

# 一种基于智能体的 G 代码生成问答系统设计与实现

郜宇博<sup>1,2</sup>, 何平<sup>1</sup>, 周正<sup>1,2</sup>, 迟连攀<sup>1,2</sup>, 陈伟强<sup>1,2</sup>, 于东<sup>1,2,3</sup>

<sup>1</sup>(中国科学院沈阳计算技术研究所, 沈阳 110168)

<sup>2</sup>(中国科学院大学, 北京 100049)

<sup>3</sup>(沈阳中科数控技术股份有限公司, 沈阳 110168)

E-mail: yudong11@sict.ac.cn

**摘要:** 智能制造与工业自动化的发展使数控加工技术成为现代制造业的关键。G 代码作为数控机床的重要指令语言, 其生成质量直接影响加工精度、生产效率和制造水平。然而, 传统 G 代码依赖人工编写, 效率低下且易出错。为此, 本文提出了一种基于智能体的 G 代码生成系统。系统通过自然语言处理解析用户需求, 自动生成精准的 G 代码。同时, 采用领域数据校正方法, 有效解决了相似词语识别错误问题。此外, 设计了 CMT-BERT 模型, 优化任务识别与参数提取。实验结果表明, 意图识别和实体抽取的 F1 分数分别为 98.0% 和 95.8%, 显著优于其他传统模型, 极大地提升了 G 代码生成效率与准确性。本研究为数控加工的智能化与自动化提供了创新解决方案, 具有重要实践与理论意义。

**关键词:** 对话系统; 智能体; 知识图谱; G 代码生成

中图分类号: TP391

文献标识码: A

文章编号: 1000-1220(2026)02-0351-10

## Design and Implementation of an Agent G-code Generation Question-answering System

GAO Yubo<sup>1,2</sup>, HE Ping<sup>1</sup>, ZHOU Zheng<sup>1,2</sup>, CHI Lianpan<sup>1,2</sup>, CHEN Weiqiang<sup>1,2</sup>, YU Dong<sup>1,2,3</sup>

<sup>1</sup>(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

<sup>2</sup>(University of Chinese Academy of Sciences, Beijing 100049, China)

<sup>3</sup>(Shenyang Zhongke CNC Technology Co., Ltd, Shenyang 110168, China)

**Abstract:** The advancement of intelligent manufacturing and industrial automation has positioned CNC machining technology as a cornerstone of modern manufacturing. As the essential programming language for CNC machines, the quality of G-code generation directly impacts machining accuracy, production efficiency, and overall manufacturing performance. However, traditional G-code generation relies heavily on manual programming, which struggles to meet the demands of complex and dynamic production environments, resulting in low efficiency and high error rates. To address these challenges, this paper proposes an intelligent agent-based G-code generation system. The system leverages natural language processing to interpret user requirements and automatically generates accurate G-code. Additionally, a domain-specific data correction method is implemented to effectively resolve issues related to the misidentification of similar terms. Furthermore, a customized CMT-BERT model is developed to optimize task recognition and parameter extraction. Experimental results demonstrate that the proposed system outperforms traditional models in intent recognition and entity extraction, significantly improving the efficiency and accuracy of G-code generation. This research offers an innovative solution for the intelligent and automated generation of G-code, providing substantial practical and theoretical value for CNC machining.

**Keywords:** dialogue system; agents; knowledge graph; G-code generation

## 0 引言

在智能制造与工业自动化的推动下, 数控加工技术已成为现代制造业的核心组成部分。随着制造业不断向更高效、精准、智能化方向发展, 数控加工技术对产品质量、生产效率和制造成本产生深远影响。作为数控机床的基本指令语言, G 代码直接控制机床操作步骤, 是制造过程中的核心命令。G 代码生成的质量不仅影响工件加工精度, 更关系到生产效率和制

造质量。因此, 提升 G 代码生成的效率、准确性和智能化水平, 已成为现代制造业亟待解决的核心问题。

然而, 传统的 G 代码生成方式依赖人工编写, 繁琐且易出错, 尤其在面对复杂加工任务时, 人工编写既耗时又难以满足多样化和高复杂度的生产需求。随着生产环境的变化和定制化需求增多, 传统方法的局限性愈加明显, 限制了生产效率和智能制造水平的提升<sup>[1]</sup>。因此, 开发自动化、智能化的 G 代码生成方式, 成为提升制造业竞争力的关键。

收稿日期: 2024-12-31 收修改稿日期: 2025-02-27 基金项目: 国家工信部高质量发展专项课题项目 (TC220H05S-003) 资助。作者简介: 郜宇博, 男, 2001 年生, 硕士研究生, 研究方向为数控技术; 何平, 男, 1973 年生, 博士, 研究员, 研究方向为计算机系统结构、嵌入式系统应用、数控技术; 周正, 男, 1996 年生, 博士研究生, 研究方向为数控与数字化车间技术; 迟连攀, 男, 2001 年生, 硕士研究生, 研究方向为数控技术; 陈伟强, 男, 2000 年生, 硕士研究生, 研究方向为数控技术; 于东 (通信作者), 男, 1966 年生, 博士, 研究员, 博士生导师, CCF 会员, 研究方向为数控技术、面向控制器的实时系统等。

近年来,虽然基于 CAD 图纸(如 DXF 文件格式)的 G 代码生成方法逐渐成熟<sup>[2]</sup>,但基于自然语言理解的 G 代码生成研究仍然较为稀缺.本文提出了一种基于智能体的 G 代码生成系统.通过模块化设计,系统能够感知用户输入的加工需求,做出实时决策与优化,生成高效、精准的 G 代码.

本文结合知识图谱技术<sup>[3]</sup>,在与用户交互过程中,系统不仅能自动生成 G 代码,还能借助深度学习算法和知识图谱进行优化与校正.知识图谱将传统加工经验转化为可复用的智能资源,提升了 G 代码生成的自动化和精准度.此外,系统引入了领域数据校正机制,有效解决了因相似词语导致的识别错误.为提高任务识别与参数提取的准确性,本文提出了一种定制化 CMT-BERT(Customized Machining Task BERT)模型,专为加工任务的自然语言处理设计.CMT-BERT 能够精准提取用户输入中的关键加工参数,显著提升任务分类和优化决策效率,进一步推动 G 代码生成的智能化.

实验结果表明,所提出的 G 代码生成系统在意图识别和实体抽取任务中表现优异,显著提高了 G 代码生成的效率和准确性.该系统不仅能适应多样化生产环境,还展现出高度的智能化与自适应能力.本研究为数控加工的智能化与自动化提供了创新解决方案,推动智能制造领域的发展,且为相关领域的研究提供了新的技术思路和实践支持.

## 1 相关工作

### 1.1 面向智能制造的自动化 G 代码研究

在智能制造和工业自动化的发展背景下,数控加工技术已成为现代制造业的关键技术,对产品质量、生产效率及制造成本具有重要影响<sup>[4]</sup>.作为数控机床的指令语言,G 代码直接决定了加工过程的精度与效率.传统的 G 代码生成方法主要依赖人工编写或基于 CAD/CAM 软件的自动化生成.然而,这些方法在面对复杂加工任务及多变的生产环境时,适应性不足、易出错等问题逐渐显现.

近年来,基于人工智能的自动化 G 代码生成技术受到广泛关注.一些学者尝试将人工智能算法引入 G 代码生成领域,例如,Mansour 等人(2020)利用遗传算法优化加工路径,提高了生成效率与路径合理性<sup>[5]</sup>;Lasluisa-Naranjo 等人(2024)通过应用 K-means 机器学习聚类算法,对 G 代码中的关键参数进行了优化,成功减少了制造时间和材料消耗<sup>[6]</sup>.国内的研究则更多关注结合制造业实际需求,Zan Lamberger(2024)提出了一种生成和组装生物制造 G 代码的标准化方法,确保了不同机器和规模之间的兼容性和融合性<sup>[7]</sup>.

Nguyen 等人(2020)则提出了一种新型的 CAPP 系统(BKCAPP),并将其与 G 代码生成模块集成,实现了完整的 CAD-CAPP-CNC 系统.该系统能够自动从三维(3D)实体模型中识别加工特征和操作,并基于宏编程自动生成 G 代码.这种方法显著减少了 CAM 模块中的人工干预,提高了系统的自动化和实用性<sup>[8]</sup>.在增强现实(AR)技术的支持下,Mourtzis 等人(2023)开发了一种基于 AR 的 G 代码生成框架.该框架通过增强现实增强了用户对加工过程的感知能力,并结合云技术,实现了 G 代码的虚拟指令生成、实时传输和自动化执行,显著提升了工业 4.0 背景下的车间操作效率<sup>[9]</sup>.

尽管上述研究取得了一定进展,但仍存在以下不足:1)大多数方法无法直接支持自然语言输入与 G 代码生成的无缝衔接,交互性较弱;2)对复杂加工任务的自适应优化能力有限,难以全面满足多样化的生产需求.

### 1.2 面向人机交互的任务型对话系统研究

任务型对话系统在智能制造、智能客服等领域得到广泛应用.这类系统专注于解决特定任务,通过自然语言交互帮助用户完成目标<sup>[10]</sup>.近年来,深度学习技术的应用显著提升了问答系统的性能.诸如 Seq2Seq、Transformer 和 BERT 等模型在自然语言生成方面展现了强大的能力.基于小样本机器学习的任务对话系统也展示了其在垂直领域对话中的优势,尤其是在意图识别与槽位填充方面的表现<sup>[11]</sup>.

此外,知识图谱的引入为任务型对话系统提供了领域特定的知识支持,从而增强了系统的推理与信息检索能力<sup>[12]</sup>.这种结合在工业领域显示出巨大潜力,支持生产管理和流程优化.例如,基于 Rasa 框架的系统能够定制化地提供工艺流程查询和反馈,提升生产线的灵活性和效率<sup>[13]</sup>.

未来,任务型对话系统将逐步向多模态交互、个性化响应生成及情感分析等方向发展.结合智能体技术,这些系统将在复杂场景中展现出更高的适应性和鲁棒性<sup>[14]</sup>.

### 1.3 基于智能体的数控系统优化设计

基于上述自动化 G 代码生成技术和任务型对话系统的研究成果,本文进一步探索了智能体技术在数控系统中的应用.智能体技术在数控系统中的应用为制造业带来了新的变革.智能体架构具有动态适应和优化能力,可根据实时数据调整加工计划与策略.Wong 等人提出了一种基于多智能体的敏捷制造计划与控制系统(AMPCS),通过集成 RFID 技术与多智能体系统,实现了制造过程的实时监控和动态调度<sup>[15]</sup>.

在知识驱动数控系统方面,王凌云等人开发了一种基于知识图谱的智能数控编程系统,专注于复杂零件的加工优化<sup>[16]</sup>.该系统通过知识获取、表示和推理,实现了工艺流程与参数的标准化,显著提高了编程效率与加工质量.

然而,传统的数控系统仍面临诸多挑战.首先,G 代码的生成往往依赖于人工编写,难以快速响应动态变化的生产环境和工艺需求.其次,传统系统在运行中缺乏智能化处理能力,无法根据实时反馈进行调整,这导致加工精度和效率难以满足现代制造业对高标准的要求.

为解决上述问题,本文设计了一种基于智能体的 G 代码生成问答系统.系统基于知识图谱通过自然语言处理技术解析用户需求,自动生成精准的 G 代码.如图 1 所示.

智能体架构以加工工艺流程为核心,成为 CAD/CAM 与 CNC 系统之间的信息桥梁.它不仅实现了信息流的双向交互,而且增强了系统的动态适应性和加工效率.为进一步提升系统性能,本文设计中结合了加工工艺知识图谱与深度学习技术,并引入拼音相似度校正方法与 CMT-BERT 模型.该系统能够高效识别用户的加工需求并生成精准的 G 代码,同时根据加工环境的动态变化和用户输入参数进行自适应优化.通过融入闭环反馈机制和动态调整能力,该智能体架构有效弥补了传统系统的信息差,大幅提升了加工质量与生产效率,为现代制造系统的智能化发展提供了全

新的解决方案.

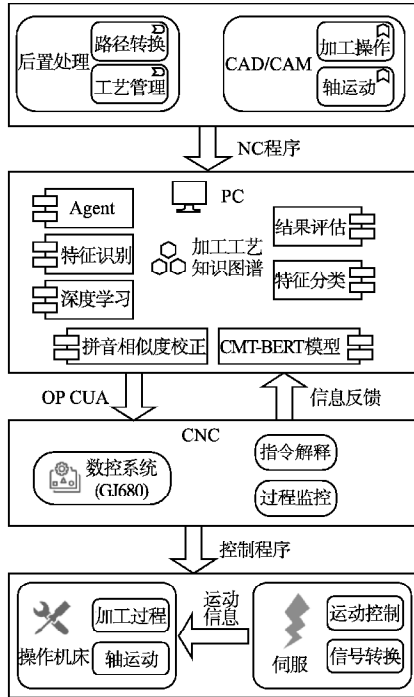


图1 基于智能体的数控系统加工工艺流程结构

Fig. 1 Agent-based CNC machining process flow structure

这套基于智能体的优化设计,不仅拓展了数控系统的應用边界,还为未来更高效、更智能的制造流程奠定了基础.

## 2 基于知识图谱的 G 代码生成方法建模

### 2.1 知识图谱构建

知识图谱的构建起源于数据整合与关系表达,并逐渐发展为信息表示与推理的重要工具.在本研究中,知识图谱作为核心框架,旨在整合数控加工中的工艺知识、所需参数及 G 代码生成规则,为自动化加工任务提供结构化和语义化支持.通过知识图谱,能够有效展示加工过程中的各种信息及其相互关系,支撑自动化决策,从而显著提升数控加工的效率 and 精确性.

在知识图谱的构建过程中,包含了多种核心实体,如加工工艺、操作、加工类别以及相关的加工参数等.这些实体具有明确的定义和属性.例如,加工类别可以划分为外圆工艺、里孔工艺、锥面工艺等,每一种加工类别下面可能包含多个加工子工艺,而每种工艺又有具体的加工参数.例如,加工类别“外圆工艺”包含子工艺“外圆”和“外圆弧”,它们都需要参数‘Cn’和‘L’.

图谱中的这些实体通过多种关系进行连接,如“包含”、“需要”、“输入”等,表达了不同种类与工艺之间的关系,以及加工参数与工艺的适配性.这些关系为后续的 G 代码推理与生成提供了关键的依据.

在数学上,加工特征的集合可以表示为:

$$P = B_L - \{M_{F_1}, \dots, M_{F_i}, \dots, M_{F_n}\} \quad (1)$$

其中, P 表示加工后的最终零件,  $B_L$  是毛坯或半成品的工件,

F 代表“加工特征”,如切削、钻孔、磨削等加工过程,它决定了最终零件的形状和尺寸,每一个  $M_{F_i}$  表示对工件进行第 i 种特定加工的操作.

为了保证知识图谱的准确性与完整性,本文采集了来自实际加工数据、数控机床操作手册及专家讲解等多种来源的数据.这些数据不仅涵盖了各种类型的加工任务,还包括了从传统手动编程到自动生成 G 代码过程中的所有相关信息.通过这些数据的整合,构建出了一个多层次、多维度的知识图谱,确保它能够应对复杂和多变的加工任务.当用户输入加工任务时,系统能够通过图谱中已有的工艺节点和其关系,自动匹配相关的参数.为了保证特征之间的关系是准确的,可以通过以下公式表示特征间的相互关系:

$$M_{F_i} \cap M_{F_j} \cup B_{M_{F_i}} \cap B_{M_{F_j}} \neq \Phi \quad (2)$$

如果  $M_{F_i}$  和  $M_{F_j}$  之间有交集,或它们的边界  $B_{M_{F_i}}$  和  $B_{M_{F_j}}$  存在交集,则意味着它们之间有一定的相互影响.整合这些数据后,构建了多层次、多维度的知识图谱,以应对复杂加工任务,如表 1 所示.

表 1 加工工艺知识图谱表

Table 1 Processing technology knowledge graph

工艺类别	子工艺	参数
外圆工艺	外圆,外圆弧,外锥面	Cn, L, Tr, Cr, F
端面工艺	端面,切槽,内端面	Cn, Lr, deltaT, CT, F
里孔工艺	内圆,内锥面,内槽,内弧,中心孔	Cn, deltaT, F, BT, L
锥面工艺	外正锥面,外反锥面,内正锥面,内反锥面	Tr, F, L, Cn, A, xDir, zDir
螺纹工艺	外直螺纹,外锥螺纹,内直螺纹,内锥螺纹	L, Tr, Tp, Cn, Cr, multi_head, tail-length, cuttingDepthSelection
倒角工艺	外圆角倒角,外倒角,内圆角倒角,内倒角	R, Tr, Cn, F, G2G3, PHi1, L, G71G73

最终构建的部分知识图谱如图 2 所示.

### 2.2 智能体架构设计

本文提出的系统中,通过模块化结构提升了系统的智能化水平,支持用户与智能制造系统的高效交互,能够根据加工任务需求自动生成精准的 G 代码.智能体模块包括感知器、数据处理模块、学习判断模块、特征标识模块、执行器和加工工艺知识图谱,如图 3 所示.

感知器是智能体与外部环境交互的桥梁,主要用于接收用户的自然语言输入和工艺信息.在本系统中,感知器基于 Rasa 对话框架,完成意图识别和实体抽取,将用户输入转化为结构化数据,并传递至数据处理模块.感知器的输入主要包括用户通过自然语言提供的加工需求(如加工类别、参数)以及相关的历史记录,为后续模块提供可靠的数据支持.

数据处理模块是整个系统的数据预处理核心,其主要任务是对感知器传递的原始数据进行格式化、清洗和转换,以适应模型训练和后续分析的需求.模块的具体功能包括:数据清洗,格式转换和上下文交互.

公式描述模块的输入和输出过程如下:

$$D_{\text{processed}} = F_{\text{process}}(D_{\text{raw}}, K) \quad (3)$$

其中,  $D_{\text{processed}}$  是处理后的数据,  $D_{\text{raw}}$  是感知器输入的原始数

据,  $K$  表示知识图谱提供的上下文信息,  $F_{process}$  表示一个函数, 它对原始数据  $D_{processed}$  和上下文信息  $K$  进行处理.

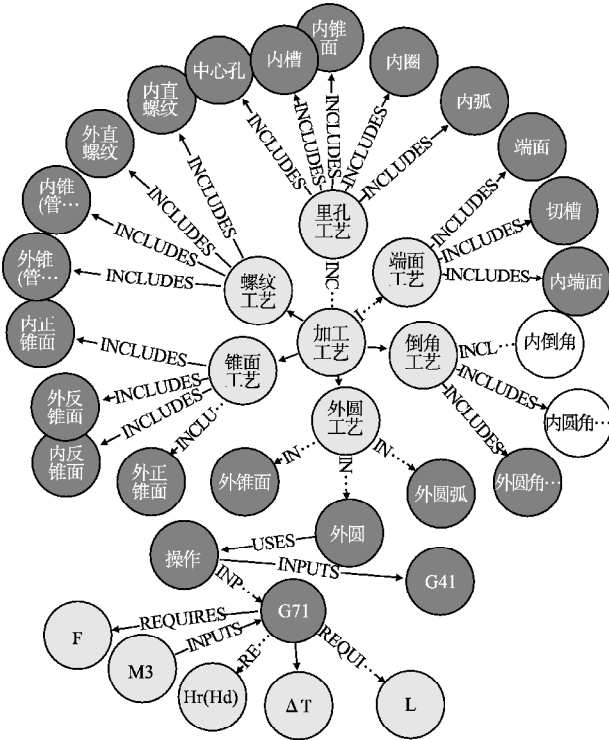


图2 知识图谱  
Fig. 2 Knowledge graph

学习判断模块是系统的核心智能单元, 负责使用深度学习技术进行特征识别和决策. 模块利用 CMT-BERT (定制化加工任务 BERT) 模型, 从知识图谱中的特征库中提取信息, 结合用户输入, 生成加工任务的精准 G 代码.

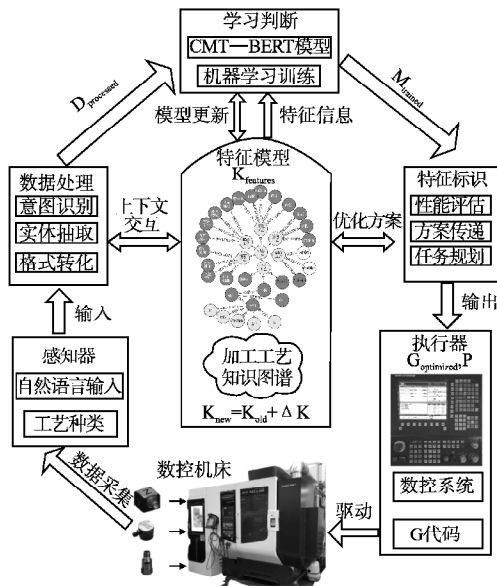


图3 智能体架构模型图  
Fig. 3 Intelligent agent architecture model

学习判断过程可以表示为:

$$M_{trained} = \text{Train}(D_{processed}, K_{features}) \quad (4)$$

其中,  $M_{trained}$  为训练好的深度学习模型,  $K_{features}$  为知识图谱提供的特征库.

模块的输出直接影响特征标识模块的任务决策, 同时可对生成的 G 代码进行动态调整以满足用户需求.

特征标识模块通过验证训练模型的效果, 判断是否需要调整或重新训练. 同时结合知识图谱信息, 对 G 代码进行优化. 最终将优化后的执行方案传输至执行器模块. 该模块在系统中的功能类似于“任务决策层”, 确保任务规划与执行的精准性和效率.

执行器是系统的物理执行单元, 负责将特征标识模块输出的 G 代码转化为具体的加工指令. 模块直接与数控系统 (CNC) 对接, 执行加工路径规划、进给速度调整等任务.

执行器的输入可以用公式(5)表示:

$$E_{command} = F_{generate}(G_{optimized}, P) \quad (5)$$

其中,  $G_{optimized}$  是优化后的 G 代码,  $P$  是数控加工参数,  $F_{generate}$  表示生成数控指令的函数.

加工工艺知识图谱是系统的核心数据模型, 用于存储和管理加工工艺、工具信息、参数配置以及 G 代码模板.

### 3 系统设计与实现

#### 3.1 系统架构

如图4所示为系统的架构图, 分为用户页面、用户交互模块、自然语言理解模块、动作执行模块、对话管理模块和 Neo4j 数据库.

#### 3.2 自然语言理解模块设计与实现

针对数控工艺加工领域的数据库特点, 本文基于数据表的常用字段, 在 domain.yml 文件中定义了 12 种意图 (Intents), 覆盖从用户交互到 G 代码生成的主要场景.

为满足用户多样化的交互需求, 设计的意图如表2所示.

#### 3.3 对话管理 (DM) 模块设计与实现

对话管理模块采用故事 (Stories) 形式进行设计, 通过定义不同场景的交互逻辑, 使系统能够根据上下文合理规划对话流程. 部分对话故事示例如下:

```

-story: request processing
steps:
  -intent: request_processing
  -action: action_request_processing
-story: request sub processing
steps:
  -intent: request_sub_processing
  -action: action_request_sub_processing
-story: generate gcode story
steps:
  -intent: request_gcode
  -action: gcode_form
  -active_loop: gcode_form
  -action: utter_goodbye

```

以上示例中, 当用户输入被识别为 request\_processing 意图时, 系统将调用 action\_request\_processing 自定义动作. 在生

成 G 代码场景中,通过 gcode\_form 表单收集用户所需参数,生成完整的 G 代码后结束会话。

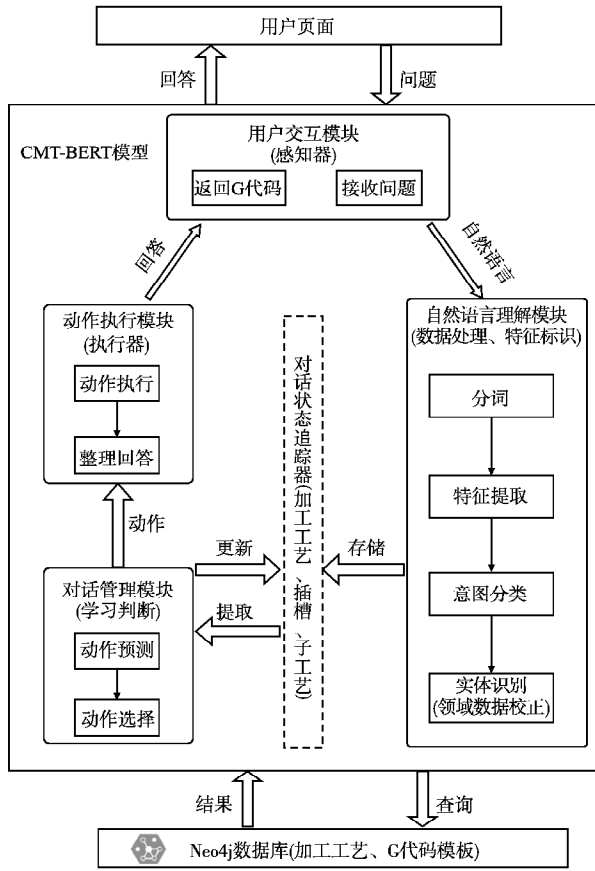


图 4 系统架构图

Fig. 4 System architecture

为增强系统的功能与扩展性,本文设计了多种自定义动作,并实现了对应的 Action 类,通过重写 run 方法实现特定逻辑。具体动作设计如表 3 所示。

表 2 意图表  
Table 2 Intent design

意图名称	描述
greet	问候,机器人开始对话时会使用
goodbye	告别,结束对话时会使用
affirm	确认,对话中表示同意时会使用
deny	拒绝,对话中表示拒绝时可使用
mood_unhappy	表示心情不好时可能会使用
bot_challenge	通常用于测试机器人的回应能力
inquire_neo4j	查询知识图谱
request_gcode	请求生成 G 代码
request_sub_processing	请求子工艺
request_processing	请求工艺种类
clear_history	清除历史记录
affirm	确认,表示同意时会使用

### 3.4 领域数据校正方法设计与实现

针对加工工艺名称模糊匹配中的复杂性与多样性问题,本文提出了一种基于索引过滤与拼音相似度结合的多层次匹配算法。该算法旨在通过优化匹配效率和准确性,解决加工工

艺名称在用户输入中的模糊匹配问题。利用加工工艺知识图谱的支持,算法首先通过长度过滤筛选出可能的候选节点,然后通过拼音相似度校正进一步缩小范围,最后通过编辑距离计算确定最终的匹配结果。

表 3 动作表

Table 3 Action design

动作名称	功能描述
ActionClearHistory	清除所有历史记录和插槽
GCodeForm	动态加载插槽并验证输入。
ActionGenerateGcode	生成 G 代码
ActionRequestProcessing	处理输入的工艺种类
ActionRequestSubProcessing	确认用户选择的子工艺类型

在该方法中,首先对加工工艺名称节点进行高效的预处理,构建了双重索引:长度索引与字符倒排表。长度索引记录了每个工艺名称的字符长度,利用输入模式串的长度  $m$  和编辑距离阈值  $k$ ,筛选出目标节点集合  $str$  中符合条件的候选节点。候选节点的长度  $x$  需要满足以下条件:

$$m(1 - \alpha) \leq x \leq m(1 + \alpha), \alpha = \frac{k}{m} \quad (6)$$

这一过程显著减少了计算量,优化了后续的匹配过程。字符倒排表进一步通过统计模式串字符与候选节点字符的命中数量,优化了候选节点的筛选效率。

针对用户输入中可能出现的拼音错误或语义识别误差,本文引入了拼音相似度校正模块。具体方法为:首先将模式串与候选节点的名称转化为拼音序列,然后通过计算拼音编辑距离  $ED_{pinyin}$  来衡量两者之间的拼音相似度。拼音相似度  $Sim_{pinyin}$  的计算公式为:

$$Sim_{pinyin} = 1 - \frac{ED_{pinyin}(P_1, P_2)}{\max(|P_1|, |P_2|)} \quad (7)$$

其中,  $P_1$  和  $P_2$  分别表示模式串与候选节点的拼音序列。通过设定拼音相似度阈值,能够有效地排除拼音差异较大的候选节点,进一步缩小匹配范围。

对于经过拼音相似度校正后的候选节点集合,本文采用编辑距离算法来计算最终的相似度。最终相似度综合了字符相似度和拼音相似度,使用加权的方式对候选节点进行排序,公式如下:

$$Sim_{final} = w_1 \cdot Sim_{char} + w_2 \cdot Sim_{pinyin} \quad (8)$$

其中,  $w_1$  和  $w_2$  分别为字符相似度和拼音相似度的权重系数,用于平衡两者在最终匹配结果中的重要性。

算法流程如算法 1 所示。

#### 算法 1.

输入: pattern, str, k, threshold\_pinyin

输出: 匹配最优的加工工艺名称节点 best\_match

1. Function LengthFilter(pattern, str, k, LengthIndex):

$m \leftarrow \text{Length}(\text{pattern}), \alpha \leftarrow k * m$

Candidates  $\leftarrow []$

For each node in str:

$\text{length} \leftarrow \text{Length}(\text{node})$

If  $(m - \alpha) \leq \text{length} \leq (m + \alpha)$  // 满足长度过滤条件

Add node to Candidates

Return Candidate // 返回长度过滤后的候选节点

```

End Function
2. Function PinyinSimilarity(pattern, candidates):
    Convert pattern to Pinyin sequence P1
    For each node in candidates:
        Convert node to Pinyin sequence P2
        // 计算拼音编辑距离、拼音相似度
        EDpinyin, Simpinyin ← Calculate(P1, P2)
        If Simpinyin ≥ threshold_pinyin:
            Add node to ValidCandidates
    Return ValidCandidates // 返回拼音校正后的候选节点
End Function
3. Function CalculateFinalSimilarity(args):
    Candidates, ValidCandidates ← GetCandidates()
    For each valid node in ValidCandidates:
        Simchar, Simpinyin ← Calculate()
        Simfinal ← w1 * Simchar + w2 * Simpinyin
    Store Simfinal for each node
    Return Node // 返回最终相似度的节点
End Function

```

End Function

该方法通过逐步细化匹配过程,显著提高了模糊匹配的效率与准确性。通过结合索引过滤、拼音相似度校正与编辑距离计算,不仅有效解决了拼音和语音输入误差带来的问题,还提高了大规模加工工艺名称匹配的性能,实现效果如图5所示。

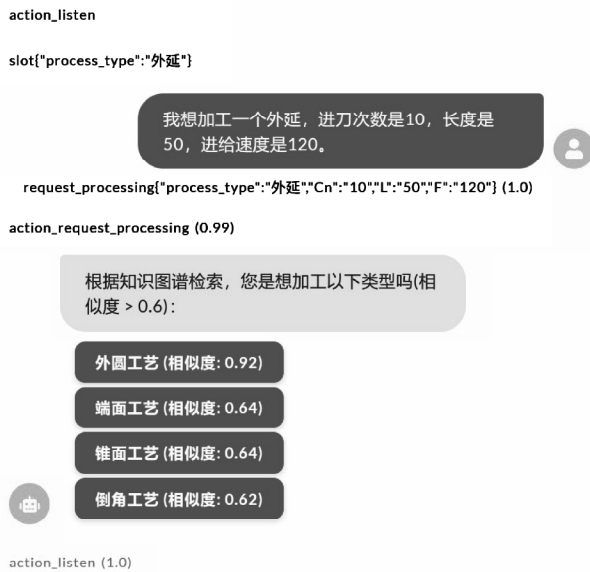


图5 领域数据纠正效果图

Fig. 5 Effect of domain data correction

### 3.5 CMT-BERT 模型设计与实现

CMT-BERT 结合了深度学习技术与领域特定的校正方法,通过引入多层次的匹配算法,对用户输入中的模糊工艺名称进行精确匹配。具体来说,CMT-BERT 在领域数据校正上做出了创新,采用了索引过滤、拼音相似度校正和编辑距离计算等多种技术,逐步缩小候选工艺范围,提升了匹配效率和准确性。这使得模型能够在复杂的加工任务中,准确识别工艺类型和关键参数,进而生成符合数控机床执行标准的 G 代码。

此外,CMT-BERT 模型不仅能够理解用户的加工需求,还能够生成实时调整的、符合特定加工要求的 G 代码,相

于之前的模型,其在任务、参数提取和生成 G 代码的能力上具有显著的优势。

#### 3.5.1 输入处理与特征提取

本文提出的 CMT-BERT 模型旨在处理用户通过自然语言描述的加工任务,这些任务通常涉及对特定加工工艺类型(如外圆工艺、端面工艺等)的请求。由于用户的表达方式各异,可能存在同义词、拼写错误或口音差异,处理这些变异是提高模型鲁棒性和准确性的重要环节。为了应对这些挑战,模型设计了针对性的预处理,即 3.4 节提到的领域数据纠正方法。提高了工艺类型的匹配效率,并增强了模型在实际应用中的准确性。

在输入处理阶段,首先对用户输入的文本进行分词和标准化处理,随后将其转化为适合 BERT 模型的 token ID 格式。为了让模型能够理解输入文本的上下文信息,特别添加了 [CLS] 和 [SEP] 特殊标记,这样做能够支持任务分类和句子间的关系理解。

在特征提取过程中,将分词后的 token 输入到特征提取器中,为每个 token 生成数字化的特征矩阵,用于后续的意图分类和实体识别。在传统的自然语言处理模型中,特征提取器通常包括稀疏特征器和稠密特征器。稀疏特征器通过 one-hot 编码和 N-gram ( $n \leq 5$ ) 编码生成稀疏特征向量,这些编码方式可能会引入冗余信息,影响模型的训练效果。为了优化特征的表示,通过 L2 正则化来控制特征维度的大小,避免过拟合。L2 正则化有助于平滑模型的训练过程,使得模型能够在多变的输入数据中更加稳健。

在稠密特征的生成方面,利用了预训练的语言模型 BERT,通过嵌入方式将每个 token 转换为高维的稠密向量。与稀疏特征相比,稠密特征能够更加精细地捕捉到文本的语义信息,并且适应性更强。基于这些稠密向量,能够获得更精确的语义表示,从而为意图分类和实体识别提供更加准确的输入。

考虑到 Transformer 模型对输入维度的一致性要求,采用了前馈神经网络(Feed-forward Network)来对稀疏特征和稠密特征进行维度对齐。通过前馈神经网络,稀疏特征的维度将被调整,与稠密特征进行拼接,形成统一的特征向量。此时,拼接后的特征向量将经过另一个前馈神经网络层,进一步调整至固定的维度(256 维)。此过程确保了模型输入维度的一致性,同时也提升了模型处理复杂输入的能力。前馈神经网络层的计算公式如下所示:

$$FF_w(x) = \text{ReLU}(0, xW_1 + b_1)W_2 + b_2 \quad (9)$$

其中,  $W_1$  和  $W_2$  为权重矩阵,  $b_1$  和  $b_2$  为偏置项。通过这种特征提取方法,CMT-BERT 能够有效地从用户输入中提取出高维度、语义丰富的特征向量,为后续的意图分类和实体识别奠定基础。

#### 3.5.2 BERT 编码器与任务分类

在 Rasa 框架下,CMT-BERT 模型利用 BERT 编码器的多层 Transformer 架构,对用户输入的自然语言进行深度语义特征提取。BERT 的自注意力机制能够捕捉输入文本中的长程依赖关系,有效地识别加工工艺名称与相关参数之间的隐含联系。这种机制特别适合加工任务中复杂的上下文关联,有助于提高模型在任务识别和参数提取中的表现。

Rasa 中的意图分类模块与实体识别模块通过与 BERT 编码器集成,进一步提升了多样化用户输入的理解能力.在意图分类任务中,Rasa 将[CLS] 标记的语义嵌入作为整个文本的摘要表示,通过该嵌入向量进行加工任务类型的分类.例如,用户输入“需要外圆加工”,模型将通过[CLS] 表示捕获整体语义,并将其送入一个全连接层,最终通过 Softmax 分类器计算出不同任务类型的概率分布.

在 Rasa 框架的对话管理模块中,CMT-BERT 的任务分类部分与槽值提取高度集成.具体而言,任务分类器基于 Softmax 输出用户的加工意图类别,同时结合实体提取模块,自动填充工艺类型和参数槽位.以下公式描述了任务分类过程:

$$P(t|x) = \text{Softmax}(W_t \cdot h_{[\text{CLS}]} + b_t) \quad (10)$$

其中, $P(t|x)$  是任务  $t$  的概率分布, $h_{[\text{CLS}]}$  是 BERT 编码器生成的[CLS] 向量, $W_t$  和  $b_t$  分别是分类器的权重矩阵和偏置项.

通过这种设计,CMT-BERT 能够在多种加工任务中精准分类用户意图,同时从输入中提取相关参数并填充至 Rasa 的槽位(Slot)管理模块.例如,用户请求“加工外圆工艺,直径 50”,模型不仅能将意图识别为“外圆加工”,还会提取直径参数并更新槽位,工作流程如算法 2 所示.

#### 算法 2.

输入:Text, BERT, Num\_labels neo4j

输出:Intent\_name, Slots

1. Initialize tokenizer using pre-trained BERT
2. Load BERT model for sequence classification with num\_labels
3. Function ProcessMessage(text):
 

```

      inputs ← Tokenize text to tensors
      NoGrad():
          outputs ← model(inputs) // 模型前向传播
          logits ← outputs.logits
          probs ← Softmax(logits) // 计算各类的概率分布
          predicted_class ← argmax(probs)
          intent_name ← GetIntentFromClass(predicted_class)
          slots ← ExtractSlots(text, neo4j) // 从文本提取槽位
          Return intent_name, slots
      End Function
      
```
4. Function ExtractSlots(text, neo4j):
 

```

      entities ← NER(text) // 调用实体识别模块
      slots ← Initialize empty slot dictionary
      For each entity in entities:
          slot_value ← QueryNeo4j(entity) // 查询知识图谱
          UpdateSlot(slots, entity, slot_value)
      End For
      Return slots
      End Function
      
```

#### 3.5.3 加工参数提取与插槽填充

在基于 Rasa 框架的 CMT-BERT 模型中,加工参数的提取与动态插槽填充通过结合命名实体识别(NER)技术与上下文敏感的动态插槽机制,实现了对用户输入的精准解析和灵活响应.CMT-BERT 依托其语义理解能力,从用户输入的自然语言描述中抽取关键加工参数(如进给速度 F、刀具半径 R、加工角度 A 等),并动态映射到 Rasa 的插槽中.这种设计不仅提高了参数提取的灵活性,也确保了不同加工任务对参

数需求的适配性.

Rasa 动态插槽填充机制根据加工任务的上下文,对不同工艺类型灵活调整参数提取策略.当用户描述中缺少某些关键参数时,系统通过任务上下文自动判断所需参数并主动发起补全询问.例如,对于“外圆加工”任务,系统将关注“刀具半径”插槽;而对于“端面加工”任务,则更关注“进给速度”和“加工角度”的补全.

此外,为确保提取参数的准确性,CMT-BERT 结合领域数据校正方法对插槽值进行动态验证.这种结合了动态插槽机制与深度语义理解的参数提取方法,为多样化加工需求提供了智能解决方案,进一步提高了对话系统的任务适应性和交互效率,具体流程如算法 3 所示.

#### 算法 3.

输入:Text, BERT, Tracker, Config

输出:填充完成的插槽参数 Slots

1. ExtractAndCorrectProcessType:
 

```

      // 调用算法 1 进行领域纠正
      corrected_process ← CalculateFinalSimilarity()
      tracker.SetSlot("sub_process_type", corrected_process)
      Return corrected_process_type
      End Function
      
```
2. Function RequiredSlots(tracker):
 

```

      config ← LoadProcessParamsConfig() // 加载工艺配置
      sub_process_type ← tracker.GetSlot("sub_process_type")
      required_params ← GetParams(sub_process_type, config)
      Return required_params // 返回动态插槽列表
      End Function
      
```
3. Function FillSlots(text, tracker, model):
 

```

      required_slots ← RequiredSlots(tracker) // 获取所需插槽
      entities ← NER(text) // 提取实体
      slots ← Initialize empty slot dictionary // 初始化插槽
      For each entity in entities:
          slot_value ← QueryNeo4j(entity) // 获取参数值
          If slot_value is None:
              slot_value ← PerformContextualInference()
          slots[entity] ← slot_value // 填充插槽
      Return slots // 返回填充完成的插槽
      End Function
      
```

#### 3.5.4 输出与后处理

在完成加工任务参数的提取与插槽填充后,CMT-BERT 模型通过工厂模式动态生成对应的 G 代码.该模块根据不同工艺类型实例化特定的加工工艺类,每种工艺类都包含独立的 G 代码生成逻辑,以确保生成代码符合数控机床的执行标准.工厂模式的引入不仅提高了系统扩展性,也为不同工艺类型提供了高度定制化的 G 代码生成能力.

具体而言,系统首先根据加工任务类型动态选择合适的工艺类,例如外圆加工工艺类、端面加工工艺类等.这些工艺类实现了一个统一的 G 代码生成接口,但在具体实现中根据工艺需求采用不同的生成策略.例如,外圆加工工艺类会根据用户输入的刀具半径和进给速度生成对应的循环代码,而端面加工工艺类则侧重于刀具路径和加工角度的精准控制.

在 G 代码生成完成后,系统通过后处理模块对代码进行格式化和优化,确保其在不同数控环境下的兼容性.最终,格

式化后的 G 代码连同加工任务的具体参数返回给用户或直接用于设备指令,保证了从用户交互到设备执行的无缝衔接。工厂模式与后处理模块的结合为 CMT-BERT 提供了一种灵活且高效的输出机制,进一步提升了其在复杂加工任务中的应用价值。

### 3.6 系统部署

针对实际场景中的基于智能体的 G 代码生成问答系统部署,本文提出了一种集成数控机床、PC 端控制与直接网线连接数控系统的本地化部署方案。该方案旨在真实生产环境中,通过直接连接数控系统,实现 G 代码的高效生成、传输与执行,使数控机床能够即时运行生成的 G 代码。系统的部署过程主要由以下 4 个关键组件构成:G 代码生成系统、数控机床、数控系统以及本地控制计算机。

本地控制计算机负责运行 G 代码生成问答系统,通过自然语言解析用户需求,实时生成满足加工要求的高精度 G 代码。

生成的 G 代码通过网线直接传输至数控系统。数控系统对接收到的 G 代码进行解释和编译,并将编译后的指令发送至数控机床。数控机床依据指令执行相应的加工操作,确保任务精准、高效完成。

## 4 实验分析

### 4.1 实验环境

实验环境采用的操作系统为 Windows 11, Python 3.8.5, Rasa 2.8.1, TensorFlow 2.3.4。实验的硬件环境为 Intel Core i9-13900K 处理器,内存 64G,显卡 NVIDIA RTX 4090。实验中,针对模型的不同部分调整了参数,最终设置如表 4 所示。

表 4 参数配置表

Table 4 Parameter configuration

参数名	参数值
Dropout	0.2
Transformer size	256
学习率 (lr)	0.001
多头注意力机制头数	4
LSTM 隐藏单元个数	128
Batch Size	6~8
Embedding Dimension	20
Epochs	100

### 4.2 数据集构建

本文系统聚焦于特定的数控加工任务(例如,生成 G 代码、设计加工工艺等),并且涉及的语句和参数与通用的自然语言处理任务存在较大差异,本文并未使用公开的自然语言理解(NLU)数据集,而是根据数控加工领域的实际需求,构建了一个自定义数据集。

该数据集涵盖了与不同加工工艺相关的多种意图和实体类型,包括但不限于外圆加工、端面加工、切槽、内锥面加工等。

数据集中的每条语句包含了特定的加工工艺类型(如外圆、端面、切槽等)及其对应的参数(如进刀次数、退刀量、进给速度等),系统会自动补全未给出的参数。数据集包含了 100 个与加工基础工艺相关的语句,这些语句从实际的数控

加工场景中收集而来。如图 6 所示为部分训练数据集。

```

-intent: request_processing
examples: |
- 我想加工一个[外圆](process_type),进刀次数是[10](Cn),长度是[50](L),进给速度是[120](F)。
- 请帮我设计一个[外圆](process_type)的G代码,参数是进刀次数[8](Cn),长度[60](L),进给速度[100](F),角度[30](A)。
- 需要加工[外锥面](process_type),参数是圆锥半径[5](R),进刀深度[4](Tr),进给速度[150](F)。
- 我想做一个[端面](process_type)工艺,进刀次数是[12](Cn),进刀量是[3](Lr),最后进刀刀量[CT: 2](CT)。
- 加工[切槽](process_type),宽度是[6](W),刀具宽度是[2.5](Tw),进给速度是[130](F)。
- 帮我生成一个[内锥面](process_type)的G代码,进刀次数[10](Cn),进刀量[0.8](deltaT),速度[100](F)。
- 我需要加工[中心孔](process_type),参数如下:进刀次数[15](Cn),长度[40](L),进刀量[3](Cr)。
- 设计一个[内槽面](process_type),进刀次数是[10](Cn),长度是[50](L),角度是[45](A)。
- 生成一个[内圆](process_type),加工宽度[8](W),刀具宽度[3](Tw),进刀深度[5](Tr)。
- 我想加工一个[内圆](process_type),进刀次数[8](Cn),圆锥半径[6](R),进给速度[140](F)。
- 我需要加工一个[内圆](process_type),进刀次数是[12](Cn),进刀量是[1.0](deltaT),进给速度[90](F)。
- 我想加工一个[外圆](process_type),进刀次数[8](Cn),长度[55](L),进给速度[110](F)。
- 设计一个[外圆](process_type),进刀次数[7](Cn),长度[65](L),角度[20](A),速度[105](F)。
- 加工[外锥面](process_type),圆锥半径[6](R),进刀深度[5](Tr),速度[145](F)。
- 帮我生成[端面](process_type),参数是进刀次数[10](Cn),进刀量[4](Lr),最后一次进刀刀量[CT: 1.5](CT)。
- 我想加工[切槽](process_type),宽度是[7](W),刀具宽度是[3](Tw),进刀深度[6](Tr),速度[125](F)。
- 请帮我做一个[内锥面](process_type),进刀次数[9](Cn),进刀量[1.2](deltaT),速度[115](F)。
- 加工[中心孔](process_type),进刀次数[15](Cn),长度[45](L),进刀量[2](Cr),进给速度[135](F)。

```

图 6 训练数据集图

Fig. 6 Training dataset

### 4.3 实验设计

本实验旨在评估不同类型的模型在加工工艺问题中的意图识别与实体提取的性能,重点考察其准确性与效率。为了全面评估模型的表现,专门设计了一个自动化测试系统,通过循环提问、获取预测结果并计算常用评估指标来进行模型评估。

构建了一个包含多种加工工艺相关问题的语句集合,每个问题涵盖了不同的工艺类型和参数设置。问题种类包括但不限于“外圆加工”、“端面加工”、“切槽”等工艺类型,每种工艺类型的参数也具有多样性,如进刀次数、进给速度、切割深度等。这些问题代表了实际生产过程中可能遇到的典型任务,确保了问题的多样性和广泛性。

实验通过编写自动化脚本进行,该脚本的主要功能包括从问题集合中随机选择问题,向系统发送 HTTP 请求并获取模型的预测结果。每次请求包括一个加工工艺问题,模型根据训练过程中学习到的意图分类和实体提取能力,对问题进行预测并返回分类结果和相关参数。

为衡量模型在任务分类和实体提取中的效果,实验采用了常见的评估指标,包括精确率(Precision)、召回率(Recall)和 F1 分数(F1-score)。具体计算公式如下:

$$\text{Precision} = \frac{\sum_{i=1}^n \text{TP}_i}{\sum_{i=1}^n (\text{TP}_i + \text{FP}_i)} \quad (11)$$

$$\text{Recall} = \frac{\sum_{i=1}^n \text{TP}_i}{\sum_{i=1}^n (\text{TP}_i + \text{FN}_i)} \quad (12)$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (13)$$

其中,  $\text{TP}_i$  为模型预测为正的样本数,  $\text{FP}_i$  为预测为正的负样本数,  $\text{FN}_i$  为预测为负的正样本数。

### 4.4 实验结果

本文提出的 CMT-BERT 模型在多工艺识别任务中的表现优于基准模型。与以下模型进行对比实验:

DIET:基础模型,无稀疏特征与稠密特征提取。

DIET (BERT):采用 BERT 提取稠密特征。

DIET (DistilBERT):采用 DistilBERT 提取稠密特征。

BiLSTM-CRF:经典序列标注模型。

LSTM-DIET 模型:将 BiLSTM 层替换为单向 LSTM 做实体识别。

XLNet:基于自回归预训练模型,提升了在文本理解上的

表现。

**RoBERTa**:通过去掉 BERT 中的部分优化技术,提升了大规模数据训练效果。

**T5**:将文本生成任务与理解任务结合的统一框架。对比结果如表 5、表 6 所示。

表 5 意图识别测试结果表  
Table 5 Intent recognition test results

模型名称	TP	FP	FN	TN	Precision	Recall	F1
DIET	80	15	10	5	0.842	0.889	0.865
BERT	81	14	9	5	0.852	0.900	0.875
DistilBERT	85	12	7	5	0.876	0.924	0.899
BiLSTM-CRF	78	17	12	5	0.821	0.867	0.843
LSTM-DIET	82	4	9	5	0.854	0.901	0.877
XLNet	84	13	8	5	0.866	0.913	0.889
RoBERTa	86	10	6	5	0.896	0.935	0.915
T5	89	8	5	5	0.918	0.947	0.932
本文模型	97	3	1	5	0.970	0.990	0.980

表 6 实体提取测试结果表  
Table 6 Entity extraction test results

模型名称	TP	FP	FN	TN	Precision	Recall	F1
DIET	75	18	12	5	0.806	0.862	0.833
BERT	78	16	10	5	0.830	0.886	0.857
DistilBERT	83	14	6	5	0.856	0.933	0.892
BiLSTM-CRF	70	19	15	5	0.787	0.824	0.805
LSTM-DIET	80	13	7	5	0.860	0.920	0.889
XLNet	84	11	8	5	0.883	0.912	0.897
RoBERTa	87	9	5	5	0.906	0.936	0.921
T5	90	7	4	5	0.926	0.957	0.941
本文模型	92	5	3	5	0.948	0.968	0.958

各模型意图识别、实体识别结果直观对比如图 7 所示,深色柱为意图识别 F1 的值,浅色柱为实体提取 F1 值。

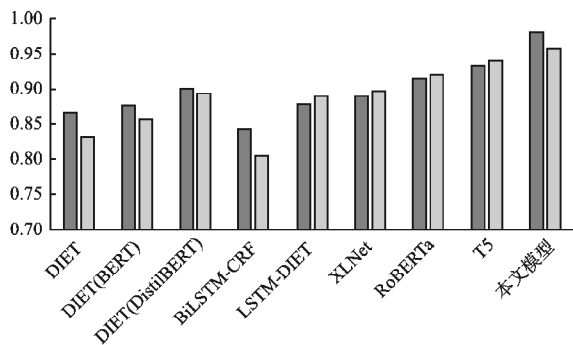


图 7 测试结果图

Fig. 7 Intent recognition test results

本文提出的 CMT-BERT 模型在多工艺识别任务中的表现明显优于现有的基准模型。通过与 DIET、DIET (BERT)、DIET (DistilBERT)、BiLSTM-CRF、LSTM-DIET、XLNet、RoBERTa 以及 T5 模型的对比实验,结果显示 CMT-BERT 在意图识别和实体识别任务中的 F1 值均显著提高。

具体而言,CMT-BERT 在意图识别任务中的 F1 分数为 0.980,显著优于本次实验中 F1 值排名第二的 T5 模型,提升

了 4.8 个百分点。此外,CMT-BERT 在精确度和召回率方面也表现突出,精确度和召回率分别达到 0.970 和 0.990,在提高识别准确性的同时,能够有效减少假阳性和假阴性,体现了其在复杂工艺识别任务中的优势。在实体识别任务中,CMT-BERT 的 F1 分数为 0.958,相较于 T5 模型提高了 1.7 个百分点。这些结果表明,CMT-BERT 模型在多工艺识别任务中具有较高的实用性和应用潜力,能够有效提升系统的智能化水平。综上所述,CMT-BERT 模型不仅在提升任务识别的精确度和召回率方面表现出色,还展现了较好的泛化能力,适用于更为复杂的自然语言处理任务。

## 5 结论

本文针对传统 G 代码生成过程中效率低、易出错等问题,提出了一种基于智能体的数控加工 G 代码生成系统解决方案。研究内容涵盖了自然语言处理、领域数据校正、知识图谱构建、深度学习模型设计与优化以及系统的部署与评估。实验结果表明,所提出的 CMT-BERT 模型在意图识别和实体提取任务中表现优异,F1 值分别达到了 98.0% 和 95.8%,显著优于其他基准模型。通过将该系统应用于实际数控加工场景,实现了 G 代码的自动化生成,在支持简单工艺加工任务中表现出较高的效率和稳定性。

本研究增强了数控系统在简单加工任务中的智能化水平,同时在提高加工路径规划的准确性和生产灵活性方面展现了显著优势。然而,目前系统主要适用于外圆加工、端面加工等简单工艺,对于多轴联动、复杂曲面加工等高难度工艺的支持尚显不足。这在一定程度上限制了系统在高精度、高复杂度加工场景中的应用。

针对这一局限性,未来的研究将聚焦于以下几个方向:首先,可以通过扩展知识图谱的工艺节点和模板库,增强系统对复杂工艺的理解与支持;其次,可引入更强大的神经网络模型,例如结合多任务学习和图神经网络(GNN),以提升模型对复杂工艺特征的提取能力。同时,考虑融合强化学习技术,通过加工过程中实时反馈的数据不断优化生成的 G 代码,从而实现复杂加工工艺的动态适配。

此外,在数据流处理方面,未来可集成 Kafka 或 Flink 等流处理框架,实现对实时数据的高效清洗与预处理,进一步提升系统对复杂工艺加工场景的适应性。随着技术的不断发展,基于智能体的 G 代码生成系统将在复杂加工任务中逐步实现突破,为智能制造和自动化生产提供更广泛的应用支持。

## References:

- [1] Odeh A, Odeh N, Mohammed A S. A comparative review of AI techniques for automated code generation in software development: advancements, challenges, and future directions [J]. TEM Journal, 2024, 13(1): 726-739.
- [2] Dodok T, Čuboňová N, Císar M, et al. Automatic generation of the machining processes for turned components in CAD/CAM system [C]//Advances in Manufacturing Engineering and Materials II, Cham: Springer, 2021: 355-362.
- [3] Ji S, Pan S, Cambria E, et al. A survey on knowledge graphs: repre-

- sensation, acquisition and applications [J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 33(4): 1181-1196.
- [4] TAN W Y, YU D, SUN N, et al. Recognition of agent machining trajectory features in numerical control system [J]. Journal of Chinese Computer Systems, 2024, 45(12): 2817-2822.
- [5] Tsagaris A, Mansour G, Sagris D. CNC machining optimization by genetic algorithms using CAD-based system [J]. International Journal of Modern Manufacturing Technologies, 2013, 5(1): 75-80.
- [6] Lasluisa Naranjo H, Rivas Lalaleo D, Vaquero López J, et al. Machine learning G-Code optimization [J]. Engineering Proceedings, 2024, 77(1): 32-43.
- [7] Zan Lamberger, Mussoni C, Chicaiza Cabezas N, et al. Painting bio: a vector-based method for precise G-code generation across scales in biofabrication [J]. International Journal of Bioprinting, 2024, doi:https://doi.org/10.36922/ijb.6239.
- [8] Nguyen T K, Phung L X, Bui N T. Novel integration of CAPP in a G-code generation module using macro programming for CNC application [J]. Machines, 2020, 8(4): 61-70.
- [9] Mourtzis D, Kaimasidis P, Angelopoulos J, et al. Design and development of a G-code generator for CNC machine tools based on augmented reality (AR) [C]//Product Lifecycle Management, PLM in Transition Times: the Place of Humans and Transformative Technologies, 2022: 378-387.
- [10] XU K, WANG Z Y, WANG X, et al. A survey on task-oriented dialogue strategies based on reinforcement learning [J]. Journal of Computer Science, 2024, 47(6): 1201-1231.
- [11] LIU J M, MENG Y L, WAN X Y. Cross-task dialogue system based on few-shot learning [J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2019, 31(3): 299-304.
- [12] Iga V I R, Silaghi G C. LLMs for knowledge-graphs enhanced task-oriented dialogue systems: challenges and opportunities [C]//International Conference on Advanced Information Systems Engineering, 2024: 168-179.
- [13] Joshi M, Sharma R K. An analytical study and review of open source chatbot framework, RASA [J]. International Journal of Engineering Research & Technology (IJERT), 2020, 9(6): 123-135.
- [14] Wang D S, Wang W M, Wang S, et al. A review of natural language understanding methods for domain-specific question-answering systems [J]. Computer Science, 2017, 44(8): 42-49.
- [15] Wang L C, Lin S K. A multi-agent based agile manufacturing planning and control system [J]. Computers & Industrial Engineering, 2009, 57(2): 620-640.
- [16] Wang L Y, Huang H H, Rae W West, et al. Intelligent manufacturing system of impeller for computer numerical control (CNC) programming based on KBE [J]. Journal of Central South University, 2014, 21(12): 4577-4584.

#### 附中文参考文献:

- [4] 谭雯月, 于东, 孙娜, 等. 一种智能体的数控系统加工轨迹特征识别方法 [J]. 小型微型计算机系统, 2024, 45(12): 2817-2822.
- [10] 徐恺, 王振宇, 王旭, 等. 基于强化学习的任务型对话策略研究综述 [J]. 计算机学报, 2024, 47(6): 1201-1231.
- [11] 刘继明, 孟亚磊, 万晓榆. 基于小样本机器学习的跨任务对话系统 [J]. 重庆邮电大学学报(自然科学版), 2019, 31(3): 299-304.