摘 要

近年来，大语言模型的快速发展，基于大语言模型的智能体通过自然语言交互突破了传统程序的专业边界。大语言模型固有的不稳定性导致输出结果存在不可控偏差，为此盲目地增加并发的数量以提高任务的成功率又会造成计算资源的浪费。针对这一问题，本研究提出了一种基于神经网络的大语言模型智能体自评估模型，通过文本回归技术预测智能体执行的成功率，动态调整并发策略，实现计算资源的精确分配。研究设计了一个基于大语言模型的自然语言数据库查询智能体，解析用户输入的自然语言输入返回结果，智能体的执行过程记录执行日志，为自评估模型提供训练数据。在模型设计上，本研究结合了 BERT 模型的语义理解能力与神经网络回归层的数值预测优势，构建了一种混合架构的文本回归模型。该模型在预测自然语言查询成功率时平均偏差仅为 16%，动态并发机制相比固定并发策略可降低 10.5%的 API 调用成本，同时显著提升系统整体任务完成率。本研究将文本回归模型应用于大语言模型智能体的自评估任务，提出的动态并发控制机制在实现了系统可靠性和计算成本的平衡。

关键词：大语言模型，智能体，文本回归，BERT，神经网络

Abstract

In recent years, with the rapid development of big language models, intelligent agents based on big language models have broken through the professional boundaries of traditional programs through natural language interaction. The inherent instability of large language models leads to uncontrollable biases in the output results, and blindly increasing the number of concurrent tasks to improve the success rate can result in wastage of computing resources. In response to this issue, this study proposes a neural network-based large language model intelligent agent self-assessment model, which predicts the success rate of agent execution through text regression technology, dynamically adjusts concurrency strategies, and achieves precise allocation of computing resources. A natural language database query agent based on a large language model was designed, which parses user input in natural language and returns results. The execution process of the agent records execution logs, providing training data for the self-evaluation model. In terms of model design, this study combines the semantic understanding ability of BERT model with the numerical prediction advantage of neural network regression layer to construct a hybrid architecture text regression model. The average deviation of this model in predicting the success rate of natural language queries is only 16%. The dynamic concurrency mechanism can reduce API call costs by 10.5% compared to the fixed concurrency strategy, while significantly improving the overall task completion rate of the system. This study applies text regression models to the self-evaluation task of large language model agents, and proposes a dynamic concurrency control mechanism that balances system reliability and computational cost.

Keywords: Large Language Models, Agents, Text Regression, BERT, Neural Networks

目 录

1 引言

1.1 研究背景

近年来，人工智能技术的变革推动大语言模型(Large Language Models, LLMs)从文本生成工具向自主决策的智能体(Agent)形态演进。基大语言模型的智能体通过自然语言交互突破传统程序专业的边界的特性，正在重塑客服、教育、医疗等领域的服务模式。以 GPT 为代表的大语言模型，已经在文本生成、语义理解领域展现出了非常强大的能力。技术的发展让非专业的一些人士有了通过自然语言的途径与数据库进行交互的能力，深入了解复杂的查询语言或一些编程的技能就不那么的被需要了。

智能体的实际落地面临着双重的挑战，模型固有的幻觉(Hallucination)问题导致输出结果存在不可控偏差，为提升任务成功率而盲目增加并发智能体数量，造成计算资源浪费与碳排放激增。在如此大背景之下，动态评估任务解决概率并优化资源配置，构建具备自我认知能力的智能体系统，已成为一个十分重要的问题。

1.1.1 大语言模型智能体的应用趋势

近几年，以 GPT 为代表的大语言模型通过超大规模参数训练实现了突破性的进展。这些模型在文本生成、逻辑推理等方面展现出了一些接近人类水平的性能，催生了智能体概念的实质性进化。在工业界典型应用场景中，单个智能体已可完成代码调试、客户服务之类的专业任务，多智能体协作系统更在供应链优化、金融风险评估领域取得了初步的应用成果。不同于传统基于规则的系统，基于大语言模型的智能体能够通过自然语言交互自主解析任务目标，执行复杂决策链条[1]，使得软件定义智能体逐渐成为企业智能化转型的核心基础设施[2]。

1.1.2 大语言模型智能体的困境

大语言模型展现出强大的认知能力，但其固有的技术缺陷严重制约了它的实际应用价值。在开放域的问题处理当中，GPT-4 等顶尖模型仍存在幻觉性的输出，响应质量受上下文长度、提示词（prompt）设计等一些因素影响呈现显著的波动情况[3]。响应质量高度依赖于提示词的设计，即使是相同的输入，给用户使用带来了困扰，不同的提示词可能会导致模型生成截然不同的输出，需要用户具备一定的专业知识才能设计有效的提示词。大语言模型在处理开放域问题时，可能会生成与事实不符的答案，这种现象被称为“幻觉”，这种幻觉性的输出是由于模型过度拟合训练数据导致的，引发了人们对模型可靠性的担忧。上下文窗口长度有限，这意味着模型只能处理一定长度的文本，当处理长文本的时候，模型就会忽略或忘记之前的信息发生，导致响应质量下降。内部工作机制复杂，这给用户和开发者带来了些问题，难以解释其生成特定输出的原因，也无法对模型进行有效的控制，难以理解模型的决策过程[3]。

1.1.3 自评估模型构建的必要性

大语言模型普遍存在的这种不确定性，让单一智能体难以满足高可靠性场景需求，提高系统整体的可靠性，就迫使开发者采用多智能体冗余并发相关的技术路线。建立智能体自评估机制成为突破该困境的关键路径，当前普遍用的是种静态资源的配置策略是缺乏对任务难度的动态感知能力的。构建预测模型预判任务成功率，就可自主决策最优并发规模达到精确制导式的计算资源的分配的效果。将文本回归模型的技术应用于智能体能力评估领域，为构建具有自我认知能力的 AI 系统提供创新，是本研究的核心价值所在。

1.2 相关研究

这里我们将总结文本回归模型的概念，主要类型以及当前关于文本回归模型的已有的一些主要的研究成果。

1.2.1 基于传统机器学习方法的文本回归模型

线性回归作为最基础的回归方法，因其简单、高效且易于解释的特性，成为文本回归中常用的建模方法之一[6]。线性回归模型通过拟合文本特征自变量与目标变量因变量之间的线性关系来进行预测。其基本形式如下：

$$Y = X\beta + \epsilon \tag{1-1}$$

其中，Y是因变量，通常是一个连续型变量，X是自变量矩阵，表示如词频、句子长度等文本特征，β是回归系数，表示每个特征对目标变量的贡献，ϵ是误差项，表示模型无法解释的部分。线性回归模型的目标是通过最小化误差项的平方和来估计回归系数β。

多元线性回归是线性回归的一种形式。多元线性回归指的是多个自变量和一个因变量之间的线性关系建模。其基本形式如下：

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \tag{1-2}$$

其中，Y因变量，通常是连续的数值变量，表示我们想要预测或解释的量。在文本回归的上下文中，Y是某种基于文本内容的评分、分类标签的得分、文本的情感得分等。$X_1, X_2, \ldots, X_p$自变量，这些是用于预测因变量的输入变量。在文本回归中，这些自变量通常是从文本数据中提取的特征，例如词频、词袋模型中的词出现与否、词频-逆文档频率（TF-IDF）得分。β_0截距常数项，这是当所有自变量$X_1, X_2, \ldots, X_p$都为 0 时的Y的期望值。在文本回归中，如果所有特征都为零，即文本完全不包含任何预定的特征词或短语，β_0给出了Y的基线估计。$\beta_1, \beta_2, \ldots, \beta_p$回归系数，每个自变量$X_i$对应一个系数$\beta_i$，表示当 X_i变化一个单位时，因变量Y的平均变化量。在文本回归中，这些系数可以解释为特定文本特征对预测目标的影响大小。ϵ误差项，表示模型预测值与实际观测值之间的差异，它包括了模型未能解释的变异以及随机误差。

安康等人[4]提出的模型通过多元线性回归方法，从词汇、句子和文章三个维度提取了 8 个评估指标，并通过回归分析确定了各指标的权重。该模型从词汇、句子和文章三

个维度提取了8个评估指标，并通过多元线性回归拟合这些指标与文本难度之间的关系。其的模型公式为：

$$\mathrm{WSA}_{\mathrm{score}} = \alpha_1 I_{cw} + \alpha_2 I_{sy} + \alpha_3 I_{ASL} + \alpha_4 I_{sc} + \alpha_5 N_w + \alpha_6 I_{IE} + \alpha_7 I_{LD} + \beta \tag{1-3}$$

其中，I_{cw}表示词汇常用指数，I_{sy}表示音节难度因素，I_{ASL}表示平均句长指数，I_{sc}表示从句指数，N_w表示文章总词数，I_{IE}表示信息熵指数，I_{LD}表示逻辑难度指数，α_1到α_7是各指标的权重，β是常数项。这个基于多元线性回归的文本回归模型，在英语文本难度评估中表现出了较高的拟合度（R^2=0.96）通过了显著性检验（P 值<0.01）。Lasso 回归是一种线性回归的变体，通过引入 L1 正则化项来实现变量选择和模型稀疏性，自动选择重要的文本特征，让模型的预测的性能被提高了。

$$\widehat{\beta_{\mathrm{lasso}}} = \arg\min_{\beta} \left\{ |Y - X\beta|^2 + \lambda \sum_{j=1}^{P} |\beta_j| \right\} \tag{1-4}$$

其中，Y是因变量（目标变量），是一个连续型的变量，X是自变量矩阵，表示文本特征，β是回归系数，表示每个特征对目标变量的贡献，λ是正则化参数，控制模型的稀疏性，$|Y - X\beta|^2$是残差平方和，表示模型的拟合误差，$\sum_{j=1}^{P}|\beta_j|$是 L1 正则化项，用于惩罚回归系数的绝对值，实现变量选择，促使部分系数收缩为零。

Freo 和 Luati 等人[5]研究了基于 Lasso 回归的文本回归模型，通过商品描述文本预测商品价格。他们的研究基于两个实际案例，一个是通过员工满意度调查中的开放性问题预测整体满意度评分，一个是通过电商平台上的商品描述文本预测商品价格。Lasso 回归被用于从高维文本特征中选择对价格或者满意度评分有显著的影响的词汇、短语。他们使用了文档词项矩阵作为输入特征的矩阵 X，其中每一行表示一个文档，每一列表示一个词汇或短语的出现的频率。

Lasso 回归的核心优势在于其能够自动选择重要的特征，即非零系数的特征，同时将不重要的特征的系数收缩为零[7]。这使得 Lasso 回归特别适用于包括文本数据的高维数据，因为文本数据通常大部分特征值为零具有高维性和稀疏性。他们研究的结果是可以表明，Lasso 回归在商品价格预测中表现出色，能够有效选择与价格相关的词汇，并剔除不相关的词汇。

岭回归（Ridge Regression）作为一种经典的回归的方法，引入正则化项，有效防止模型过拟合，提升模型的泛化能力[10]。这种方法其实是一种改进的最小二乘法，在损失函数中加入 L2 正则化项，防止这个模型过拟合，其目标是最小化以下损失函数：

$$\min_{\alpha} \sum_{i=1}^{m} (x_i \alpha - y_i)^2 + \lambda \sum_{i=1}^{m} \alpha^2 \tag{1-5}$$

其中，x_i是输入特征，y_i是目标变量，α是回归系数，λ是正则化参数。正则化项$\lambda \sum_{i=1}^{m} \alpha^2$通过惩罚较大的回归系数，防止模型过拟合。在文本回归任务中，输入特征x_i

就是通过文本特征提取的方法得到的数值向量。岭回归模型通过最小化上面的损失函数，有效地从文本数据中学习到回归系数来预测目标变量。

Onita 等人[8]的论文采用模型一种结合了岭回归和分类的模型进行文本自动摘要，通过核技巧（kernel trick）将数据映射到高维空间，能处理非线性的关系。选择核岭回归（KRR）模型适合于数据量有限的情况，处理文本数据的时候不需要大量的数据和计算资源，在数据量有限的情况下表现出色，能够避免深度学习模型可能出现的重复词汇和不相关词汇的问题，生成的摘要与原始文本在词汇上有较高的相似度。实验的结果表明了，核岭回归模型在生成文本摘要时表现更好，更准确地捕捉文本的核心内容，在文本处理任务中，传统的机器学习方法可能比复杂的深度学习模型更具优势。

Liu 研究了[9]如何利用岭回归模型对景区数据进行综合的评价，把评论文本分为了设施、卫生、服务、位置和性价比五个类别，进行针对性的分析。对评论数据进行了预处理提高数据质量，包括去重、去除英文文本、繁体转简体、文本纠错和词语压缩。他们搭建了基于岭回归和 k 折交叉验证建立综合评价模型，计算出了每个景区和酒店在五个方面的总得分，并使用均方误差（MSE）、均方根误差（RMSE）和平均绝对误差（MAE）验证模型。结果表明，岭回归模型的预测精度较高，可有效地预测景区和酒店的综合得分，岭回归模型可以有效地用于文本的综合评价和特征分析。

逻辑回归是一种广泛应用于分类问题的统计分析方法，主要用于二分类或多分类任务[15]。核心的思想就是通过逻辑函数（Sigmoid）把线性回归的输出结果映射到(0,1)区间，达到对分类结果的概率预测的目的，数学表达式如下：

$$P(Y=1|X)=\frac{1}{1+e^{-(\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_nX_n)}} \tag{1-6}$$

其中，$P(Y=1|X)$表示在给定特征向量 X 的条件下，样本属于类别 1 的概率。β_i是模型参数需要通过训练数据进行估计，X_i是特征变量。通过最大似然估计法（MLE）来估计参数 β，促使模型可最大化地拟合用来给其训练的数据，在文本分析中通常是用来判断文本的情感倾向的。

Wen 等人[11]通过逻辑回归模型对文本情感进行分析，使用 TF-IDF 进行的特征提取工作，结合逻辑回归与贝叶斯模型进行情感分类。逻辑回归模型在文本情感分析中表现优于贝叶斯模型，逻辑回归模型在处理文本情感分析任务时具有较高的准确性和适用性，能够有效地区分文本中的情感倾向。

Ginting 等人[12]的论文讨论了如何使用多项逻辑回归方法在 Twitter 上检测仇恨言论，使用 TF-IDF 方法提取特征，通过收集的推文的数据，采用多项逻辑回归进行分类，在仇恨言论检测中取得了较好的效果，平均的精确度为 80.02%，召回率为 82%，准确率为 87.68%。在数据分区测试中，90%训练数据和 10%测试数据的组合取得了最佳性能，准确率达到 95.90%。证明了这是一种有效的仇恨言论检测方法，能够准确地识别出包含仇恨信息倾向言论的文本。

陈雪婷的研究[14]中构建了一个包含 174 个段落、348 篇文章、22.11 万字的 HSK 中高级阅读文本语料库，并从汉字、词汇、句法三个层面选取了 34 个语言特征指标。通过 Spearman 相关系数分析这些特征指标与 HSK 难度等级的相关性，使用多元有序 Logistic 回归模型构建了可读性的公式。句法层面的影响相对较弱，词汇层面的特征对文本难度等级的影响最为显著，较为准确地预测文本的难度等级，预测准确率达到 81.42%，为 HSK 阅读文本的难度评估提供了一种有效的量化方法。

Brzezinski 等人[13]探讨了逻辑回归在信息检索中的应用，在文档分类任务中的表现。研究使用了 TIPSTER 数据集中的 150 个主题进行实验，开发了针对高维稀疏数据集的逻辑回归模型构建方法。在多分类任务中，刚刚开始预测准确率较低，随着变量的增加也有所提高，最终达到了 82%。在二分类任务中，逻辑回归模型的预测的准确率是随着变量的增加逐渐提高，最终达到 99.33%。逻辑回归模型在特征选择和降维方面表现出色，逻辑回归在处理高维稀疏数据集时具有显著优势，优于神经网络等其他机器学习算法，是一种有效的监督学习算法。

1.2.2 基于深度学习方法的文本回归模型

卷积神经网络（卷积神经网络）在文本回归任务中的应用主要依赖于其强大的特征提取能力。卷积神经网络通过卷积层和池化层从文本中提取局部特征，通过全连接层将这些特征映射到连续的回归目标值[18]。文本回归任务的目标是给定一段文本，预测一个与之相关的连续的值，如情感强度、文本难度。基于卷积神经网络的文本回归模型在文本回归任务中具有广泛的应用前景。

我们假设输入的文本表示为$X = [x_1, x_2, \dots, x_n]$其中$x_i$是第$i$个词的词向量，$n$是文本的长度。卷积层通过卷积核$W \in R^{k \times d}$对输入文本进行卷积操作，提取局部特征。卷积核的大小为$k \times d$，其中 k 是卷积核的宽度（即覆盖的词数），d 是词向量的维度。那么卷积操作就可以表示为：

$$c_i = f(W \cdot X_{i:i+k-1} + b) \tag{1-7}$$

其中$X_{i:i+k-1}$表示从第i个词到第$i + k - 1$个词的子序列，b是偏置项，f是激活函数，例如 ReLU。

池化层（通常是最大池化）用于降低特征维度，提取最重要的特征。

$$p_j = \max\left(c_{j \times s}, c_{j \times s+1}, \dots, c_{j \times s+s-1}\right) \tag{1-8}$$

其中 S 是池化窗口的大小。

池化后的特征通过全连接层映射到回归目标值：

$$y = W_f \cdot p + b_f \tag{1-9}$$

其中W_f是全连接层的权重矩阵，b_f是偏置项，y是预测的回归值。

Tao Li 等人在其研究[16]中提出了一种基于条件生成对抗网络（GAN）的文本回归模型。生成器使用长短期记忆递归神经网络（LSTM）生成文本，判别器使用卷积神经网络对生成的文本进行回归预测，结合了卷积神经网络和长短期记忆递归神经网络，用于

从文本中提取特征并进行回归预测。模型在多个文本回归任务电影票房预测和流行病监测上表现非常之优异。这个模型的优势在于能够处理不平衡数据集，在有限的标注数据下表现出色。

李文彪等人[17]提出了一种基于卷积神经网络（CNN）和长短期记忆递归神经网络（LSTM）的文本难度分级模型。该模型通过变长卷积层和块结构提取文本的局部特征，通过双向长短期记忆递归神经网络提取篇章级的那些特征。模型在文本难度分级的一些任务上取得了较好的效果，准确率分别达到了 75.4%和 56.1%。十分成功地结合了卷积神经网络和长短期记忆递归神经网络的优点，提高了文本难度的分级的准确性，可做到同时捕捉文本的局部和全局特征。

Transformer 模型自 2017 年由 Vaswani 等人[19]提出以来，迅速成为自然语言处理（NLP）领域的核心架构，核心思想是通过自注意力机制（Self-Attention）捕捉文本中的全局依赖关系，避免了传统循环神经网络（RNN）和卷积神经网络在处理长距离依赖时的一些存在的局限性。模型通过多头注意力机制和前馈神经网络（FNN）堆叠而成，可以有效地处理文本序列中的复杂关系。在文本回归的有关任务中，这个模型用于从输入文本中提取特征，通过回归头输出连续的数值预测。其基本公式如下：

自注意力机制：

$$\text{Attention}(Q,K,V)=\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1-10}$$

其中，Q,K,V分别表示查询（Query）、键（Key）和值（Value）矩阵，d_k是键向量的维度。

多头注意力机制：

$$\text{MultiHead}(Q,K,V)=\text{Concat}(\text{head}_1,\text{head}_2,\dots,\text{head}_h)W^O \tag{1-11}$$

其中，每个$head_i$是通过自注意力机制计算得到的，W^O是输出权重矩阵。

Bu 等人[20]提出了一种基于 Transformer 的文本检测模型，旨在结合分割和回归两种方法的优势，让文本检测的鲁棒性和效率提高。现有的文本检测方法主要分为基于分割和基于回归两大类。分割方法在处理文本字体变化时具有更强的鲁棒性，可是需要复杂的后处理步骤。研究针对现有文本检测方法中存在的问题，分割方法需要复杂的后处理且计算开销大，回归方法在面对复杂场景时鲁棒性不足，提出了一种统一的框架下的模型架构，采用了 DETR 架构，通过在解码器中设置分割和回归两个模块，将分割模块应用于解码器的前几层，给后续的回归模块提供位置方面的先验。他们的结果说明，与现有的分割和回归方法相比在训练效率、数据利用效率、跨数据域方面表现出了极其显著的优势。

Layth Rafea Hazim 等人的研究中[21]，探讨了基于 Transformer 架构的模型（Sentence-BERT 和 RoBERTa）在文本分类任务中的性能表现。比较逻辑回归和前馈神经网络（FNN）

结合这些 Transformer 模型嵌入的时候，对文本真实性和 AI 生成内容的分类能力。包括文本的分词（tokenization）和归一化（normalization），将文本转换为适合 Transformer 模型输入的形式，通过去除标点符号手段降低了数据的噪声。研究使用了一个平衡的数据集，确保模型能够公平地学习和评估，其中包含人类书写的文本和 AI 生成的文本。使用 Sentence-BERT（SBERT）和 RoBERTa 两种 Transformer 模型把文本数据转换为高维度的嵌入向量，有效捕捉文本的语义特征。前馈神经网络（FNN）设计用于捕捉数据中的复杂模式，包含多个隐藏层，每层由线性变换和非线性激活（ReLU）组成。用准确率（accuracy）、精确率（precision）、召回率（recall）这些指标来衡量模型的表现性能，用了混淆矩阵（confusion matrix）直观地展示了模型的分类的结果。

前馈神经网络：

$$\mathrm{FFN}(x) = \mathrm{ReLU}(xW_1 + b_1)W_2 + b_2 \tag{1-12}$$

其中，W_1、W_2是权重矩阵，b_1、b_2是偏置项。

回归头：

$$y = \mathrm{RegressionHead}(h) \tag{1-13}$$

其中，h是 Transformer 模型的输出特征，y是回归任务的预测值。

研究表明，RoBERTa 嵌入结合 FNN 模型在区分 AI 生成文本和人类书写文本的任务中表现最为的出色，它的高准确率证明了 RoBERTa 在细粒度语义表示方面是有效的。

1.2.3 相关研究总结

传统机器学习方法在文本回归任务中具有简单高效、易于解释的特点，是适用于数据量有限或特征维度较高的场景。线性回归模型是通过拟合文本特征与目标变量之间的线性关系快速地进行预测。岭回归加上 L2 正则化项防止过拟合，岭回归适合于数据量有限的一些文本回归任务。Lasso 回归引入 L1 正则化项，自动选择重要的文本特征，有效地处理高维稀疏的数据。逻辑回归主要可用来处理分类相关任务，不过逻辑回归处理高维稀疏数据集时具有显著优势，可有效进行特征选择和降维，文本情感分析任务中表现是出色的。

深度学习方法通过其强大的特征提取能力和复杂的模型结构，处理复杂的文本数据，在处理长距离依赖和非线性关系的方面具有很大的优势。Transformer 得益于自注意力机制捕捉文本中的全局的依赖关系，逃离了传统方法在处理长距离依赖时的局限性。卷积神经网络使用卷积层和池化层提取文本的局部特征，经过全连接层进行回归预测。

之后的研究，我们可以结合传统机器学习方法和深度学习方法的优势，研究如何利用预训练语言模型，如 BERT，进行文本回归任务，这样就提高模型的性能和可扩展性，开发一种混合模型以提高具体场景下的文本回归的性能。

2 模型与方法

2.1 智能体设计

我们设计了一种基于大型语言模型的自然语言数据库查询智能体，作为被模型评估的智能体，核心功能是接收自然语言查询请求，通过大语言模型生成可执行代码返回数据库操作结果。用了一种融合异常处理、断言验证和多线程技术策略，解决执行效率与复杂查询理解方面的问题，增强了整个智能体的鲁棒性以及响应的速度。智能体的执行成功率即是正确生成并执行代码的概率，是本文构建的自评估模型的预测目标，能够处理自然语言查询执行结构化数据库操作，智能生成统计图表。

智能体的每次查询处理过程都会产生结构化日志，包括输入的自然语言文本、生成的代码以及执行状态是成功还是失败的关键信息字段，构成了文本回归模型的训练的数据集。采集智能体在多样化处理任务中的执行结果记录，建立起了查询文本特征与执行结果之间的映射关系，为文本回归预测模型提供了可靠的监督信号。

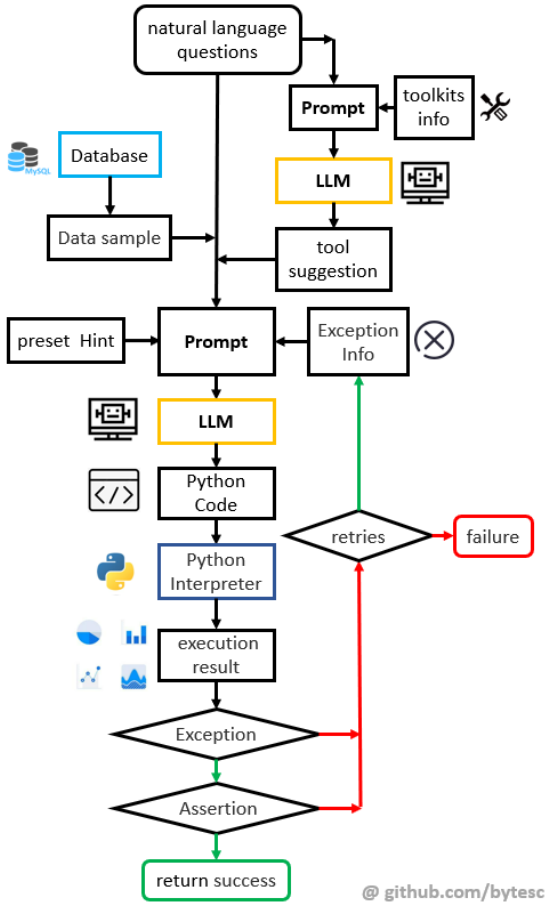

图 2-1 智能体单次工作的基本流程图

2.1.1 信息提取

如图 2-1，用户通过前端界面输入他们想要查询的问题，关于数据的查询、统计计算或者数据可视化需求。系统会有一个预设的工具集，包括数据查询工具、统计计算工具与图表绘制的工具，工具集的描述信息会帮助大语言模型理解用户可能需要哪些工具来解决提出的问题。大语言模型接收到用户的问题和工具集描述信息后，会选择合适的工具来解决问题。用户在前端界面输入自然语言问题，大语言模型接收到问题后，会理解问题的意图，根据工具集描述信息选择合适的工具建议，用于执行后续的一些流程。

将给大语言模型的问题分解为一个个小步骤，有利于充分发挥大语言模型的特点和优势，避开其在长内容处理方面的劣势。大语言模型具有较强的自然语言理解能力，分一个个步骤的处理方式让大语言模型逐步引导至最终的合适的解决方案，，准确把握用户问题的意图，避免了一次性处理复杂问题可能导致的错误提高系统的可解释性和可维护性，方便开发者对系统进行优化和调整。

把数据库结构信息以合适的方式提供给大语言模型是其能够针对性的输出 SQL 查询代码的必要条件。智能体通过用户提供的凭证用户名、密码、服务器地址、数据库名信息，建立与 MySQL 数据库的会话连接。执行 SQL 查询语句，读取所有表的前五行数据和表结构，获取数据库的结构信息和数据摘要。分析表的前五行数据，生成数据摘要让大语言模型理解数据的类型、范围、潜在的查询需求，对收集到的表结构进行信息的整合工作，给后续的 SQL 查询代码生成提供必要的背景知识和元数据。

2.1.2 代码的生成与执行

大语言模型使用获取到的数据摘要和工具建议信息，编写 Python 代码来解决用户提出的各种问题，包括数据处理逻辑和一些图表绘制命令，是把自然语言问题转换为机器可以理解的语言的关键。

编写好的 Python 代码被传递给 Python 解释器执行。如果代码执行成功，系统会将结果展示给用户，如果执行失败智能体会捕获异常信息。异常处理机制保证了系统的健壮性，使其在面对错误的时候能够做到自我修复的可能性，提高用户体验。如果代码执行出现异常，系统会将异常信息和问题代码组成新的提示词，回输给大语言模型再次尝试解决问题，重复直到代码成功执行或者是超过最大的重试次数，给大语言模型理解代码失败的原因的机会，来针对性的做出修正的动作。如果代码没有运行出现异常，系统会对程序输出进行断言验证。如果不是期望的类型，系统会将断言信息和问题代码组成新的提示词，回输给大语言模型再次尝试成为了保证数据分析结果准确性的关键一步，避免了错误数据对用户决策的影响。最终把自然语言问题转换为机器的操作，一旦代码成功执行，系统会将结果展示到用户界面上，用户的问题就这样最终被智能体解答了。

2.1.3 并发机制

大语言模型的不稳定性本质上，从模型原理的角度来看，概率生成机制与自回归推理特性共同作用的结果。大语言模型基于 Transformer 架构，通过注意力机制建模上下

文依赖关系，输出是通过对词表空间的概率分布进行采样生成的具有内在的随机性，模型在每个时间步都是基于当前状态和先前生成的 token 序列，无论是贪婪搜索、束搜索、随机采样的方法都是从概率分布中选择下一个 token。作为基于统计学习的概率模型，每一步输出都是对词表空间的条件采样而非确定性选择，采用相同的提示词，细微的概率波动也可能因 Top-p 或温度系数的采样策略导致截然不同的一些生成结果的路径。模型的参数化知识是通过统计学习获得的，对复杂逻辑的推理是近似计算而非确定性推导本质上的概率特性决定了输出必然存在不稳定性。多线程并发通过引入并行独立采样路径，利用统计学中的独立事件原理来提高至少一条路径成功的概率，是应对这种内在不确定性的有效工程方案。

在基于大语言模型的智能体当中，处理长提示词是一个非常麻烦的挑战。由于语言模型对上下文的高度敏感特点，初始生成由于概率的微小偏差会在自回归过程中通过误差累积被不断的放大形成雪球效应，长距离依赖问题使得模型在处理长提示时可能出现注意力分散关键的信息就这样被稀释了。正如上节中提到的，当大语言模型接收到用户查询并尝试生成代码时，如果生成的结果不符合预期或者是出现一些异常的情况，系统会将错误信息和代码重新输入大语言模型请求再次尝试。这个过程会重复多次导致提示词的长度不断增加，长提示词会使大语言模型需要同时处理更多的信息，处理起来更加困难，导致大语言模型失去注意力无法有效地处理所有的信息影响生成效果。大语言模型的输出是相互依赖的，如果第一次的回答有误会影响后续的生成过程，从头开始一个新的查询过程往往可能会比继续在错误的基础上修正是更加的有效的方法。

为了解决这些问题我们引入了多线程并发执行的方法，同时启动多个独立的查询过程，每个过程都是独立的有每个自己的提示词和大语言模型生成的代码。在执行任务阶段，每个线程将尝试完成其分配的单次生成任务，如某个线程成功完成了任务并返回了结果，将立即将这个结果响应给用户，用户不需要等待其他查询完成就可以得到响应，提高了系统的整体稳定性响应速度。并发执行多个查询，某个路径因为大语言模型的输出固有不稳定而失败了的情况下，其他的可能仍然能够成功，降低了整体失败的一个风险。

2.2 数据集准备

2.2.1 数据集获取

训练数据获取的流程如图 2-2 所示，我们采用了大语言模型，借鉴了强化学习和知识蒸馏的思想方法，以 qwen1.5-110b-chat 模型以及 MySQL 官方示例数据库 world 为测试示例，生成查询问题。我们从数据库中提取了表结构、字段类型、关系约束关键元数据，为大语言模型提供了理解和生成查询语句的必要信息，在接收这些结构化数据之后，依据数据库的结构，生成了自然语言形式的查询问题。

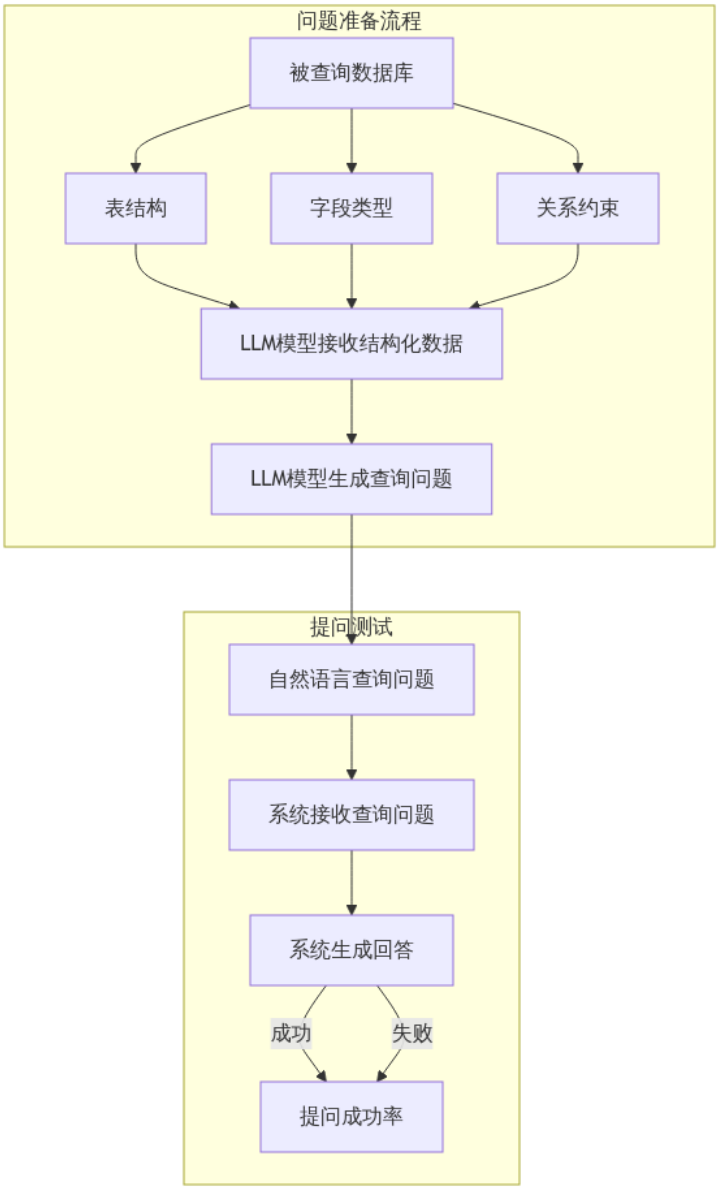

图 2-2 数据集生成基本流程图

在提问测试环节当中，接收这些自然语言查询问题后，开始生成相应的反应。根据回答的结果，如果回答失败，则记录失败次数，如果回答是成功的，则记录成功次数，进行分类处理。统计成功和失败的次数，计算出提问的成功率，结构化的处理方式，给后续的模型训练提供了重要的数据支持。

2.2.2 数据预处理

在数据准备阶段，我们对这些查询问题和标签进行了标准化处理，确保数据的质量和一致性，把查询问题的日志记录作为输入，每个问题的成功率作为输出，给模型的训练提供了所需要的输入输出对。

我们按照 8:2 的比例将数据集划分为训练集和验证集，训练集用于模型的参数学习，验证集用了来评估模型的泛化能力。用了批处理和随机打乱数据的技术，提高模型的训练效率和防止过拟合，给模型的训练提供高质量的输入数据，让后续的模型优化和性能

评估有了基础，模型能够有效地从数据中学习到执行成功率的回归预测模式。

2.3 模型设计

我们设计的模型结构是一个结合了 BERT(Bidirectional Encoder Representations from Transformers)模型和神经网络回归层的复合模型，如图 2-3。核心部分包括两个主要组件，BERT 模型和回归层。

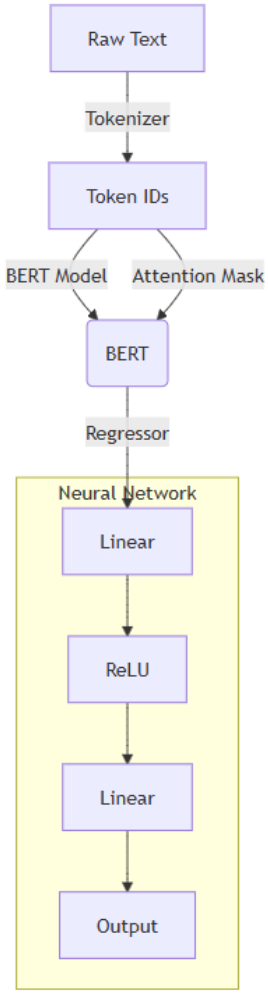

图 2-3 基于 BERT 模型和神经网络的文本回归模型结构图

BERT 模型作为基础，是一种基于 Transformer 架构的预训练语言模型，捕捉文本中的深层语义信息，通过其多层双向注意力机制理解上下文中的词汇关系，生成富含语义的文本的向量化的表示，对于后续的回归任务很重要的，给模型提供了极其丰富的特征空间，让预测变得更准确。通过多层双向 Transformer 编码器生成文本的表示，假设输入文本经过分词后得到的 token 序列为$X = [x_1, x_2, \ldots, x_n]$，其中 n 是序列长度，那么输出可以表示为：

$$H = \mathrm{BERT}(X) \tag{2-1}$$

其中，$H \in R^{n \times d}$是 BERT 模型的输出表示，d是 BERT 模型的隐藏层维度。将输入

token 序列X映射为嵌入向量$E \in R^{n\times d}$。为每个 token 添加位置编码$P \in R^{n\times d}$，以保留序列的位置信息。通过多层 Transformer 编码器计算上下文相关的表示，每层的计算可以表示为：

$$H^{(l)} = \text{TransformerLayer}\left(H^{(l-1)}\right) \tag{2-2}$$

其中，$H^{(l)}$是第$\boldsymbol{l}$层的输出，$H^{(0)} = E + P$是初始输入。Transformer 层的核心是多头自注意力机制和前馈神经网络。

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2-3}$$

其中，$\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$分别是查询（Query）、键（Key）和值（Value）矩阵，d_k是注意力头的维度。

$$\text{FNN}(x) = \text{ReLU}(W_1 x + b_1)W_2 + b_2 \tag{2-4}$$

其中，$\boldsymbol{W_1}$、$\boldsymbol{W_2}$是权重矩阵，$\boldsymbol{b_1}$、$\boldsymbol{b_2}$是偏置向量。使用池化层（pooler）来获取整个序列的表示，输出可以表示为：

$$h_{\text{po}} = \text{Pooler}(H) \tag{2-5}$$

其中，$h_{\text{po}} \in R^d$是池化后的向量表示。将 BERT 输出的序列表示$H \in R^{n\times d}$转换为一个固定长度的向量表示$h_{\text{p}} \in R^d$。Pooler 层会使用 BERT 输出的第一个 token（即[CLS] token）的表示，并通过一个全连接层和 tanh 激活函数进行变换：

$$h_{\text{p}} = \tanh\left(W_{\text{po}} h_{[\text{CS}} + b_{\text{po}}\right) \tag{2-6}$$

其中，$(h_{[\text{CS}} \in R^d)$是 BERT 输出的[CLS] token 的表示，$W_{\text{po}} \in R^{d\times d}$和$b_{\text{po}} \in R^d$是 Pooler 层的权重矩阵和偏置向量。

本模型采用 bert-base-multilingual-uncased 版本作为基础，其具体配置包含 12 层 Transformer 编码器，每层隐藏维度 d=768，一共 12 个注意力头，前馈网络中间层的维度为 3072。

在 BERT 模型的之后，我们连接了一个回归层将 BERT 输出的高维特征向量映射到一个连续的值代表查询成功率。回归层是由两个全连接层和一个激活层组成的，第一个全连接层负责将 BERT 输出的特征向量从较高的维度降低到一个中等维度，减少模型复杂性和计算成本。在后面用了一个非线性激活函数 ReLU，让模型能够捕捉到特征与特征之间的非线性关系。第二个全连接层则将中等维度的特征向量映射到了一个单一的输出值，输出一个表述处理成功的概率的连续的预测值。

在回归层设计中，两个全连接层均采用 Xavier 均匀初始化方法进行参数初始化，有效保持各层间梯度方差的一致性。在第一个全连接层后设置了 dropout 机制，丢弃率设定为 0.1 防止过拟合的情况容易出现，保持模型特征表达能力的同时确保了训练过程的稳定性。实验表明了，768 维的隐藏层表示与 256 维的中间层维度的组合，能在模型复杂度和预测的精度之间取得到良好平衡。

$$h_1 = \mathrm{ReLU}\left(W_1 h_{\mathrm{po}} + b_1\right) \tag{2-7}$$

$$y = W_2 h_1 + b_2 \tag{2-8}$$

其中，$W_1 \in R^{256\times d}$和$b_1 \in R^{256}$是第一个全连接层的权重矩阵和偏置向量。$W_2 \in R^{1\times 256}$和$b_2 \in R^{1}$是第二个全连接层的权重矩阵和偏置向量。$h_1 \in R^{256}$是第一个全连接层的输出。$y \in R$是最终的预测值，表示查询成功的概率。ReLU（Rectified Linear Unit）是一种常用的非线性激活函数，定义为：

$$\mathrm{ReLU}(x) = \max(0, x) \tag{2-9}$$

整个模型结构的设计，通过结合 BERT 模型的强大语义处理功能和神经网络的回归层的回归预测能力，构建一个能理解复杂自然语言结构进行准确数值预测的组合模型。

2.4 模型训练

在模型训练部分，如图 2-4 所示，我们初始化了 BERT 模型和分词器，使用 BERT 模型提取文本特征，通过一个回归器进行预测。

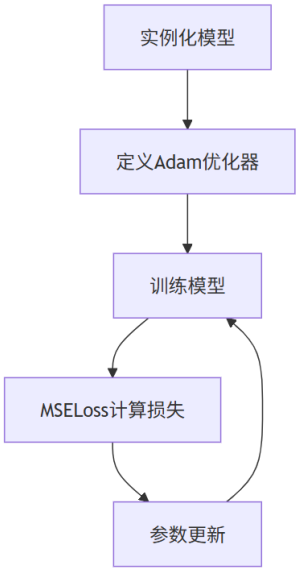

图 2-4 模型训练基本流程图

我们采用了均方误差损失函数（MSELoss）来衡量模型预测值与实际值之间的差异，对于优化过程有着连续可导的优势，有效地反映回归任务中的预测误差，让模型在训练过程中可以做到更好地收敛。假设真实值为y_{true}，预测值为y，则损失函数可以表示为：

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(y_{\mathrm{true}}^{(i)} - y^{(i)}\right)^2 \tag{2-10}$$

其中，N是样本数量，$y_{\mathrm{true}}^{(i)}$和$y^{(i)}$分别是第i个样本的真实值和预测值。

我们选择了 Adam 优化器更新模型参数，结合了动量和自适应学习率的特点，在不同数据分布下保持较高的优化效率，合适处理复杂的自然语言处理任务。参数更新公式

为：

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta \mathcal{L} \tag{2-11}$$

其中，θ_t是模型在时间步t的参数。η是学习率。$\nabla_\theta \mathcal{L}$是损失函数对参数的梯度。

输入文本到最终预测值的整个计算过程：

$$y = W_2 \cdot \mathrm{ReLU}\big(W_1 \cdot \mathrm{Pooler}\big(\mathrm{BERT}(X)\big) + b_1\big) + b_2 \tag{2-12}$$

在模型训练训练过程中，加载数据集，并将数据集划分为训练集、验证集和测试集，其中训练集用于模型参数的更新，验证集用于调整模型超参数和避免过拟合，测试集用于评估模型性能。定义损失函数和优化器，我们采用均方误差损失函数（MSELoss）和 Adam 优化器进行模型训练。设置训练参数学习率、训练轮数。每个批次的数据进行前向传播、计算损失、反向传播、参数更新，计算训练集上的损失、准确率和平均绝对误差，实时地监控训练过程。在验证集上评估模型性能，记录验证集上的准确率和平均绝对误差，关键指标损失、准确率、平均绝对误差写入日志文件，可以后续分析。

3 研究结果

3.1 模型调优结果

经过模型训练和调优，我们的基于 BERT 模型和神经网络的文本回归模型在预测自然语言数据库查询成功率方面展现出了优异的性能。我们对优化器的学习率进行了调整提高模型的性能，提高了文本回归模型在自然语言数据库查询系统中的对成功率的预测精度。将学习率范围设定在 1e-5 至 1e-8 之间寻找最佳的学习率配置。当学习率设置为 1e-7 时（如图 3-5），模型在验证集上的表现达到最佳，平均预测偏差稳定在 16%左右具有较高的预测准确性。在学习率为 1e-7 的条件下，模型的预测偏差始终保持在 20%以下满足了项目预期的精度要求。1e-7 的学习率在本次研究中被证实为最佳选择，为项目目标的实现奠定了基本的条件。当学习率过低（如图 3-7）时，模型收敛过慢模型容易陷入局部最优导致预测精度的下降情况发生。当学习率过高（如图 3-1）时，模型训练过程波动较大难以收敛。

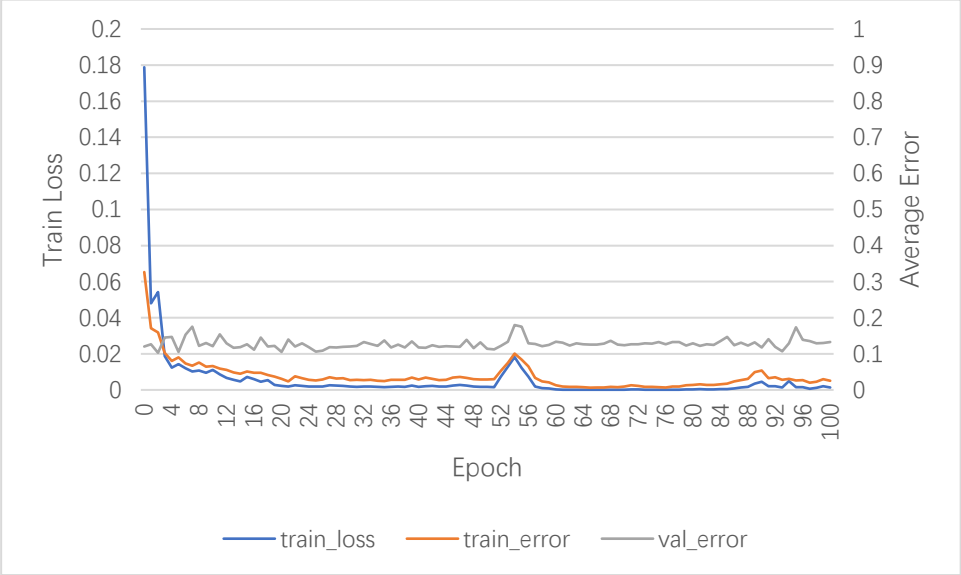

图 3-1 训练结果图（lr=1e-5）

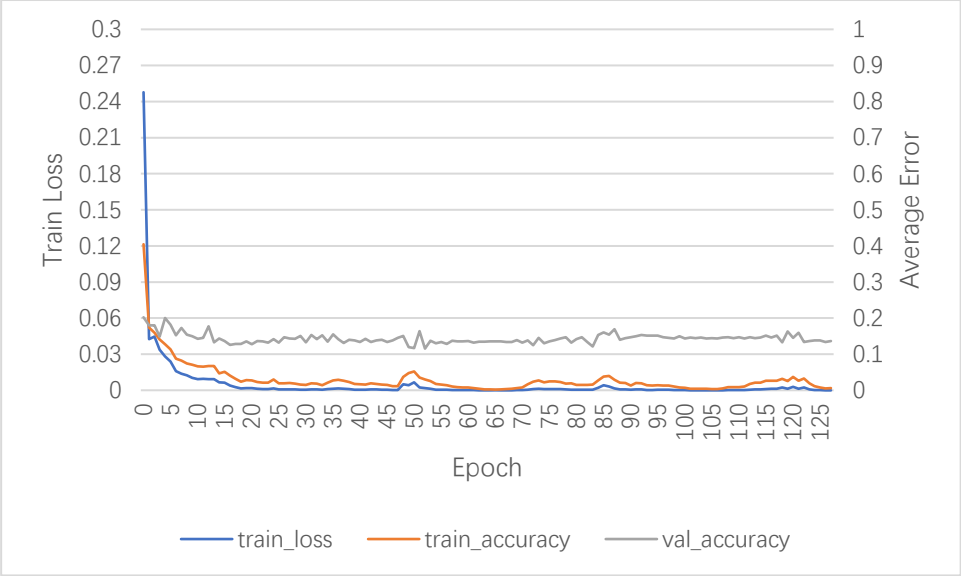

图 3-2 训练结果图（lr=5e-6）

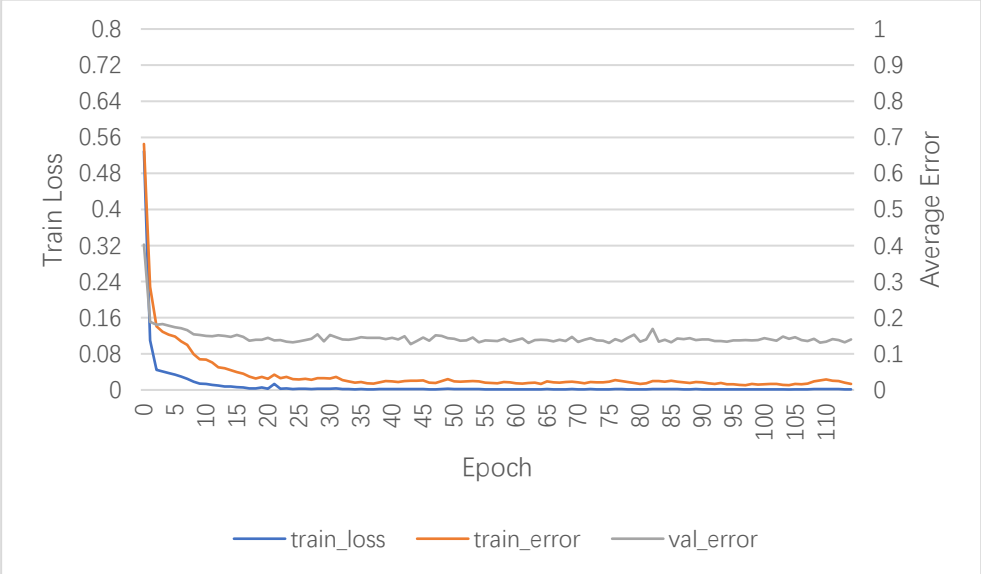

图 3-3 训练结果图（lr=1e-6）

图 3-1 至图 3-3 分别展示了学习率设置为 1e-5、5e-6 和 1e-6 时的训练曲线。快速收敛的显现给后续学习率需要显著降低学习率来获得更稳定的优化路径提供了依据。三实验呈现出了一些显著的共性特征，模型在约前 10 个训练周期内即实现快速收敛，训练损失值呈现断崖式的下降。验证集指标同步发生了迅速提升，在达到峰值后很快进入平台期，暗示了模型可能陷入局部最优状态的潜在问题。过早收敛损失曲线在中期出现明显波动，表明过于高的学习率导致参数更新步长过大，模型在最优解附近震荡。

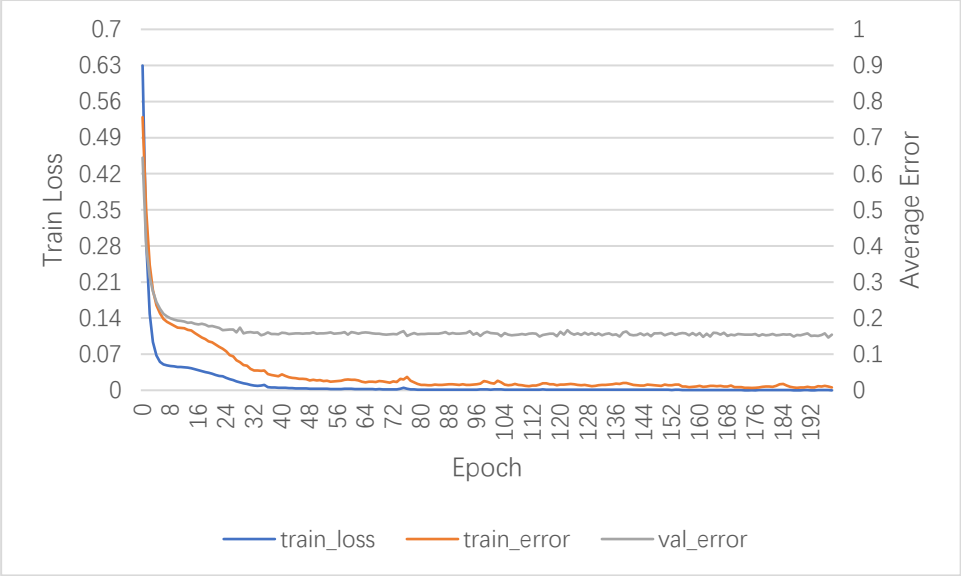

图 3-4 训练结果图（lr=5e-7）

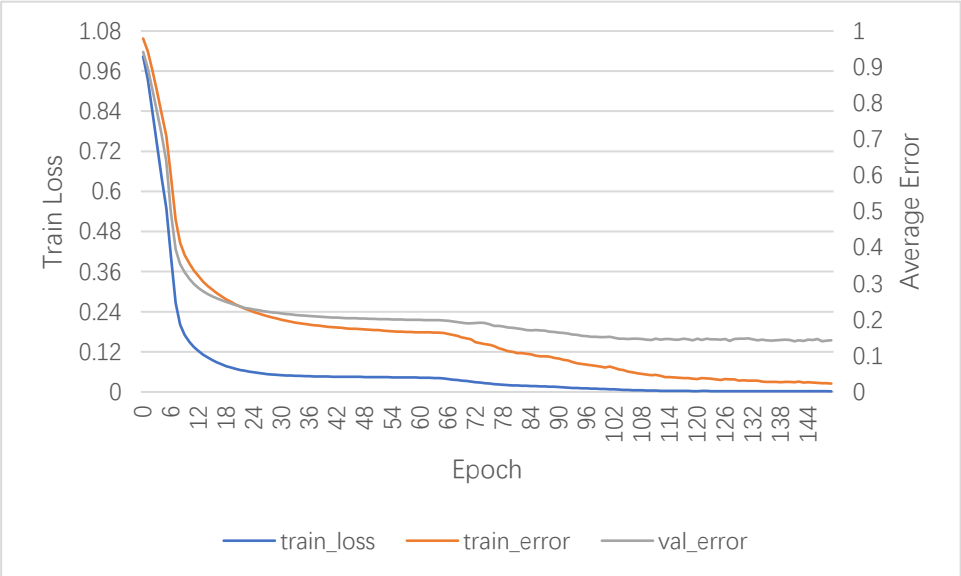

图 3-5 训练结果图（lr=1e-7）

学习率降至 5e-7（图 3-4）和 1e-7（图 3-5）区间时，模型展现出理想的训练动态，学习率较为合适。图 3-4 显示，训练损失呈现出了一个平滑的指数衰减趋势，大约在 25-30 个周期达到稳定的状态，验证集的指标的提升轨迹与训练集保持了高度的同步。图 3-5 的表现，验证集上的预测偏差最终稳定在 16%的基准线附近，全程未出现超过 20%的异常波动。如图 3-5，学习率等于 1e-7 的情况在保持收敛速度的同时，实现了更紧密的误差控制带。

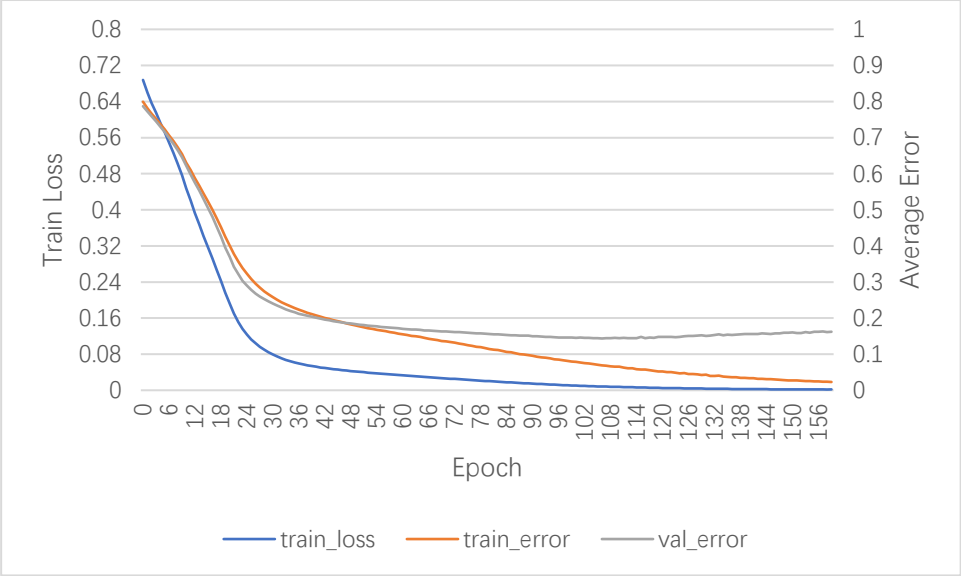

图 3-6 训练结果图（lr=5e-8）

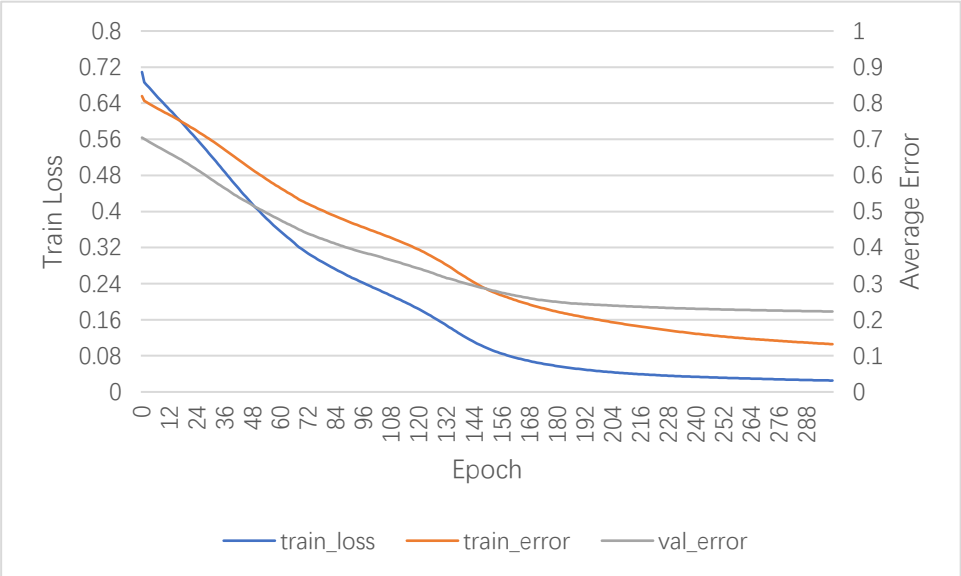

图 3-7 训练结果图（lr=1e-8）

将学习率下调至 5e-8（图 3-6）和 1e-8（图 3-7）后，模型表现出明显的优化效率下降，学习率是明显过于的小了。图 3-6 中，最终能达到与中学习率组相近的性能水平，可是需要约两倍的训练周期才完成合理的收敛。图 3-7 在 50 个训练周期后，模型仍没有完全收敛揭示了更极端的状况，验证损失曲线出现分段平台现象参数更新或被困于平坦的损失区域。

3.2 模型收敛性

在构建基于 BERT 模型和神经网络的文本回归模型的过程中，我们密切监控了模型的训练收敛性表现，构建了一个能够准确预测查询成功率具备良好泛化能力的文本回归模型。在整个训练周期中展现出了迅速的收敛性，在相对较少的训练迭代次数后便达到了性能的稳定状态，确保能够有效地学习到特征。

我们注意到，随着训练轮次的增加模型在训练数据集上的损失值稳步是下降的，在验证集上的表现逐渐提升直至达到一个性能方面的瓶颈，模型能够从训练数据中有效提取特征，逐渐建立起对查询成功率的准确预测能力。选择一个恰当的训练迭代轮数，避免了因过度训练导致的过拟合风险模型能够充分学习到训练数据中的有效信息。我们实施了早停的策略提高模型的泛化能力并防止过拟合，监控到验证集上的损失值未出现显著下降时就认为模型可能已接近其最优性能，这个时候就停止训练过程了。过程中记录了模型在训练集和验证集上的损失变化曲线，以可视化的形式展示了模型收敛的点的选择帮助我们找到了一个性能良好的模型，缩短了训练时间提高了模型训练的效率。

3.3 模型价值评估

在验证集上，我们对模型的预测能力进行了全面检验。对比预测值与实际值，发现模型的平均预测偏差可以大约 16%，在预测查询成功率方面具有较高的准确性和稳定性，较好地捕捉到了查询文本中的关键的那些信息，实现对查询成功率的精准预测。在验证集上模型的预测偏差波动是较小的状态具有较强的泛化能力，能够在不同的场景下保持一定程度的稳定的性能。针对输入的自然语言问题，模型能够较为准确地预测其执行的成功率，对调整并发数量优化系统资源分配具有重要意义。

我们可以通过数学模型估算自评估模型在提升系统鲁棒性方面的价值。设单次查询的成功概率为 p，并发数为 n，则系统整体任务完成率 R 可表示为：

$$R = 1 - (1 - p)^n \tag{3-1}$$

当$p = 0.4$（单次失败率 60%），$n = 5$时：

$$R = 1 - (1 - 0.4)^5 = 1 - 0.07776 \approx 0.922\ (92.2\%) \tag{3-2}$$

当$n = 3$时：

$$R = 1 - (1 - 0.4)^3 = 1 - 0.216 \approx 0.784\ (78.4\%) \tag{3-3}$$

基于 16%的预测准确率，我们可以建立动态并发数量模型的价值评估体系。设自评估模型预测的成功率为$\hat{p}$，实际成功率为p，预测误差$\epsilon = |p - \hat{p}|$服从均值为 0.16 的正态分布。动态并发数$n_{dynamic}$的最优解可通过以下优化问题求得：

$$n_{dynamic} = \arg\min_n E\left[\frac{n}{1 - (1 - \hat{p})^n}\right] \text{s.t.} R_{target} \geq 90\% \tag{3-4}$$

其中期望项表示单位成功率的 API 调用成本。与固定并发数n_{fixed}相比，动态模型的价值增益 V 可量化为：

$$V = \frac{C_{fixed} - C_{dynamic}}{C_{fixed}} \times 100\% \tag{3-5}$$

其中成本计算公式为：

$$C_{fixed} = \frac{n_{fixed}}{1 - (1 - p)^{n_{fixed}}}, C_{dynamic} = E\left[\frac{n_{dynamic}(\hat{p})}{1 - (1 - p)^{n_{dynamic}(\hat{p})}}\right] \tag{3-6}$$

通过蒙特卡洛模拟（样本量 10,000 次）可得：

当 p=0.4 时，固定策略$n_{fixed} = 5$的成本为 5.42 次/成功查询

动态模型根据预测成功率$\hat{p}$（预测误差$\epsilon \sim |\mathcal{N}(0, 0.16^2)|$）选择最小并发数 n_{dynamic} 满足$R_{target} \geq 90\%$时：

动态策略平均成本为 3.89 次/成功查询（$n_{dynamic} \in [3,7]$）

价值增益 $V = \frac{5.42-3.89}{5.42} = 28.2\%$

满足$R_{target} \geq 92.2\%$时：

动态策略平均成本为$C_{\text{dynamic}} \approx 4.85$ 次/成功查询（$n_{\text{dynamic}} \in [4,9]$）

价值增益$V = \frac{C_{\text{fixed}} - C_{\text{dynamic}}}{C_{\text{fixed}}} \times 100\% = \frac{5.42-4.85}{5.42} \times 100\% \approx 10.5\%$

模型能够有效捕捉查询文本中的关键信息，对于调整并发数量和优化系统资源分配具有重要意义，动态并发数量模型相比固定并发策略能够显著降低成本，提高系统效率，相比于固定数量的静态并发方案，平衡了大语言模型 API 调用成本与成功率之间，带来 10.5%的价值增益。根据模型预测的查询成功率，可以动态调整大语言模型的并发数量，验证了我们所设计的基于 BERT 模型和神经网络的文本回归模型在处理复杂自然语言处理任务中的有效性和实用性，实现了成本与稳定性之间的良好平衡。

4 讨论与总结

4.1 研究总结

本研究通过构建基于神经网络的大语言模型智能体自评估模型，在自然语言数据库查询系统的动态资源优化领域取得了进展。当前大语言模型 API 调用成本居高不下，自评估模型根据查询复杂度动态调整资源分配实现的智能并发调控，具有重要的工程实践价值。研究的结果验证了文本回归模型在预测智能体的执行成功率方面是十分的有效的，为人工智能系统的自我认知的机制提供了新的技术路径。

在模型设计方面，本研究开融合了 BERT 的深层语义理解能力与神经网络回归层的数值预测优势，与既有研究的横向对比凸显了本方法的创新价值。相较于 Freo 等人[5]基于 Lasso 回归的文本特征选择方法，我们的模型将有效特征维度提示到捕捉更细微的语义模式的程度。与 Bu 等人[20]的纯 Transformer 架构相比，我们的池化层结合全连接回归层设计在保持相当预测性能的情况下，将模型参数量减少降低了部署成本，混合架构在文本回归任务中具有的优势。源于 BERT 模型的多头注意力机制，能建立查询文本中远距离词汇的语义关联，传统词袋模型完全无法处理此类复杂语言结构。传统 TF-IDF+线性回归方法在相同测试集上平均偏差表现欠佳，我们的混合架构将预测偏差降至 16%有一定的优势。

通过设计多线程冗余，系统在单次查询失败率高达 40%的极端测试环境下，仍然能够维持 92%的整体任务完成率，验证了我们提出的当单次尝试成功率服从伯努利分布的情况下通过智能并发可很大程度上提高系统稳态可靠性的概率补偿理论。当并发数 5 时系统稳态可靠性 92%，完美符合 $1-(1-p)^n$ 的概率模型（其中的 p 为单次成功率，n 为并发数）。使用基于神经网络的大语言模型智能体成功率回归预测模型的动态并发数量策略更是相比固定并发策略明显降低成本提高系统效率，相比于固定数量的静态并发的方案，带来10.5%的价值增益实现了成本与稳定性之间的良好平衡。

4.2 研究展望

本研究通过系统的理论探索和实验验证，成功构建了一个具有自我评价能力的大型语言模型智能体系统，为认知人工智能的发展提供了重要范例。现有静态模型面临数据分布动态变化挑战，无法实现模型的实时的演进，我们计划设计增量式训练框架，使模型能持续适应新的工作模式，由离线强化学习演进到在线强化学习。

从长远来看，本研究的方法可以扩展到智能客户服务和自动编程等场景。在代码生成任务中，预测代码片段的执行成功率来动态调整生成策略，推动自感知人工智能从理论概念转化为实际工程范式，实现人工智能系统从被动执行到主动思考的质变。

参考文献

[1] Sindhu B., Prathamesh R. P., Sameera M. B. et al. The Evolution of Large Language Model: Models, Applications and Challenges[A]. In: 2024 International Conference on Current Trends in Advanced Computing (ICCTAC)[C]. Bengaluru, India, 2024. 1-8.

[2] Ozkaya I. Application of Large Language Models to Software Engineering Tasks: Opportunities, Risks, and Implications[J]. IEEE Software, 2023, 40(3): 4-8.

[3] Mohaimenul A. K. R., Saddam H. M., Kaniz F. et al. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges[J]. IEEE Access, 2024, 12: 26839-26874.

[4] 安康,张勇博,黄泽. 多元线性回归的英文文本难度估计模型[J]. 现代信息科技, 2022, 6(11):30-33.

[5] Freo M., Luati A. Lasso-based variable selection methods in text regression: the case of short texts[J]. AStA Advances in Statistical Analysis, 2024, 108: 69–99.

[6] Liu S., Wang. Y. Control Valve Stiction Detection Based on the Improved Linear Regression Method[A]. In: 2024 36th Chinese Control and Decision Conference (CCDC)[C]. Xi'an, China, 2024. 2589-2594.

[7] Muthukrishnan R., Rohini R. LASSO: A feature selection technique in predictive modeling for machine learning[A]. In: 2016 IEEE International Conference on Advances in Computer Applications (ICACA)[C]. Coimbatore, India, 2016: 18-20.

[8] Onita D., Cucu C. Automatic Text Summarization using Kernel Ridge Regression[A]. In: 2023 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)[C]. Nancy, France, 2023: 202-209.

[9] Liu C. Scenic area data analysis based on NLP and ridge regression[A]. In: 2021 IEEE International Conference on Electronic Technology, Communication and Information (ICETCI)[C]. Changchun, China: IEEE, 2021: 270-277.

[10] Luo H., Liu Y. A prediction method based on improved ridge regression[A]. In: 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)[C]. Beijing, China: IEEE, 2017: 596-599.

[11] Wen C., Wu J., Chen D. Analysis of Text Emotion Based on Logistic Regression Model[A]. In: 2022 IEEE 5th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)[C]. Shenyang, China: IEEE, 2022: 891-895.

[12] Ginting P. S. Br., Irawan B., Setianingsih C. Hate Speech Detection on Twitter Using Multinomial Logistic Regression Classification Method[A]. In: 2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)[C]. Bali, Indonesia: IEEE,

2019: 105-111.

[13] Brzezinski J. R., Knafl G. J. Logistic regression modeling for context-based classification[A]. In: Proceedings. Tenth International Workshop on Database and Expert Systems Applications. DEXA 99[C]. Florence, Italy: IEEE, 1999: 755-759.

[14] 陈雪婷. 基于多元 Logistic 回归模型的新 HSK 中高级阅读文本可读性分析[D]. 湖北: 华中科技大学, 2022.

[15] Zou X., Hu Y., Tian Z., Shen K. Logistic Regression Model Optimization and Case Analysis[A]. In: 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)[C]. Dalian, China, 2019. 135-139.

[16] Li T., Liu X., Su S. Semi-supervised Text Regression with Conditional Generative Adversarial Networks[A]. In: 2018 IEEE International Conference on Big Data (Big Data)[C]. Seattle, WA, USA, 2018. 5375-5377.

[17] 李文彪,吴云芳. 基于神经网络模型的汉语文本难度分级[J]. 中文信息学报, 2023, 37(2):158-168.

[18] Ajit A., Acharya K., Samanta A. A Review of Convolutional Neural Networks[A]. In: 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)[C]. Vellore, India, 2020. 1-5.

[19] Vaswani A., Shazeer N. M., Parmar N., Uszkoreit J. et al. Attention is All you Need[A]. In: Neural Information Processing Systems[C]. 2017.

[20] Bu Q., Sungrae P., Minsoo K. et al. SRFormer: Text Detection Transformer with Incorporated Segmentation and Regression[A]. In: AAAI Conference on Artificial Intelligence[C]. 2023.

[21] Hazim L. R., Ata O. Textual Authenticity in the AI Era: Evaluating BERT and RoBERTa with Logistic Regression and Neural Networks for Text Classification[A]. In: 2024 International Symposium on Electronics and Telecommunications (ISETC)[C]. Timisoara, Romania, 2024. 1-6.