软件学报 ISSN 1000-9825, CODEN RUXUEW
2025,36(10):4753−4767 [doi: 10.13328/j.cnki.jos.007365] [CSTR: 32375.14.jos.007365]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

针对 LLM 对话属性情感理解的多代理一致性反思*

刘一丁, 王晶晶, 罗佳敏, 周国栋

(苏州大学 计算机科学与技术学院, 江苏 苏州 215006)

通信作者: 周国栋, E-mail: gdzhou@suda.edu.cn

摘　要: 近年来, 针对对话文本的属性情感理解吸引了越来越多研究者的关注, 取得了一定的研究进展. 与已有的研究工作不同, 致力于探索大语言模型在对话属性情感理解任务上的性能, 并且认为对话属性情感理解任务存在属性指代映射问题和属性情感映射问题两个关键挑战, 严重制约对话结构下的属性情感理解的精度. 基于此, 提出大语言模型对话属性情感理解任务. 该任务致力于利用大语言模型抽取包含属性指代映射关系和属性情感映射关系的四元组, 并且标注了一个高质量的对话属性情感理解四元组数据集用于评估大语言模型在该任务上的性能. 进一步地, 针对上述对话属性情感理解存在的两个关键映射关系挑战以及大语言模型固有的幻觉问题挑战, 提出了一种多代理一致性反思方法. 该方法首先设计了 3 个子任务代理, 目的在于通过多代理的方式帮助模型捕捉对话结构下的上述两种映射关系; 其次提出了一致性增强的反思方法, 目的在于让模型通过多代理一致反思生成最优的结果, 以缓解大语言模型幻觉问题. 实验结果表明, 该方法在多个评估指标上优于当前主流的基准方法. 此外, 该方法相较于其他基准方法具有最优的对话属性指代关系抽取和属性情感抽取能力, 这将有力地促进大语言模型在对话结构下的细粒度情感理解方面的研究.

关键词: 对话属性情感理解; 属性指代映射; 大语言模型; 多代理机制; 一致性反思

中图法分类号: TP18

中文引用格式: 刘一丁, 王晶晶, 罗佳敏, 周国栋. 针对LLM对话属性情感理解的多代理一致性反思. 软件学报, 2025, 36(10): 4753–4767. http://www.jos.org.cn/1000-9825/7365.htm

英文引用格式: Liu YD, Wang JJ, Luo JM, Zhou GD. Multi-agent Consistency Reflection for LLM-grounded Conversational Aspect Sentiment Understanding. Ruan Jian Xue Bao/Journal of Software, 2025, 36(10): 4753–4767 (in Chinese). http://www.jos.org.cn/1000-9825/7365.htm

Multi-agent Consistency Reflection for LLM-grounded Conversational Aspect Sentiment Understanding

LIU Yi-Ding, WANG Jing-Jing, LUO Jia-Min, ZHOU Guo-Dong

(School of Computer Science & Technology, Soochow University, Suzhou 215006, China)

Abstract: Recently, aspect sentiment understanding in conversational texts has received growing attention from researchers, leading to notable progress in the field. Unlike prior research, this study focuses on the performance of large language models (LLMs) in conversational aspect sentiment understanding tasks and highlights two primary challenges: aspect-coreference mapping and aspect-sentiment mapping. These challenges pose significant constraints on achieving accurate aspect sentiment understanding within conversational structures. To address these issues, the study defines a new task for LLMs in conversational aspect sentiment understanding, aiming to extract quadruples that encapsulate both aspect-coreference mapping and aspect-sentiment mapping relationships. A high-quality annotated dataset of conversational aspect sentiment quadruples has been created to evaluate the performance of LLMs on this task. To tackle the challenges of mapping relationships and mitigate the inherent hallucination issues of LLMs, this study introduces a multi-agent

* 基金项目: 国家自然科学基金 (62006166, 62376178, 62076175); 江苏高校优势学科建设工程资助项目; 软件新技术与产业化协同创新中心项目
刘一丁和王晶晶为共同第一作者.
收稿时间: 2024-05-12; 修改时间: 2024-07-21, 2024-10-04; 采用时间: 2024-11-13; jos 在线出版时间: 2025-08-01
CNKI 网络首发时间: 2025-00-00

consistency reflection approach. This method involves the design of three sub-task agents to aid in capturing the complex mapping relationships within conversational structures. In addition, an enhanced consistency reflection is proposed, enabling the model to generate optimal results through multi-agent consensus, thus alleviating hallucination problems. Experimental results demonstrate that the proposed approach significantly outperforms state-of-the-art benchmarks. Furthermore, it exhibits superior capabilities in extracting aspect referential relationships and aspect sentiments, contributing to advancements in fine-grained sentiment understanding within conversational contexts using LLMs.

Key words: conversational aspect sentiment understanding; aspect-coreference mapping; large language model (LLM); multi-agent mechanism; consistency reflection

属性级情感理解又称属性级情感分析 (aspect-based sentiment analysis, ABSA), 该任务是一个细粒度的情感分析任务, ABSA 任务在很多领域中都有着较为广泛的应用, 比如电子商务领域和社交舆论挖掘[1]. 早期 ABSA 相关的工作主要集中在普通文本[2,3], 例如评论数据[4], 最近越来越多的工作注意到普通文本的局限性, 例如人们可能在社交媒体以多轮对话的方式谈论某些产品, 明星或者政治内容. 因此目前 ABSA 工作的研究重点也从普通文本转移到对话文本[5,6], 例如问答[7]以及对话[5]等. 近期大语言模型 (large language model, LLM) 的提出, 使得模型对于对话有较强的理解能力, 这也鼓励我们使用 LLM 处理对话场景下的属性级情感理解任务.

在对话场景中, 存在属性指代映射关系和属性情感映射关系这两个极其重要的关系, 这两个关系严重制约对话结构下的属性情感理解的精度. 基于此, 本文提出了大语言模型对话属性情感理解任务. 该任务不仅包括传统的三元组抽取 (即属性-实体、观点描述语、情感极性), 还扩展至属性指代映射关系的识别. 图 1 中的对话所示, 这段对话的第 2 句中出现了观点描述语“很漂亮”, “很漂亮”指向的属性实体为“杨幂”, 然而第 2 句话中还存在“她”, 在模型理解对话时这种代指提及是至关重要的, 因为在对话中属性指代映射关系可能存在较大的跨度, 这加大了模型理解对话文本情感的难度, 但传统的 ABSA 任务忽略了这一点, 因此大语言模型对话属性情感理解任务额外的抽取属性实体“杨幂”的代指提及“她”, 从而得到大语言模型对话属性情感理解任务需要提取的目标四元组 (杨幂, 她, 很漂亮, 积极). 为了评估大语言模型对话属性情感理解任务, 我们标注了一个高质量的对话属性情感理解四元组数据集用于评估大语言模型在该任务上的性能.

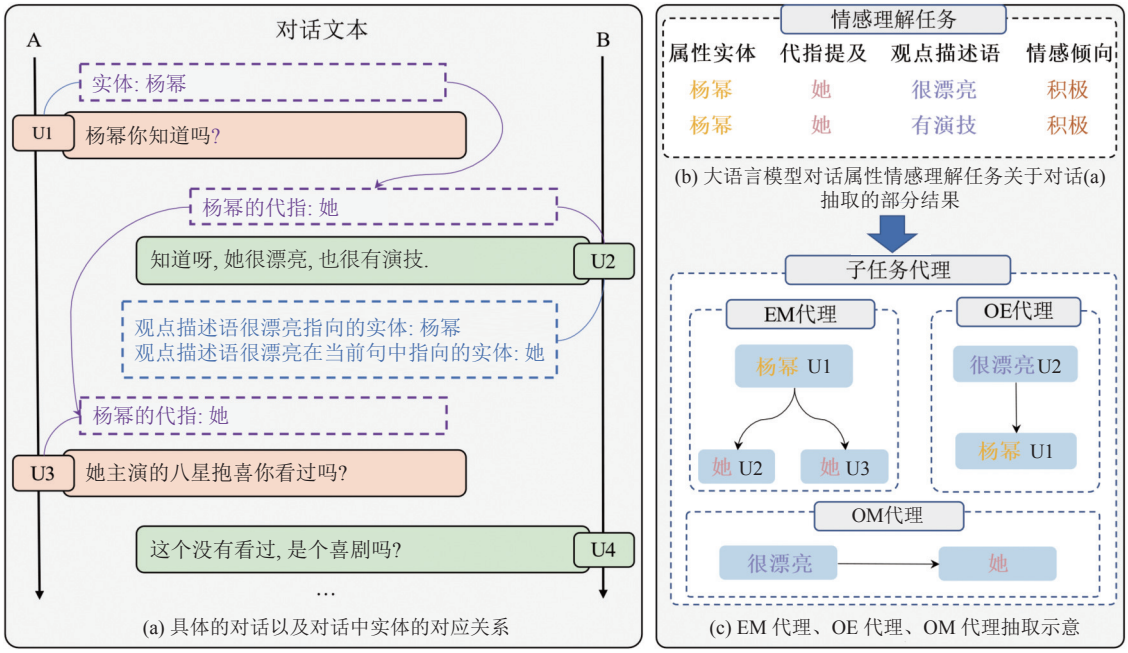

图 1 对话示例以及任务图

本文认为该任务目前存在两大挑战: (1) 对话场景存在的两个重要映射关系, 分别是对话属性实体和观点描述语之间的映射 (属性情感映射), 以及对话属性实体与其指代之间的映射 (属性指代映射), 能否有效地捕捉这两个映射关系决定着模型性能的好坏. (2) LLM 在处理不同任务时存在固有的幻觉现象[8,9], 模型在生成过程中常常会出现幻觉导致模型输出的结果与我们期望不一致. 这两大挑战严重制约对话属性情感理解的性能.

为了解决上述挑战, 本文提出多代理一致性反思方法 (multi-agent consistency reflection approach, MACR). 该方法首先设计了 3 个子任务代理, 通过多代理的方式帮助模型捕捉对话场景下的这两个映射关系, 之后为了缓解大模型的幻觉问题, 该方法设计了一致性增强的反思模块, 该模块通过让模型通过评估情感理解任务和任务代理的一致性来让模型进行反思, 从而让模型生成正确的结果. 具体来说, 我们的方法首先从任务的角度出发, 基于大语言模型对话属性情感理解任务设计出 3 个子任务代理, 通过让模型对子任务进行学习来提升模型在大语言模型对话属性情感理解任务上的性能, 3 个任务代理的抽取目标如图 1 所示. 此外, 我们提出了一致性反思方法, 该方法采用强化学习的思想, 首先根据情感理解任务和子任务代理的结果得到奖励值, 当奖励值低于设定好的阈值时让模型进行反思, 通过反思促使模型生成正确的结果. 我们在标注好的数据集上评估了 MACR 的有效性, 实验结果与分析表明: MACR 的性能显著超过目前主流的基准方法.

综上所述, 本文的主要贡献有以下 3 点.

(1) 为了捕捉对话中的属性指代映射和属性情感映射, 本文提出了大语言模型对话属性情感理解任务, 该任务旨在利用大语言模型抽取包含属性指代映射关系和属性情感映射关系的四元组. 为了评估该任务, 我们标注了一个高质量的对话属性情感理解四元组数据集.

(2) 为了解决对话情感理解任务中的两个挑战, 本文提出了多代理一致性反思方法, 该方法首先设计了 3 个子任务代理, 通过让模型对子任务进行学习提升模型在情感理解任务上的性能. 其次提出了一致性增强的反思模块, 该方法通过分析子任务代理和情感理解任务上的一致性让模型进行反思, 从而促使模型生成一致的结果.

(3) 本文通过实验验证了 MACR 的有效性, MACR 相较于其他基准方法具有最优的对话属性指代关系抽取和属性情感抽取能力, 这将有力地促进大语言模型在对话结构下的细粒度情感理解方面的研究.

1 相关工作

1.1 属性级情感理解

属性级情感理解又称为属性级情感分析. 该任务是一种细粒度的情感分析任务, 旨在对文本中的属性实体、观点描述语、情感极性以及它们之间的关系进行提取. 从任务形式来看, ABSA 任务衍生出了很多工作, 这些工作定义了不同的抽取对象[2,3,10]. 例如 ASTE[2]提出了一个三元组抽取任务, 抽取文本中属性词、观点描述语以及情感极性. ASQP[11]提出了一个四元组抽取任务, 抽取文本中属性词、属性词的种类、观点描述语和情感极性. 但上述工作均基于普通文本, 但在现实中对话场景更为普遍, 近些年很多研究人员注意到了这一问题, 提出了基于对话文本的 ABSA 任务. 例如 DiaASQ[6]从微博评论中构建了一个对话数据集并从中提取目标, 属性, 观点描述语和情感极性四元组. CASA[5]构建了一个对话数据集并从中提取观点描述语, 情感极性以及相应的属性词.

从任务方法来看, 之前很多研究采用序列标注进行处理 ABSA 任务[6,12]. 随着 T5 [13]等生成模型的提出, 许多研究[14,15]开始尝试通过生成方式直接完成抽取任务. 目前解决 ABSA 任务的方法主要分为两类: 基于序列的方法以及基于生成模型的方法.

基于序列的方法将 ABSA 任务转化为序列标注任务或者通过特殊的标记来抽取结果. Xu 等人[16]扩展了 BIOES 标记, 在 B 标签以及 S 标签中添加了位置信息. Wang 等人[17]将双向强化注意力模块作为解码器来捕捉上下文中实体之间的关系. Li 等人[18]注意到当模型遇到属性词有多个观点描述语的问题时, 处理 ASTE 任务的方法可能发生混乱, 为此该工作设计了一个方法对属性词、观点描述语以及对应的情感极性单独进行抽取. Chen 等人[19]注意到之前的方法忽略了词与词之间的关系, 因此他们设计了增强多通道的图卷积网络模型, 通过加入图卷积网

络来捕捉词与词之间的关系, 提升模型的效果. Liang 等人[20]将 ASTE 任务视为多类别 span 标记分类问题, 该方法设计了 3 个标签集合, 通过在标签集合中进行贪心推理来得到三元组. 以上方法将 ABSA 任务视为序列级分类问题, 这类方法虽然取得了不错的效果, 但性能很大程度上受限于预训练模型, 并且存在模型预测过程中没有标签的语义信息的缺陷.

基于生成模型的方法一般采用 T5[13]等生成模型. 这类方法通过模型直接得到需要抽取的信息. Bao 等人[14]通过树型结构的输出来强调需要抽取的实体间的关系. Zhang 等人[11]通过释义的方式得到输出. Mao 等人[21]将元组的生成顺序转化为树上的路径, 采用这种生成方式来减少元组之间的依赖性以及解决一对多问题 (一个属性词对应多个观点描述语). Gou 等人[12]引入一种基于元素顺序的 prompt 学习方法, 通过聚合多视图的结果来提升模型性能. 以上工作通过从数据的角度出发, 让模型输出结果中附带额外信息来提升模型性能.

近些年来很多 ABSA 相关的工作关注于对话的数据, 但遗憾的是, 上述工作大多只考虑了属性情感映射, 其所构建的数据集并没有同时考虑到对话场景下属性指代映射和属性情感映射, 显式地建模这些映射能够帮助模型更好地理解对话中的情感. 因此我们在 ABSA 任务中加入指代映射, 并通过设置子任务代理让模型更好地理解这些实体间的关系.

1.2 基于反思的大语言模型

LLM 在庞大的参数量基础上训练了海量的数据, 因此表现出强大的语言理解能力, 并在广泛的自然语言处理任务中取得了令人印象深刻的结果. 如今 LLM 已经成为 NLP 领域新的范式, 目前许多高校和机构都开源了大语言模型, 比如 Meta 开源的 LLaMA[22]以及智谱 AI 开源的 ChatGLM[23]等. 随着大模型的出现, 近期提出的 agent 大多以 Zero-Shot 的方法通过设计提示词来挖掘 LLM 的能力, Zhang 等人[24]提出一种自动的协作学习方法, 该方法将用户和项目同时作为 agent, 通过两个 agent 之间的交互来解决复杂任务. Xu 等人[25]通过多 agent 之间相互纠正来提升 LLM 处理复杂任务的能力. Liu 等人[26] 采用分而治之的方法, 将复杂任务拆分为若干个难度较小的子任务, 通过解决拆分后的子任务来解决复杂任务, 让代理能够解决现实世界中复杂的任务.

尽管 LLM 对语言有着极强的理解能力, 但其仍然存在较为严重的幻觉问题. 目前很多工作都着重于解决这一问题. ICL (in-context learning)[27]在输入的时候给模型一个示例, 让模型在示例的指导下得到输出. ReAct[28]通过将推理与行动相结合的方法缓解 LLM 中的幻觉问题. Chain-of-thought (CoT)[29]让模型输出答案的同时输出得到答案的推导过程, 通过让模型思考来缓解幻觉问题. ToT[30]通过对模型的思考结果进行评估, 在将生成过程转换为在树形结构上的搜索过程来减少幻觉. Reflexion[31]设计了启发式规则, 当模型输出的答案质量较低时执行反思操作, 通过该方式减少幻觉. 上述工作都能有效地减少大模型的幻觉问题, 其中让模型进行反思的方式挖掘了模型自身的能力, 成为近期研究的热点. 反思是一种让模型评估生成结果并让模型在某些条件下重新生成答案的方法. 目前, 反思成为处理大模型幻觉问题一种常见的方式, 很多工作利用反思减少幻觉现象的发生. Shinn 等人[32]通过设置一个评估器, 当模型的动作评估结果不通过时进行反思. Huang 等人[33]设计了一个评估模型的输出的方法, 该方法通过反思减少了大模型中的幻觉现象.

上述方法极大程度地减少了 LLM 的幻觉问题, 但遗憾的是, 并没有方法考虑到对话场景中 LLM 的幻觉问题. 基于此, 本文引入多个子任务来降低提升模型对于属性级情感任务的理解能力. 此外, 为了解决 LLM 中固有的幻觉问题, 本文提出多代理一致性反思方法. 该方法对模型生成结果的一致性进行评估, 当评估不通过时让模型进行反思, 通过反思让模型生成一致的结果.

2 多代理一致性反思

目前大模型仍然存在难以捕捉对话文本中细粒度的信息的问题[34], 这导致 LLM 并不能在大语言模型对话属性情感理解任务上取得令人满意的性能. 为了解决上述问题, 本文提出了多代理一致性反思方法, 其结构如图 2 所示, 其中, 一致性感知的奖励表示一致性奖励的构建过程. 因为情感理解任务数据集是中文对话数据, 在目前的开源社区中 ChatGLM 具有较强的中文理解能力, 因此本文采用 ChatGLM3 (https://github.com/THUDM/ChatGLM3)[23]

作为基座模型, 通过微调的方式让模型有效地理解和捕捉上下文中的细粒度信息. 同时本文的方法从任务出发, 设计出 3 个子任务代理. 这 3 个代理通过让模型捕捉上下文与句子间的信息来提升对对话的理解能力. 此外, 受 Reflexion[32]的启发, 本文引入强化学习的思想设计出一致性反思模块. 并且为了保证模型在反思过程的稳定性, 我们在反思阶段不修改模型的参数, 而是将存储单元中的内容与 prompt 结合, 以 Zero-Shot 的方式让模型进行反思. 下面就子任务代理和一致性反思的细节予以介绍.

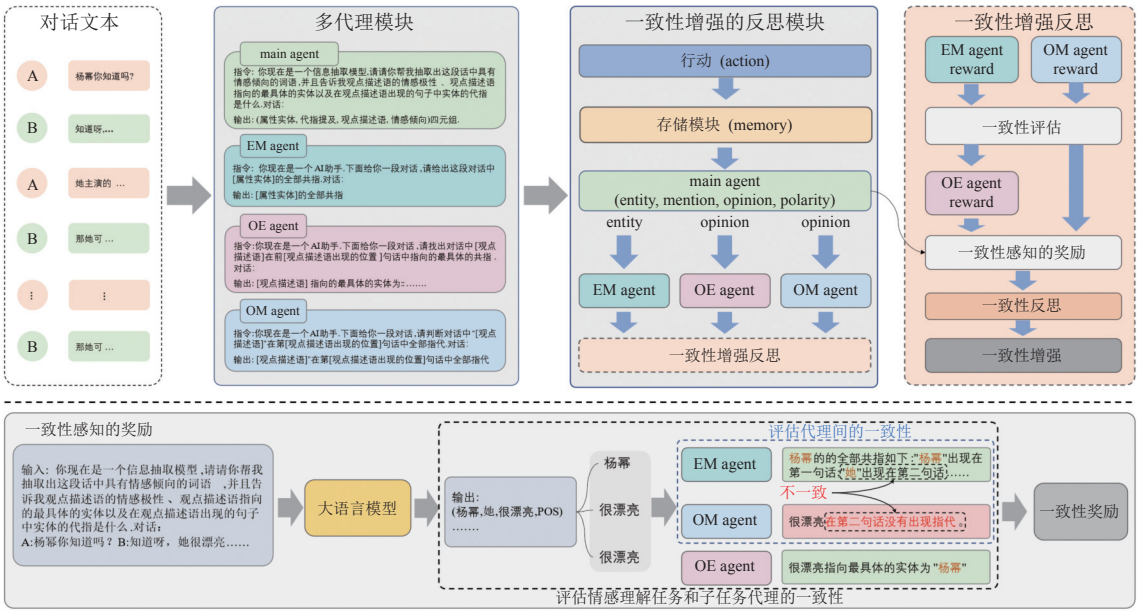

图 2　多代理一致性反思模型结构图

2.1　子任务代理

在介绍本节内容之前, 我们给出大语言模型对话属性情感理解任务的形式化描述. 给定一段对话 $C=\{c_1,c_2,\ldots,c_n\}$, 其中, c_i 表示第 i 句话, n 表示这段对话中话语个数, 大语言模型对话属性情感理解任务的目的是抽取所有的四元组 (e_i,r_i,o_i,p_i), 其中, o_i 是观点描述语, e_i 是 o_i 指向的最具体的属性实体, r_i 是观点描述语出现的句子中 e_i 的指代提及, p_i 是观点描述语 o_i 的情感极性. 特别地, 当观点描述语出现的句子中没有出现属性实体 e_i 的代指时, 我们将指代提及标记为“null”.

为了让模型更好地理解任务, 我们设计了 3 个子任务代理, 通过将难度较高的情感理解为若干个难度较低的子任务, 降低模型的学习难度. 我们通过任务的定义, 从四元组抽取任务中划分出 3 个子任务: (1) 抽取对话中的属性实体 (entity) 与代指提及 (mention) 之间的关系 (EM 代理); (2) 抽取对话中的观点描述语 (opinion) 与属性实体 (entity) 之间的关系 (OE 代理); (3) 抽取观点描述语出现的句子中观点描述语 (opinion) 与指代提及 (mention) 之间的关系 (OM 代理). 这 3 个代理旨在让模型更好地理解任务定义. 具体而言, EM 代理和 OE 代理捕捉对话中的属性指代映射. OM 代理捕捉当前语句中的属性情感映射. 每个代理抽取的信息如图 2 中多代理模块所示, 形式化描述如下.

● 属性实体-代指提及代理 (entity-mention agent, EM agent). 给定一段对话 $C=\{c_1,c_2,\ldots,c_n\}$, 其中, c_i 表示第 i 句话, n 表示这段对话中对话的个数. EM 代理的目的是在给定属性实体的情况下, 找出这段对话 C 中全部的指代以及找出指代出现的位置.

● 观点描述语-属性实体代理 (opinion-entity agent, OE agent). 给定一段对话 $C=\{c_1,c_2,\ldots,c_n\}$, 其中, c_i 表示第 i 句话, n 表示这段对话中对话的个数. OE 代理的目的是在给定观点描述语 o_i 的情况下, 找到 o_i 指向的属性实体 e_i

以及 e_i 最早出现的位置.

● 观点描述语-代指提及代理 (opinion-mention agent, OM agent). 给定一段对话 $C=\{c_1,c_2,\ldots,c_n\}$, 其中, c_i 表示第 i 句话, n 表示这段对话中对话的个数. OM 代理的目的是在给定观点描述语 o_i 的情况下, 找到 o_i 出现的这句话中所有的指代关系 (包括属性实体 e_i 以及相应的指代提及 r_i).

Agent 解决复杂任务时都凭借大模型自身的能力以 Zero-Shot 的方式处理任务, 这种方法需要设计复杂的 prompt 以及处理流程, 并且 prompt 设计的好坏极大地影响了模型的性能. 与 agent 不同, 本文采用低资源微调的方式, 通过一个 Adapter 将任务本身集成到模块中. 这种方法相较于 Zero-Shot 模型性能更加稳定. 同时受指令学习[35]的启发, 本文通过向模型加入指令显式地列出任务定义, 提升模型对任务的理解. 我们将指令和输入的文本进行拼接, 作为微调模型的输入, 大语言模型对话属性情感理解任务以及各个子任务的具体指令如下.

● 大语言模型对话属性情感理解代理 (主代理 (main agent)). “你现在是一个信息抽取模型, 请你帮我抽取出这段话中的观点描述语, 并且抽取出观点描述语的情感极性、观点描述语指向的最具体的实体以及实体在观点描述语出现句子中的代指. 你需要将抽取结果封装到四元组中返回给我, 四元组中的内容分别是: (最具体的实体, 实体的代指, 观点描述语, 观点描述语的情感极性), 如果抽取出多个四元组, 每个四元组之间以分号进行分隔.”

● EM 代理. “你现在是一个信息抽取模型. 下面给你一段对话, 你要抽取出这段对话中 [属性实体] 的全部共指.”

● OE 代理. “ 你现在是一个信息抽取模型. 下面给你一段对话, 你要抽取出对话中观点描述语 [观点描述语] 在前 [观点描述语出现的位置] 句话中指向的最具体的实体.”

● OM 代理. “ 你现在是一个信息抽取模型. 下面给你一段对话, 你要抽取出对话中观点描述语 [观点描述语] 在第 [观点描述语出现的位置] 句话中的指向的全部实体.”

大语言模型对话属性情感理解任务以及各个子任务的输出如下.

● 大语言模型对话属性情感理解任务. (属性实体, 代指提及, 观点描述语, 情感倾向) 四元组, 四元组的数量为对话文本中观点描述语的个数.

● EM 代理. “对 [属性实体] 的共指抽取结果如下: \n[指代词 1] 出现在第 [指代词 1 出现的位置] 句话\n[指代词 2] 出现在第 [指代词 2 出现的位置] 句话”. 指代词指观点描述语指向的所有实体, 包括属性实体以及代指提及.

● OE 代理. “[观点描述语] 指向的最具体的实体为: [属性实体]”.

● OM 代理. “这句话中 [观点描述语] 指向的实体为: [指代词 1], [指代词 2]”. 指代词指观点描述语指向的所有实体, 包括属性实体以及代指提及.

2.2 一致性增强的反思

当模型在一次生成中无法产生良好的响应, 反思是一种有效的补救措施. 一致性增强的反思模块采用强化学习的思想, 通过生成奖励值来调节模型的动作. 同时为了解决模型在强化学习中训练不稳定的情况, 受 Reflexion[32] 的启发, 我们采用 Zero-Shot 的方式通过设计 prompt 来让模型实现反思. 一致性增强的反思过程如图 2 一致性增强的反思模块所示, 对于给定的输入, 模型执行相应的动作, 存储模块存储本次的输入以及模型的生成结果, 之后模型抽取四元组中的属性实体以及观点描述语, 得到各个代理的输入, 通过评估属性情感理解任务以及代理生成结果的一致性来生成奖励, 越低的一致性就分配越低的奖励值, 当奖励值低于阈值时让模型根据生成的结果进行反思. 例如, 当模型抽取的四元组中属性实体与观点描述语之间的对应关系错误时, 评估器生成一个较低的奖励值从而触发阈值, 之后我们通过 prompt 诱导模型根据之前的历史信息进行反思, 重新生成结果. 形式化如下: 在两个状态 s^t_{main} 和 s^t_{oe} 下, 其中 t 代表着第 t 个时间步, 模型根据策略 $\pi(a_{\text{main}}|s^t_{\text{main}})$ 和 $\pi(a_{\text{oe}}|s^t_{\text{oe}})$ 执行两个不同的动作 a_{main} 和 a_{oe}, 模型根据动作 a_{main} 和 a_{oe} 的一致性来确定是否反思. 代理的反思过程如下.

● 动作 (action). 我们通过微调后的模型得到大语言模型对话属性情感理解任务的结果, 形式化如下:

$$\mathrm{Q}_{\mathrm{main}} = \mathrm{LLM}(\mathrm{I}_{\mathrm{main}}) \tag{1}$$

其中, $\mathrm{I}_{\mathrm{main}}$ 是大语言模型对话属性情感理解任务的输入, $\mathrm{Q}_{\mathrm{main}}$ 是主任务代理生成的结果.

● 存储模块 (*memory*). 该模块是多代理反思的核心, 模型需要根据存储模块储存的历史信息进行反思, 该模块用于存储模型生成的历史信息, 因此我们设计了存储模块来保存用户的输入以及模型的输出, 模块的形式化描述如下:

$$memory = Concat(\mathrm{I}_{\mathrm{main}}, \mathrm{Q}_{\mathrm{main}}) \tag{2}$$

其中, $\mathrm{I}_{\mathrm{main}}$ 和 $\mathrm{Q}_{\mathrm{main}}$ 分别是大语言模型对话属性情感理解任务的输出和输入, *Concat*() 是字符串拼接操作.

在得到 $\mathrm{Q}_{\mathrm{main}}$ 之后我们对四元组中的观点描述语和属性实体进行提取, 得到观点描述语集合 $O = \{o_1, o_2, \ldots, o_m\}$ 以及属性实体集合 $E = \{e_1, e_2, \ldots, e_m\}$, 其中 o_i 表示第 i 个四元组的观点描述语, e_i 表示第 i 个四元组的属性实体, m 表示生成结果中四元组的个数. 之后我们将 O 和 E 与对话文本进行合并, 作为 EM 代理、OE 代理、OM 代理的输入得到输出 Q_{oe}、Q_{om}、Q_{em}, 其中, $\mathrm{Q}_{\mathrm{oe}} = \mathrm{LLM}(O)$, $\mathrm{Q}_{\mathrm{om}} = \mathrm{LLM}(O)$, $\mathrm{Q}_{\mathrm{em}} = \mathrm{LLM}(E)$.

● 一致性感知的奖励 (consistency-aware reward). 奖励值的构建过程如图 2 中一致性感知的奖励所示. 奖励值大语言模型对话属性情感理解任务的输出 $\mathrm{Q}_{\mathrm{main}}$ 以及各个代理的输出 Q_{oe}、Q_{om}、Q_{em}. 我们从 $\mathrm{Q}_{\mathrm{main}}$ 中抽取出大语言模型对话属性情感理解任务中属性实体与观点描述语的二元关系 $(O_{\mathrm{main}}, E_{\mathrm{main}})$, 属性实体与观点描述语与代指提及的二元关系 $(O_{\mathrm{main}}, M_{\mathrm{main}})$ 以及代指提及的二元关系 $(E_{\mathrm{main}}, M_{\mathrm{main}})$, 将这些二元关系分别与 Q_{oe}、Q_{om}、Q_{em} 进行对比, 得到评估结果. 具体来说, 从 Q_{oe} 中抽取出 $(O_{\mathrm{main}}, E_{\mathrm{oe}})$, 从 Q_{om} 中抽取出 $(O_{\mathrm{main}}, E_{\mathrm{om}})$ 以及从 Q_{em} 中抽取出 $(E_{\mathrm{main}}, M_{\mathrm{em}})$. 奖励值的计算公式如下:

$$R_{\{\mathrm{oe,om,em}\}} = \frac{\{OE_{\mathrm{right}}, OM_{\mathrm{right}}, EM_{\mathrm{right}}\}}{\{|OE_{\mathrm{main}}|, |M_{\mathrm{main}}|, |M_{\mathrm{main}}|\}} \tag{3}$$

其中, OE_{right} 指 E_{main} 与 OE 代理中属性实体正确的匹配个数, OM_{right} 和 EM_{right} 分别表示 M_{main} 与 OM 代理和 EM 代理中代指提及正确的匹配个数.

● 一致性评估 (consistency evaluation). 因为 EM 代理与 OM 代理都是解决对话中的指代关系, 因此对 EM 代理和 OM 代理进行一致性评估, 当 EM 代理与 OM 代理不一致时, 我们认为模型在指代关系的抽取中发生了不一致的情况, 并修改最终的奖励值, 经过一致性评估的奖励值形式化如下:

$$\begin{cases} R = 0.5 \times R_{\mathrm{oe}} + 0.25 \times (R_{\mathrm{om}} + R_{\mathrm{em}}), & \text{if } OM_{\mathrm{right}} = EM_{\mathrm{right}} \\ R = R_{\mathrm{oe}}, & \text{else} \end{cases} \tag{4}$$

一致性评估过程如上述公式所示, 当 OM 代理和 EM 代理预测结果一致时, 我们将其产生的奖励加入最终的奖励中, 否则我们任务子任务代理间发生了不一致, 此时只采用 OE 代理的奖励值作为最终的奖励值. 在得到奖励值后, 我们通过阈值 α 来决定模型是否进行反思, 当评估器的输出低于 α 时, 执行反思操作, 其中, α 是超参数, 本文设置为 0.5. 具体的反思以及存储模块如下.

● 一致性增强的反思 (consistency reflection). 当奖励值低于 α 时, 我们认为模型发生了不一致的情况, 这时候调用一致性反思模块进行反思. 一致性反思通过 prompt 在不改变模型参数的情况下让模型对之前的输入进行反思. prompt 如下: “经过评估, 你之前抽取的结果存在较大的偏差, 请你确认之前抽取的结果正确性, 删除没有把握的抽取结果并重新抽取有把握的结果.”之后将 prompt 与模型输入拼接, 得到最终的输入 $\mathrm{I}_{\mathrm{self}}$.

最后, 我们将一致性反思的输入与存储模块的输入拼接作为模型的输入, 通过 prompt 让模型进行反思, 得到最终一致性反思后的结果, 形式化如下:

$$\mathrm{Q}_{\mathrm{reflection}} = \mathrm{LLM}(Concat(\mathrm{I}_{\mathrm{self}}, memory)) \tag{5}$$

● 一致性增强 (consistency-enhancing): 在模型进行反思后, 不一定能抽取正确的结果, 为了避免这种现象, 我们通过设计强制反思来确保生成结果的一致. 具体来说, 我们对 $\mathrm{Q}_{\mathrm{reflection}}$ 进行评估, 当模型输出仍然不一致时, 对各个代理生成的结果进行合并, 使用 agent 生成的结果替换掉情感理解任务的结果, 从而提高模型生成结果的一致性. 算法流程如算法 1 所示.

算法 1. 多代理一致性反思.

1. 初始化 actor、EM 代理、OE 代理、OM 代理: A、A_{em}、A_{oe}、A_{om}
2. 初始化策略 $\pi_\theta(a_i|s_i)$, $\theta=\{A, memory\}$
3. 基于 π_θ 生成轨迹 Q, 得到四元组 Q_{main}, 其中, Q_{main} 为微调后模型抽取四元组的结果
4. 设置 $memory$←Q, 将 Q 保存到存储模块当中
5. 对四元组 Q_{main} 进行抽取, 得到 Q_{main} 中所有的实体和观点描述语: $E=\{e_1,e_2,\ldots,e_n\}, O=\{o_1,o_2,\ldots,o_n\}$
6. 将 E 和 O 作为输入, 通过 EM 代理, OE 代理和 OM 代理得到代理的输出: Q_{em}、Q_{oe}、Q_{om}
7. 评估 $\text{Output}_{\text{main}}$ 与 $\text{Output}_{\text{em}}$、$\text{Output}_{\text{oe}}$、$\text{Output}_{\text{om}}$, 生成奖励 R
8. **if** $R<\alpha$ //α 为模型是否反思的阈值
   基于 π_θ 以及 $memory$, 生成 Q′, Q′表示模型进行反思后的结果
   使用 $\text{Output}_{\text{em}}$、$\text{Output}_{\text{oe}}$、$\text{Output}_{\text{om}}$ 修改 Q′, 得到 Q″, Q″表示通过子任务代理修改后的结果
   将 Q″作为模型的预测结果, 输出 Q″
9. **else**
   将 Q 作为模型的预测结果, 输出 Q

2.3 模型优化目标

我们在主代理以及 3 个子任务代理上使用交叉熵损失微调 ChatGLM3, 损失函数如下所示:

$$L^{(\{\text{main,em,oe,om}\})}=-\sum_{i=1}^{N}\sum_{j=1}^{K}\{y_{ij}\log(\hat{y}_{ij}),w_{ij}\log(\hat{w}_{ij}),p_{ij}\log(\hat{p}_{ij}),q_{ij}\log(\hat{q}_{ij})\} \tag{6}$$

其中, y_{ij} 和 $\hat{y}_{ij}$ 分别是大语言模型对话属性情感理解任务的标签和预测结果, w_{ij} 和 $\hat{w}_{ij}$ 分别是 EM 的标签和预测结果, p_{ij} 和 $\hat{p}_{ij}$ 分别是 OE 的标签和预测结果, q_{ij} 和 $\hat{q}_{ij}$ 分别是 OM 的标签和预测结果, $L^{(\text{main})}$、$L^{(\text{em})}$、$L^{(\text{oe})}$、$L^{(\text{om})}$ 分别是大语言模型对话属性情感理解任务、EM 任务、OE 任务和 OM 任务的损失函数, 最终的损失函数为 $L=L^{(\text{main})}+\beta(L^{(\text{em})}+L^{(\text{oe})}+L^{(\text{om})})$, 其中, β 为超参数, 防止辅助任务训练数据过多导致大语言模型对话属性情感理解任务训练效果不佳.

3 实验设置

3.1 大语言模型对话属性情感理解数据集构建

大语言模型对话属性情感理解任务需要同时抽取属性实体、代指提及、观点描述语以及情感极性这 4 部分的信息. 为了实现简单而有效的评估, 本文在 CASA[5]的基础上重新标注构建了一个高质量的四元组抽取数据集. CASA[5]数据集共包含 3 000 段中文的二元对话以及 27 062 句话. 数据集中的对话主要涉及娱乐领域, 包括电影、明星以及电视节目等, 因此该数据集中存在的大量的属性实体以及代指关系. 该数据集能够有效地评估模型在对话场景下对细粒度信息的抽取能力. 值得注意的是, DiaASQ[6]也是对话领域的数据集, 不过该数据集数据来源于微博, 该场景下的对话具有明显的层级关系, 因此并不能将其归于对话场景. 下面就该数据集以及数据集的构建过程予以介绍.

● 属性实体

1) 我们只标注那些有观点描述语指向的实体, 例如图 1 的对话中没有出现观点描述语指向“八星报喜”, 因此我们不标注该实体.

2) 随着对话的进行, 如果在一个属性实体之后出现了另一个更具体的属性实体, 我们也将其标为属性实体. 例如图 1 对话 U5 中, 第 1 次出现的属性实体为“有个演员”, 但随着对话的进行, U5 中出现了指代更明确的属性实体“田蕊妮”, 因此我们将“田蕊妮”标注为属性实体. 特别地, 因为对话具有时序性, 在我们标注“田蕊妮”为属性实体

后, “有个演员”不会被修改为代指提及.

3) 本文只标注评价对象, 对于对话者本身的一些属性不做标注, 比如对话中对话者关于自身的评价“我真笨”, 那么我们不会标注“我”, 并且也不会标注观点描述语“真笨”.

● 代指提及

我们将对话中具体程度不如属性实体的提及标注为代指提及, 比如图 1 的对话中, 在第 2 句和第 3 句中出现了“她”, 但是“她”的具体程度不如第 1 句出现的“杨幂”, 因此我们将其标注为代指提及.

● 情感倾向

我们根据观点描述语的情感程度将情感倾向划分为积极、消极和中性 3 大类.

特别地, 因为 OE 代理的定义, 一段话中相同的属性实体可能被标注多次 (因为一句话中可能会多次出现同一属性实体). 例如一句话如果出现了两次“杨幂”, 那么我们对这两个“杨幂”都进行标注.

数据集由 10 名领域内专业人员进行标注, 当标注结果发生不一致时, 有一名评估专家会做出最后的评估. 标注好的数据集中一共有 11 300 个四元组, 其中包含 11 266 个属性实体以及 7 675 个代指提及, 为了验证模型的能力, 我们按照 8:1:1 的比例从数据集中划分出训练集, 验证集和测试集. 最终的数据集中大语言模型对话属性情感理解任务的四元组数量以及子任务数量如表 1 所示, 其中, Main agent、EM agent、OE agent、OM agent 表示情感理解任务、EM 任务代理、OE 任务代理、OM 任务代理的训练数据量. 参考 Cai 等人[36]的工作, 我们采用两位标注者之间的 *Macro-F*1 作为数据标注中一致性衡量指标, 标注好的数据集中属性实体, 代指提及以及观点描述语的 *F*1 分数为分别为 89.50%、72.51% 以及 86.77%.

表 1 数据集统计表

Split	Utterances (dialogue)	Sentiment	Main agent	EM agent	OE agent	OM agent
Train	21 612 (2 400)	8 967	8 967	5 852	8 970	8 970
Dev	2 727 (300)	1 131	1 131	758	1 131	1 131
Test	2 723 (300)	1 202	1 202	757	1 203	1 203

3.2 超参数设置以及评估标准

我们使用第 3.1 节中标注的数据集来评估模型在大语言模型对话属性情感理解任务上的性能. 为了验证 MACR 框架的有效性, 我们采用 LoRA[37]的方式微调 ChatGLM3-6B-Base (https://huggingface.co/THUDM/chatglm3-6b-base). 在 LoRA 模块中, 矩阵的秩, 缩放因子和丢弃率分别被设置为 12、32 和 0.1. 模型采用单张 NVIDIA Tesla A100 40 GB GPU 在数据集中迭代 5 次, 模型每一批处理的数据量为 2, 训练时采用 DeepSpeed (https://github.com/microsoft/DeepSpeed) 进行训练. 同时模型训练采用 Adam[38]优化器, 学习率为 2E–4, 权重衰减为 5E–4, 模型采用半精度进行训练. 在反思阶段, 我们设置阈值 α 为 0.5. 在最终的损失函数中 β 设置为 0.2.

参考之前的工作[6], 我们使用 *Macro-F*1 作为评估模型的指标, 使用 t 检验[39]评价两种方法之间性能差异的显著性. 同时, 我们通过 3 个方面来评估模型的性能. (1) 单实体匹配: 单独地抽取四元组中的每一个元素, 即单独地对属性实体、代指提及、观点描述语以及情感倾向进行抽取, 单实体匹配用于评估模型的实体抽取能力. (2) 对匹配: 抽取四元组中的元素对 (属性实体-代指提及对、属性实体-观点描述语对以及代指提及-观点描述语对), 对匹配用于评估模型理解属性指代映射和属性情感映射的能力. (3) 四元组匹配: 抽取完整的四元组. 在衡量模型性能时, 只有四元组完全匹配才被认为正确, 四元组匹配用于评估模型在大语言模型对话属性情感理解任务上的性能. 模型对单个元素抽取时, 可能会出现内容不同但是表达相同含义的情况, 比如对于观点描述语“很好看”、 模型抽取“很好看”和“好看”都表达了积极的情感, 因此在四元组的单个元素中, 我们采用模糊匹配的方式. 在本次实验中, 我们将模糊匹配的阈值设置为 2, 即如果预测结果和标注结果字符差异小于等于 2, 我们会认为模型单实体预测正确.

3.3 基准方法

本文在第 3.1 节构建的数据集中比较 MACR 与其他基准方法, 以全面评估 MACR 的有效性. 因为选取的方法包括基于 RoBERTa 等预训练模型, 这些模型在参数量上远远小于大模型, 将这些方法放在一起比较是不公平的,

因此我们将选取的基准方法分为基于传统预训练语言模型的方法和基于大语言模型的方法两大类, 具体如下所述.

(1) 基于传统预训练语言模型的方法

基于传统预训练语言模型的方法使用参数量较小的模型以微调全部参数的方式训练模型.

● T5[13]. T5 使用 encoder-decoder 架构采用生成的方式来得到结果, 在本次实验中我们采用和 Zhang 等人[40]一样的实验设置.

● DiaASQ[6]. DiaASQ 将 RoBERTa 作为编码器, 引入了一个多视图的交互层来增强模型对对话的理解, 在多视图的交互层中引入 3 个注意力掩码矩阵来捕捉对话中的信息. 此外, 该文将四元组关系抽取转化为不同词元对的关系矩阵. 本次实验中我们采用和该工作一样的实验设置.

(2) 基于大语言模型的方法

基于大语言模型的方法主要被分为两类: 1) 以 Zero-Shot 的方式通过提示词直接获得结果; 2) 通过低资源微调的方式获得结果. 具体介绍如下.

● ChatGPT (https://platform.openai.com/docs/overview) (Zero-Shot). 在这条基线中, 我们采用 GPT-3.5 Turbo 复现了 ChatIE[41]中提出的方法. 在此方法中, 先对观点描述语进行提取, 之后分别提取出观点描述语指向的属性实体、代指提及以及观点描述语的情感极性.

● ChatGPT (ICL). 在这条基线中, 我们采用 GPT-3.5 Turbo, 以 ICL 的方式得到结果. 具体来说, 在每一条输入数据中, 我们从训练集选取一条数据作为上下文示例.

● MACR (Zero-Shot). 在这条基线中, 我们采用 GPT-3.5 Turbo, 以 Zero-Shot 的设定进行多代理反思. 具体来说, 首先通过 ICL 的方式得到出主任务和 3 个代理任务的结果, 之后采用一致性增强的反思来得到最终的抽取结果.

● ChatGLM3. ChatGLM3 基于 GLM[23]的一款大模型, 在 GLM 中, 模型通过自回归空白填充进行训练. ChatGLM3 在训练过程中训练了大量的中文数据, 因此该模型对中文有着较强的理解能力, 在这条基线中, 我们采用与 T5 一样的范式, 通过 LoRA 来微调 ChatGLM3.

4 实验分析

在实验结果中, 不同的初始化值对模型性能有一定的影响. 因此本文的每条基线中采用不同的随机种子训练 3 次, 采用 3 次结果的平均值作为最终的实验结果. MACR 模型和其他基准方法在数据集上的性能如表 2 所示, 其中, entity、mention 表示属性实体以及代指提及的抽取结果, OE、EM、OM 分别表示属性实体-观点描述语对、属性实体-代指提及对、代指提及-观点描述语对的抽取结果, OM、EM、OE 分别表示 OM 代理、EM 代理以及 OE 代理. 通过实验结果, 可以得到以下信息.

表 2 MACR 与其他基准方法以及消融实验的实验结果 (%)

方法	单实体匹配				对匹配			四元组匹配
	entity	mention	opinion	ploarity	OE	EM	OM	
T5	65.39	70.00	66.67	80.90	55.75	56.18	57.63	48.25
DiaASQ	60.36	59.38	61.58	73.42	52.16	43.43	44.50	37.21
ChatGPT (Zero-Shot)	47.65	55.60	41.88	64.99	27.43	31.05	26.71	22.38
ChatGPT (ICL)	50.39	45.67	56.69	61.42	44.09	48.82	42.52	40.94
MACR (Zero-Shot)	52.35	67.53	59.74	75.32	46.75	49.35	44.15	43.15
ChatGLM3	68.17	70.74	68.89	82.82	58.35	57.95	59.24	50.46
MACR	**73.29**	**73.21**	70.58	82.83	**62.39**	**62.12**	**61.79**	**54.31**
MACR w/o reflection	72.23	**73.46**	70.59	82.86	61.19	60.78	61.51	53.02
MACR w/o OM agent	72.28	71.10	**74.09**	**85.91**	62.38	57.95	61.47	51.49
MACR w/o EM agent	73.18	73.26	70.70	82.48	60.39	60.70	59.84	52.48
MACR w/o OE agent	70.04	67.26	69.33	79.01	61.70	58.21	59.37	52.64

4.1 实验结果与分析

(1) 采用训练的方式微调模型的方法 (T5、DiaASQ、ChatGLM3、MACR) 性能优于基于 Zero-Shot 的方法, 这鼓励我们采用微调的方式来处理大语言模型对话属性情感理解任务.

实验结果表明, 基于微调的方法 (ChatGLM3、MACR、T5) 的性能显著超越基于 Zero-Shot 的方法 (ChatIE、ICL). 其中, T5 在大语言模型对话属性情感理解任务上的性能比 ChatGPT (Zero-Shot) 高 25.87%, 比 ChatGPT (ICL) 高 7.31%. 这说明, 在大语言模型对话属性情感理解任务中, 因为需要从对话中抽取出四元组, 因此任务难度较高, 采用 Zero-Shot 的方式模型不能很好地理解任务. 值得注意的是, 相较于传统的 ICL 设定, MACR 的 Zero-Shot 实现方式虽然取得了不错的效果, 但并不能得到令人满意的性能, 这鼓励我们采用微调的方式来解决大语言模型对话属性情感理解任务.

(2) 模型加入子任务代理后, 模型在单实体匹配中性能远远超过基于预训练模型的方法以及基于 LLM 的方法 (p-*value*<0.01), 这说明子任务代理帮助模型有了最优的实体抽取能力.

ChatGLM3 加入辅助任务后, 模型在单实体匹配中分别比 T5 和 DiaASQ 高 4.98% 和 12.03%, 这也验证了大模型具有优秀的信息抽取能力. 此外, 在 ChatGLM3 加入子任务代理后, 单实体匹配相较于 ChatGLM3 提升了 2.13%, 其中属性实体以及代指提及的性能提升了 4.06% 和 2.72%, 这与子任务代理的目的相符, 即抽取出观点描述语后抽取出观点描述语指向的属性实体以及代指提及, 更表明子任务代理范式的有效性.

(3) MACR 在对匹配中的性能相较于其他基线方法有显著提升 (p-*value*<0.01), 这表明 MACR 有着最优的实体关系匹配能力.

对匹配中, MACR 相较于基于预训练模型的方法 T5 和 DiaASQ, 平均性能分别提升了 5.58% 和 15.40%, 相较于基于大模型 Zero-Shot 的方法, 性能分别提升了 33.70% 和 16.96%, 这表明了 MACR 优秀的关系匹配能力. 相较于 MACR 在单实体匹配中性能比 ChatGLM3 高 2.32%, 在对匹配中性能分别比 ChtaGLM3 高 3.59%, 这充分表明了子任务代理以及反思能够有效地提升帮助模型捕捉对话中的属性指代关系和属性情感关系, 也表明了子任务代理以及反思的有效性.

(4) MACR 在四元组抽取中 $F1$ 分数显著超越其他基准方法, 这说明子任务代理以及反思能有效地帮助模型理解任务, 从而提升模型的性能.

在四元组抽取结果中, 基于 MACR 框架的模型性能显著超过其他方法 (p-*value*<0.01), 这说明 MACR 有着最优的情感融合能力. 当模型加入子代理时, 模型在大语言模型对话属性情感理解任务的性能相较于 ChatGLM3 提升了 2.56%, 这表明了子任务代理能够有效地让模型通过对子任务进行学习来提升大语言模型对话属性情感理解任务的性能. 并且我们可以注意到, MACR 在单实体匹配中性能相较于 ChatGLM3 提升了 2.32%, 但在四元组抽取的结果比 ChatGLM3 高 3.85%, 这表明我们的方法可以让模型更有效地捕捉对话文本实体间关系, 也验证了 MACR 的有效性.

4.2 消融实验与分析

为了验证我们提出模块的有效性, 本节针对 MACR 核心模块以及每个子任务进行消融实验, 从而进一步验证 MACR 各个模块的效果, 消融实验的结果如表 2 所示. 具体而言, MACR w/o reflection 即只采用子任务代理微调模型得到结果, MACR w/o OM agent、MACR w/o EM agent、MACR w/o OE agent 表示我们分别删除了 OM 代理、EM 代理、OE 代理以及反思模块. 我们通过删除单个子任务来评估单个任务对模型性能的影响. 接下来展开具体分析.

当反思模块被删除时, 单实体匹配的性能平均下降了 0.19%, 对匹配的性能分别平均下降了 0.94%, 四元组抽取的结果下降了 1.29%, 这说明反思模块能有效评估模型生成结果中实体关系, 通过提升实体关系之间的一致性来提升模型四元组抽取的一致性, 从而提升模型在大语言模型对话属性情感理解任务上的性能.

当反思模块以及 OM 代理被删除时, 在观点描述语抽取的性能达到 74.09% 的情况下, 抽取代指提及的性能只有 71.1%, 相较于删除其他代理, OM 任务在抽取代指提及的性能有最大幅度的下降, 这说明模型在没有 OM 任务的指导下, 出现了观点描述语与代指提及匹配难度上升的情况. 当反思模块以及 EM 代理被删除时, 模型虽然在属性实体以及代指提及这两个单实体匹配中取得了不错的效果, 但是在 EM 对匹配中性能较差. 这说明当没有

EM 代理时, 模型对于指代关系匹配难度加大, 这造成了 EM 对匹配性能不佳. 值得注意的是, 当 EM 代理被删除时, 模型在 OM 对匹配中性能有着最大的下降, 这说明 EM 代理能够有效地指导模型理解对话中的指代关系. 当反思模块以及 OE 代理被删除时, 相较于 MACR 删除反思模块, 模型对属性实体的抽取中性能有最直观的下降, 这也说明了 OE 代理能够指导模型抽取出观点描述语对应的属性实体.

4.3 不同基座以及 MACR 方法的进一步有效性分析

为了验证 MACR 框架的有效性, 本节采用和微调 ChatGLM3 一样的实验设置来微调 Baichuan2-7B-Chat (https://huggingface.co/baichuan-inc/Baichuan2-7B-Chat), 实验的结果如表 3 所示. 从表中我们可以看到, Baichuan2 在微调后性能优于 ChatGLM3, 这是可能因为 Baichuan2-7B 因为参数量多于 ChatGLM3-6B, 所以 Baichuan2 有更好的中文理解能力. 当 Baichuan2 基座模型加入 MACR 框架后, 模型的性能在单实体匹配, 对匹配和四元组匹配上的性能平均提升了 2.80%、3.27% 和 3.13%. 这说明 MACR 框架能够有效地帮助模型更好地抽取和理解对话中的实体以及实体间的关系, 这进一步说明了我们方法的有效性.

表 3　不同基座模型下引入与不引入 MACR 框架的模型实验结果 (%)

模型	单实体匹配				对匹配			四元组匹配
	entity	mention	opinion	ploarity	OE	EM	OM	
ChatGLM3	68.17	70.74	68.89	82.82	58.35	57.95	59.24	50.46
Baichuan 2[42]	69.75	68.61	69.06	79.74	61.03	58.23	58.39	51.79
MACR (ChatGLM3)	**73.29**	**73.21**	70.58	82.83	62.39	62.12	**61.79**	54.31
MACR (Baichuan2)	72.78	71.78	**70.81**	**82.97**	**63.62**	**62.67**	61.16	**54.92**

4.4 结合案例的多代理一致性反思方法有效性分析

为了进一步验证我们模型的有效性, 我们选取了相关案例来验证反思的有效性. 对于图 3 中给定的文本, 当不进行反思时, 模型错误地认为观点描述语“太怂”指向的属性实体为“刘文卿”以及观点描述语“口碑一般”指向的属性实体为“他主演的一部电影”, 模型出现了较为严重的幻觉现象, 但是并没有一种方法能够监督模型的输出进而纠正模型. 当加入一致性增强的反思模块后, 模块中的评估器能够发现这种错误, 产生较低的奖励值, 从而触发阈值让模型进行反思. 反思后的结果如图 3 中 MACR 所示, w/o reflection 表示 MACR 模型去掉一致性增强的反思模块. 当加入反思后, 模型能够在 prompt 的指导下根据历史信息进行反思, 根据历史内容重新抽取四元组, 从而产生正确的结果.

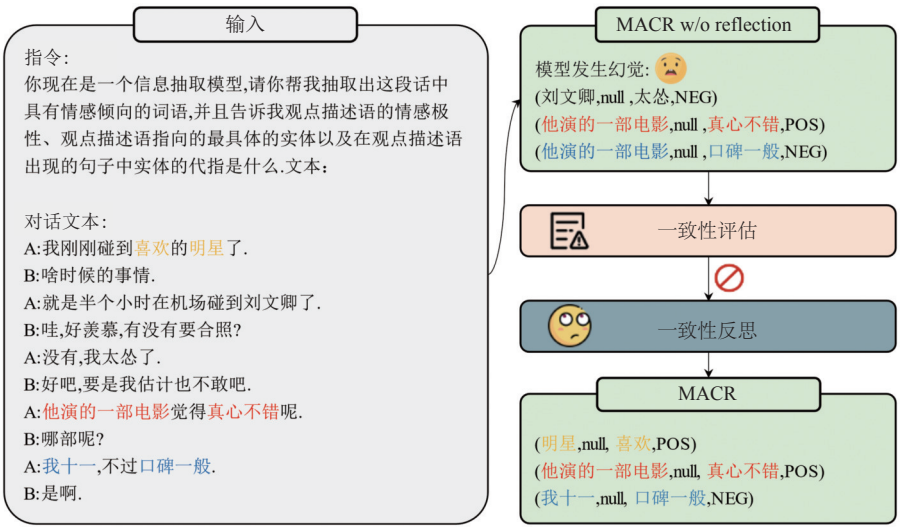

图 3　案例分析图

4.5 多代理一致性反思方法关键超参数的有效性分析

● 不同阈值下反思对模型的影响. 本文对不同阈值下让模型进行反思的性能进行了评估. 如图 4(a) 所示, 当阈值较小时, 只有发生非常严重的不一致, 模型才会反思, 这时模型相较于不加反思模块提升较少, 性能接近于不加反思模块, 这与我们的直觉相符. 当阈值较大时, 因为特别小的错误也有可能引起模型反思, 并且评估器的预测结果也存在误差, 当 α 大于 0.5 时模型性能快速下降, 甚至低于不加入反思模块, 这说明过度的反思对模型是有害的. 根据图 4(a), 当 α 的值为 0.5 时模型性能最佳, 这样既能让模型进行反思, 也能容忍较小的不一致性, 从而让模型生成正确的结果.

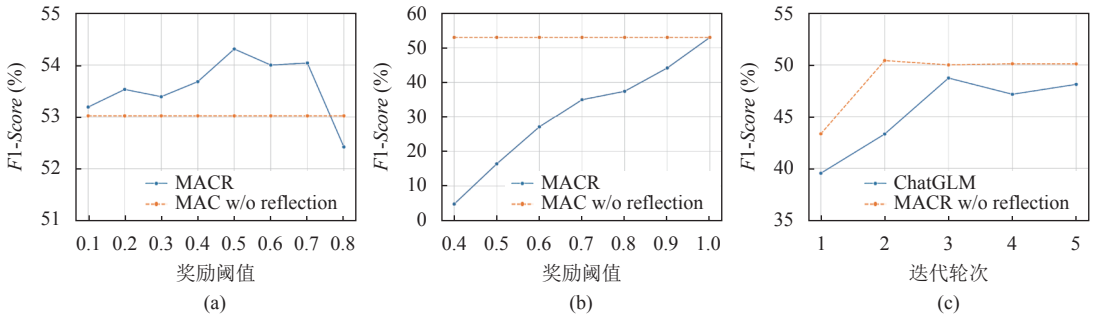

图 4 不同超参数对模型性能的影响

● 基于一致性进行反思的合理性. 本文根据模型预测结果的一致性分别计算了这些结果的 $F1$ 分数. 如图 4(b) 所示, 模型预测结果的性能随着模型生成过程一致性的提高而提高. 特别地, 当模型预测结果的一致性小于 0.4 时, 预测结果的 $F1$ 分数为 4.76%, 这远远低于模型的性能. 图 4(c) 的折线图也表明了一致性对于模型的意义, 这鼓励我们通过反思的方式让模型生成一致的结果.

● 子任务代理对模型在情感理解任务上的影响. 本文记录了每个迭代次数中模型在验证集中的 $F1$ 分数, 具体的分数如图 4(c) 所示. 根据图 4(c) 可知, 当训练中加入子任务时, 模型能够更快地拟合, 并且模型有更优的性能. 这也说明了子任务能够从关系匹配的角度帮助模型处理大语言模型对话属性情感理解任务, 从而提升模型在大语言模型对话属性情感理解任务上的性能. 这鼓励我们在训练模型阶段采用多代理的范式.

5 总 结

近些年来, 大语言模型的提出开辟了 NLP 领域新的范式, 刷新了许多任务的新性能, 这鼓励我们采用大语言模型处理属性级情感理解任务, 因此本文提出大语言模型对话属性情感理解任务并标注了一个高质量的对话属性情感理解数据集, 该任务在传统的三元组抽取中额外抽取对话中的指代关系, 旨在帮助模型更好地理解对话中的情感. 在属性情感理解任务的基础上, 本文针对该任务的两个挑战提出了多代理一致性反思方法. 该方法通过多代理机制来捕捉对话场景中两个重要的映射关系, 即属性指代映射关系和属性情感映射关系. 除此之外, 为了缓解大模型的幻觉问题, 本文提出一致性增强的反思模块. 该模块通过情感理解任务和 3 个子任务代理的一致性来得到奖励, 当奖励低于某一阈值时让模型进行反思, 通过反思让模型生成正确的结果. 我们在本文标注的数据集上评估了 MACR 的有效性, 实验结果与分析表明: MACR 的性能显著超过目前主流的基准方法. 在未来的工作中, 我们将会进一步研究模型该如何有效地理解上下文中的细粒度信息, 并将我们的方法迁移到多模态场景, 比如多模态属性级情感理解, 这在大模型时代对模型理解人类语言有至关重要的作用. 除此之外, 我们将探索如何通过反思机制更好地帮助模型缓解幻觉问题.

References:

[1] Chambers N, Bowen V, Genco E, Tian XS, Young E, Harihara G, Yang E. Identifying political sentiment between nation states with social media. In: Proc. of the 2015 Conf. on Empirical Methods in Natural Language Processing. Lisbon: ACL, 2015. 65–75. [doi: 10.

18653/v1/D15-1007]

[2] Peng HY, Xu L, Bing LD, Huang F, Lu W, Si L. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In: Proc. of the 34th AAAI Conf. on Artificial Intelligence. New York: AAAI, 2020. 8600–8607. [doi: 10.1609/aaai.v34i05.6383]

[3] Wan H, Yang YF, Du JF, Liu Y, Qi KX, Pan JZ. Target-aspect-sentiment joint detection for aspect-based sentiment analysis. In: Proc. of the 34th AAAI Conf. on Artificial Intelligence. New York: AAAI, 2020. 9122–9129. [doi: 10.1609/aaai.v34i05.6447]

[4] Pontiki M, Galanis D, Papageorgiou H, Manandhar S, Androutsopoulos I. SemEval-2015 task 12: Aspect based sentiment analysis. In: Proc. of the 9th Int'l Workshop on Semantic Evaluation (SemEval 2015). Denver: ACL, 2015. 486–495. [doi: 10.18653/v1/S15-2082]

[5] Song LF, Xin CL, Lai SP, Wang AT, Su JS, Xu K. CASA: Conversational aspect sentiment analysis for dialogue understanding. Journal of Artificial Intelligence Research, 2022, 73: 511–533. [doi: 10.1613/jair.1.12802]

[6] Li BB, Fei H, Li F, Wu YH, Zhang JS, Wu SQ, Li JY, Liu YJ, Liao LZ, Chua TS, Ji DH. DiaASQ: A benchmark of conversational aspect-based sentiment quadruple analysis. In: Findings of the Association for Computational Linguistics: ACL 2023. Toronto: ACL, 2023. 13449–13467. [doi: 10.18653/v1/2023.findings-acl.849]

[7] Shen CL, Sun CL, Wang JJ, Kang YY, Li SS, Liu XZ, Si L, Zhang M, Zhou GD. Sentiment classification towards question-answering with hierarchical matching network. In: Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing. Brussels: ACL, 2018. 3654–3663. [doi: 10.18653/v1/D18-1401]

[8] Ji ZW, Lee N, Feiseke R, Yu TZ, Su D, Xu Y, Ishii E, Bang YJ, Madotto A, Fung P. Survey of hallucination in natural language generation. ACM Computing Surveys, 2023, 55(12): 248. [doi: 10.1145/3571730]

[9] He ZW, Liang T, Jiao WX, Zhang ZS, Yang YJ, Wang R, Tu ZP, Shi SM, Wang X. Exploring human-like translation strategy with large language models. Trans. of the Association for Computational Linguistics, 2024, 12: 229–246. [doi: 10.1162/tacl_a_00642]

[10] Zhao H, Huang LT, Zhang R, Lu Q, Xue H. SpanMlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In: Proc. of the 58th Annual Meeting of the Association for Computational Linguistics. ACL, 2020. 3239–3248. [doi: 10.18653/v1/2020.acl-main.296]

[11] Zhang WX, Deng Y, Li X, Yuan YF, Bing LD, Lam W. Aspect sentiment quad prediction as paraphrase generation. In: Proc. of the 2021 Conf. on Empirical Methods in Natural Language Processing. Punta Cana: ACL, 2021. 9209–9219. [doi: 10.18653/v1/2021.emnlp-main.726]

[12] Gou ZB, Guo QY, Yang YJ. MvP: Multi-view prompting improves aspect sentiment tuple prediction. In: Proc. of the 61st Annual Meeting of the Association for Computational Linguistics, Vol. 1 (Long Papers). Toronto: ACL, 2023. 4380–4397. [doi: 10.18653/v1/2023.acl-long.240]

[13] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou YQ, Li W, Liu PJ. Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research, 2020, 21(1): 5485–5551.

[14] Bao XY, Wang ZQ, Jiang XT, Xiao R, Li SS. Aspect-based sentiment analysis with opinion tree generation. In: Proc. of the 31st Int'l Joint Conf. on Artificial Intelligence. 2022. 4044–4050. [doi: 10.24963/ijcai.2022/561]

[15] Bao XT, Jiang XT, Wang ZQ, Zhang Y, Zhou GD. Opinion tree parsing for aspect-based sentiment analysis. In: Findings of the Association for Computational Linguistics: ACL 2023. Toronto: ACL, 2023. 7971–7984. [doi: 10.18653/v1/2023.findings-acl.505]

[16] Xu L, Li H, Lu W, Bing LD. Position-aware tagging for aspect sentiment triplet extraction. In: Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP). ACL, 2020. 2339–2349. [doi: 10.18653/v1/2020.emnlp-main.183]

[17] Wang JJ, Sun CL, Li SS, Liu XZ, Si L, Zhang M, Zhou GD. Aspect sentiment classification towards question-answering with reinforced bidirectional attention network. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: ACL, 2019. 3548–3557. [doi: 10.18653/v1/P19-1345]

[18] Wang F, Li YC, Zhang WJ, Zhong SH. A more fine-grained aspect-sentiment-opinion triplet extraction task. arXiv:2103.15255, 2021.

[19] Chen H, Zhai ZP, Feng FX, Li RF, Wang XJ. Enhanced multi-channel graph convolutional network for aspect sentiment triplet extraction. In: Proc. of the 60th Annual Meeting of the Association for Computational Linguistics, Vol. 1 (Long Papers). Dublin: ACL, 2022. 2974–2985. [doi: 10.18653/v1/2022.acl-long.21]

[20] Liang S, Wei W, Mao XL, Fu YY, Fang R, Chen DY. STAGE: Span tagging and greedy inference scheme for aspect sentiment triplet extraction. In: Proc. of the 37th AAAI Conf. on Artificial Intelligence. Washington: AAAI, 2023. 13174–13182. [doi: 10.1609/aaai.v37i11.26547]

[21] Mao Y, Shen Y, Yang JC, Zhu XY, Cai LJ. Seq2Path: Generating sentiment tuples as paths of a tree. In: Findings of the Association for Computational Linguistics: ACL 2022. Dublin: ACL, 2022. 2215–2225. [doi: 10.18653/v1/2022.findings-acl.174]

[22] Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, Lacroix T, Rozière B, Goyal N, Hambro E, Azhar F, Rodriguez A, Joulin A, Grave E, Lample G. LLaMA: Open and efficient foundation language models. arXiv:2302.13971, 2023.

[23] Du ZX, Qian YJ, Liu X, Ding M, Qiu JZ, Yang ZL, Tang J. GLM: General language model pretraining with autoregressive blank infilling. In: Proc. of the 60th Annual Meeting of the Association for Computational Linguistics, Vol. 1 (Long Papers). Dublin: ACL, 2022. 320–335. [doi: 10.18653/v1/2022.acl-long.26]

[24] Zhang JJ, Hou YP, Xie RB, Sun WQ, Mcauley J, Zhao WX, Lin LY, Wen JR. AgentCF: Collaborative learning with autonomous language agents for recommender systems. arXiv:2310.09233, 2023.
[25] Xu ZR, Shi SB, Hu BT, Yu JD, Li DF, Zhang M, Wu YX. Towards reasoning in large language models via multi-agent peer review collaboration. arXiv:2311.08152, 2023.
[26] Liu ZY, Lai ZQ, Gao ZW, Cui EF, Li ZH, Zhu XZ, Lu LW, Chen QF, Qiao Y, Dai JF, Wang WH. ControlLLM: Augment language models with tools by searching on graphs. arXiv:2310.17796, 2023.
[27] Liu JC, Shen DH, Zhang YZ, Dolan B, Carin L, Chen WZ. What makes good in-context examples for GPT-3? In: Proc. of the Deep Learning Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures. Dublin: ACL, 2022. 100–114. [doi: 10.18653/v1/2022.deelio-1.10]
[28] Yao SY, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, Cao Y. ReAct: Synergizing reasoning and acting in language models. arXiv:2210.03629, 2023.
[29] Wei J, Wang XZ, Schuurmans D, Bosma M, Ichter B, Xia F, Chi EH, Le QV, Zhou D. Chain-of-thought prompting elicits reasoning in large language models. In: Proc. of the 36th Int'l Conf. on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2022. 24824–24837.
[30] Yao SY, Yu D, Zhao J, Shafran I, Griffiths TL, Cao Y, Narasimhan K. Tree of thoughts: Deliberate problem solving with large language models. In: Proc. of the 37th Int'l Conf. on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2023. 11809–11822.
[31] Shinn N, Labash B, Gopinath A. Reflexion: An autonomous agent with dynamic memory and self-reflection. arXiv:2303.11366, 2023.
[32] Shinn N, Cassano F, Gopinath A, Narasimhan K, Yao SY. Reflexion: Language agents with verbal reinforcement learning. In: Proc. of the 37th Int'l Conf. on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2023. 8634–8652.
[33] Huang X, Lian JX, Lei YX, Yao J, Lian DF, Xie X. Recommender AI agent: Integrating large language models for interactive recommendations. arXiv:2308.16505, 2024.
[34] Zhang WX, Deng Y, Liu B, Pan SJ, Bing LD. Sentiment analysis in the era of large language models: A reality check. arXiv:2305.15005, 2023.
[35] Wei J, Bosma M, Zhao VY, Guu K, Yu AW, Lester B, Du N, Dai AM, Le QV. Finetuned language models are zero-shot learners. arXiv:2109.01652, 2022. [DOI: 10.48550/arXiv.2109.01652]
[36] Cai HJ, Xia R, Yu JF. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In: Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th Int'l Joint Conf. on Natural Language Processing, Vol. 1 (Long Papers). ACL, 2021. 340–350. [doi: 10.18653/v1/2021.acl-long.29]
[37] Hu EJ, Shen YL, Wallis P, Allen-Zhu A, Li YZ, Wang SA, Wang L, Chen WZ. LoRA: Low-rank adaptation of large language models. arXiv:2106.09685, 2021.
[38] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv:1412.6980, 2017.
[39] Yang YM, Liu X. A re-examination of text categorization methods. In: Proc. of the 22nd Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Berkeley: ACM, 1999. 42–49.
[40] Zhang WX, Li X, Deng Y, Bing LD, Lam W. Towards generative aspect-based sentiment analysis. In: Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th Int'l Joint Conf. on Natural Language Processing, Vol. 2 (Short Papers). ACL, 2021. 504–510. [doi: 10.18653/v1/2021.acl-short.64]
[41] Wei X, Cui XY, Cheng N, Wang XB, Zhang X, Huang S, Xie PJ, Xu J, Chen YF, Zhang MS, Jiang Y, Han WJ. ChatIE: Zero-shot information extraction via chatting with ChatGPT. arXiv:2302.10205, 2024.
[42] Yang AY, Xiao B, Wang BN, *et al*. Baichuan 2: Open large-scale language models. arXiv:2309.10305, 2023.

刘一丁(1999－), 男, 硕士生, 主要研究领域为自然语言处理.

罗佳敏(1997－), 女, 博士生, CCF 学生会员, 主要研究领域为自然语言处理.

王晶晶(1990－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为自然语言处理.

周国栋(1965－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为自然语言处理.