套大语言模型与推荐系统融合的规范范式。最后本工作进行了上述混合推荐系统的工程实现，在 MovieLens 数据集上实现行为推荐最多 26.63% 的召回率提升并在最终的混合架构中实现了 NDCG 最多 148.02% 的推荐质量提升。

关键词：推荐系统；大语言模型；机器学习；语义理解；自然语言处理

ABSTRACT

A recommender system extracts user preferences from past interactions to suggest items, widely used in platforms like user-generated content, online shopping, and urban services. These systems aim to provide accurate recommendations, reduce user interaction burden, and enhance user experience while improving socio-economic benefits. Current technologies include content-based, behavior-based, and hybrid recommendations. Behavior-based technologies, like collaborative filtering, are mature due to extensive user-item interaction data but focus on behavior over intrinsic features, lacking comprehensive understanding.

Advancements in natural language processing (NLP), text vectorization, and large language models (LLMs) have made content-based recommendations based on semantic understanding more prominent. This study proposes a recommender system framework based on large language model technology, which enables semantic understanding of item content and integrates both short-term and long-term user interaction behaviors for recommendation. Firstly, it optimizes collaborative filtering models to recommend both long-term and short-term interests. LLM-based agents identify changes in user preferences for adaptive recommendations. Secondly, it vectorizes content using NLP and LLMs to generate similar retrieval terms, enhancing existing retrieval systems. Thirdly, it aligns LLM-optimized content recommendations with user behavior, integrating both aspects. A standardized paradigm for LLM integration with recommender systems is also proposed. This hybrid system achieved a recall rate improvement of up to 26.63% in behavior recommendation and an NDCG improvement of up to 148.02% in the final framework on the MovieLens dataset.

Keywords: recommendation system; large language model; machine learning; semantic understanding; natural language processing

目 录

第 1 章 引言	1
1.1 研究背景	1
1.2 研究现状	2
1.2.1 基于用户交互的推荐系统	3
1.2.2 基于内容的推荐系统	5
1.2.3 基于大语言模型及其智能体的推荐系统	7
1.2.4 混合推荐系统框架	8
1.3 研究难点	8
1.4 研究方法	9
第 2 章 相关工作	11
2.1 基于用户交互的推荐系统算法	11
2.1.1 基于用户交互的推荐系统建模	11
2.1.2 基于矩阵分解的机器学习方法	12
2.2 基于内容的推荐系统算法	14
2.2.1 基于内容的向量化方法	14
2.2.2 基于相似度排序的推荐检索	15
2.3 小结	16
第 3 章 基于大语言模型智能体的混合推荐系统设计	17
3.1 概述	17
3.2 方法	18
3.2.1 物品推荐概述	18
3.2.2 智能体记忆模块存储及更新	18
3.2.3 长短期兴趣变化识别及自适应推荐	24
3.2.4 基于语义理解的内容推荐	27
3.2.5 基于大语言模型的混合意图识别推荐系统框架	33
3.3 小结	35

第 4 章 实验	36
4.1 实验设置	36
4.1.1 数据集	36
4.1.2 评价指标	36
4.2 实验内容	38
4.2.1 基于交互的长短期兴趣变化识别及自适应推荐系统实现	38
4.2.2 基于语义理解的内容推荐系统实现	39
4.2.3 基于大语言模型的混合意图识别推荐系统实现	40
4.3 实验结论	40
第 5 章 总结与未来工作	42
5.1 内容总结	42
5.2 未来工作	42
插图索引	44
表格索引	45
参考文献	46
致 谢	51
声 明	53
附录 A 外文资料的书面翻译	55
在学期间参加课题的研究成果	67

第1章 引言

1.1 研究背景

在现代信息检索系统设计中,待检索物品、用户数量及交需求规模急剧增大,使得准确识别用户偏好、快速精准地为用户提供信息检索结果成为需要格外着重优化的目标。早在公元前三世纪,人类便开始为产生的大量信息设计信息检索系统。例如人类第一座国际性图书馆—古希腊的亚历山大图书馆,涵盖了当时世界上绝大多数书籍或复制品,便已设计了初具雏形的信息分类检索系统^[1]。随着交互规模急剧及更新频率加快,这一被动式信息检索系统已经难以满足设计初衷,因此为减轻用户检索负担同时也继续满足高效精准的信息检索的需要,R.Armstrong^[2]等在1995年第一次提到了一种主动式信息检索系统,也是如今推荐系统的原始形态。随着移动互联网的发展以及用户-物品交互频次的进一步提升,业界急需更高效更精准的推荐系统技术。例如,2024年美团公司公布的财报^[3]中提到,其外卖业务全年产生了21.89亿次用户-物品交互,同比增长了23.9%。信息数量的增长以及交互类别的快速更新也对推荐系统的设计带来了巨大且全新的挑战。

为更好地应对交互数据过载带来的推荐系统负荷、更精准地识别用户意图同时更高效且贴近用户真实交互行为的推荐结果,研究者们开始探究推荐系统(Recommender System),希望通过数据挖掘的方式从大量诸如用户评分、用户交互序列等行为数据得到用户偏好的特征表示,并对用户未来的交互行为进行预测。例如在电影网站里,通过收集既往用户对电影的评分对用户尚未观看的电影可能做出的评分进行预测从而给出推荐列表^[4],以帮助用户更快地找到偏好的电影。在城市生活数字化领域,用户与POI(Point Of Interest,即用户可能产生交互的兴趣点)的交互,例如居民在吃穿住行方面的大量交互行为,可以用于预测其未来行为,从而为城市生活带来便利。

以上种种应用都极大程度的依赖过往交互数据的质量及数量,力求推荐系统能够预测出与推真实交互行为接近的推荐结果。移动互联网时代的大量交互记录为基于行为统计的未来预测技术提供了数据基础。而得益于自然语言处理技术的蓬勃发展,基于内容语义理解的推荐技术在近年来逐渐受到重视。为此,本工作提出了一种利用大语言模型对内容及行为进行融合的混合推荐系统框架及实现。

1.2 研究现状

根据推荐系统数据类型及建模方式的不同，推荐系统大致上可以分为基于用户交互的推荐系统 (Interaction-Based RS)、基于内容的推荐系统 (Content-Based RS) 以及混合推荐系统 (Hybrid RS)^[5]。随着例如 ChatGPT 等的一系列大语言模型的发布，依靠着其对自然语言的出色理解能力以及对先验知识出色的泛化能力，逐渐形成了一种基于大语言模型技术逐渐成型的基于智能体的推荐系统 (Agent-Based RS)。其中基于用户交互的推荐系统的基础是大量的用户与物品交互记录以及用户对物品的评分，比较具有代表性的一类算法是基于用户交互评分的协同过滤算法 (Collaborative Filtering, CF)^[6]，能够识别出有相似交互行为的用户，并将相似用户之间差异化的剩余物品进行交叉推荐；或反之对有相似用户群体的物品产生交互的用户之间进行交叉推荐。这一技术及其依赖于既有交互数据的数量以及质量，因此在推荐系统搭建初期会面临及其严峻的冷启动问题，同时面对跨领域的推荐也会有先验数据缺乏的问题。协同过滤算法作为最具影响力的推荐系统算法之一也逐渐有了很多优化和变体，例如基于模型 (Model-Based CF)^[7] 及基于记忆 (Memory-Based CF) 的协同过滤算法，后者又可以细分为基于用户 (User-Based CF)^[8] 或物品 (Item-Based CF)^[9] 的不同变体。与之相对应的是基于内容语义理解的推荐系统，这类推荐系统的核心是对推荐系统中的物品以及用户本身作为内容进行特征提取以及向量化，从内容本身获取相似性。这一类算法从原理上就一定程度上规避了基于用户交互所产生的交互数据依赖从而能够对于新的物品以及新的用户基于语义理解能够有一定的泛化能力。同时由于其极大程度上依赖特征提取的质量，因此特征提取质量不佳会极大程度影响最终推荐系统的推荐效果。

随着推荐系统技术的逐步发展，一些研究者^[5]逐渐意识到了两种推荐系统算法各自有独到的优势以及劣势，因此一个更高质量的推荐系统应当能够很好的兼顾用户行为以及推荐系统数据库中的用户及物品的内容信息，因此可以对基于不同原理的推荐系统进行一些融合对齐。常见的方法例如对不同的推荐系统生成的评分进行加权 (Weighted)、根据一定的规则进行模型的调用 (Weighted) 以及推荐系统之间的级联 (Cascaded) 或混合 (Mixed) 的方式对多个推荐系统进行模型级别或特征级别的整合。随着大语言模型技术的成熟，可以利用基于大语言模型的智能体的出色推理能力，在推荐系统中深入理解推荐内容的语义、深入理解用户画像同时理解用户交互的情景从而给出更灵活精准的推荐结果。

1.2.1 基于用户交互的推荐系统

Paul Resnick^[6]等人最早将协同过滤这一概念引入至推荐系统技术之中。具体而言，协同过滤的设计初衷大致是出于以下对交互行为的假设：具有相似性的用户会对相似的物品感兴趣，而反之具有相似性的物品也会吸引相似的用户产生交互行为。这一朴素但直观的思想贯穿了协同过滤算法以及之后的几种变体。最常见的协同过滤算法主要有基于用户 (User-Based CF)^[8]、基于物品 (Item-Based CF)^[9]及基于模型 (Model-Based CF)^[7]的协同过滤算法。

基于用户的协同过滤算法^[8] 在已有大量用户物品交互评分数据的情况下，基于用户的协同过滤首先得到各个用户对所有物品的交互评分记录作为向量表示，计算用户间的相似度，对有更高相似度之间的用户之间采取交叉推荐，其中相似度的计算参考了统计学中常用的皮尔森相似度^[10]。

具体而言用户 U_a 和用户 U_b 的相似度可以表示为：

$$\text{sim}(U_a, U_b) = \frac{\sum_{i \in I_a \cap I_b} (S_{a,i} - \bar{S}_a)(S_{b,i} - \bar{S}_b)}{\sqrt{\sum_{i \in I_a \cap I_b} (S_{a,i} - \bar{S}_a)^2 \sum_{i \in I_a \cap I_b} (S_{b,i} - \bar{S}_b)^2}} \quad (1.1)$$

其中 $\text{sim}(U_a, U_b)$ 是用户 U_a 和用户 U_b 之间的相似度， $S_{a,i}$ 和 $S_{b,i}$ 分别是用户 U_a 和 U_b 对项目 i 的评分， \bar{S}_a 和 \bar{S}_b 是用户 U_a 和 U_b 的平均评分， $I_a \cap I_b$ 是用户 U_a 和 U_b 都评分的项目集合。在定义了相似度之后对于单个用户就可以得到其邻域 N ，可以基于领域中的用户对当前用户未有交互的物品评分进行预测，从而给出推荐列表。

具体而言，用户 U_a 对项目 i_j 的预测评分可以表示为：

$$\hat{S}_{a,j} = \bar{S}_a + \frac{\sum_{b \in N(a)} \text{sim}(U_a, U_b) \cdot (S_{b,j} - \bar{S}_b)}{\sum_{b \in N(a)} |\text{sim}(U_a, U_b)|} \quad (1.2)$$

其中， $\hat{S}_{a,j}$ 是用户 U_a 对项目 i_j 的预测评分 $N(a)$ 是与用户 U_a 最相似的一组用户集合。

基于物品的协同过滤算法^[9] 除了基于用户的协同过滤算法以外，与之对应的还有基于物品的协同过滤算法。两种算法的核心思想是相似的，区别是基于物品的协同过滤将物品用用户交互的向量表示，而在实际的推荐系统环境中，用户数量是远远大于物品数量的，因此在减小向量数量的同时，增大了向量维度，具体使用哪一种依赖于具体的数据场景。在将物品向量化后可计算物品之间的相似度。

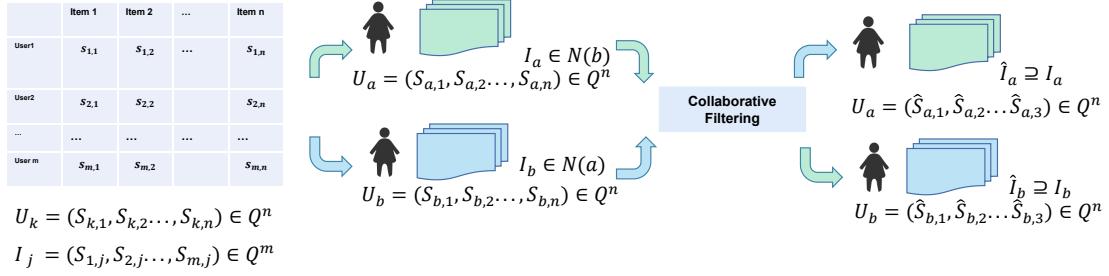


图 1.1 协同过滤示意图

具体而言，物品 I_i 和物品 I_j 的相似度可以表示为：

$$\text{sim}(I_i, I_j) = \frac{\sum_{u \in U_i \cap U_j} (S_{u,i} - \bar{S}_i)(S_{u,j} - \bar{S}_j)}{\sqrt{\sum_{u \in U_i \cap U_j} (S_{u,i} - \bar{S}_i)^2 \sum_{u \in U_i \cap U_j} (S_{u,j} - \bar{S}_j)^2}} \quad (1.3)$$

其中 $\text{sim}(I_i, I_j)$ 是物品 I_i 和物品 I_j 之间的相似度， $S_{u,i}$ 和 $S_{u,j}$ 分别是用户 U_u 对物品 I_i 和物品 I_j 的评分， \bar{S}_i 和 \bar{S}_j 是物品 I_i 和物品 I_j 的平均评分， $U_i \cap U_j$ 是对物品 I_i 和 I_j 都评分的用户集合。在定义了相似度之后，对于单个物品就可以得到其邻域 N ，可以基于邻域中的物品对当前用户未有交互的物品评分进行预测，从而生成推荐列表。

具体而言，用户 U_u 对物品 I_i 的预测评分可以表示为：

$$\hat{S}_{u,i} = \frac{\sum_{j \in N(i)} \text{sim}(I_i, I_j) \cdot S_{u,j}}{\sum_{j \in N(i)} |\text{sim}(I_i, I_j)|} \quad (1.4)$$

其中， $\hat{S}_{u,i}$ 是用户 U_u 对物品 I_i 的预测评分， $N(i)$ 是与物品 I_i 最相似的一组物品集合。可以看到两种推荐算法的核心思想是对称的，区别在于具体数据应用场景中物品和用户哪一个维度更高，更有利于快速计算。

基于模型的协同过滤算法^[7] 在实际的推荐系统中，随着用户数量以及物品数量急剧增大，实际的交互矩阵是极为稀疏的，使得基于物品以及基于用户的协同过滤算法效率极为低下。同时也并非所有的交互矩阵都显式的包含用户的评分。基于模型的协同过滤算法通过对交互矩阵的特征进行学习从而给出推荐结果，如矩阵分解 (Matrix Factorization)、隐语义模型 (Latent Semantic Models)、及深度学习 (Deep Learning)。

矩阵分解是最常用且高效的一种实现方法方法，例如通过将交互矩阵通过

SVD 分解得到用户和物品的潜在向量特征表示, 这些特征相比于原先的交互矩阵相比有更小的维度。

具体而言, 假设我们有一个用户-物品评分矩阵 R , 其中 R_{ui} 表示用户 u 对物品 i 的评分。矩阵分解方法通过将矩阵 R 分解为两个低维矩阵 P 和 Q , 即:

$$R \approx PQ^T \quad (1.5)$$

其中, P 是用户的潜在特征矩阵, Q 是物品的潜在特征矩阵。对于用户 u 对物品 i 的预测评分 \hat{R}_{ui} 可以表示为:

$$\hat{R}_{ui} = P_u^T Q_i \quad (1.6)$$

为了找到最佳的 P 和 Q , 我们可以通过优化以下损失函数:

$$L = \sum_{(u,i) \in R} (R_{ui} - P_u^T Q_i)^2 + \lambda(\|P_u\|^2 + \|Q_i\|^2) \quad (1.7)$$

其中, λ 是正则化参数, 用于防止过拟合。基于矩阵方法由于其简单高效, 并且随着高性能计算技术不断发展, 这一过程的计算效率也在不断提升。本工作中基于用户行为的推荐优化业主要基于此算法。

综上, 基于用户交互的推荐系统更多的关注于用户交互行为, 从行为中提取行为特征并给出预测, 往往不需要关注推荐内容本身的特征, 因此在面临全新的推荐环境时会面临交互数据缺乏稀疏产生的冷启动问题。同时用户偏好短期产生变化时, 模型固定可能无法及时产生长短期变化的推荐结果, 也是需要着重优化的目标。

1.2.2 基于内容的推荐系统

推荐系统最早的思想就是基于内容本身进行推荐, 但碍于特征提取技术发展缓慢, 并没有基于用户交互行为的推荐系统高效。Gerard Salton^[11]首次提到了信息检索 (Information Retrieval) 与信息过滤 (Information Filtering) 之间的关联与区别。前者主要关注在大型数据库中被动提供检索结果, 而后者则是现推荐系统的雏形。信息过滤系统会持续检测系统中的信息流以及数据变化, 并且主动的提供检索结果给用户。在 Gerard 在信息过滤技术里提出了向量空间模型 (Vector Space Model), 开创性的将内容以及检索都映射到了向量空间中, 检索结果由检索向量

以及内容向量之间的相似度给出。

因此基于内容的推荐算法的核心就是对用户和物品本身进行特征提取^[12], 利用向量嵌入技术 (embedding) 将用户、物品以及检索行为映射到向量空间, 以用户向量化为例, 通过分析用户档案及交互记录来构建用户偏好的模型, 可以表示为一个加权向量:

$$\mathbf{p} = (p_1, p_2, \dots, p_m) \quad (1.8)$$

其中, p_j 表示用户对特征 j 的偏好权重。在得到不同用户的向量表示之后就可以计算相似度, 相似度的计算是基于向量的余弦相似度,

$$\text{sim}(\mathbf{p}, \mathbf{x}_i) = \frac{\mathbf{p} \cdot \mathbf{x}_i}{\|\mathbf{p}\| \|\mathbf{x}_i\|} \quad (1.9)$$

其中, $\mathbf{p} \cdot \mathbf{x}_i$ 是向量的点积, $\|\mathbf{p}\|$ 和 $\|\mathbf{x}_i\|$ 分别是向量的范数。

向量点积的公式为:

$$\mathbf{p} \cdot \mathbf{x}_i = \sum_{j=1}^m p_j \cdot x_{ij} \quad (1.10)$$

向量范数的公式为:

$$\|\mathbf{p}\| = \sqrt{\sum_{j=1}^m p_j^2} \quad (1.11)$$

$$\|\mathbf{x}_i\| = \sqrt{\sum_{j=1}^m x_{ij}^2} \quad (1.12)$$

为了生成推荐, 系统为每个项目 i 计算一个得分, 然后根据得分从高到低排序, 选择得分最高的项目推荐给用户。基于内容的推荐系统的核心就是特征提取及向量化, 这一部分的介绍及优化将在第二章及第三章着重展开。因此基于内容的推荐系统很大程度上依赖于特征提取以及向量化的质量。尽管如此, 基于内容的推荐系统算法有着很多独特的有点。首先推荐结果能够更加个性化, 因为结果基于对用户及交互内容的语义分析而非统计特性。其次由于向量化模型本身蕴含了隐藏特征, 因此及时面对全新的推荐系统或跨领域推荐, 也可以极大的缓解冷启动问题。同时, 由于基于内容的推荐系统保留了很丰富的语义特征, 因此推荐系统整体的可解释性也是比较强的。

1.2.3 基于大语言模型及其智能体的推荐系统

随着大语言模型技术的发展，一系列自然语言处理领域的公开问题得到了解决。主要有基于大语言模型及基于大语言模型智能体的两大类推荐系统。大语言模型例如 ChatGPT、Llama 等在推荐系统中凭借出色的语义理解以及长序列预测能力使其能够帮助推荐系统更好地获取语义特征并在较长序列上给出较准确的预测效果。大语言模型能够为推荐系统带来性能提升，主要体现在通过大量语料训练得到预训练模型能够提供准确的语义理解以及高质量的特征提取，例如可以从用户的档案、评论交互、交互列表中准确提取用户兴趣；此外大语言模型一定程度上可以兼顾基于行为以及基于内容的推荐系统，从而在各自应用场景中取长补短，具体而言，类似 Transformer^[13]的架构能够给出基于长序列的序列化预测，可以很好的基于交互序列给出继续预测的结果；同时对于内容的强大语义理解也可以在缺乏交互数据或交互矩阵稀疏的情况下很大程度上基于语料库知识缓解冷启动问题。基于大语言模型或智能体的推荐系统已经有一系列颇具影响力成果发布，例如基于 Bert 模型的 Bert4Rec^[14]序列预测推荐系统、基于 Transformer 架构的 Transformer4Rec^[15]、基于大语言模型智能体的 GPT4Rec^[16]等等。

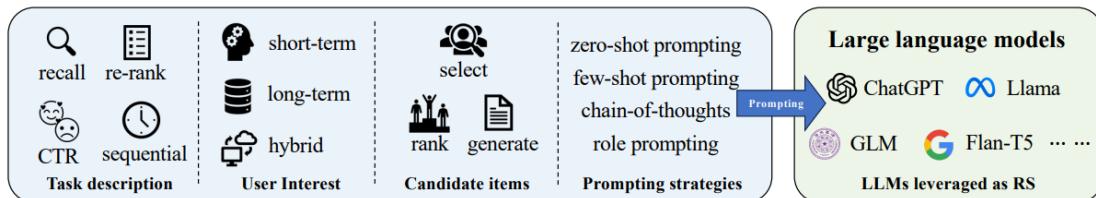


图 1.2 基于大语言模型的推荐系统示意图，该图来自文献^[17]

事实上大语言模型在推荐系统中的角色^[17]并不是单一的。整体而言，大语言模型在推荐系统中可以用于语义理解，直接为用户或物品进行向量嵌入；也可以利用其语义理解能力总结概括用户意图，在不同的推荐场景之下进行混合意图识别并进行不同的推荐；还可以基于上下文理解对已经召回的物品做重排、粗排及细排等操作；也可以作为代理人 (agent) 进行流程控制等。

大语言模型能力强大且具有极强的泛化能力，基于大语言模型的智能体能够利用其强大的推理能力，对传统的推荐系统各个环节存在的短板进行优化，也可以综合行为模型和内容模型的优点进行融合推荐。因此基于大模型进行行为-内容混合架构设计是可行的，通过大语言模型或其智能体在不同的环节参与推荐召回并且承担不同的角色，对用户内容以及交互上下文进行更深入的语义理解并给出更高质量的推荐结果，是本工作的重点以及未来工作需要着重优化的目标。

1.2.4 混合推荐系统框架

基于上述的多种推荐系统以外，还有将多种推荐系统进行混合从而综合提升性能、结合多种推荐系统的优点并一定程度上规避各自缺点的混合推荐系统架构^[5]。常见的混合策略^[18-21]主要包括加权融合、切换混合、级联融合、混合模型、特征组合、特征增强和元级混合等方法。

基于上述混合推荐系统技术，本工作提出了一种利用大语言模型对内容和行为进行融合的混合推荐系统框架及其实现。

具体而言，在基于用户行为的推荐系统中，主要使用了不同注意力范围的线性加权网络从而可以调整兴趣特征提取范围，并结合大语言模型智能体的推理能力进行模型切换混合。

在基于内容的推荐系统中主要使用大语言模型智能体的上下文推理预测能力，对用户进行兴趣理解以及未来短期兴趣预测，产生相关检索词进行特征增强，再利用大语言模型智能体进行模型调度从而给出推荐结果。

在最终的行为-内容融合架构中主要使用了模型间对齐修正的混合技术，在基于内容进行物品召回以后，对结果增加行为特征混合从而进行行为特征的补充，从而更加全面的总结用户交互行为及内容特征并综合给出推荐结果。

1.3 研究难点

如上一节综述中所提及的，在推荐系统问题上，已经有了基于用户交互、基于内容以及最新出现的基于大语言模型以及种种算法互相混合的多种策略。然而，将大语言模型很好的融入已有的推荐系统框架任然是一个极为前沿的问题。具体而言，各个推荐系统算法经常会有各自的局限性：基于用户交互行为的推荐系统算法及其依赖大量的先验交互数据以及由此带来的冷启动问题；基于内容的推荐系统算法会受限于特征提取质量以及特征泛化问题；基于大语言模型的推荐系统受限于算力以及远程运算资源开销，难以成为单一的推荐系统组成。这些工作也正说明，急需一种将大语言模型涵盖其中，能够很好的综合已有推荐系统优点，同时能够以合理的开销让大语言模型在关键节点释放性能的平衡架构。

为此，本工作设计了一套基于大语言模型的混合推荐系统框架，并且对框架的不同部分基于大语言模型分别进行优化，这一问题的研究难点主要可归纳如下：

第一，基于用户交互行为的模型缺乏对于推荐对象个性化的推荐修正，用户存在长短期兴趣偏差，而调用模型是统一的，因此将在部分用户的推荐结果上产

生较大偏差。而由于基于用户交互行为本身缺乏对内容本身的理解，识别兴趣变化会产生额外的开销，现代推荐系统中的交互矩阵规模是十分庞大的，将会带来巨大的计算开销。

第二，基于内容的推荐系统能够更好的提取用户及物品特征，但是由于不包含交互行为特征，因此对于一些实时热点产生短时间激增的大规模类似交互行为无法产生响应，因而产生推荐偏差。同时推荐过程里若召回数量过大则会导致单次计算开销增大，而最终也需要精细排序，因此会产生较大的无用计算资源开销，减缓推荐系统整体性能。

第三，大语言模型在推荐系统工作流中的定位模糊以及范式缺乏。推荐系统本身是一个庞大的工程系统实现，同时也需要实现一整套具备信息检索功能的系统。在大语言模型加入后，需要尽可能的在语义理解及特征提取的过程里发挥大语言模型的性能，同时也不使系统计算开销及复杂度大幅度提升，同时也需要对齐基于交互行为及基于内容的推荐系统。因此需要一套普适性更强同时平衡性能提升及开销的适用于不同数据集的推荐系统框架及大语言模型基座。

1.4 研究方法

基于第 1.2 部分中已有的相关研究，针对第 1.3 部分提出的研究难点，本工作提出了一套基于大语言模型的混合推荐系统框架，融合基于行为的推荐系统以及基于内容的推荐系统，强化推荐系统的语义理解以及特征提取功能，增强了整体模型的交互预测能力，研究方法主要包含以下内容：

第一，本工作设计了基于用户交互及大语言模型智能体的长短期兴趣变化识别及自适应推荐组件，并在传统的矩阵分解算法上增加了权重调整的线性层使得模型的注意力范围可以人为控制从而适应长短期推荐的不同情景。同时设计了智能体的记忆更新机制，能够更准确的识别出用户兴趣变化情况并且一定程度上细分量化出变化程度并作出模型调度的决策，从而弥补了传统基于行为的推荐系统对于内容理解的缺失以及模型更新的开销过大。

第二，本工作基于 Li 等人的 GPT4Rec^[16]架构基础上修正了某些由于内容中过长干扰项产生的向量化偏差以及相似度计算误差。本工作在此基础上提出了基于正则阈值过滤的 Query-Search 框架用于对基于内容的推荐系统的特征识别增强。依靠大语言模型的语义理解及泛化能力，生成未来检索词预测并快速生成粗排结果用于后续细化，规避了传统算法中计算资源浪费的问题。

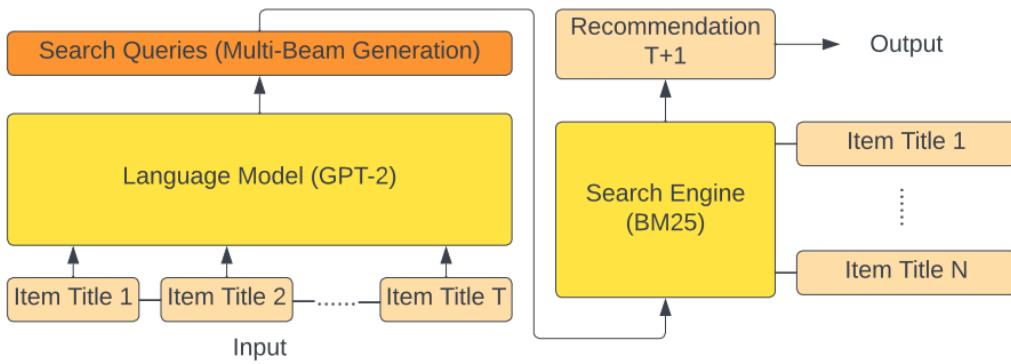


图 1.3 GPT4Rec 流程示意图，该图来自文献^[16]

第三，本工作提出了一整套基于大语言模型的完整的混合推荐系统对齐框架，通过锚定基于内容推荐系统推荐结果并利用大语言模型的语义理解能力用基于交互的长短时自适应网络进行修正，从而使得最终结果能够兼顾交互行为及内容理解，同时也能够节省大语言模型的资源开销。这一框架提出了一种将大语言模型融入现有推荐系统体系的方法，并且在未来存在高度可扩展性及泛化性。

第2章 相关工作

2.1 基于用户交互的推荐系统算法

基于第一章所提到的推荐系统技术概论，本章将基于大语言模型技术特点选取基于交互行为以及基于内容的合适的推荐系统算法简要介绍重要的相关工作以及其实现原理。主要介绍基于交互行为进行推荐的问题建模以及较基础且常用的基于矩阵分解进行推荐模型训练的算法。而后对于常见的内容向量化方法以及基于相似度的检索技术进行介绍。

2.1.1 基于用户交互的推荐系统建模

对于基于用户交互的推荐系统而言，最核心的要素是对于大量用户交互行为的统计特性进行提取，而对于用户本身以及物品本身的关注则较少。现代推荐系统中基于矩阵分解进行用户交互特征提取基于一系列基本的假设以及现状：1) 推荐系统中存在大量的交互数据。这些交互数据数量极其庞大，为后续基于交互的推荐系统算法设计提供了数据基础。同时交互种类是多样的，最常见的有对物品进行评分，本工作中业主要采取评分作为特征。此外还有点击以及购买等不同的交互种类，不同的交互种类也可以用强度进行区分，例如点击交互视为程度最低的交互，而购买视为程度最高的交互并给出评分量化表等。2) 实际的交互矩阵由用户及物品的序列组成，因此维度等于用户数量与物品数量之积，而在现代推荐系统中，物品数量与用户数量本身就十分庞大且日益更新，直接导致最终的交互矩阵的维度十分庞大并且会以极快的速度更新。在这一前提下，将原先高纬度的交互矩阵进行矩阵分解，在保留原先物品以及用户数量的前提下，能够按照所需精度保留各自向量维度，从而大大降低用户向量及物品向量的总体维度，使得计算以及更新的开销得到降低。3) 在实际的推荐系统之中，用户实际只会各自与极小一部分的物品产生过交互行为，而大多数物品也只有小部分用户的交互记录。同时推荐系统中有着极为明显的长尾效应^[22-23]，也就是头部的少部分物品占据了整体交互数据的绝大部分，而大多数普通物品整体的交互总和仅仅占了小部分。因此交互矩阵虽然维度巨大，但实际十分稀疏^[24]，绝大多数矩阵元素都是未有交互。因此在原先矩阵上直接进行特征提取运算效率十分低下，而转换为物品和用户的低维向量后数据存储与运算效率都得到了提升。4) 基于行为的推荐系统假设用户基于自身喜好进行交互行为，这种隐藏的喜好特征在一定程度上能够在

时间序列中维持稳定特性，反映在交互序列中便是依据过往交互行为可以通过特征提取得到这一种隐藏喜好，同时基于其稳定性能够用于预测未来交互。

基于上述假设前提，对于基于用户交互的推荐系统可以抽象成如下问题描述：给定一个用户集合 $\mathcal{U} = \{u_1, u_2, \dots, u_m\}$ 和一个项目集合 $\mathcal{I} = \{i_1, i_2, \dots, i_n\}$ 。交互状态空间可以表示为一个矩阵 $\mathbf{R} \in \mathbb{R}^{m \times n}$ ，其中 \mathbf{R}_{ui} 表示用户 u 对项目 i 的评分。如果用户 u 没有对项目 i 评分，则 \mathbf{R}_{ui} 为缺失值。推荐系统目标为根据用户在前一部分项目上的评分数据 $\{I_1, I_2, \dots, I_k\}$ ，预测用户在后一部分项目上的评分 $\{I_{k+1}, \dots, I_n\}$ 。即，利用前 k 个项目的评分记录来预测剩余项目的评分，从而为用户推荐他们可能感兴趣的项目。为了评估推荐系统的效果，我们可以将预测值 $\{I'_{k+1}, \dots, I'_n\}$ 与真实值 $\{I_{k+1}, \dots, I_n\}$ 进行比较。相似程度越高的代表推荐系统效果越好。

当然这一评价方式虽然是惯用的，但也确实存在局限性。由于推荐系统介入改变了交互序列，因此无法排除曝光行为本身的因果效应与实际交互的耦合，碍于实时数据集收集困难，这一点有望在真实推荐系统环境及数据集中得到进一步改善。

2.1.2 基于矩阵分解的机器学习方法

基于用户交互行为的模型推荐算法中最常用的是基于矩阵奇异值分解的方法。首先，我们将评分矩阵 \mathbf{R} 分解为三个矩阵：用户特征矩阵 \mathbf{U} 、奇异值矩阵 Σ 、项目特征矩阵 \mathbf{V} 。公式表示如下：

$$\mathbf{R} \approx \mathbf{U} \Sigma \mathbf{V}^T \quad (2.1)$$

实际推荐系统中由于维度过大，为了降维，取前 k 个奇异值及向量，得到截断交互矩阵，记为 $\hat{\mathbf{R}}$ ，实际的推荐行为基于该矩阵：

$$\mathbf{R} \approx \mathbf{U}_k \Sigma_k \mathbf{V}_k^T \triangleq \hat{\mathbf{R}} \quad (2.2)$$

为了提高预测的准确性，通常需要最小化以下损失函数：

$$\mathcal{L} = \sum_{(u,i) \in \mathcal{K}} (\mathbf{R}_{ui} - (\mathbf{U}_k \Sigma_k \mathbf{V}_k^T)_{ui})^2 + \lambda(\|\mathbf{U}_k\|^2 + \|\mathbf{V}_k\|^2) \quad (2.3)$$

其中， \mathcal{K} 是所有已知评分的集合， λ 是正则化参数， $\|\cdot\|^2$ 代表 Frobenius 范数，用于正则化，防止过拟合。

在学习过程中，首先随机初始化矩阵 \mathbf{U}_k 和 \mathbf{V}_k 。然后，基于梯度下降^[25]优化损失函数 \mathcal{L} 。

计算梯度并更新参数， η 是学习率：

$$U'_k = U_k - \eta \frac{\partial \mathcal{L}}{\partial U_k} = U_k - 2 \sum_{(u,i) \in \mathcal{K}} (R_{ui} - (U_k \Sigma_k V_k^T)_{ui}) V_k \Sigma_k + 2\lambda U_k \quad (2.4)$$

$$V'_k = V_k - \eta \frac{\partial \mathcal{L}}{\partial V_k} = V_k - 2 \sum_{(u,i) \in \mathcal{K}} (R_{ui} - (U_k \Sigma_k V_k^T)_{ui}) \Sigma_k^T U_k^T + 2\lambda V_k \quad (2.5)$$

将原始数据集 \mathcal{K} 分割成训练集 $\mathcal{K}_{\text{train}}$ 、验证集 \mathcal{K}_{val} 和测试集 $\mathcal{K}_{\text{test}}$ ：

$$\mathcal{K} = \mathcal{K}_{\text{train}} \cup \mathcal{K}_{\text{val}} \cup \mathcal{K}_{\text{test}} \quad (2.6)$$

定义训练集误差、验证集误差及测试集误差^[26]：

$$\text{MSE}_{\text{train}} = \frac{1}{|\mathcal{K}|} \sum_{(u,i) \in \mathcal{K}} (R_{ui} - \hat{R}_{ui})^2 \quad (2.7)$$

$$\text{MSE}_{\text{val}} = \frac{1}{|\mathcal{K}_{\text{val}}|} \sum_{(u,i) \in \mathcal{K}_{\text{val}}} (R_{ui} - \hat{R}_{ui})^2 \quad (2.8)$$

$$\text{MSE}_{\text{test}} = \frac{1}{|\mathcal{K}_{\text{test}}|} \sum_{(u,i) \in \mathcal{K}_{\text{test}}} (R_{ui} - \hat{R}_{ui})^2 \quad (2.9)$$

整个流程的伪代码如下：

算法 2.1 基于 SVD 的推荐系统算法

Require: 用户集合 U ，项目集合 I ，评分矩阵 R ，奇异值个数 k ，学习率 η ，正则化参数 λ ，迭代次数 num_epochs

Ensure: 预测评分矩阵 \hat{R}

- 1: 随机初始化 $U_k \in \mathbb{R}^{m \times k}$ 和 $V_k \in \mathbb{R}^{n \times k}$
 - 2: **for** epoch = 1 to num_epochs **do**
 - 3: **for each** $(u, i) \in R$ **do**
 - 4: 计算误差 $e_{ui} = R_{ui} - (U_k \Sigma_k V_k^T)_{ui}$
 - 5: 更新 U_k 和 V_k
 - 6: $U_k[u, :] = U_k[u, :] + \eta (e_{ui} V_k[i, :] - \lambda U_k[u, :])$
 - 7: $V_k[i, :] = V_k[i, :] + \eta (e_{ui} U_k[u, :] - \lambda V_k[i, :])$
 - 8: **end for**
 - 9: **end for**
 - 10: 计算预测评分矩阵 $\hat{R} = U_k \Sigma_k V_k^T$
 - 11: **return** \hat{R}
-

2.2 基于内容的推荐系统算法

与基于交互行为的推荐系统相对应，基于内容的推荐系统算法并不会将交互记录等视为行为信号并以统计学的视角对损失函数进行优化，而是从语义分析的角度给予用户及物品向量化，因此一个合理的特征提取策略以及高质量的向量化嵌入是极为重要的。高质量的向量化直接决定了推荐系统的推荐效果。本节将介绍后续实验中用于优化的三种向量化方法。

2.2.1 基于内容的向量化方法

符号同前，本节介绍三种向量化方法在基于内容的推荐系统中的具体应用。

2.2.1.1 TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency)^[27]在自然语言处理领域常用于计算单个 token 对于文档的重要程度，而在推荐系统中可以用来衡量单个项目项目在用户交互历史中的重要程度。TF-IDF 包括两个组成：词频 (TF) 和逆文档频率 (IDF)。在推荐系统中可以引申为项目频率和逆用户频率。

项目频率 (TF) 表示项目 i 在用户 u 中出现的次数。具体为：

$$\text{TF}(i, u) = \frac{f_{i,u}}{\sum_{i' \in u} f_{i',u}} \quad (2.10)$$

其中 $f_{i,u}$ 是项目 i 在用户 u 中出现的次数，分母代表用户 u 项目总数。

逆用户频率 (IDF) 表示项目 i 的普遍重要性。具体为：

$$\text{IDF}(i, \mathcal{U}) = \log \left(\frac{|\mathcal{U}|}{|\{u \in \mathcal{U} : i \in u\}|} \right) \quad (2.11)$$

其中， $|\mathcal{U}|$ 是用户集 \mathcal{U} 中用户的总数，分母是包含项目 i 的用户数。

TF-IDF 结合项目频率及逆用户频率，具体为：

$$\text{TF-IDF}(i, u, \mathcal{U}) = \text{TF}(i, u) \times \text{IDF}(i, \mathcal{U}) \quad (2.12)$$

TF-IDF 值越高，项目在用户交互记录中的重要性越大。将每个项目的特征表示为一个向量。对于每个项目 i ，我们计算所有用户 \mathcal{U} 中该项目的 TF-IDF 值，形成一个向量 \mathbf{v}_i ，其第 j 个分量为：

$$\mathbf{v}_i[j] = \text{TF-IDF}(i, u_j, \mathcal{U}) \quad (2.13)$$

其中， u_j 表示第 j 个用户。每个项目 i 都可以表示为一个 TF-IDF 向量 \mathbf{v}_i ，在后续用于计算相似度，从而生成推荐结果。

2.2.1.2 BERT

BERT (Bidirectional Encoder Representations from Transformers)^[28]在自然语言处理领域用于文本上下文表示，而在基于内容的推荐系统中也可以用于物品的语义表示。Bert 是一个大模型，通常从零开始训练是比较大的，一般应用会包含预训练及微调，本工作中主要使用预训练的 Bert 来验证框架正确性。使用 Bert 进行物品语义向量化通常包含以下步骤：

预处理 将项目 i 的档案信息 Doc_i 进行分词和标准化处理。

向量化 将档案信息作为预训练 BERT 输入，从而得到上下文特征表示 \mathbf{v}_i 。

$$\mathbf{v}_i = \text{BERT}(\text{Doc}_i) \quad (2.14)$$

用户特征建模 对于每个用户 u ，基于其历史交互项目的特征向量来建模其偏好。假设用户 u 交互过的项目集合为 \mathcal{I}_u ，则用户的偏好向量 \mathbf{p}_u 可以表示为这些项目特征向量的加权平均：

$$\mathbf{p}_u = \frac{1}{|\mathcal{I}_u|} \sum_{i \in \mathcal{I}_u} \mathbf{v}_i \quad (2.15)$$

得到用户向量 \mathbf{p}_u 与候选项目特征向量 \mathbf{v}_i 后通过相似度计算得到推荐结果。

2.2.1.3 SpaCy

SpaCy 是一个自然语言处理开源库，同样提供了语义理解及向量化的工具。以下将简单介绍其原理及如何运用于推荐系统：

预处理 通过预定义词典对档案信息进行分词。

词性标注 通过统计数据以及预训练模型为每个分词标注词性。

依存句法分析 构建依存树以识别句子中词汇之间的依存关系并建立句子结构

命名实体识别 (NER) 通过预训练模型和规则识别文本中具有特定意义的实体。

词嵌入 通过预训练模型诸如 GloVe (Global Vectors for Word Representation)^[29] 和 fastText^[30] 将分词 w 映射为固定长度向量 \mathbf{e}_w 。

与 Bert 类似，通过输入档案信息得到用户向量 \mathbf{p}_u 与候选项目特征向量 \mathbf{v}_i 用于后续相似度计算以及推荐。

2.2.2 基于相似度排序的推荐检索

基于上节得到了用户及物品的特征向量表示，就可以计算相似度并进行推荐，这一过程的伪代码如下：

算法 2.2 基于相似度排序的推荐系统算法

Require: 用户偏好向量 \mathbf{p}_u , 项目特征向量集合 $\{\mathbf{v}_i\}$, 推荐数量 N

Ensure: 推荐项目列表

- 1: **for** 每个候选项目 i **do**
 - 2: 计算相似度 $\text{sim}(u, i) = \mathbf{p}_u \cdot \mathbf{v}_i$
 - 3: **end for**
 - 4: 按相似度从高到低排序候选项目
 - 5: 返回前 N 个项目作为推荐结果
-

2.3 小结

本章介绍了基于用户交互行为以及基于内容的推荐系统的基本思想以及几个常用的实现。本章介绍内容也作为第三章本工作设计的基于大语言模型的混合推荐系统框架及实现的基础。后续将介绍如何通过大语言模型对基础实现进行优化并最终将基于用户交互行为的推荐系统以及基于内容的推荐系统进行对齐融合的一种实现。

第3章 基于大语言模型智能体的混合推荐系统设计

3.1 概述

近年来，大语言模型^[31]技术飞速崛起，解决了一系列自然语言处理领域的公开问题。在本工作中主要采用 GPT3.5-turbo^[32]作为预训练大语言模型提供语义理解的接口。大语言模型依靠强大的预训练知识库以及出色的语义理解能力为推荐系统在语义特征提取、流程理解以及情感分析等功能上提供出色的功能接口，是本工作中混合推荐系统的核心设计框架的基础。

本工作中采用大语言模型的必要性及设计优势的论述可以概括如下：

兴趣理解 在推荐系统中，有很多交互信息并非以量化交互得分存储在交互矩阵中的，这些信息包括但不限于用户个人的档案信息、物品的档案信息及用户文本评价、用户历史交互信息的文本数据等。这些数据的形态各异，同时也没有传统自然语言处理技术的统一处理方法，此时大语言模型提供了一个稳定且统一的处理接口。在本工作中，基于大语言模型的代理人^[33]用于基于用户基础信息档案分析并基于智能体的记忆绘制用户画像并更新。用户的动态交互记录也可以参与记忆更新。

少样本交互预测 虽然大语言模型的训练需要依靠大量的语料，但是在使用预训练大模型时，依靠已经存在在预训练模型中的知识信息，能够在新的问题上仅仅提供少量或者零样本就足以进行泛化学习。具体在推荐系统框架中对于用户已有的交互系统，需要判断用户的短期兴趣是否产生了强烈的变化就可以给予推荐系统少量的学习样本从而让预训练模型给出泛化结果。

增强兴趣特征生成 在推荐系统中，直接使用大语言模型进行推荐往往是效率比较低的方式。一方面预训练模型中并没有当前推荐系统中特定的用户及物品还有交互记录的语料用于训练。另一方面大语言模型的运算开销比普通模型更大，因此在即时的推荐系统中对于实时性要求较高的情形下，直接使用大语言模型无法得到准确的推荐结果同时时效性表现会更差。在此基础上，Li 等人在 GPT4Rec^[16]中提到一种利用大语言模型进行粗召回的方法，即利用大语言模型对用户交互历史进行语义分析，让大语言模型进行推理，得到其下次可能使用的检索词。此时再使用新的推理词使用检索系统检索，得到粗召回结果，可以大幅度减少预训练模型运算开销，也可以在粗召回结果上继续精细排序提高排序质量。

3.2 方法

3.2.1 物品推荐概述

本节主要基于物品推荐进行智能体框架设计。城市生活中物品的推荐涵盖了日常生活的方方面面，包括衣食住行。在本节框架设计中尽可能的包含了真实物品所可能包含的全部字段。不仅包含了常见在线推荐系统中的交互特征及主体信息，更重要的包含了城市空间真实信息，例如地理位置、地图信息以及城市综合数据库。在完整框架的基础上，一些常见的线上推荐系统的智能体记忆更新框架可以视为物品推荐场景的子集，即去除真实地理信息的物品推荐系统。因此推荐的主体可以是常见的线上可交互物品，也可以是真实地理空间中涵盖城市信息的物品。

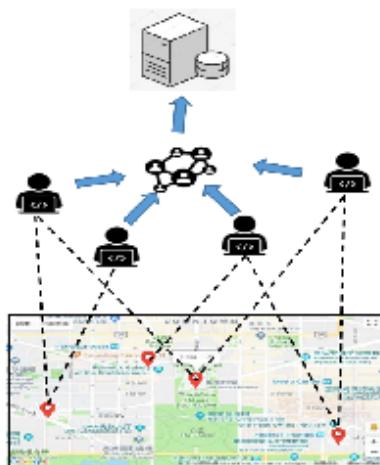


图 3.1 物品推荐示意图

3.2.2 智能体记忆模块存储及更新

3.2.2.1 动静态混合记忆模块

外部记忆模块是智能体的核心。对于单个用户而言需要用用户的个人档案以及交互历史进行初始化，作为静态记忆。而随着推荐过程的进行，会有实时内容更新形成动态记忆，同时需要与之配套的更新机制。因此一个全面的键值以及更新方式是记忆模块设计的核心。实际应用场景中可能会有缺失值，则相应方法则需要对所缺失的数据字段进行针对性处理，或直接对智能体记忆模块更新机制进

行通用性处理。与传统推荐系统中以交互矩阵以及用户对物品的评分不同，首先该记忆模块对单个用户存储了更多的用户基础信息以及过往交互信息，能够保留单个用户单次交互行为的更多特征，从而允许推荐系统在推荐过程里挖掘更多信息从而给出更加精准的推荐结果。此外由于该记忆更多的是以显式文本的形式进行存储的，因此在大语言模型诞生以前对显式文本直接进行语义判断是有困难的，而依靠大语言模型技术能够在语义层面对用户档案进行更加详细的理解推断，同时对于开发者而言，最终呈现的大语言模型的可解释性也将显著增强。

以下将从智能体记忆模块数据结构以及常用操作进行详细介绍：

a) 数据结构

智能体的记忆数据主要可以分为两大部分，即静态信息和动态信息。静态信息不会在推荐过程中发生变化，但会随着推荐系统更新进行维护。与之相对的动态信息则与用户当前交互时空状态相关。动态信息与静态信息都用于形成智能体对于当前用户的喜好偏好的描绘，且承担着不同的角色。静态信息中描绘的主要是用户长期交互以及基础偏好共同形成的偏向静态且长期稳定的用户喜好特征，在推荐系统需要做出长期兴趣推断及预测推断时以及被动式推荐的过程里能够提供充足的用户特征信息。而真实的推荐系统中用户常常会做出与长期推荐系统并不统一的交互请求，在实际系统中这类信息常常包含特殊时间，比如特殊节日、特殊地点，如用户在非常规地点的差率以及偏好短期剧烈变化，例如用突然产生探索性推荐的请求等。在这些情况下需要短期推荐的介入，此时动态信息可以对推荐结果进行必要的修正。

静态信息 用户的静态信息在数据库中显式存储。主要包含两部分：用户档案信息（Identification Info）以及交互信息（Interaction Info）。用户档案信息包含姓名、年龄、性别等用户基础信息，而同时也包含了用户工作工作、工资、存款等与社会生活直接相关的信息，这些信息可以辅助进行用户画像的描绘。交互信息则与传统的交互矩阵不同，存储了用户交互偏好字段外还显式地存储了用户交互记录，包括交互的时空信息以及数据库中存储的物品评价信息等。具体的数据结构以 Json^[34] 格式记录在算法 3.1 中。

动态信息 动态信息产生与用户交互时，与普通推荐系统最大的区别是，依靠于大语言模型的语义理解能力，能够接受用户主动搜索的 Query 字段，此外还包含发送推荐请求时的时空信息以及附加搜索条件。具体的数据结构以 Json^[34] 格式记录在算法 3.2 中。

算法 3.1 用户静态信息表示

```
1: { "Identification Info": {
2:     "Name": 用户的姓名,
3:     "UserID": 用户的唯一标识符,
4:     "Age": 用户的年龄,
5:     "Gender": 用户的性别,
6:     "Salary": 用户的工资,
7:     "Federation": 用户所属的组织或团体,
8:     "Work age": 用户的工作年限,
9:     "Deposit": 用户的存款,
10:    "Company": 用户的公司,
11:    "Home Address": 用户的家庭住址
12: },
13: "Interaction Info": {
14:     "Description of Favor": 用户的偏好描述,
15:     "Interaction History": {
16:         "Inter1": {
17:             "Time": 交互发生的时间,
18:             "Query": 用户的查询,
19:             "Filter": 应用的过滤条件,
20:             "Final decision": {
21:                 "Item name": 物品的名称,
22:                 "Item category": 物品的类别,
23:                 "Item comment grade": 物品的评论评分,
24:                 "Item price average": 物品的平均价格,
25:                 "Item comment count": 物品的评论数量,
26:                 "Item distance": 兴趣点的距离
27:             }
28:         }
29:     }
30: }
```

算法 3.2 用户动态信息表示

```
1: { "Query": 发送给推荐系统的用户查询信息,
2:     "Filter": {
3:         "filter type": 过滤类型 (距离, 评论数, 价格, 评论等级, 分类),
4:         "rank type": 排序类型 (1: 上升, 0: 下降, none: 无)
5:     },
6:     "Location": 查询时用户的位置,
7:     "Time": 查询发送时间
8: }
```

b) 数据操作

基于上一节所提到的智能体记忆模块的数据结构，本节将提出基于所设计的数据机构所设计的一系列数据操作以及智能体记忆模块更新机制。操作由一系列智能体代理完成，每一个代理都完成特定操作，最终组合并彼此配合完成推荐。主要的记忆模块智能体代理组及各自操作包含如下：

静态记忆初始化 (Passive intent recognition,Pir Agent) 在初始化智能体记忆模块时，主要读取记忆模块中的静态数据。Pir Agent 主要读取用户基础信息以及过往交互记录，形成对当前用户偏好的被动式意图识别，并更新在”Interaction Info.Description Favour” 字段中。

动态记忆更新 (Active intent recognition,Air Agent) 在收到用户主动的返回推荐结果请求时，Air Agent 首先在用户的检索词中获取用户主要意图，并识别用户请求中的时空信息分解并推断其中的混合意图，并更新到智能体记忆中。

混合记忆交叉更新 (Supplement and widen, Sew Agent) 在一轮推荐中，Pir Agent 和 Air Agent 除了各自读取更新外还涉及到交叉更新的过程 (PAir)，在这个过程里，新的动态交互会写入静态信息中的交互记录以及用户静态交互偏好中，同时静态信息中提取的语义特征也可以作为动态信息缺失时的默认值。

推荐物品召回 (Gathering 、 Linkage out-of Versatile Explorer,Glove Agent) 智能体记忆完成更新后，可以依靠记忆中的数据获取智能体暂时的人格信息，并仿照人类交互从推荐系统中获取返回结果，具体召回过程会在后续章节详细展开。

记忆反思 (Reasoning in General and Specific,Rings Agent) 为进一步相比传统推荐系统提升推荐结果的可解释性以及提高推荐质量，对于每一个推荐结果进行理解，并推测推荐结果是否合乎用户喜好，并依据反思结果提高或降低排序，从而提升排序质量。

探索性推荐 (PLAN-B Agent) 在一些特定情境下，用户可能不会喜欢一尘不变的推荐，例如特殊时间，比如特殊节日、特殊地点，如用户在非常规地点等，有时用户可能有特殊交互请求变化或者特殊情况，或者只是单纯的长期兴趣波动较小，此时就可以采用探索性推荐。探索性推荐还可以用于应对推荐系统常见的长尾性问题，在内容生产平台中为生产优质内容但没有得到足够曝光度的物品提供特殊推荐。

反馈接收 (Shake Agent) 传统推荐系统在返回推荐结果后很难大幅度地为每一个用户进一步调整推荐结果，在本工作框架之中可以接收用户反馈调整的需求并进行理解，从而对推荐结果进行优化修正。

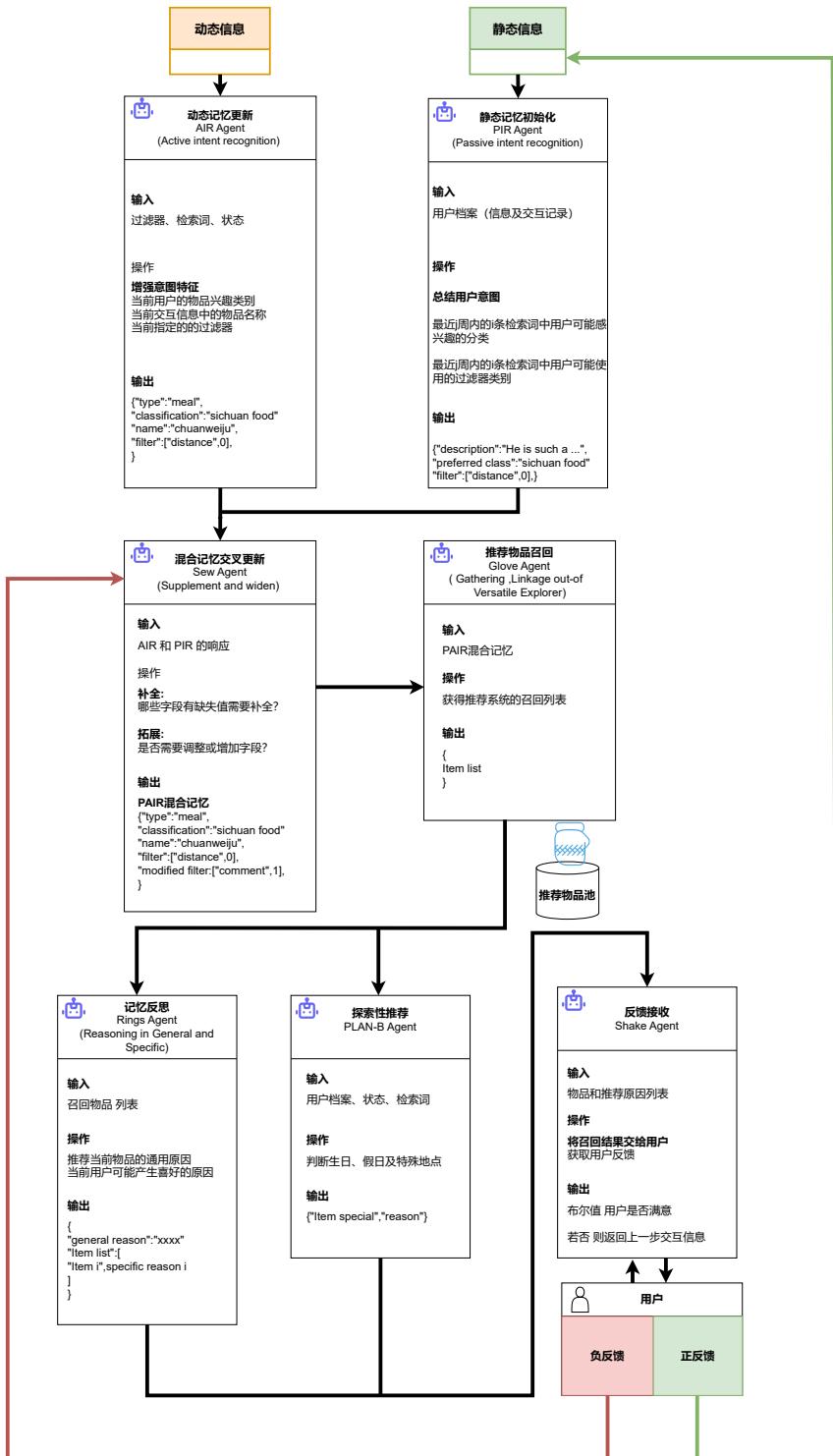


图 3.2 基于大语言模型智能体的完整的 POI 推荐系统推荐流程示意图

3.2.2.2 基于大语言模型智能体的 POI 推荐流程

本节在介绍了各个推荐系统组成智能体的各自功能后，将介绍一个大致的推荐框架。后续的优化工作将主要基于其中的部分组件进行优化。正如各个智能体名字简称所揭示的一样，推荐系统就是一个缝制手套、带上戒指并与用户握手的过程。完整的推荐系统主要包含：智能体记忆初始化阶段，主要包含 PAIR Agent 以及 Sew Agent、推荐召回环节，主要包含 Glove Agent、Rings Agent 及 Plan-B Agent、后反馈环节，主要包含 Shake Agent。

记忆初始化 在推荐的初始化阶段，Pir Agent 以及 Air Agent 分别进行初始化，并分别将用户的动态推荐请求及静态档案进行初始化，随后将档案输入 Sew Agent 进行交叉更新。更新后维护静态记忆，并将强化后的动态请求输入后续推荐流程。

推荐召回 在对用户信息进行充分的语义理解后，就可以从推荐系统中获取召回结果，召回也分为常规召回及探索性召回，分别对应用户普通的长短期兴趣探索以及特殊情形下的探索性推荐。

后反馈 获取推荐结果后会与用户进行一次交互，用户若接受推荐结果，则结束推荐流程，并对记忆体进行更新，若不接受，则会着重调用反思模块调整推荐结果，直至返回用户满意的推荐结果。

3.2.3 长短期兴趣变化识别及自适应推荐

3.2.3.1 基于线性神经网络的长短期推荐系统

基于前一章节的描述，在关注大量用户过往记录时往往会采用基于模型的用户交互推荐系统算法，这一算法最重要的前提是用户在长期交互过程中有稳定且容易提取的特征。但是事实情况是，这种传统的基于矩阵的算法并未考虑交互记录的时效性。具体而言用户可能在近期产生了重大的兴趣偏移，但是由于时间上长期以前的交互记录与近期交互记录的权重相同，因此长期交互记录也产生了相同的影响，导致对近期的关注度没有显著。因此需要基于时间进行权重调整，调整注意力范围，在一定时间范围内识别成高注意力范围，对这些交互记录采取更多的权重。而对更早时间的交互记录采用低注意力以作为区分。本节将着重介绍线性权重层的引入和结构以及对算法的影响。

考虑基于奇异值分解的基于用户交互推荐系统，为了关注近期的交互行为，为了关注近期的交互行为，引入多个权重调制层。每个权重矩阵 W^n 的每个元素 W_{ij}^n 由以下公式定义：

$$W_{ij}^n = \left(\frac{j_n}{i} \right)^{-1} \quad (3.1)$$

更一般地，每个权重调制函数可以抽象为一个激活函数形式：

$$W_{ij}^n = \frac{1}{1 + e^{-\alpha_n(j_n/i - \beta_n)}} \quad (3.2)$$

其中， α_n 和 β_n 是第 n 个调制函数的可调参数，控制函数的形状和中心位置。这里， j_n 表示第 n 个高注意力时间，即用户在第 n 个时间尺度上的近期交互时间， i 表示用户有交互记录的总时间，计算梯度并更新参数， η 是学习率：

$$U'_k = U_k - \eta \left(\frac{\partial \mathcal{L}}{\partial U_k} \right) = U_k - \eta \left(-2 \sum_{(u,i) \in \mathcal{K}} (R_{ui} - \sum_{n=1}^N W_{ui}^n \cdot (U_k \Sigma_k V_k^T)_{ui}) \sum_{n=1}^N W_{ui}^n V_k \Sigma_k + 2\lambda U_k \right) \quad (3.3)$$

$$V'_k = V_k - \eta \left(\frac{\partial \mathcal{L}}{\partial V_k} \right) = V_k - \eta \left(-2 \sum_{(u,i) \in \mathcal{K}} (R_{ui} - \sum_{n=1}^N W_{ui}^n \cdot (U_k \Sigma_k V_k^T)_{ui}) \sum_{n=1}^N W_{ui}^n \Sigma_k^T U_k^T + 2\lambda V_k \right) \quad (3.4)$$

为简洁及利用 cuda 的并行计算优势, 上述运算可以转换为张量运算, 具体而言: 预测评分矩阵及梯度更新变为张量形式, \circ 表示逐元素乘法:

$$\hat{\mathcal{R}} = \sum_{n=1}^N \mathcal{W}_n \circ (\mathcal{U} \Sigma \mathcal{V}^T) \quad (3.5)$$

梯度更新规则为: $\mathcal{U} \leftarrow \mathcal{U} - \eta \frac{\partial \mathcal{L}}{\partial \mathcal{U}}$ 和 $\mathcal{V} \leftarrow \mathcal{V} - \eta \frac{\partial \mathcal{L}}{\partial \mathcal{V}}$ 。具体形式为:

$$\frac{\partial \mathcal{L}}{\partial \mathcal{U}} = -2 \left(\mathcal{R} - \sum_{n=1}^N \mathcal{W}_n \circ (\mathcal{U} \Sigma \mathcal{V}^T) \right) \sum_{n=1}^N \mathcal{W}_n \circ (\mathcal{V} \Sigma) + 2\lambda \mathcal{U} \quad (3.6)$$

$$\frac{\partial \mathcal{L}}{\partial \mathcal{V}} = -2 \left(\mathcal{R} - \sum_{n=1}^N \mathcal{W}_n \circ (\mathcal{U} \Sigma \mathcal{V}^T) \right)^T \sum_{n=1}^N \mathcal{W}_n \circ (\Sigma \mathcal{U}^T) + 2\lambda \mathcal{V} \quad (3.7)$$

基于并行运算优化的加入权重层的推荐系统算法流程基于如下伪代码描述:

算法 3.3 基于权重层控制注意力范围的推荐系统 (GPU)

Require: 用户集合 U , 项目集合 I , 评分矩阵 R , 奇异值个数 k , 学习率 η , 正则化参数 λ , 迭代次数 num_epochs , 注意力范围集合 $\{K_n\}$

Ensure: 不同注意力范围 K_n 的模型集合 $\{\hat{\mathcal{R}}^n\}$

```

1: for 每个  $K_n$  在  $\{K_n\}$  中 do
2:   将  $\mathcal{U}^n \in \mathbb{R}^{m \times k}$  和  $\mathcal{V}^n \in \mathbb{R}^{n \times k}$  随机初始化并传输到 GPU
3:   for epoch = 1 到 num_epochs do
4:     计算预测评分矩阵  $\hat{\mathcal{R}}^n = \sum_{i=1}^N \mathcal{W}_i^n \circ (\mathcal{U}^n \Sigma \mathcal{V}^{nT})$ 
5:     计算梯度  $\frac{\partial \mathcal{L}^n}{\partial \mathcal{U}^n}$  和  $\frac{\partial \mathcal{L}^n}{\partial \mathcal{V}^n}$ 
6:     更新  $\mathcal{U}^n \leftarrow \mathcal{U}^n - \eta \frac{\partial \mathcal{L}^n}{\partial \mathcal{U}^n}$ 
7:     更新  $\mathcal{V}^n \leftarrow \mathcal{V}^n - \eta \frac{\partial \mathcal{L}^n}{\partial \mathcal{V}^n}$ 
8:   end for
9:   将模型  $\hat{\mathcal{R}}^n$  从 GPU 保存到 CPU
10: end for
11: return 模型集合  $\{\hat{\mathcal{R}}^n\}$ 

```

训练得到子模型组的过程可以用下图简单表述

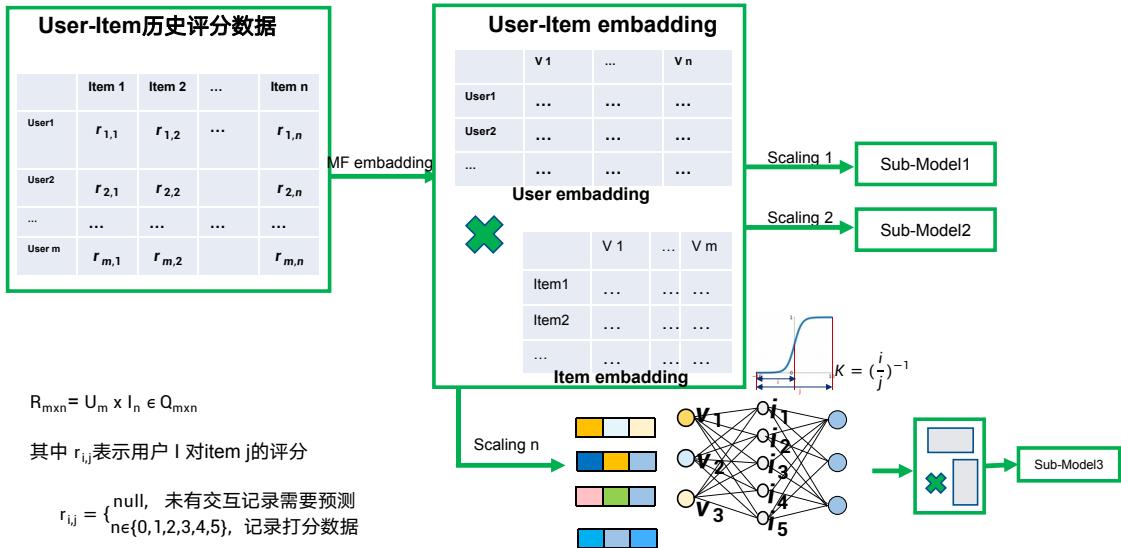


图 3.3 权重长短期推荐网络示意图

3.2.3.2 基于大语言模型的用户偏好变化识别

在得到一系列不同注意力范围的子模型后，调度何种模型的关键在于对于用户偏好变化识别。在这里主要提供两种思路，分别是定性识别兴趣变化以及定量识别兴趣变化。

对于定性挖掘用户偏好变化，利用大语言模型的语义分析能力并不能准确得到高注意力范围的准确比例，只能利用少量样本提示大语言模型并大致定性分析用户的长短期变化是否发生了较大的变化。在这种情形下，每个用户的短期模型参数是统一的，大语言模型仅仅从一个长期模型以及一个短期模型中选择更适合当前推荐情形的模型进行调度。具体而言，若用户长期兴趣与短期兴趣之间差异过大，则优先调用短期模型。若长期以来偏好保持不变，则以一定概率调度探索性推荐模型^[35]，大概率维持长期推荐模型。

对于定量挖掘用户偏好，则需要定量利用大语言模型推测用户大致发生兴趣偏移的时间节点计算高注意力比例，并调用刚好超过对应参数的子模型用于进行推荐。定量挖掘偏好依赖于大语言模型准确的兴趣变化识别能力以及少样本学习能力，实际性能在本工作实验环境中并不如定性挖掘。

3.2.3.3 基于大语言模型的自适应长短期推荐系统

基于算法 3.3 以及上节中定量判断用户兴趣偏好变化的方法，本工作提出一种基于大语言模型的自适应长短期推荐系统的算法实现如下：

算法 3.4 用户兴趣变化检测与模型调度

Require: 用户档案集合 $\{P_u\}$, 模型集合 $\{\hat{\mathcal{R}}^n\}$
Ensure: 最优模型调度 $\{\hat{\mathcal{R}}_u^*\}$

```
1: for 每个用户档案  $P_u$  在  $\{P_u\}$  中 do
2:   Sew Agent 分析  $u$  的兴趣变化情况并计算用户兴趣变化量  $\Delta I_u$ 
3:   if  $\Delta I_u$  大于预设阈值 then
4:     选择最适合的模型  $\hat{\mathcal{R}}_u^* = \arg \max_n$  适应度( $K_n, \Delta I_u$ )
5:   else
6:     保持当前模型  $\hat{\mathcal{R}}_u^*$ 
7:   end if
8:   保存最优模型  $\hat{\mathcal{R}}_u^*$ 
9: end for
10: return 最优模型调度  $\{\hat{\mathcal{R}}_u^*\}$ 
```

3.2.4 基于语义理解的内容推荐

3.2.4.1 基于向量化及基于相似度召回方法

在第二章章节中，已经介绍了基于内容的推荐方法。本工作选择了三种常见的自然语言向量化模型或代码库：TF-IDF、BERT 和 SpaCy。首先，通过后续基于大语言模型的框架生成一系列检索词，并对这些检索词进行向量化表示。这些向量表示将用于计算用户与物品之间的相似度，生成推荐列表。结合大模型的优点在于，它能提供更强大的语义理解和生成能力，捕捉用户偏好的细微变化，从而实现更高层次的语义分析和用户偏好建模，提供更加精准和个性化的推荐服务。以下是本工作中对于自然语言检索词进行向量化的大致框架：

算法 3.5 向量化与相似度召回

Require: 用户集合 $\{u\}$, 项目集合 $\{i\}$, 推荐数量 N , 选择的向量化模型 (TF-IDF, BERT, SpaCy)

Ensure: 推荐项目列表

```
1: for 每个项目  $i$  do
2:   if 选择 TF-IDF then
3:     计算  $\text{TF-IDF}(i, u, \mathcal{U})$ 
4:      $\boldsymbol{v}_i = [\text{TF-IDF}(i, u_1, \mathcal{U}), \dots, \text{TF-IDF}(i, u_m, \mathcal{U})]$ 
5:   else if 选择 BERT then
6:     预处理档案信息  $\text{Doc}_i$ 
7:      $\boldsymbol{v}_i = \text{BERT}(\text{Doc}_i)$ 
8:   else if 选择 SpaCy then
9:     预处理档案信息
10:    标注词性, 进行依存句法分析和命名实体识别
11:     $\boldsymbol{v}_i = \text{词嵌入}(\text{Doc}_i)$ 
12:   end if
13: end for
14: for 每个用户  $u$  do
15:   初始化用户偏好向量  $\boldsymbol{p}_u$ 
16:    $\boldsymbol{p}_u = \frac{1}{|\mathcal{I}_u|} \sum_{i \in \mathcal{I}_u} \boldsymbol{v}_i$ 
17:   for 每个候选项目  $i$  do
18:     计算相似度  $\text{sim}(u, i) = \boldsymbol{p}_u \cdot \boldsymbol{v}_i$ 
19:   end for
20:   按相似度从高到低排序候选项目
21:   返回前  $N$  个项目作为推荐结果
22: end for
23: return 推荐项目列表
```

3.2.4.2 基于正则匹配及阈值过滤对召回的修正

在实际向量化并检索的过程里, 本工作关注到了一些特例并予以修正^[36-39]。一般而言对于相似度召回有三种方式, 即基于字符串正则匹配直接召回、基于特征向量相似度召回以及基于交互记录协同过滤召回。基于交互记录协同过滤的召回方式以及其基于大语言模型的优化方法已经在上节介绍, 本节主要介绍基于正则匹配及阈值过滤对内容向量相似度召回的修正。例如在现有的检索系统中, 使用 “Star War”^[36] 进行检索, 会将 “Star” 及 “War” 分词化后平均, 再在数据库中检索与该向量最接近的物品进行召回。而实际上, 星球大战的后续作品, 例如《Star Wars:The Force Awakens》由于标题中含有较多与原始检索词偏离较远的副标题分词, 导致在基于向量相似度的检索中无法以较高优先级进行召回。恰巧相反, 与星球大战题材和标题较为接近, 但实际上内容没有直接相关的银河护卫队



图 3.4 Star Wars: 图 3.5 Guardians of the Galaxy
Episode I - The Phantom Menace

图 3.6 Star Wars: 图 3.7 The Mandalorian
Episode VII - The Force Awakens

由于“Guardian”与“War”、“Star”与“Galaxy”更为接近，却能够以很高的优先级进行召回。此外，《The Mandalorian》作为星球大战的外传，备受星球大战粉丝追捧，但是由于标题分词与原始分词几乎没有任何关联也不包含真实语义，也很难通过向量化相似度进行召回。关于《The Mandalorian》的例子，将通过基于交互行为的推荐系统进行修正并在后续融合框架中提到具体解决办法。本节将基于直接正则匹配以及阈值过滤方法修正基于向量相似度进行召回的语义偏离的召回偏差问题。

下面给出基于正则匹配及阈值过滤对召回的修正的建模：假设我们有一系列检索词 Q_1, Q_2, \dots, Q_n ，每个检索词在数据库中查找相似物品，并获得其召回结果 $R_{Q_i} = \{R_{Q_i1}, R_{Q_i2}, \dots, R_{Q_im}\}$ 。参数定义如下：检索词 Q_i 是第 i 个检索词，向量表示为 \mathbf{v}_{Q_i} ；数据库中的第 j 个物品为 $item_j$ ，向量表示为 \mathbf{v}_j ；检索词 Q_i 和物品 j 之间的相似度表示为 $sim(\mathbf{v}_{Q_i}, \mathbf{v}_j)$ ，相似度阈值为 θ 。按相似度排序后的检索词 Q_i 的召回结果为 $Sorted_{R_{Q_i}}$ ，正则匹配结果为 Match，最终的物品召回结果为 $Final_{R_{Q_i}}$ 。则算法可以用如下步骤概括：

1. 物品向量化：

向量化检索词和数据库中的物品，具体而言对每个检索词 Q_i 和数据库中的物品 $item_j$ 进行向量化表示：

$$\mathbf{v}_{Q_i} = \text{Vectorize}(Q_i) \quad (3.8)$$

$$\mathbf{v}_j = \text{Vectorize}(item_j) \quad \forall j \in \text{Database} \quad (3.9)$$

2. 相似度计算及阈值过滤:

对于每个检索词 Q_i , 计算其与数据库中物品 $item_j$ 的相似度, 并过滤掉相似度小于阈值 θ 的结果:

$$\text{sim}(\mathbf{v}_{Q_i}, \mathbf{v}_j) > \theta \quad (3.10)$$

3. 排序:

将相似度超过阈值的检索词 Q_i 的召回结果按相似度从高到低排序:

$$\text{Sorted}_{R_{Q_i}} = \{R_{Q_i1}, R_{Q_i2}, \dots\} \quad \text{where} \quad \text{sim}(\mathbf{v}_{Q_i}, \mathbf{v}_{R_{Q_i j}}) \text{ is decreasing} \quad (3.11)$$

4. 正则匹配修正:

对于召回结果不足的部分, 用正则匹配补齐, 并按相似度进行排序并插入:

$$\text{Match} = \{M_1, M_2, \dots\} \quad \text{where} \quad M_k \text{ matches the regular expression} \quad (3.12)$$

5. 融合插入:

基于上述阈值过滤以及正则匹配的结果进行插入重排, 得到最终的找回列表:

$$\text{Final}_{R_{Q_i}} = \begin{cases} \text{Sorted}_{R_{Q_i}} + \text{Match} & \text{if } \exists Q_i \text{ s.t. } \text{sim}(\mathbf{v}_{Q_i}, \mathbf{v}_j) > \theta \\ \text{Match} & \text{otherwise} \end{cases} \quad (3.13)$$

其中检索词 Q_i 是第 i 个检索词, 向量表示为 \mathbf{v}_{Q_i} ; 数据库中的第 j 个物品为 $item_j$, 向量表示为 \mathbf{v}_j ; 检索词 Q_i 和物品 j 之间的相似度表示为 $\text{sim}(\mathbf{v}_{Q_i}, \mathbf{v}_j)$, 相似度阈值为 θ ; 按相似度排序后的检索词 Q_i 的召回结果为 $\text{Sorted}_{R_{Q_i}}$, 正则匹配结果为 Match , 最终的物品召回结果为 $\text{Final}_{R_{Q_i}}$ 。

这个过程的伪代码描述如下:

算法 3.6 基于检索词的物品召回与排序算法

Require: 检索词集合 $\{Q_1, Q_2, \dots, Q_n\}$, 数据库中的物品集合, 阈值 θ

Ensure: 召回物品列表

```
1: 初始化结果集合 Final_R = []
2: for 每个检索词  $Q_i$  do
3:   将  $Q_i$  向量化为  $v_{Q_i}$ 
4:   初始化临时结果集合 Temp_R = []
5:   for 数据库中的每个物品  $j$  do
6:     将物品  $j$  向量化为  $v_j$ 
7:     计算相似度  $\text{sim}(v_{Q_i}, v_j)$ 
8:     if  $\text{sim}(v_{Q_i}, v_j) > \theta$  then
9:       将物品  $j$  加入 Temp_R
10:    end if
11:   end for
12:   if Temp_R ≠ [] then
13:     按照相似度从高到低排序 Temp_R
14:     将 Temp_R 合并入 Final_R
15:   end if
16: end for
17: if Final_R == [] then
18:   使用正则表达式匹配补齐物品
19:   按照相似度从高到低排序匹配结果
20:   将匹配结果插入 Final_R
21: end if
22: return Final_R
```

3.2.4.3 基于大语言模型的 Query-Search 推荐架构

预训练大语言模型可以通过分析用户的交互历史, 预测用户下一次可能进行检索的若干关键词。这些关键词可以接入检索与排序步骤, 从而优化推荐效果。直接使用大语言模型进行检索效率低下, 而通过生成检索词进行检索可以显著提升效率。以下是该过程的详细描述:

1. 用户交互历史初始化:

用户的交互历史序列可以表示为 $H = \{h_1, h_2, \dots, h_k\}$, 其中每个 h_i 表示用户交互过的物品序列。

2. 用户交互历史分析:

使用预训练大语言模型读取并分析用户的交互历史序列 H , 生成用户可能在未来会观看的电影名称中的关键词或短语。:

$$Q = \text{PTM}(H) \quad (3.14)$$

其中， $Q = \{q_1, q_2, \dots, q_n\}$ 表示预测的检索关键词集合。

3. 将得到的检索词用于算法 3.6：

使用推荐系统根据预测的关键词查找与之相似的物品，并返回推荐结果。具体地，利用推荐系统查找与每个预测关键词 q_i 相似的物品：

$$R_{q_i} = \text{FindSimilarItems}(q_i, \theta, \text{top_n}) \quad (3.15)$$

其中， R_{q_i} 表示与预测关键词 q_i 相似的物品集合， θ 为相似度阈值， top_n 为返回的推荐数量。

上述过程也可以用伪代码进行描述：

算法 3.7 基于预训练大语言模型的 Query-Search 推荐架构

Require: 用户交互历史序列 $H = \{h_1, h_2, \dots, h_k\}$ ，数据库中的物品集合，阈值 θ

Ensure: 召回物品列表

- 1: 初始化结果集合 $\text{Final_R} = []$
 - 2: 使用预训练大语言模型分析用户交互历史 H ，生成可能的检索词 $Q = \{q_1, q_2, \dots, q_n\}$
 - 3: **for** 每个预测的检索词 q 在 Q **do**
 - 4: 根据预测的检索词 q 调用基于正则匹配及阈值过滤的修正召回
 - 5: **end for**
 - 6: **return** Final_R
-

通过上述步骤，预训练大语言模型能够基于用户的交互历史，生成未来可能使用的检索关键词，并通过这些关键词进行检索过滤以及相似度匹配，从而实现利用大语言模型技术通过基于阈值的正则匹配修正实现基于内容相似度的推荐。

3.2.5 基于大语言模型的混合意图识别推荐系统框架

3.2.5.1 内容与交互混合推荐系统的对齐召回

本节将介绍一种基于 3.2.3 中基于用户行为的优化推荐系统模型对 3.2.4 中基于内容相似度的模型找回结果缺乏行为数据的推荐修正方法。

用户的交互历史序列可以表示为 $H = \{h_1, h_2, \dots, h_k\}$, 其中每个 h_i 表示用户交互过的物品序列。基于上节提到使用预训练大语言模型读取并分析用户的交互历史序列 H , 预测检索词序列 $Q = \{q_1, q_2, \dots, q_n\}$ 并得到初步召回结果 $R_{q_i} = \text{FindSimilarItems}(q_i, \theta, \text{top_n})$, 以下算法将介绍基于粗召回结果基于交互行为进行对齐修正的步骤:

基于行为推荐系统的修正召回 对每个初步召回的物品 $r \in R_{q_i}$, 使用基于权重层控制注意力范围的推荐系统方法 (算法 3.3) 生成的模型 $\hat{\mathcal{R}}_n$ 进行进一步召回, 并将其结果进行融合。假设初步召回的物品集合为 $R = \bigcup_{i=1}^n R_{q_i}$, 对于每个 $r \in R$, 得到进一步召回的物品集合 V_r :

$$V_r = \hat{\mathcal{R}}_n(r)$$

这里的 $\hat{\mathcal{R}}_n(r)$ 表示使用算法 3.3 生成的模型对物品 r 进行进一步召回操作。

1. 召回结果对齐 设初步召回的物品集合为 $R = \bigcup_{i=1}^n R_{q_i}$, 对于每个 $r \in R$, 得到进一步召回的物品集合 V_r 。将所有 V_r 融合成最终的召回物品列表, 并对其进行重排:

2. 融合召回结果: 将所有进一步召回的物品集合 V_r 融合成集合 F :

$$F = \bigcup_{r \in R} V_r$$

3. 计算综合评分: 对于 F 中的每个物品 v , 计算其综合评分。假设评分函数为 $S(v)$:

$$S(v) = \sum_{r \in R} w_r \cdot \text{score}(v, r)$$

其中: w_r 是物品 r 的权重, $\text{score}(v, r)$ 是物品 v 与 r 的相似度评分。

4. 排序: 按照综合评分 $S(v)$ 对 F 进行排序, 得到最终的推荐列表。

基于上述流程就可以对基于内容的推荐系统召回结果做到基于行为数据的召回对齐即修正, 最终的推荐结果会在一定程度上兼顾物品内容以及历史交互数据。完整的算法流程可以用如下伪代码进行描述:

算法 3.8 基于预训练大语言模型的 Query-Search 推荐架构

Require: 用户交互历史序列 $H = \{h_1, h_2, \dots, h_k\}$, 数据库中的物品集合, 阈值 θ

Ensure: 召回物品列表

```
1: 初始化结果集合 Final_R = []
2: 使用预训练大语言模型分析用户交互历史  $H$ , 生成可能的检索词  $Q = \{q_1, q_2, \dots, q_n\}$ 
3: for 每个预测检索词  $q$  在  $Q$  中 do
4:    $R_q = \text{FindSimilarItems}(q, \theta, \text{top\_n})$ 
5:   初始化中间结果集合 Intermediate_R = []
6:   for 每个物品  $r$  在  $R_q$  中 do
7:      $V_r = \hat{\mathcal{R}}_n(r)$ 
8:     将  $V_r$  中的物品加入 Intermediate_R
9:   end for
10:  将 Intermediate_R 融合至 Final_R
11: end for
12: 初始化最终结果集合 Final_F = []
13: for 每个物品  $r$  在 Final_R 中 do
14:   for 物品  $v$  在  $V_r$  中 do
15:     计算综合评分  $S(v) = \sum_{r \in R} w_r \cdot \text{score}(v, r)$ 
16:     将  $v$  插入 Final_F
17:   end for
18: end for
19: 对 Final_F 按照综合评分  $S(v)$  进行排序
20: return 排序后的 Final_F
```

3.2.5.2 大语言模型驱动的自适应混合推荐系统

基于先前系统组件的设计, 本节将详细介绍最终的设计组合。

首先基于 3.2.3 中基于自适应矩阵分解网络, 可以得到一系列注意力范围不同的子模型, 利用这些模型可以得到物品在不同时间尺度量化下的向量化模型。随后大语言模型基于语义理解或基于少量样本学习大致推断出用户的短期兴趣是否相对于长期兴趣产生了剧烈的变化, 或者更进一步在显著情形下推断产生兴趣变化的高注意力范围比例, 从而选用更加合适的子模型进行基于行为的推荐结果召回。

其次基于 3.2.4 中基于正则匹配及阈值过滤的内容推荐模型会首先基于用户交互历史的文本信息预测用户接下来可能使用的检索词, 随后使用预测结果在检索系统中基于相似度进行粗召回。

在得到行为修正模型和基于内容的召回结果后, 使用上节所提到的融合框架用行为模型对基于内容的召回结果进行修正从而使得最终结果能够同时兼顾召回物品的内容特征以及用户交互特征, 对最后的召回组之间进行排序插入后返回

最终的推荐召回结果并进行性能评估。

完整的大语言模型驱动的自适应混合推荐系统框架如下如所展示：

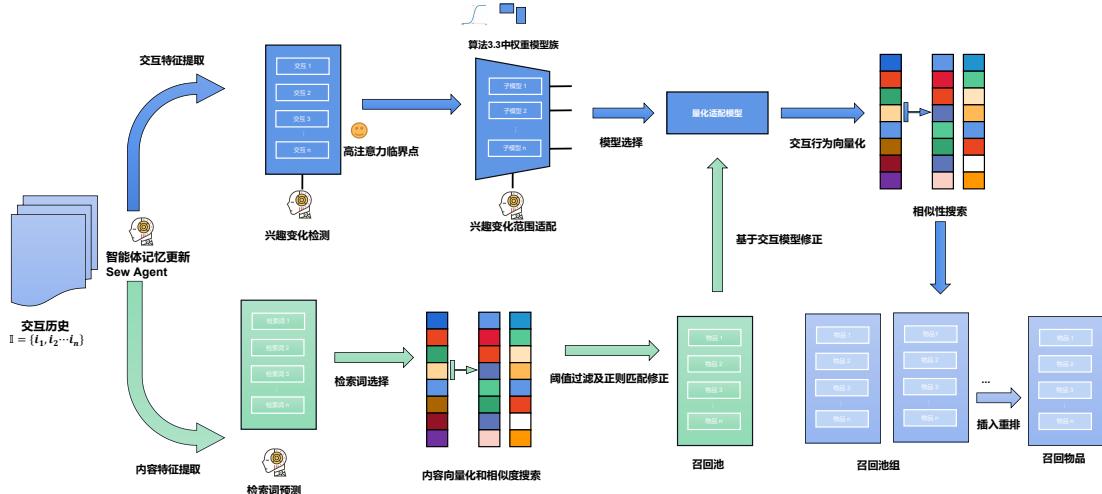


图 3.8 大语言模型驱动的自适应混合推荐系统框架

3.3 小结

本节提出了基于大语言模型的基于交互行为的长短期自适应推荐网络、基于正则匹配阈值修正的基于内容推荐网络，以及使用 Query-Search 及行为内容对齐修正的推荐系统框架并给出了各部分的数学建模、算法设计以及最终整体架构的设计框架。本节作为本工作的核心也是后续实验的直接理论基础。

第 4 章 实验

4.1 实验设置

4.1.1 数据集

为了验证推荐系统的推荐效果，本实验需要一个同时包含有用户信息物品文本信息以及用户与物品交互信息的真实数据集。基于此本工作选择了 MovieLens 数据集^[4]。该数据集由 GroupLens 组织^[40]在电影推荐平台上收集的用户对电影的真实评分数据。自发布至今涵盖了接近 20 年的时间跨度，同时发布了 100K、1M、10M 及 20M^[41-44] 等不同数据规模的数据集。出于运算成本以及特征提取难度的考量，本实验主要采用的数据集是 MovieLens-1M 数据集。本工作当前使用数据集为离线数据集，同时电影数据集所包含的字段是上一章节所提到的完整物品数据集的子集，在当前阶段数据已经足以验证框架的合理性。本工作后续迭代版本将基于真实生产环境中的在线实时数据以及完整的数据结构进行在线版本的开发测试。

MovieLens-1M 数据集是从 MovieLens 网站收集而来，包含了 6,040 名用户对 3,883 部电影的 1,000,209 条评分记录。每位用户至少为 20 部电影进行了评分。该数据集涵盖了多种类型的电影，并且用户的评分分布在时间上较为均匀。本研究利用 MovieLens-1M 数据集来评估推荐系统的性能。

在本工作中，将数据集分为训练集、验证集和测试集，比例为 8:1:1，以用于基于模型的推荐系统训练。数据集中还包括用户的基本信息（如年龄、性别、职业）和电影的基本信息（如标题、类型）。

此外，为了处理评分数据，本工作对评分的时间戳进行了预处理，转换为年、月、日等格式，从而能够在后续时间尺度上对模型进行处理时能够更加方便的处理数据。同时加入对用户信息转换为完整句子的接口方便后续进行语义分析。

关于数据集的一些其他统计学特性在如下表格中体现：

4.1.2 评价指标

在本节中将介绍本工作中所使用的推荐系统的评价指标，具体而言，对于最后的推荐系统需要量化评价准确率以及质量，从而使用 Recall@k 和 NDCG@k^[45-46]作为评价指标，分别评价排序召回率及排序质量。本节将介绍本工作中使用的这两个指标以及具体计算方法。

表 4.1 MovieLens-1M 数据集的统计特性

统计量	数值
用户数	6,040
电影数	3,883
评分数	1,000,209
用户平均评分数	165.5
电影平均评分数	257.6
平均评分	3.58

先前章节对于推荐问题进行了一些抽象建模，与先前符号相同，在推荐系统中给定用户集合 $\mathcal{U} = \{u_1, u_2, \dots, u_m\}$ 和一个项目集合 $\mathcal{I} = \{i_1, i_2, \dots, i_n\}$ ，交互状态空间可以表示为一个矩阵 $R \in \mathbb{R}^{m \times n}$ ，其中 R_{ui} 表示用户 u 对项目 i 的评分。如果用户 u 没有对项目 i 评分，则 R_{ui} 为缺失值。推荐系统目标为根据用户在前一部分项目上的评分数据 $\{I_1, I_2, \dots, I_{n-k}\}$ ，预测用户在后一部分项目上的评分 $\{I_{n-k+1}, \dots, I_n\}$ 。即，利用前 $n - k$ 个项目的评分记录来预测剩余项目的评分，由推荐系统给出预测值 $\{I'_{n-k+1}, \dots, I'_n\}$ 与真实值 $\{I_{n-k+1}, \dots, I_n\}$ 进行比较。

Recall@k 召回率用于衡量推荐列表与真实交互记录的重合度，仅衡量推荐的准确率而不反映推荐质量。度量范围为 0 到 1，数值越高则重合度越高，参数 k 衡量度量范围。

$$\text{Recall}@k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\{I_{n-k+1}, I_{n-k+2}, \dots, I_n\} \cap \{I'_1, I'_2, \dots, I'_k\}|}{|\{I_{n-k+1}, I_{n-k+2}, \dots, I_n\}|} \quad (4.1)$$

NDCG (Normalized Discounted Cumulative Gain) 归一化折损累计增益用于计算召回结果的推荐质量，若用户高评价物品排序更考前，则可以得到更高的分数。度量范围为 0 到 1，数值越高则推荐质量越高，参数 k 衡量度量范围。

1. DCG@k (Discounted Cumulative Gain) :

$$\text{DCG}@k = \sum_{i=1}^k \frac{2^{R_{u,I'_i}} - 1}{\log_2(i + 1)} \quad (4.2)$$

2. IDCG@k (Ideal DCG) :

$$\text{IDCG}@k = \sum_{i=1}^k \frac{2^{R_{u,I_i^*}} - 1}{\log_2(i + 1)} \quad (4.3)$$

3. NDCG@k:

$$\text{NDCG}@k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{\text{DCG}@k}{\text{IDCG}@k} \quad (4.4)$$

其中, R_{u,I'_i} 是用户 u 对推荐项目 I'_i 的评分, I'_i 表示推荐系统生成的第 i 个推荐项目, I_i^* 表示按真实评分排序的第 i 个项目。

4.2 实验内容

4.2.1 基于交互的长短期兴趣变化识别及自适应推荐系统实现

基于 3.2.3 中所提及的基于线性权重层的长短期自适应推荐系统网络以及算法 3.3 中的训练方式, 对 K 取 1, 2, 3, 4 时的模型分别进行训练, 其中 K=1 的就是原始 MF 算法的模型作为基线, 经过训练后得到 MF@Kfold_agent_free 模型。随后对训练好的模型接入大语言模型记忆体进行模型调用的推断, 得到最终的 MF@Kfold_agent 模型, 并在 MovieLens-1M 的最后 600 名用户组成的验证集上进行验证, 得到的结果如下图所示:

Experiment	Recall@1	Recall@5	Recall@20	Recall@100
MF@1fold(baseline)	0.00522	0.02642	0.09155	0.25371
MF@2fold_agent_free	0.00466	0.02621	0.08443	0.26017
MF@2fold_agent	0.00631 (+20.88%)	0.03345 (+26.63%)	0.10639 (+16.19%)	0.29908 (+17.88%)
MF@3fold_agent_free	0.00510	0.02617	0.09087	0.25734
MF@3fold_agent	0.00489 (-6.31%)	0.02878 (+8.93%)	0.09788 (+6.92%)	0.27161 (+7.05%)
MF@4fold_agent_free	0.00477	0.02596	0.09040	0.26133
MF@4fold_agent	0.00518 (-0.77%)	0.02766 (+4.69%)	0.09702 (+5.98%)	0.27142 (+6.98%)

表 4.2 K-fold MF 实验结果

从实验结果中可以推断出的结论:

引入量化注意力范围对召回率的改善 通过实验结果可以看出, 在 K=2, 3, 4 时, 相比与基线模型, 召回率都有了显著的提升。由于 Recall@1 的召回条件较为苛刻, 存在性能下降以外, 其他召回规模都得到了显著的提升。提升最大出现在 MF@2fold_agent 模型, 在原始基线模型的基础上得到了 26.63% 的性能提升。

改变 K 值对长期预测的影响 通过横向对比不同基础模型的结果可以看到高注意力范围大致是总交互时长一半时, 对于性能提升是最为显著的。但是随着 K 值增

大，高注意力范围逐渐缩短，因而模型一方面在模型中提取的特征减少，另一方面注意力时长简短，导致长期预测性能逐渐降低。

大语言模型对于语义理解的增强 在各组实验中，相比与没有大语言模型接入的量化模型而言，由大语言模型进行语义推理以及模型调度的模型都有性能上的提升，因此可以证明大语言模型出色的用户偏好理解能力以及模型调度能力。

其中性能最好的 MF@2fold_agent 模型在最常用的召回 5 个物品的情景之下 的累计召回率变化如下图所示：

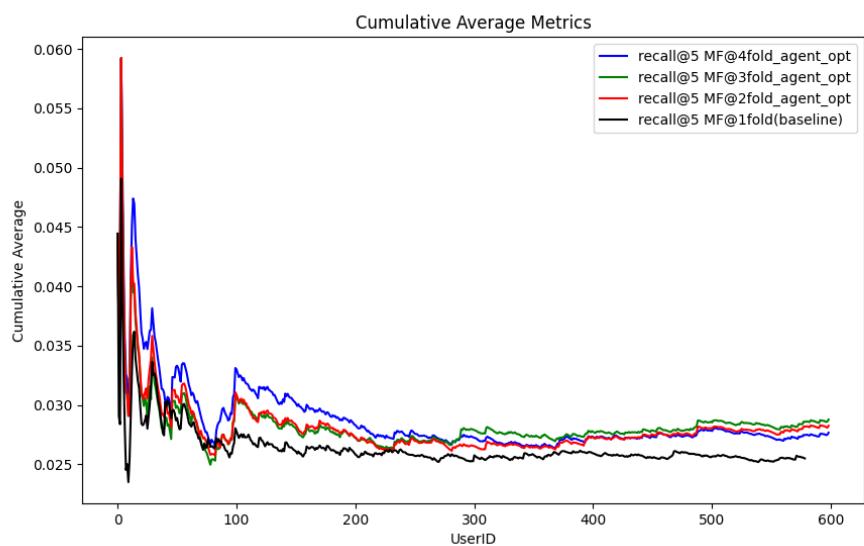


图 4.1 MF@2fold_agent Recall@5 实验结果

4.2.2 基于语义理解的内容推荐系统实现

基于 3.2.4 中的系统设计以及算法 3.5 的设计，本节实现了基于向量化进行检索的检索系统。接口方面，测试了三种向量化模型或软件库：Bert、TF-IDF 及 Spacy 库，并基于大语言模型对用户未来检索词进行预测，最后使用检索词基于向量化接口进行物品召回。由于检索词较多，需要多组召回，因此最小的 @k 使用了 10 作为参数。最终得到的实验结果如下图所示：

基于实验结果，选取性能更好的 TF-IDF 以及 Spacy 库进行后续实验。由于 Bert 性能较差且在后续实验中由于显存占用过大与其他模型无法兼容，因此后续实验基于算力以及性能的考虑不会使用 Bert，而是用其余的向量化方法验证框架的性能。

	NDCG@k				Recall@k			
	k=10	k=20	k=50	k=100	k=10	k=20	k=50	k=100
Bert	0.1695	0.2215	0.2789	0.3305	0.0034	0.0073	0.0171	0.0341
TF-IDF	0.2151	0.2489	0.3007	0.3541	0.0046	0.0080	0.0184	0.0416
Spacy	0.1912	0.2324	0.3017	0.3380	0.0044	0.0082	0.0206	0.0361

表 4.3 基于不同向量化方法的 Query-Search 推荐系统性能

4.2.3 基于大语言模型的混合意图识别推荐系统实现

在本节中基于 3.2.5 中所设计的混合意图识别推荐系统框架以及算法 3.8 进行算法实现。在本实验中，作为基线的是多臂老虎机算法（Multi-Armed Bandit, MAB）用于探索行为规律以及探索发掘之间的平衡。并且使用 Epsilon-Greedy 进行探索从而实现已有信息和新信息的平衡^[47]，最大化累积奖励^[48-49]，在本工作中进行探索性发掘的概率值设置为 0.1。

	NDCG@k				Recall@k			
	k=10	k=20	k=50	k=100	k=10	k=20	k=50	k=100
MAB	0.1101	0.1120	0.1245	0.1428	0.0206	0.0373	0.0784	0.1293
-	0.1912 (+73.59%)	0.2324 (+107.14%)	0.3017 (+142.29%)	0.3380 (+136.73%)	0.0044 (-78.64%)	0.0082 (-78.00%)	0.0206 (-73.72%)	0.0361 (-72.08%)
Spacy	K1_align 0.2435 (+121.06%)	0.2514 (+124.52%)	0.3019 (+142.49%)	0.3461 (+142.43%)	0.0042 (-79.61%)	0.0061 (-83.65%)	0.0142 (-81.89%)	0.0291 (-77.49%)
-	0.1937 (+75.88%)	0.2387 (+113.19%)	0.2834 (+127.59%)	0.3304 (+131.38%)	0.0028 (-86.41%)	0.0057 (-84.73%)	0.0138 (-82.40%)	0.0286 (-77.87%)
Spacy	K2_align 0.1820 (+65.04%)	0.2060 (+83.93%)	0.2626 (+110.85%)	0.3221 (+125.56%)	0.0033 (-83.98%)	0.0055 (-85.27%)	0.0124 (-84.18%)	0.0272 (-78.95%)
-	0.1759 (+59.76%)	0.2057 (+83.71%)	0.2609 (+109.54%)	0.3202 (+124.21%)	0.0033 (-83.98%)	0.0058 (-84.47%)	0.0126 (-83.93%)	0.0265 (-79.49%)
Spacy	K3_align 0.1538 (+39.68%)	0.1985 (+77.23%)	0.2520 (+102.39%)	0.3110 (+117.86%)	0.0022 (-89.32%)	0.0047 (-87.40%)	0.0113 (-85.60%)	0.0245 (-81.04%)
-	0.1530 (+38.87%)	0.2020 (+80.36%)	0.2543 (+104.21%)	0.3134 (+119.45%)	0.0024 (-88.35%)	0.0049 (-86.86%)	0.0114 (-85.46%)	0.0248 (-80.83%)
TF-IDF	K2_align 0.1927 (+75.03%)	0.2265 (+102.24%)	0.2729 (+119.17%)	0.3281 (+129.91%)	0.0031 (-84.95%)	0.0056 (-84.98%)	0.0126 (-83.93%)	0.0274 (-78.80%)
-	0.2151 (+95.23%)	0.2489 (+122.23%)	0.3007 (+141.48%)	0.3541 (+148.02%)	0.0046 (-77.67%)	0.0080 (-78.56%)	0.0184 (-76.55%)	0.0416 (-67.76%)
TF-IDF	K1_align 0.2190 (+98.19%)	0.2482 (+121.61%)	0.3019 (+142.49%)	0.3343 (+134.14%)	0.0039 (-81.07%)	0.0066 (-82.30%)	0.0150 (-80.87%)	0.0279 (-78.43%)

表 4.4 推荐系统性能比较

从实验结果来看，虽然在召回率上相比原先的基线模型性能有所下降，但在召回质量上均有较大的提升，说明模型在提升推荐效果上是有较为优秀的效果的。

4.3 实验结论

在本节中进行了第三章中所设计的基于大语言模型的基于交互行为的长短期自适应推荐网络、基于正则匹配阈值修正的基于内容推荐网络，以及使用 Query-

Search 及行为内容对齐修正的推荐系统框架的实验验证，最终在自适应长短期推荐模块中提高了至多 26.63% 的召回率表现并在最终的整体框架上相比已有的混合框架提升了整体功能性以及框架对混合意图的识别能力，证明框架的有效性。

第 5 章 总结与未来工作

5.1 内容总结

本工作基于大语言模型技术以及基于内容及基于行为的推荐系统的分析，分别提出了基于交互数据加权及大语言模型系统的长短期自适应推荐系统、基于阈值过滤以及阈值匹配的内容推荐系统、以及基于大语言模型及两种推荐方法修正对齐的混合意图识别推荐系统框架。具体而言，本工作首先提出了推荐智能体的记忆结构用于形成对推荐系统主体的语义理解和认知。其次，提出了基于矩阵分解的加入线性权重层以及基于大语言模型智能体的模型调用框架用于强化基于用户交互行为数据的推荐流程。随后提出了一种基于阈值过滤以及正则匹配增强的基于内容推荐系统。最后基于上述推荐系统及智能体给出了完整的基于大语言模型的内容-交互修正对齐的混合意图识别推荐系统，并给出了完整的系统设计框架图。

同时本工作基于上述推荐系统框架给出了实验验证，证明了混合意图识别的推荐系统框架对于推荐召回率以及推荐质量评价指标的提升，同时增加了推荐系统的整体可解释性以及语义推理能力。作为大语言模型与传统推荐系统算法融合的前沿课题研究，本工作为基于大语言模型的推荐系统设计提供了一套全新的范式，对于推荐系统领域后续深入研究以及社会经济综合发展都会有更加深远的意义。

5.2 未来工作

未来的工作更加关注基于大语言模型基于真实交互情景的交互式推荐以及更多的基于语义理解推理的推荐技术研究。具体而言现在的推荐系统指标都基于收集的既往推荐系统行为数据以及知识库，而实际上推荐行为本身也是会对被推荐主体的行为产生影响的。基于大语言模型技术以及其他语义理解技术，有望在之后的推荐系统设计中加入更多与被推荐主体交互以及基于真实推荐环境中产生的实时数据集进行推荐系统技术优化研究，并基于全新的交互场景提出召回率、ndcg 以外的更加合适的评价指标。

此外，基于大语言模型的推荐系统能够对更加混合式的推荐环境产生理解。诸如第三章开头所提到的城市空间情景下的推荐问题，相比与网络空间的物品有

更多的真实时空信息，因此对 POI 的推荐问题会更加依赖于对于语义以及自然语言、时空信息的理解能力，也是现在推荐系统技术难以解决的前沿难题。未来基于大语言模型以及智能体技术的研究有望对城市空间推荐技术有更加深入且有价值的研究。

插图索引

图 1.1 协同过滤示意图	4
图 1.2 基于大语言模型的推荐系统示意图，该图来自文献 ^[17]	7
图 1.3 GPT4Rec 流程示意图，该图来自文献 ^[16]	10
图 3.1 物品推荐示意图	18
图 3.2 基于大语言模型智能体的完整的 POI 推荐系统推荐流程示意图	22
图 3.3 权重长短期推荐网络示意图	26
图 3.4 Star Wars: Episode I - The Phantom Menace.....	29
图 3.5 Guardians of the Galaxy	29
图 3.6 Star Wars: Episode VII - The Force Awakens	29
图 3.7 The Mandalorian	29
图 3.8 大语言模型驱动的自适应混合推荐系统框架	35
图 4.1 MF@2fold_agent Recall@5 实验结果.....	39

表格索引

表 4.1	MovieLens-1M 数据集的统计特性	37
表 4.2	K-fold MF 实验结果	38
表 4.3	基于不同向量化方法的 Query-Search 推荐系统性能	40
表 4.4	推荐系统性能比较	40

参考文献

- [1] Phillips H. Great library of alexandria[J]. Library philosophy and practice, 2010(Aug.).
- [2] Armstrong R, Freitag D, Joachims T, et al. Webwatcher: A learning apprentice for the world wide web[C]//AAAI Spring symposium on Information gathering from Heterogeneous, distributed environments: volume 93. Stanford, 1995: 107.
- [3] Meituan. Announcement of the results for the year ended december 31, 2023[EB/OL]. 2024. https://media-meituan.todayir.com/202403221654401765350700_en.pdf.
- [4] Harper F M, Konstan J A. The movielens datasets: History and context[J]. Acm transactions on interactive intelligent systems (tiis), 2015, 5(4): 1-19.
- [5] Ko H, Lee S, Park Y, et al. A survey of recommendation systems: recommendation models, techniques, and application fields[J]. Electronics, 2022, 11(1): 141.
- [6] Resnick P, Iacovou N, Suchak M, et al. GroupLens: An open architecture for collaborative filtering of netnews[C]//Proceedings of the 1994 ACM conference on Computer supported cooperative work. 1994: 175-186.
- [7] CarlKadie J B D. Empirical analysis of predictive algorithms for collaborative filtering[J]. Microsoft Research Microsoft Corporation One Microsoft Way Redmond, WA, 1998, 98052.
- [8] Goldberg D, Nichols D, Oki B M, et al. Using collaborative filtering to weave an information tapestry[J]. Communications of the ACM, 1992, 35(12): 61-70.
- [9] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th international conference on World Wide Web. 2001: 285-295.
- [10] Pearson K. Vii. mathematical contributions to the theory of evolution.—iii. regression, heredity, and panmixia[J]. Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character, 1896(187): 253-318.
- [11] Salton G, Buckley C. Information filtering and information retrieval: Two sides of the same coin?[C]//Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR). ACM, 1989: 11-18.
- [12] Pazzani M J, Billsus D. Content-based recommendation systems[M]//The Adaptive Web: Methods and Strategies of Web Personalization. Springer-Verlag, 2007: 325-341.
- [13] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems: volume 30. Curran Associates, Inc., 2017.

- [14] Sun F, Liu J, Wu J, et al. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer[C]//Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019: 1441-1450.
- [15] Khorasani S M H, Huber S, Falk J, et al. Transformers4rec: Bridging the gap between nlp and sequential/session-based recommendation[C]//Fifteenth ACM Conference on Recommender Systems. 2021: 143-149.
- [16] Li J, Zhang W, Wang T, et al. Gpt4rec: A generative framework for personalized recommendation and user interests interpretation[A]. 2023.
- [17] Xu L, Zhang J, Li B, et al. Prompting large language models for recommender systems: A comprehensive framework and empirical analysis[A]. 2024.
- [18] Burke R. Hybrid recommender systems: Survey and experiments[J]. User Modeling and User-Adapted Interaction, 2002, 12(4): 331-370.
- [19] Burke R. Hybrid web recommender systems[M]//The Adaptive Web. Springer, 2007: 377-408.
- [20] Burke R. Knowledge-based recommender systems[M]//Encyclopedia of Library and Information Science. Marcel Dekker, Inc., 2000: 180-200.
- [21] Rendle S. Factorization machines with libfm[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2012, 3(3): 57.
- [22] Anderson C. The long tail: Why the future of business is selling less of more[M]. New York: Hyperion, 2006.
- [23] Brynjolfsson E, Hu Y J, Smith M D. From niches to riches: Anatomy of the long tail[J]. MIT Sloan Management Review, 2006, 47(4): 67-71.
- [24] Steck H. Embarrassingly shallow autoencoders for sparse data[C]//The World Wide Web Conference. 2019: 3251-3257.
- [25] Cauchy A L. Méthode générale pour la résolution des systèmes d'équations simultanées[J]. Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences, 1847, 25: 536-538.
- [26] Robbins H, Monro S. A stochastic approximation method[J/OL]. The Annals of Mathematical Statistics, 1951, 22(3): 400-407. DOI: 10.1214/aoms/1177729586.
- [27] Jones K S. A statistical interpretation of term specificity and its application in retrieval[J]. Journal of Documentation, 1972, 28(1): 11-21.
- [28] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[A]. 2019.
- [29] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation[C]// Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.

- [30] Bojanowski P, Grave E, Joulin A, et al. Enriching word vectors with subword information[J]. Transactions of the Association for Computational Linguistics, 2017, 5: 135-146.
- [31] Wang H, Li J, Wu H, et al. Pre-trained language models and their applications[J]. Engineering, 2022.
- [32] Brown T B, Mann B, Ryder N, et al. Language models are few-shot learners[A]. 2020.
- [33] Wang L, Ma C, Feng X, et al. A survey on large language model based autonomous agents[J]. Frontiers of Computer Science, 2024, 18(6): 1-26.
- [34] Crockford D. The application/json media type for javascript object notation (json)[EB/OL]. 2006. <https://tools.ietf.org/html/rfc4627>.
- [35] Robbins H. Some aspects of the sequential design of experiments[J]. Bulletin of the American Mathematical Society, 1952, 58(5): 527-535.
- [36] Star wars: Episode i - the phantom menace[EB/OL]. <https://www.starwars.com/films/star-wars-episode-i-the-phantom-menace>.
- [37] Guardians of the galaxy[EB/OL]. <https://www.marvel.com/movies/guardians-of-the-galaxy>.
- [38] Star wars: Episode vii - the force awakens[EB/OL]. <https://www.starwars.com/films/star-wars-episode-vii-the-force-awakens>.
- [39] The mandalorian[EB/OL]. <https://www.starwars.com/series/the-mandalorian>.
- [40] GroupLens Research. Grouplens research project[EB/OL]. <https://grouplens.org/>.
- [41] GroupLens Research. MovieLens 100k dataset[EB/OL]. 1998. <https://grouplens.org/datasets/movielens/100k/>.
- [42] GroupLens Research. MovieLens 1m dataset[EB/OL]. 2003. <https://grouplens.org/datasets/movielens/1m/>.
- [43] GroupLens Research. MovieLens 10m dataset[EB/OL]. 2009. <https://grouplens.org/datasets/movielens/10m/>.
- [44] GroupLens Research. MovieLens 20m dataset[EB/OL]. 2016. <https://grouplens.org/datasets/movielens/20m/>.
- [45] Herlocker J L, Konstan J A, Terveen L G, et al. Evaluating collaborative filtering recommender systems[J]. ACM Transactions on Information Systems (TOIS), 2004, 22(1): 5-53.
- [46] Järvelin K, Kekäläinen J. Cumulated gain-based evaluation of ir techniques[C]//ACM Transactions on Information Systems (TOIS): volume 20. ACM New York, NY, USA, 2002: 422-446.
- [47] Sutton R S, Barto A G. Reinforcement learning: An introduction[C/OL]//Cambridge, MA, USA: MIT Press, 1998. <http://incompleteideas.net/book/the-book-2nd.html>.

- [48] Lattimore T, Szepesvári C. Bandit algorithms[M/OL]. Cambridge, UK: Cambridge University Press, 2020. <https://tor-lattimore.com/downloads/book/book.pdf>.
- [49] Auer P, Cesa-Bianchi N, Fischer P. Finite-time analysis of the multiarmed bandit problem [J/OL]. Machine Learning, 2002, 47(2-3): 235-256. <https://link.springer.com/article/10.1023/A:1013689704352>.

致 谢

本科五年的故事，至此终了。羞愧于不才，庆幸于所遇之人皆不偏不倚，或同行，或引领，将永远铭记。

衷心感谢杨毅老师以及韩竹老师对我科研之路的启蒙，感谢冯昊学长一起奋斗撰写论文的日子。从零到一的步骤从来都是最难的，但我很感恩科研路上初出茅庐的日子里有耐心的指导者和同行者。起点虽然平凡，但是未来艰险都会怀念这段轻松、快乐又充实的日子。

感谢 Fiblab 实验室的同僚的并肩作战。感谢李勇老师高屋建瓴的意见指导和方向引领，感谢高宸老师和兰孝翀学长带领我勇攀高峰的奋斗时光，没有 Fiblab 这段艰苦的岁月，我难有今天钢铁一般的意志。

难忘吕晖老师本科期间对我的照顾，无论是心理上的还是精神上的。病重期间，绝望之时，Attention is all I need。

感谢法学院和统计辅修的朋友们，短暂同行过，给了我本科生活不一样的光亮。

谢谢老妈一路对我的精神鼓舞，谢谢吐司和傅里叶，半夜奋斗的日子有家人和猫的陪伴是无比温馨的。

最后感谢在黑暗中未曾放弃过的自己，谢谢你在绝境中没有放弃过哪怕一次。堂吉诃德不是愚昧的傻子，是这个世界上最勇敢最伟大的骑士。

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名： _____ 日 期： _____

附录 A 外文资料的书面翻译

GPT4Rec: A Generative Framework for Personalized Recommendation and User Interests Interpretation

目录

A.1 摘要	55
A.2 计算机分类系统概念	56
A.3 引言	56
A.4 方法	57
A.4.1 使用语言模型进行查询生成	57
A.4.2 使用搜索引擎进行项目检索	58
A.4.3 训练策略	58
A.5 实验	59
A.5.1 实验设置	59
A.6 定量分析	61
A.6.1 总体性能	61
A.6.2 多查询生成的优势	61
A.6.3 定性分析	62
A.7 结论	62
参考文献	64

A.1 摘要

近期的研究者在自然语言处理（NLP）方面取得了较大进展，并且将这些技术应用于推荐系统，显示出卓越的性能。然而，目前的模型通常将项目仅视为 ID，并采用判别建模，导致以下限制：(1) 无法充分利用项目的内容信息和 NLP 模型的语言建模能力；(2) 无法解释用户兴趣以提高相关性和多样性；(3) 无法适应项目库存增长等实际情况。为了解决这些限制，我们提出了 GPT4Rec，这是一种受搜索引擎启发的新颖且灵活的生成框架。它首先根据用户历史中的项目标题生成假设的“搜索查询”，然后通过搜索这些查询来检索推荐项目。该框架通过在语

言空间中学习用户和项目嵌入，克服了先前所提到的限制。为了更好地捕捉用户兴趣的不同方面和精细程度，以提高相关性和多样性，我们提出了一种基于束搜索的多查询生成技术。生成的检索词自然地作为用户兴趣的可解释表示，并且可以通过搜索推荐冷启动项目。本工作中使用 GPT-2 语言模型和 BM25 搜索引擎，最终在两个公共数据集上的 Recall@K 分别比最先进的方法高出 75.7% 和 22.2%。实验结果进一步表明，基于束搜索的多查询生成提高了检索项目的多样性和用户多兴趣的覆盖率。最终我们还通过定性案例研究讨论了生成查询的适应性和可解释性。

A.2 计算机分类系统概念

信息系统 → 推荐系统

A.3 引言

推荐系统是信息过滤系统，用于提供个性化的建议，以推荐与特定用户相关的项目，广泛应用于电子商务和社交媒体服务^[1-4]。随着自然语言处理（NLP）的发展，许多基于 NLP 的模型通过建模用户-项目交互序列来进行个性化推荐^[5-7]。大型语言模型（LLM，^[8-11]）在各种 NLP 任务或对话（如 ChatGPT^[12]）中取得了卓越的表现，推动了基于 LLM 的推荐系统的研究^[13-18]。特别是，BERT4Rec^[17]采用了 BERT^[9]的双向自注意力结构，超越了其他基于 NLP 的模型和序列模型^[5,14,19]。

尽管模型性能有所提升，但现有的基于 NLP 的模型^[13-18]通常将项目仅视为 ID 并采用判别建模，存在以下限制：首先，这些模型无法充分利用项目的内容信息和 NLP 模型的语言建模能力。其次，这些模型无法适应不断变化和增长的项目库存，而这些都是重要的实际问题^[20-21]。此外，判别模型难以解释用户兴趣，这对于提高推荐的多样性和质量至关重要^[22-25]。

为了解决这些限制，我们提出了 GPT4Rec，这是一种新颖且灵活的生成框架，用于个性化推荐，并同时提供用户兴趣的解释。我们的框架灵感来自搜索引擎：GPT4Rec 首先使用生成语言模型生成假设的“搜索查询”，该模型将用户历史中的项目标题与生成提示结合作为输入。然后，GPT4Rec 通过搜索生成的查询来检索推荐项目。GPT4Rec 的关键组件是一个强大的生成语言模型，它在语言空间中学习用户和项目的嵌入，使我们能够利用项目标题中的语义信息并捕捉用户的多样兴趣。为了解码用户的多重兴趣并提高推荐的多样性，我们在查询生成中

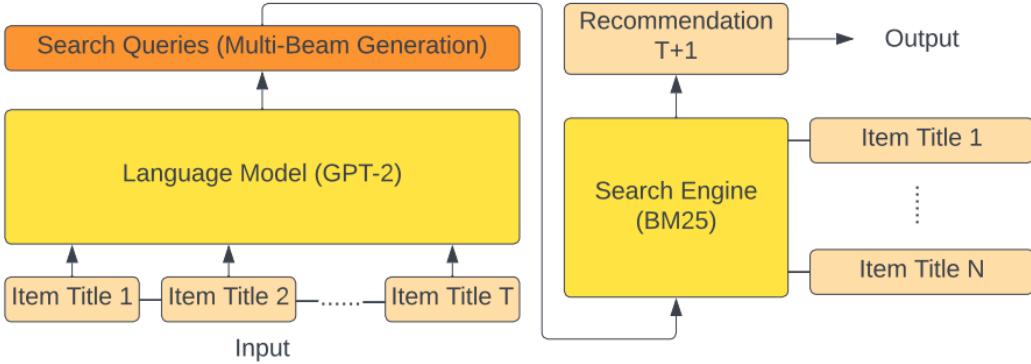


图 A.1 GPT4Rec 框架的架构

采用了多查询束搜索技术。这些查询是人类可理解的，并且在解释用户兴趣方面具有独立价值。此外，通过搜索代表用户兴趣的查询进行推荐，可以自然地解决项目冷启动问题和项目库存变化问题。最后，GPT4Rec 在实践中具有灵活性，可以结合更先进的 LLM 或搜索引擎以提高性能。总之，我们的贡献包括：(1) 我们提出了 GPT4Rec，一种新颖且灵活的生成框架，将推荐任务视为查询生成 + 搜索；(2) 我们采用多查询束搜索策略，生成多样且可解释的用户兴趣表示；(3) 我们在两个公共数据集上进行了实验，展示了 Recall@K 相对于最先进方法的显著改进；(4) 通过定量和定性分析，我们调查并展示了生成查询的数量提高了检索项目的多样性以及用户多重兴趣的覆盖率。

A.4 方法

我们的框架的架构如图 1 所示。给定用户的项目交互序列，GPT4Rec 通过提示格式化项目标题，并使用生成语言模型在语言空间中学习项目和用户嵌入。然后，该模型生成多个代表用户兴趣的查询，这些查询将被输入到搜索引擎中以检索推荐项目。在本文中，我们选择了 GPT-2 语言模型和 BM25 搜索引擎，而该框架可以灵活地结合更先进的模型。

A.4.1 使用语言模型进行查询生成

GPT4Rec 的第一个组件是生成语言模型，其目标是从项目交互序列中学习语言空间中的用户表示，然后生成多个代表用户兴趣的查询。通过微调选择的 GPT-2 模型（该模型具有 117M 参数，具有复杂的 transformer 架构，并在庞大的

语言语料库上进行了预训练)，实现捕捉用户兴趣和项目内容信息的目的。通过实验，我们决定使用以下提示格式化模型输入：

之前，顾客购买了：

<项目标题 1>。<项目标题 2>...

未来，顾客想购买

这个提示包含项目标题中的语义信息，并被记作每个用户的 W_u 。GPT-2 从 W_u 学习用户表示，然后能够基于条件分布 $P(\cdot|W_u)$ 以序列方式生成查询。

为了更好地表示用户的多样兴趣并增加推荐结果的多样性，我们建议使用束搜索技术生成多个查询^[19,26]。给定束宽 m ，生成评分函数 $S(\cdot)$ ，以及长度为 l 的候选查询 $Q_l = (q_{l1}, q_{l2}, \dots, q_{lm})$ ，束搜索算法将 Q_{l+1} 更新为长度为 $l+1$ 的前 m 个查询，这些查询最大化 $S(W_u, q)$ ，其中 $q \in \{|q| = l+1, q[1:l] \in Q_l\}$ 。最重要的是，这种束搜索策略生成了在不同方面和粒度上多样化的用户兴趣表示。

A.4.2 使用搜索引擎进行项目检索

我们框架的第二个组件是一个搜索引擎，它作为“判别器”使用。它将每个生成的查询作为输入，并使用匹配评分函数从库存中检索最相关的项目。匹配评分函数测量语言空间中的相似性，发挥类似于向量嵌入方法中的内积相似性的作用^[1,4]。我们选择采用 BM25 匹配评分函数^[27]，因为它是最广泛使用的基准搜索引擎之一，考虑了词频饱和和文档长度，具有两个相应的参数 k_1 和 b 。

其中 K 表示推荐项目的总数， m 表示生成的查询数量，我们提出一种基于排名的策略来结合每个查询的搜索结果。首先使用生成评分最高的查询从搜索结果中检索前 K/m 个项目，然后按分数排名依次添加其余查询中 K/m 个不重复的项目。所提出的策略能够平衡检索项目的相关性和多样性。

A.4.3 训练策略

我们提出了一种灵活的两步训练过程，分别优化语言模型和搜索引擎，这在模型迭代中非常有利。给定每个用户 u 的项目交互序列 i_1, i_2, \dots, i_T ，我们在 2.1 节所示的提示中取前 $T - 1$ 个项目标题，然后将其与最后一个项目 i_T 的标题连接起来，形成训练语料库，以微调预训练的 GPT-2 模型。这种策略受到对比学习文献^[28]的启发，其基本思想是最细粒度和最准确的搜索查询是目标项目标题本身。在训练完语言模型后，我们通过网格搜索参数 k_1 和 b 优化 BM25 参数，以在搜

索生成查询以检索目标项目时获得最佳性能。

A.5 实验

A.5.1 实验设置

A.5.1.1 数据

我们在两个公共数据集上进行实验，分别是 5-core Amazon Review 数据集^[29]中的美容和电子产品类别。标题缺失或噪声标题（超过 400 个字符）的项目被丢弃。每个用户的项目交互序列去重并截取到最大长度为 15。预处理数据集的描述性统计如表 2 所示。按照下一项预测任务设置^[17]，用户序列按 0.8/0.1/0.1 的比例分割为训练集、验证集和测试集，并将每个测试序列中的最后一项视为预测目标。

表 A.1 基线方法和提出框架的整体性能，生成不同数量查询的情况下。最佳性能以粗体显示，最佳基线结果下划线标记。

Dataset	Recall@K	FM-BPR	ContentRec	YouTubeDNN	BERT4Rec	GPT4Rec			
						5 Queries	10 Queries	20 Queries	40 Queries
Beauty	K=5	0.0356	0.0254	<u>0.0376</u>	0.0355	0.0653	—	—	—
	K=10	0.0499	0.0440	0.0549	0.0513	0.0810	0.1036	—	—
	K=20	0.0716	0.0644	0.0753	0.0816	0.1207	0.1252	0.1454	—
	K=40	0.1040	0.0952	0.1066	0.1161	0.1297	0.1522	0.1743	0.2000
Electronics	K=5	0.0345	0.0241	0.0352	<u>0.0362</u>	0.0434	0.0545	—	—
	K=10	0.0387	0.0307	0.0435	<u>0.0451</u>	0.0480	0.0515	—	—
	K=20	0.0441	0.0391	0.0539	0.0573	0.0524	0.0607	0.0705	—
	K=40	0.0505	0.0494	0.0684	0.0751	0.0574	0.0672	0.0788	0.0918

表 A.2 数据集统计信息

Name	User	Item	Interaction	Ave. Length	Meta Info.
Beauty	22,254	11,778	190,726	7.439	Cate.
Electronics	728,719	159,456	6,724,582	7.297	Cate., Brand

A.5.1.2 评估指标

Recall@K 是预测下一项的主要兴趣指标，衡量推荐的前 K 项中是否包含目标项，并在所有用户中取平均值。除了 Recall@K，我们还对以下指标感兴趣：

Diversity@K 衡量项目之间的差异性，定义如下：

$$\text{Diversity@K} = \text{Average} \left[1 - \frac{\sum_{i_1 \neq i_2} \text{Sim}(i_1, i_2)}{K(K-1)} \right] \quad (\text{A.1})$$

其中采用 Jaccard 相似度^[8]来测量项目类别或品牌之间的相似性。

Coverage@K 衡量推荐项目的覆盖范围，定义如下：

$$\text{Coverage@K} = \text{Average} \left[\frac{|\bigcup_{i \in R} \text{Cate}(i) \cap \bigcup_{i \in U} \text{Cate}(i)|}{|\bigcup_{i \in U} \text{Cate}(i)|} \right] \quad (\text{A.2})$$

这是一种更合理的度量用户多重兴趣覆盖率的方法，与 Diversity@K 完全不同^[22]。

A.5.1.3 基线方法

FM-BPR^[2]是一种使用贝叶斯个性化排序标准的因子分解机器，学习用户-项目交互矩阵的协同信息。ContentRec 是一种基于内容的模型，广泛应用于行业，学习项目标题的词嵌入，并将这些嵌入应用于用户交互项的词嵌入。YouTubeDNN^[1]是行业中最流行的深度学习模型之一，采用网络来学习用户-项目嵌入。BERT4Rec^[17]是一种最先进的 BERT 模型，具有双向自注意力架构和掩码训练策略。

A.5.1.4 实现细节

我们实现了 GPT-2 模型，由 HuggingFace 提供^[30]，具有 117M 参数，并通过 Adam 优化器和学习率衰减进行微调。我们将学习率设置为 0.0001 并使用 2000 步预热。BM25 通过网格搜索 $k_1 \in [0, 3], b \in [0, 1]$ 优化。ContentRec 由 StarSpace^[6]实现。所有基线方法在公共 GitHub 库^[23]上可用。对于基线方法，我们选择了最佳维度 641,128,256 基于原论文作者的意见。

A.6 定量分析

A.6.1 总体性能

表 1 总结了 GPT4Rec 在两个数据集上不同生成查询数的 Recall@K 与基线方法的比较。我们的框架在两个数据集上都优于所有基线方法，在较小的美容数据集上实现了 75.7% 的相对改进，在较大的电子产品数据集上实现了 Recall@K@40 上的 22.2% 的改进。

与基线方法的比较表明，项目信息和现代语言建模是实现卓越性能的关键因素。一方面，BERT4Rec 在两种数据集上表现最好，但由于将项目视为 ID 而未能充分利用项目的内容信息。另一方面，ContentRec 的词嵌入和多项式建模不足以实现可比性能。

A.6.2 多查询生成的优势

表 1 和表 3 中的数据表明，随着生成查询数的增加，推荐结果的相关性和多样性显著提高。特别是，生成多查询的束搜索策略显著提高了推荐的相关性。生成 K 个查询并检索每个查询的前 K 项项目获得了最佳的 Recall@K 性能。表 3 中的 Diversity@K 和 Coverage@K 的增加趋势也证明了多查询生成策略能够产生更综合的用户兴趣表示。

进一步调查生成查询的多样性和用户兴趣的覆盖率表明，多查询生成策略能捕捉用户更广泛的兴趣。相反，增加查询数量而不生成多样性则无法提高性能。

这些结果表明，多查询生成在提高推荐多样性和相关性方面具有显著优势。我们的框架灵活地结合了更先进的生成语言模型和搜索引擎，提供了进一步探索的有趣方向。

表 A.3 用户兴趣的多样性和覆盖率与生成查询数量的对比。关于类别/品牌信息的最高值以粗体显示。

Dataset	Beauty (Category)				Electronics (Category)				Electronics (Brand)			
	5	10	20	40	5	10	20	40	5	10	20	40
Diversity@K	K=5	0.679	—	—	—	0.671	—	—	0.534	—	—	—
	K=10	0.654	0.716	—	—	0.617	0.733	—	—	0.529	0.643	—
	K=20	0.659	0.706	0.749	—	0.605	0.703	0.778	—	0.559	0.645	0.717
	K=40	0.679	0.715	0.749	0.783	0.601	0.696	0.762	0.811	0.604	0.669	0.724
Coverage@K	K=5	0.417	—	—	—	0.321	—	—	0.173	—	—	—
	K=10	0.472	0.547	—	—	0.340	0.425	—	—	0.177	0.239	—
	K=20	0.535	0.602	0.674	—	0.364	0.447	0.537	—	0.184	0.245	0.317
	K=40	0.614	0.669	0.726	0.787	0.389	0.474	0.562	0.653	0.197	0.255	0.324



图 A.2 具有多样兴趣的美容产品测试用户示例，目标项目以红色圈出。



图 A.3 具有具体兴趣的电子产品测试用户示例，目标项目以红色圈出。

A.6.3 定性分析

在本节中，我们探讨了生成查询在捕捉用户兴趣和解释用户兴趣方面的有效性，并通过案例研究展示了其用处。我们在图 2 和图 3 中展示了两个测试用户序列和模型输出。

在第一个例子中，测试用户对美容产品的多个类别和品牌有多样的兴趣。在这种情况下，GPT4Rec 能够生成不仅覆盖用户历史中的一些产品或品牌的多样查询，还生成用户历史中从未出现的相关项目，例如“化妆盘”。这表明 GPT4Rec 能够基于用户-项目交互和项目标题语义信息，捕捉项目之间的关联性。在第二个例子中，测试用户对罗技无线鼠标有非常具体的兴趣。GPT4Rec 能够捕捉这一特征，所有生成的查询都集中在同一个品牌上，相同类别但产品细节不同。

我们已经证明了 GPT4Rec 在生成捕捉用户兴趣的查询方面的有效性，这些查询跨越了不同的方面和粒度水平，如上面两个案例研究所示。这些查询直接为解释用户兴趣服务。此外，两个例子的比较表明，生成查询的多样性水平适应了用户序列中的水平，这有助于进一步了解他们的行为。

A.7 结论

在这项工作中，我们提出了 GPT4Rec，一个新颖的生成框架，可以同时生成个性化推荐和可解释的用户兴趣表示。利用先进的语言模型和项目内容信息，GPT4Rec 实现了卓越的性能，并自然地解决了项目冷启动问题。所提出的多查询

束搜索技术生成了在不同方面和粒度上多样的用户兴趣表示，提高了推荐结果的相关性和多样性。我们的框架具有灵活性，可以结合更先进的生成语言模型或搜索引擎，以及更好的生成和检索策略，这些都是未来探索的有趣方向。

参考文献

- [1] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.
- [2] Maciej Kula. Metadata embeddings for user and item cold-start recommendations. In *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems (RecSys 2015), Vienna, Austria, September 16-20, 2015. (CEUR Workshop Proceedings, Vol. 1448)*, pages 14–21. CEUR-WS.org, 2015. URL <http://ceur-ws.org/Vol-1448/paper4.pdf>.
- [3] Aditya Pal, Chantat Eksombatchai, Yitong Zhou, Bo Zhao, Charles Rosenberg, and Jure Leskovec. Pinnersage: Multi-modal user embedding framework for recommendations at pinterest. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2311–2320, 2020.
- [4] Steffen Rendle. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE, 2010.
- [5] Dietmar Jannach and Malte Ludewig. When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 306–310, 2017.
- [6] Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. Starspace: Embed all the things! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [7] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1893–1902, 2020.
- [8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [10] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

- [11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [12] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. (*2018*), 2018.
- [13] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). *arXiv preprint arXiv:2203.13366*, 2022.
- [14] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.
- [15] Gustavo Penha and Claudia Hauff. What does bert know about books, movies and music? probing bert for conversational recommendation. In *Fourteenth ACM Conference on Recommender Systems*, pages 388–397, 2020.
- [16] Damien Sileo, Wout Vossen, and Robbe Raymaekers. Zero-shot recommendation as language modeling. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part II*, pages 223–230. Springer, 2022.
- [17] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019.
- [18] Yuhui Zhang, Hao Ding, Zeren Shui, Yifei Ma, James Zou, Anoop Deoras, and Hao Wang. Language models as recommender systems: Evaluations and limitations. 2021.
- [19] Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.
- [20] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pages 208–211, 2008.
- [21] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260, 2002.
- [22] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2942–2951, 2020.

- [23] Minmin Chen, Yuyan Wang, Can Xu, Ya Le, Mohit Sharma, Lee Richardson, Su-Lin Wu, and Ed Chi. Values of user exploration in recommender systems. In *Fifteenth ACM Conference on Recommender Systems*, pages 85–95, 2021.
- [24] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. Multi-interest network with dynamic routing for recommendation at tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2615–2623, 2019.
- [25] Yongfeng Zhang, Xu Chen, et al. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101, 2020.
- [26] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016.
- [27] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.
- [28] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.
- [29] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th international conference on world wide web*, pages 507–517, 2016.
- [30] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierrick Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

书面翻译对应的原文索引

在学期间参加课题的研究成果

学术论文：

- [1] Yang, Y., Feng, H., Cheng, Y., & Han, Z. (2024). Emotion-Aware Scene Adaptation: A Bandwidth-Efficient Approach for Generating Animated Shorts. *Sensors*, 24(5), 1660.
- [2] Lan, X., Cheng, Y., Sheng, L., Gao, C., & Li, Y. (2024). Depression Detection on Social Media with Large Language Models. *arXiv preprint arXiv:2403.10750*.
- [3] Lan, X., Piao, J., Cheng, Y., Gao, C., Li, Y., Niu, Y., Song, Y., Gai, K., & Li, Y. (2024). Recommendation for Inclusivity of Underrepresented Producers in User-generated Content Platforms.

其中 [1] 为基于大语言模型的语义通信相关研究, 发表于 2024 年的 MDPI-Sensors 会议;[2] 为基于大语言模型的抑郁症检测研究, 挂载于 Arxiv 网站上待投稿;[3] 为基于公平性的用户内容生产平台推荐系统研究, 2024 年 ACMMM 会议在投中。

专利：

- [4] 杨毅, 程逸鸣, 冯昊等. 一种将视觉内容转换为虚拟动画视觉表示的语义编解码框架方法.

其中 [4] 为基于学术论文 [1] 的成果转化专利, 正在进行专利申请

综合论文训练记录表

学生姓名		学号		班级	
论文题目					
主要内容以及进度安排	<p style="text-align: right;">指导教师签字: _____</p> <p style="text-align: right;">考核组组长签字: _____</p> <p style="text-align: right;">年 月 日</p>				
中期考核意见	<p style="text-align: right;">考核组组长签字: _____</p> <p style="text-align: right;">年 月 日</p>				

指导教师评语	
	指导教师签字: _____ 年 月 日
评阅教师评语	
	评阅教师签字: _____ 年 月 日
答辩小组评语	
	答辩小组组长签字: _____ 年 月 日

总成绩: _____

教学负责人签字: _____

年 月 日