基于 Agent-Chain 的统计摘要生成研究

郭利荣[1] 梁玉琪[1] 廖文亦[1]
GUO Lirong LIANG Yuqi LIAO Wenyi

摘 要 在过去很长一段时间内，文本生成式模型给世界带来了巨大的影响，在传统的自然语言处理领域（NLP）根据一段文本生成摘要一直是一个火热的问题，但是生成的摘要缺乏分析思维和数据展现。因此文本摘要生成存在着很大的缺陷，为了解决这一问题涌现了许多基于表格和文本联合的多模态工作，但是训练是昂贵的。基于 LLM（large language model）应用的 Langchain 框架中的 Agent 能力是值得尝试的工具，基于 Agent 实现 LLM 的自动调用统计，实现数据统计摘要生成，最后使用科大讯飞的星火 spark 作为评测模型，在本地构建的数据集中汇报自评 7.54 分，讯飞星火 Spark 评价 7.52 分（满分为 10 分）。

关键词 LLM；Agent；数据统计；摘要生成

doi：10.3969/j.issn.1672-9528.2024.02.046

0 引言

在信息化时代的背景下，文本数据呈现爆炸式增长，同质化文本过滤，有毒文本剔除，快速提取文章主要内容成为一个重要的课题，而这些工作的背后也来自自然语言处理（NLP）领域的重要基础工作：自动摘要生成。

在早期的自动摘要生成主要是基于抽取式摘要作为主要的思想，在这一方向上有来自曲阜师范大学的朱玉佳、祝永志、董兆安等人的工作，他们构建了一种无监督抽取式联合打分模型，使用 word2vec 的方法进行句子向量化，基于 TextRank 算法 [1] 通过词频逆句频余弦相似度与词向量余弦相似度共同计算句子得分得到初步摘要，通过最大边缘相关度算法（MMR）进行摘要冗余去除 [2]。随着深度学习时代的到来，特别是以 BERT 为代表的以 NLP 预训练模型为基础的语言表征模型，催生了河南财经政法大学黄菲菲等学者的工作，她通过使用预训练 BERT 模型的方法获取词向量，再使用 TextRank 算法成功克服了传统词向量获取存在一词多义的问题。但是抽取式的摘要具有天生的缺点，最后生成的文本内容必须来自给定的文本中，缺少分析性和总结性能力 [3]。

在生成式摘要方面，序列到序列（seq2seq）模型成为首选的模型，基于 seq2seq 的文本摘要生成方法有来自北京林业大学的周健等人，他们基于循环神经网络（RNN）和注意力机制（Attention）实现了一种 seq2seq 模型，通过拷贝词机制缓解了摘要生成过程中的未登录问题，基于这些技术提出 Copy-Generator 模型 [4]。来自陕西科技大学的党宏社等人基于 RNN 模型，提出基于存储注意力和解码自注意力构成的混合注意力，通过注意力计指增强对序列历史单词的注意力，并通过强化学习手段使用新的训练方式来解决曝光偏差问题实现对损失函数的修正 [5]。也有改进方向为基于语句融合的方法，根据语句融合理论中的信息联系点概念对数据集进行预处理，设计以信息联系点为最小语义单元的排列语言模型，增强上下文信息捕捉能力，同时基于融合信息的注意力掩码策略控制模型在文本生成阶段的信息摄入 [6]，同时也有相关的工作基于全词注意力的文本摘要生成方法，详细对比抽取式任务和生成式任务，在不同层次、不同角度对文本摘要生成领域进行了探索，通过语料和知识库实现摘要生成优化 [7]。

在最近的相近工作是来自浙江大学的 TableGPT，它们通过构建多模态模型，使用 TableEncoder 对表格进行数据表征，并和用户问题理解联合训练了一个表格 + 文本的多模态模型，实现了数据摘要自动化功能 [8]，预训练是昂贵的，由于表格文本对数据的数据集十分稀少的，因此通过预训练来实现这一目的缺乏必要的数据支持。

通过基于基础大模型 LLM，挖掘 LLM 的摘要总结生成能力和思维推理能力，通过 LLM 推理和 Agent 实现代理任务，构造代理任务链 Agent-Chain，实现对数据的自动统计总结，实现具有数据分析能力的自动摘要生成，与传统模型的优缺点被归纳为表 1。基于预训练大语言模型（PLM）实现的智能体（Agent）任务，通过多阶段的激励性对话，为模型构建了类似思维链的能力，实现了模型处理复杂任务的能力，因此这为模型通过逐步分解复杂任务逐步与外界实现交互提供了一个很好的借鉴思路 [9]。同样的，来自谷歌的团队更进一步提出了一种闭环的方法，让大模型自行打造工具（Python

1. 中数通信息有限公司郭利荣大数据工作室 广东广州 510630

脚本）自行使用工具，实现了任务的自动实现、自动构建功能 [10]。

表 1 各类摘要模型的生成能力评价表

模型类型	优点	缺点
TextRank 抽取式摘要	模型成熟，计算量小	无监督评分模型，缺乏总结能力
Seq2Seq 序列生成	模型成熟，生成摘要	语义表征能力一般，无逻辑生成
TableGPT 多模态生成	天然的理解表格	需要训练，统计和总结能力一般

由于本文的模型旨在为了生成一段带有数据分析和数据统计的摘要报告，因此很难使用 STS-B、QQA、GLUE 等传统 NLP 指标作为模型生成结果的衡量，在这里借鉴了使用 GPT-4 为其他模型生成结果打分的方法 [11]，选择了来自科大讯飞的星火大模型对模型生成的摘要进行评价，并计算 100 条结果的平均分作为汇报指标。

1 实现方案

接下来将按照两个部分来介绍工作。（1）是如何让模型自动选择的工具构建代理链（Agent-Chain）对数据进行处理并构建推理模板，实际上这里说的代理链在系统的背后是一系列只要提供参数就可以运行的代码脚本，通过这些代码脚本可以实现大语言模型（LLM）跟外界的数据交互和状态感知。（2）设计评价模式得到客观的模型生成结果评价。同时将给出完整的数据流转过程并进行结构分析。

1.1 任务拆解实现意图理解构建 Agent-Chain

首先将 Agent-Chain 的构建拆解为两个子任务。（1）用户意图理解，构建执行逻辑，实现简单 Agent-Chain 的构建。（2）基于构建的 Agent-Chain 依靠 LLM 进行参数辅助设定并实现摘要生成。

通过 1 和 2 实现了用户意图理解，从代理池（Agent Lake）中进行代理（Agent）选择并排序，基于 LLM 的理解对脚本进行参数传递实现参数同步按要求构建。接下来将进行详细介绍并给出实现路径。

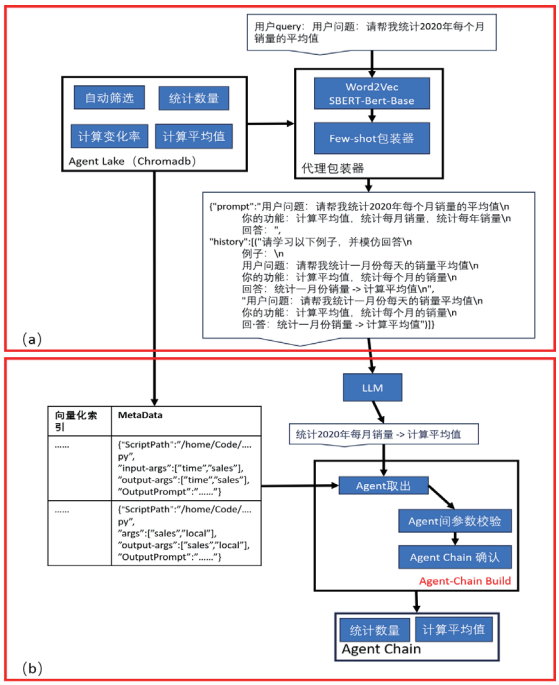
Part1：用户意图理解，构建执行逻辑，实现简单 Agent-Chain 的构建。

对于该任务，首先设计了一个代理包装器，有别于传统的 Prompt，在实践中可以发现，将用户问题和 Prompt 模板整合后在一次对话中一次性提交给模型，有很大概率会导致由于文本长度过长，小参数模型存在长距离建模能力弱。然后随着序列过长，模型的文本处理能力出现大幅度下降，因此基于多轮对话的思想，将 Prompt（事实上是一些 few-shot 文本）与期待模型实现的标准化输出构建成为模型的历史对话参数。最后在实验中相比较整合型的 Prompt 模板构建，基于多轮对话机制的 Prompt 模板，在 Agent-Chain 的构建上至少提高 10% 的精确率。

当用户问题进入 Agent-Chain 构建模型时，将通过代理包装器，搜索用户问题中可能涉及的代理任务，在模型加载期已经将模型的代理任务池通过任务描述的形式对模型描述进行向量化，存储在向量数据库中，通过代理包装器的方法，对用户 query 进行向量化，将 query 向量作为目标通过余弦相似度式（1）进行搜索。Q 表示 query 的 Embeddeding，K 表达向量数据库中的某个索引 Embeddeding。

$$\cos(\theta)=\frac{\sum_{i=1}^{n}(Q_i\times K_i)}{\sqrt{\sum_{i=1}^{n}Q_i^2}\times\sqrt{\sum_{i=1}^{n}K_i^2}} \quad (1)$$

通过搜索得到 Agent Lake 中最近似的 10 个代理任务，提取到代理包装器中与原设定的 few-shot 模板进行结合，提交模型进行推理，在上述方法中的代理包装器可以被如图 1（a）所描述。通过代理包装器，将包装后的用户 Query 传递到大模型处进行推理得到初步的执行流程，通过 Python 中的字典实现从 Agent Lake 中进行筛选，并对代理任务（Agent）进行组合，通过检测代理任务脚本的要求，判断脚本是否可以进行组合，若可以组合，则快速地构建初步的代理链 Agent-Chain，下面的流程图 1（b）将展示 Agent Lake 实际的存储模式和 Agent-Chain 的构建流程。



（a）代理包装器结构　（b）Agent Chain 构建结构

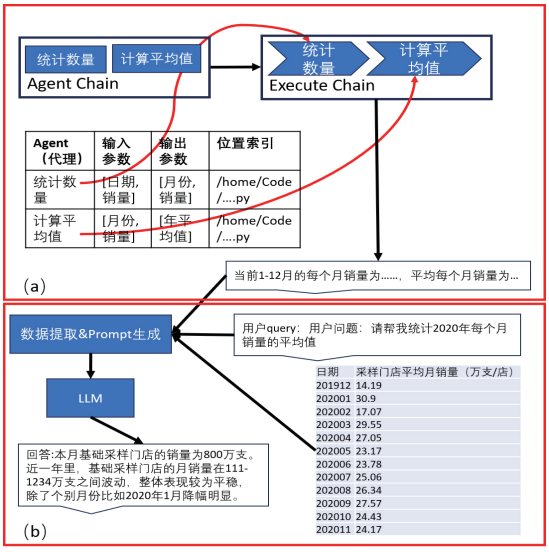
图 1 用户提问到 AgentChain 构建流程

基于图 1 梳理了代理包装器和 Agent-Chain Build 两个重要模块，通过上述代理包装器和代理任务链构建，在代理包装器中通过计算得到相关的代理任务，并通过 LLM 推理得到基础的概念链，通过代理任务链构建和代理任务参数检查，输出构建成功的代理任务链（Agent-Chain）。

Part2：基于构建的 Agent-Chain 依靠 LLM 进行参数辅助设定并实现摘要生成。

通过 Part1 基于通用大模型（LLM）、代理任务池（agent lake）、代理包装器、代理任务链构建器（Agent-Chain Build）实现了从用户初步问题到代理包装后的半结构性文本，使用 LLM 推理得到概念链，并通过代理任务链构建器，获得最后的代理任务链，但事实上这是不足的，因此还需要通过代理任务链到实际运行的程序任务上的链条来保证实际的运行。

在代理任务链构建中（图 2 中的 Agent-Chain Build）通过筛选从 Chromadb（向量数据库）中获得了代理任务脚本的参数要求和位置索引，所以后面的任务就变得尤为简单，只需要将脚本与参数整合并提交执行就可以获得对应的数据输出与提示性文本，如图 2（a）所示。接下来将获得的数据统计结果、用户模板、多轮对话设定通过模板整合器，交给模型进行推理，生成最后具有统计分析意义的分析报告，该过程的任务流程如图 2（b）所示。



（a）Execute-Chain 执行　（b）数据分析摘要生成

图 2 自动统计执行与摘要生成

1.2 构建评价模式

本文提出的是一种基于 LLM 的自动代理任务链构建，从而实现对数据的自动统计，将数据结果进行半结构的文本嵌入，并基于原始数据统计结果和用户问题进行拼装，提交给 LLM 进行代理推理。因此，该模式是难以进行对比的，本文使用与参考文献 [9] 中一致的方法，通过引入一个超大参数规模的大模型作为评分准则，在这里选择了来自科大讯飞公司的大模型讯飞星火（https://xinghuo.xfyun.）。但事实上单纯地使用大模型的评价存在偏差的可能，因此引入了一种计算相对偏差的方法，用于在一定程度上表达大模型评分中可能出现的偏好，实际上的主要做法是：D 表示实验所用的数据，使用 T_{Spark} 表示讯飞星火大模型 *Spark* 的评分，使用 T_{method} 表示模型使用的数据摘要生成模型的评分，使用 $S_{Spark}(D)$ 代表 *Spark* 使用数据 D 生成的结果，使用 $S_{method}(D)$ 代表模型使用数据 D 生成的结果，因此可以构建自动数据摘要的评价指标 F_{exam} 为：

$$F_{exam} = \frac{T_{Spark}(S_{Method}(D))}{T_{Method}(S_{Spark}(D))} \quad (2)$$

式（2）的最终取值会稳定在 (0,2) 之间，当公式 2 的评分趋近于 1，那么即可说明模型生成的数据摘要具有稳定可靠的特点。

2 实验

首先将简单介绍硬件设备，包括语言解释器、操作系统、GPU、CPU、内存等信息如下。

操作系统为 Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-82-generic x86_64)，Python 解释器 Python 3.9.6，核心库版本为 torch==2.0.1，Langchain==0.0.316，Chromadb==0.4.14，Pandas==2.0.3，GPU 为 NVIDIA A800 80 GB PCIe，CPU 与内存为 Intel(R) Xeon(R) Gold 6330 CPU/512 GB。

使用相同数据下，Qwen-14B 生成结果和科大讯飞模型生成结果的对比，横向比较在系统中接入不同模型，通过不同的大模型选择探究模型尺寸对数据摘要生成结果的影响，同时探究模型之间对彼此生成的结果的评分偏好，在文章的结尾作者给出了一部分消融实验进行证明，下面展示的是基于同样的数据讯飞星火 Spark 和阿里通义千问 -14B 的回答结果，如图 3 所示。

日期	基础采样门店销量
201912	2272.36
202001	4666.91
202002	2553.81
202003	5298.33
202004	5240.8
202005	5440.02
202006	5381.77
202007	5716.79
202008	5853.88
202009	6446.39
202010	5305.61
202011	5240.46

通义千问-14B生成

本月基础采样门店的销量为5240.46万支。近一年里，基础采样门店的月销量在2272.36-6446.39万支之间波动，整体表现较为波动，除了个别月份，比如2019年12月降幅明显。本月表现为低于正常水平。

讯飞星火生成

本月基础采样门店销量为5240.46万支，较均值4951.43稍高。近一年里，销量在2272.36至6446.39万支之间波动，整体表现为上升且波动较大。销量在2019年12月达到最小值，2020年09月达到最大值。

图 3 同样的数据下，Qwen-14B 和 Spark 的生成结果

基于评测方法的交叉评价模型，讯飞星火对通义千问 -14B 的生成进行评分，通义千问 -14B(Qwen-14B) 对讯飞星火 (Spark) 生成的评分，由于评测的数据集涉及实际的生产环境，因此无法对数据集进行纰漏，数据集大小为 50 条带有

相关数据表格的用户提问，在该数据集上基于构建的评价方法的得分如下，T_{method} 代表通义千问对生成摘要的评分，T_{Spark} 表达讯飞星火对生成摘要的评分，结果如表 2 所示（得分满分为 10 分）。

表 2 生成模型与评价模型的得分评价

生成模型	F_{exam}	T_{method}	T_{Spark}
Qwen-14B	0.998	7.54	7.52
Spark		7.53	8.5

对比表 2 数据可以发现，在使用评分模型时，通义千问 -14 B 和讯飞星火 Spark 在互为生成模型和评分模型时，它们的评分没有出现明显的差异，因此可以确定当给出具体的数据指标和数据时，带数据分析的摘要生成不需要模型进行二次的计算和排序，模型仅需要对文本内容进行浓缩和有机整合，因此此时带数据分析的统计摘要就被归约为简单的文本生成式摘要，仅考虑模型的文本序列建模能力。

可以发现生成模型的自评得分存在较为明显的差异，通义千问 -14 B 对自己的回答评分与讯飞星火 Spark 对其回答的评分相近，不存在明显的差别，但是讯飞星火 Spark 对自己的回答较为自信与评分模型给出的评估存在差异，所以可以认为产生该结果的可能性有两个。（1）通义千问 -14 B 由于模型参数量较少，对于文本建模的粒度不如讯飞星火小，因此无法进行更细粒度的注意。（2）模型对语料的评估存在偏好，对已知的或者分布近似的文本存在偏好归纳。另外，F_{exam} 为 0.998，十分近似 1，因此可以认为在本次实验中讯飞星火对通义千问的生成是不具备偏见的，但是其对自身的回答存在不一致的情况也十分值得进步深究。

3 结论与分析

本文提出的方法有区别于一般的摘要生成和自动统计报告，该方法的贡献归纳为几点。（1）提出了一种基于链式的数据分析流程，通过 LLM 对任务的理解实现代理任务的选取和编排，得到代理任务链（agent-chain）和执行链（execute-chain）。（2）将复杂的数据分析报告摘要生成拆解为了两个基础任务，数据统计与传统文本摘要归纳任务，在没有复杂训练和微调的情况下实现数据摘要报告的生成。

同时本文的方法也存在一些问题。（1）由于使用了简单的链式方式来进行数据处理，那么导致本文方法对数据样式具有比较大的要求，无法很好地理解用户提供的表格，具有一定的限制性。（2）仅能提供一些简单的数据分析和概括能力，没有办法同时给出一些基于私有领域或者知识背景的类专家回答。

4 未来展望

对于未来的发展，团队将致力于提升模型性能，将主要从几个方面来实现。（1）将代理任务链的构建从简单的脚本选取和组合，更换为具有智能代码生成能力的模型进行自动的适应数据模式的脚本生成。（2）通过 langchain 外挂知识库，或者使用图数据库的形式，将知识图谱融合到模型体系中，通过模型的自动数据分析报告摘要生成，并根据结果进行具有策略推荐的自动数据分析和策略分析的强摘要文本生成。（3）改进评估方法，通过盲评等方式来判断模型数据摘要的生成能力。

参考文献：

[1] MIHALCEA R,TARAU P.Textrank: bringing order into texts[EB/OL].[2023-08-26].https://aclanthology.org/W04-3252/.

[2] 朱玉佳 , 祝永志 , 董兆安 . 基于 TextRank 算法的联合打分文本摘要生成 [J]. 通信技术 ,2021,54(2):323-326.

[3] 黄菲菲 .BERT 的图模型文本摘要生成方法研究 [J]. 现代信息科技 , 2022,6(2):91-95+100.

[4] 周健 , 田萱 , 崔晓晖 . 基于改进 Sequence-to-Sequence 模型的文本摘要生成方法 [J]. 计算机工程与应用 , 2019, 55(1): 128-134.

[5] 党宏社 , 陶亚凡 , 张选德 . 基于混合注意力与强化学习的文本摘要生成 [J]. 计算机工程与应用 ,2020,56(1):185-190.

[6] 邹傲 , 郝文宁 , 靳大尉 , 等 . 基于语句融合和自监督训练的文本摘要生成模型 [J]. 模式识别与人工智能 , 2022, 35(5): 401-411.

[7] 汪先慈 . 基于全词注意力的文本摘要生成方法研究 [D]. 哈尔滨 : 哈尔滨工程大学 ,2021.

[8] ZHA L Y. TableGPT: Towards Unifying Tables, Nature Language and Commands into One GPT[EB/OL].(2023-07-17)[2023-09-01].https://arxiv.org/abs/2307.08674.

[9] WEI J, WANG X, SCHUURMANS D,et al.Chain of thought prompting elicits reasoning in large language models[EB/OL].(2022-01-28)[2023-09-02].https://arxiv.org/abs/2201.11903.

[10]CAI T.Large language models as tool makers[EB/OL].(2023-05-26)[2023-09-05].https://arxiv.org/abs/2305.17126.

[11]ZHENG L M. Judging LLM-as-a-judge with MT-bench and chatbot arena[EB/OL].(2023-06-29)[2023-08-29].https://arxiv.org/abs/2306.05685.

【作者简介】

郭利荣（1977—），男，广东汕头人，硕士，部门技术总监，助理工程师，研究方向：大数据、深度学习技术、大规模预训练语言模型（PLM）。

梁玉琪（1995—），女，广东肇庆人，本科，研究方向：大规模预训练语言模型（PLM）。

廖文亦（2001—），男，广东韶关人，本科，研究方向：大规模预训练语言模型（PLM）。

（收稿日期：2023-11-24）