



中国科学院大学  
University of Chinese Academy of Sciences

# 硕士学位论文

基于分类和一致性验证的 AIGC 文本检测方法研究

作者姓名：杨竞

指导教师：王石 正高级工程师

中国科学院计算技术研究所

学位类别：电子信息硕士硕士

学科专业：人工智能

培养单位：中国科学院计算技术研究所

2025 年 6 月



**Research on AIGC Text Detection Methods Based on**  
**Classification and Consistency Verification**

**A thesis submitted to**  
**University of Chinese Academy of Sciences**  
**in partial fulfillment of the requirement**  
**for the degree of**  
**Master of Electronic and Information Engineering**  
**in Artificial Intelligence**

**By**

**Yang Jing**

**Supervisor: Professor Wang Shi**

**Institute of Computing Technology, Chinese Academy of Sciences**

**June, 2025**



## **中国科学院大学 学位论文原创性声明**

本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究工作所取得的成果。承诺除文中已经注明引用的内容外，本论文不包含任何其他个人或集体享有著作权的研究成果，未在以往任何学位申请中全部或部分提交。对本论文所涉及的研究工作做出贡献的其他个人或集体，均已在文中以明确方式标明或致谢。本人完全意识到本声明的法律结果由本人承担。

作者签名：

日 期：

## **中国科学院大学 学位论文授权使用声明**

本人完全了解并同意遵守中国科学院大学有关收集、保存和使用学位论文的规定，即中国科学院大学有权按照学术研究公开原则和保护知识产权的原则，保留并向国家指定或中国科学院指定机构送交学位论文的电子版和印刷版文件，且电子版与印刷版内容应完全相同，允许该论文被检索、查阅和借阅，公布本学位论文的全部或部分内容，可以采用扫描、影印、缩印等复制手段以及其他法律许可的方式保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：

日 期：

导师签名：

日 期：



## 摘要

随着大模型（Large Language Models, LLMs）的迅猛发展，人工智能内容生成（Artificial Intelligence-Generated Content, AIGC）技术在广告、新闻、故事创作及代码生成等多个领域得到了广泛应用，其生成的文本质量与人类写作水平相当，甚至在某些情况下难以区分。然而，AIGC 技术的广泛应用也带来了诸多挑战，如虚假信息的制造与传播、学术不端行为的增加以及法律和伦理问题的复杂化。因此，开发高效的 AIGC 文本检测技术显得尤为重要。

本文针对 AIGC 文本检测中的挑战，分别从形式语言和功能性语言两个角度，提出了三项研究内容：

**基于结构信息熵的 AIGC 文本检测方法** 提出了一种基于结构信息熵的分类方法，该方法通过构建一个文本多关系图，将文本的特征表示转换为图中的节点，然后利用结构信息理论，通过最小化结构熵的方法，可以将文本节点划分成不同的簇。最小化结构熵的过程有助于减少噪声和异常值对聚类结果的影响，从而得到更稳定的簇。最后，通过聚类结果的质量和簇内文本之间的相似度，进行簇内标签推断得到最终的分类结果。在创作和问答数据集上，对 6 个大模型的平均检测性能相较于其他方法最高提升了 5%，证明了算法对非形式语言文本检测的有效性。

**基于数学运算一致性验证的 AIGC 文本检测方法** 提出了一种基于数学运算一致性验证的 AIGC 文本检测方法，该方法通过将文本转换为数学表达式，并利用数学运算规则验证其逻辑合理性，有效识别文本中的数学推理错误和逻辑矛盾。本方法不仅限于简单等式，还能处理复杂的数学运算，如嵌套的加减乘除和数值大小关系。为提升模型在数学运算文本上的泛化能力，引入对比学习技术，通过构建正负样本对，增强模型对细微差异的区分能力。在公开数据集和新建数据集上相比于基线方法提升了 9%，证明了算法在检测功能性语言文本——数学运算时的有效性。

**融合逻辑推理一致性验证的 AIGC 文本检测方法** 提出了一种基于主题一致性与逻辑推理的 AIGC 文本检测方法。该方法构建多维度检测框架，将文本的主题一致性与逻辑合理性相结合，以弥补现有方法的局限性。方法包括两阶段的逻辑检测：首先，设计自然语言到一阶谓词逻辑（FOL）转换模块，将文本形式化为可验证的逻辑表达式；其次，利用 FOL 求解器验证推理正确性，识别逻辑错误。此外，提出针对转换过程的修正策略，以减少错误传播影响，提升文本质量。实验结果表明，该方法在 AIGC 文本检测任务中表现出较高的准确性和稳定性，尤其在涉及主题一致性分析和逻辑推理验证的场景下，显著优于现有检测方

法。

基于结构信息熵的方法在形式语言特征检测方面表现出色，显著提高了检测的准确性和鲁棒性。而基于数学运算和逻辑推理的反事实检测方法有效弥补了功能性语言检测的不足，能够准确识别出 AIGC 文本中的逻辑错误和数学漏洞。本研究为 AIGC 文本检测提供了新的技术路径，通过形式语言和功能性语言的检测方法，实现了对 AIGC 生成文本的全面和准确识别。

**关键词：**人工智能生成内容, 大模型, AIGC 文本检测, 结构信息熵

## **Abstract**

With the rapid advancement of large language models (LLMs), Artificial Intelligence-Generated Content (AIGC) technology has been widely applied in various fields, such as advertising, news reporting, story creation, and code generation. The quality of the generated text is comparable to human writing and, in some cases, indistinguishable. However, the widespread adoption of AIGC technology has also introduced numerous challenges, including the proliferation of misinformation, increased instances of academic misconduct, and more complex legal and ethical issues. Therefore, developing effective AIGC text detection techniques has become increasingly important.

**Key Words:** Artificial Intelligence-Generated Content, AIGC Text Detection, Large Language Models, Structural Entropy



## 目 录

|                                  |    |
|----------------------------------|----|
| 第 1 章 绪论 .....                   | 1  |
| 1.1 研究背景 .....                   | 1  |
| 1.2 研究意义 .....                   | 2  |
| 1.3 主要工作 .....                   | 3  |
| 1.4 论文组织架构 .....                 | 4  |
| 第 2 章 相关工作与研究现状 .....            | 7  |
| 2.1 基于水印的 AIGC 文本检测方法 .....      | 7  |
| 2.1.1 数据驱动水印 .....               | 7  |
| 2.1.2 模型驱动水印 .....               | 7  |
| 2.1.3 后处理水印 .....                | 9  |
| 2.2 基于统计分析的 AIGC 文本检测方法 .....    | 11 |
| 2.2.1 白盒统计 .....                 | 11 |
| 2.2.2 黑盒统计 .....                 | 13 |
| 2.3 基于神经网络的 AIGC 文本检测方法 .....    | 14 |
| 2.3.1 基于特征的 AIGC 文本检测方法 .....    | 14 |
| 2.3.2 基于预训练模型的 AIGC 文本检测方法 ..... | 15 |
| 2.3.3 基于大模型的 AIGC 文本检测方法 .....   | 18 |
| 2.4 AIGC 文本检测研究趋势与挑战 .....       | 19 |
| 第 3 章 基于结构信息熵的 AIGC 文本检测方法 ..... | 21 |
| 3.1 引言 .....                     | 21 |
| 3.2 相关工作 .....                   | 22 |
| 3.3 方法介绍 .....                   | 24 |
| 3.3.1 文本图构建 .....                | 25 |
| 3.3.2 基于结构熵最小化的图聚类 .....         | 26 |
| 3.3.3 聚类内标签推断 .....              | 27 |
| 3.3.4 时间复杂度分析 .....              | 29 |
| 3.4 实验结果与分析 .....                | 29 |
| 3.4.1 数据集 .....                  | 29 |
| 3.4.2 相关研究工作 .....               | 30 |
| 3.4.3 实验结果 .....                 | 32 |
| 3.4.4 分析实验 .....                 | 36 |
| 3.5 本章小结 .....                   | 38 |

|                                      |    |
|--------------------------------------|----|
| 第 4 章 基于数学运算一致性验证的 AIGC 文本检测方法 ····· | 39 |
| 4.1 引言 ·····                         | 39 |
| 4.2 相关工作 ·····                       | 40 |
| 4.3 方法介绍 ·····                       | 42 |
| 4.3.1 问题定义 ·····                     | 42 |
| 4.3.2 NL2MATH 模块设计 ·····             | 43 |
| 4.3.3 训练过程 ·····                     | 45 |
| 4.3.4 数学运算一致性验证 ·····                | 46 |
| 4.4 实验结果与分析 ·····                    | 47 |
| 4.4.1 数据集 ·····                      | 47 |
| 4.4.2 实验结果 ·····                     | 49 |
| 4.5 本章小结 ·····                       | 51 |
| 第 5 章 融合逻辑推理一致性验证的 AIGC 文本检测方法 ····· | 53 |
| 5.1 引言 ·····                         | 53 |
| 5.2 相关工作 ·····                       | 53 |
| 5.2.1 自然语言-一阶谓词逻辑翻译 ·····            | 53 |
| 5.2.2 FOL 求解器 ·····                  | 54 |
| 5.3 方法介绍 ·····                       | 54 |
| 5.3.1 问题定义 ·····                     | 54 |
| 5.3.2 主题相关性检测模块 ·····                | 55 |
| 5.3.3 逻辑推理验证模块 ·····                 | 56 |
| 5.4 实验结果与分析 ·····                    | 57 |
| 5.4.1 数据集 ·····                      | 57 |
| 5.4.2 实验结果 ·····                     | 58 |
| 5.5 本章小结 ·····                       | 61 |
| 第 6 章 总结与展望 ·····                    | 63 |
| 6.1 本文工作总结 ·····                     | 63 |
| 6.2 未来工作展望 ·····                     | 64 |
| 参考文献 ·····                           | 67 |
| 致谢 ·····                             | 77 |
| 作者简历及攻读学位期间发表的学术论文与其他相关学术成果 ·        | 79 |

## 图目录

|       |                                         |    |
|-------|-----------------------------------------|----|
| 图 1-1 | AIGC 文本检测任务 .....                       | 2  |
| 图 2-1 | 基于字符采样的 AIGC 检测方法 <sup>[1]</sup> .....  | 9  |
| 图 2-2 | 基于 AWT 的 AIGC 文本检测 <sup>[2]</sup> ..... | 11 |
| 图 2-3 | 基于扰动的 AIGC 文本检测 <sup>[3,4]</sup> .....  | 13 |
| 图 2-4 | 基于对比学习的 AIGC 文本检测 <sup>[5]</sup> .....  | 16 |
| 图 2-5 | 基于对抗学习的 AIGC 文本检测 <sup>[6]</sup> .....  | 17 |
| 图 2-6 | 基于特征增强的 AIGC 文本检测 <sup>[7]</sup> .....  | 18 |
| 图 2-7 | 基于大模型的 AIGC 文本检测 <sup>[8]</sup> .....   | 19 |
| 图 3-1 | 基于结构信息熵的 AIGC 文本检测方法示意图 .....           | 25 |
| 图 3-2 | 不同数据集下 NMI 和 ARI 指标 .....               | 35 |
| 图 3-3 | SILTD 方法在检测不同模型时的 AUROC 指标 .....        | 36 |
| 图 3-4 | 超参数分析 .....                             | 37 |
| 图 3-5 | 在论文数据集中的检测示例 .....                      | 38 |
| 图 4-1 | 大模型在简单数学问题上的出错率 <sup>[9]</sup> .....    | 40 |
| 图 4-2 | 基于数学运算一致性验证的 AIGC 文本检测方法示意图 .....       | 42 |
| 图 4-3 | 构建数据集的提示模板 .....                        | 47 |
| 图 5-1 | 融合逻辑推理一致性验证的 AIGC 文本检测方法示意图 .....       | 55 |
| 图 5-2 | 超参数实验 .....                             | 61 |

## 表目录

|       |                                  |    |
|-------|----------------------------------|----|
| 表 3-1 | 常用数据集数据统计 .....                  | 30 |
| 表 3-2 | 不同检测方法在创作数据集上的 F1 分数 .....       | 33 |
| 表 3-3 | 不同检测方法在问答数据集上的 F1 分数 .....       | 34 |
| 表 3-4 | 使用 LL 和 LR 特征的消融实验结果 .....       | 35 |
| 表 3-5 | 使用 NPR 和 DG 特征的消融实验结果 .....      | 36 |
| 表 4-1 | 数据集数据统计 .....                    | 48 |
| 表 4-2 | 大模型生成文本检测指标 .....                | 49 |
| 表 4-3 | 自然语言文本转数学表达式模块的准确率 .....         | 50 |
| 表 5-1 | 融合逻辑推理一致性验证的 AIGC 文本检测实验结果 ..... | 59 |
| 表 5-2 | 消融实验结果 .....                     | 60 |



## 第1章 绪论

### 1.1 研究背景

随着大语言模型（LLM）的快速发展，人工智能内容生成（AIGC）的能力已经达到了与人类写作相媲美的水平<sup>[10-12]</sup>。大模型已经渗透到日常生活的各个方面，并在许多专业工作流程中发挥着至关重要的作用，促进了广告标语创作、新闻撰写、故事生成和代码生成等任务。2023年 Hanley 等人<sup>[13]</sup>发表的一项研究表明，主流网站上由人工智能（AI）生成的新闻文章的相对数量增加了 55.4%，而在已知散布虚假信息的网站上，这一数字在 16 个月内增加了 457%。此外，它们的影响促进了包括教育、法律、生物学和医学在内的众多领域和学科的进步。大模型强大的生成能力使得个人很难区分人工智能生成的文本和人类编写的文本，进而引发了公众的广泛关注和担忧。

首先，AIGC 技术容易制造虚假信息，依赖于过时的信息，并且对提示信息极为敏感。这些缺陷可能会加剧错误知识的传播，削弱技术专长，并助长抄袭行为。其次，存在恶意利用大模型进行虚假信息传播<sup>[14]</sup>、网络诈骗<sup>[15]</sup>、垃圾邮件生成<sup>[16]</sup>和学术不端行为的风险，尤其是学生使用 AIGC 进行论文写作<sup>[17]</sup>。与此同时，Li 等人<sup>[18]</sup>提出大模型可以自我评估甚至将自己的性能作为基准。由于其出色的文本生成性能，大模型也用于通过预设指令构建许多训练数据集<sup>[19]</sup>。然而，即大模型在训练时严重依赖网络来源的数据，而这些数据中有相当一部分来自大模型的输出，这可能会阻碍它们的长期进步，这种现象被称为“大模型自噬障碍”（MAD）<sup>[20]</sup>。AIGC 的法律地位尚不明确，尤其是其知识产权保护问题。AIGC 生成的内容是否应享有版权，以及如何界定人类在创作过程中的贡献，这仍然是一个有争议的话题<sup>[21]</sup>。

然而，对于 AIGC 文本检测任务，之前的一些检测技术<sup>[22]</sup>是不可靠的。它们主要偏向于将输出分类为人类文本，而不是检测由 AI 生成的文本<sup>[23,24]</sup>。依赖人类的检测方法也是不可靠的，准确度非常低，甚至只比随机分类略好<sup>[25-28]</sup>。此外，在各种环境设置中，人类识别人工智能生成文本的能力通常低于检测器或检测算法<sup>[28,29]</sup>。因此，迫切需要强大的检测器来有效识别 AIGC 文本，减轻大模型滥用风险和促进大模型时代的负责任 AI 治理<sup>[30-32]</sup>。在 ChatGPT 出现之前，AIGC 文本的检测研究已经受到很多关注，特别是在早期识别深度伪造文本<sup>[33]</sup>、机器生成文本检测<sup>[34]</sup>和作者归属<sup>[35]</sup>等领域。ChatGPT 的出现加速了 AIGC 技术的应用普及，成为了 AIGC 领域的重要代表。它不仅能够生成流畅自然的文本，还能根据上下文进行合理推理和回答。这标志着人工智能生成文本的质量和复杂度达到了一个新的高度，也促使研究者对如何检测 AIGC 生成文本展开了更

加深入的探索。在这一过程中，检测 AIGC 生成文本的任务逐渐从传统的基于规则的方法转向了更加智能化的技术，尤其是在深度学习和自然语言处理（NLP）领域的应用。当前，AIGC 文本检测通常被视为一种分类任务（如图1-1）。研究者们尝试通过训练分类器，基于各种特征（如词向量、上下文关联等）来区分由大模型生成的文本与由人类编写的文本。这一过程不仅依赖于文本本身的表面特征，还需要深入挖掘文本的潜在规律，例如生成模型可能存在的“偏差”或“幻觉”，以及与人工文本在风格、逻辑、连贯性等方面的差异。面对 AIGC 文本带来的挑战，研究者们也在不断创新解决方案。一方面，他们使用更加先进的机器学习和深度学习方法，训练出更强大的分类器；另一方面，他们探索新的特征提取方式，提升模型对复杂文本的感知和辨识能力。此外，为了应对不断更新的生成模型，检测方法也需要保持灵活性和适应性，从而有效应对未来可能出现的更先进的 AIGC 技术。

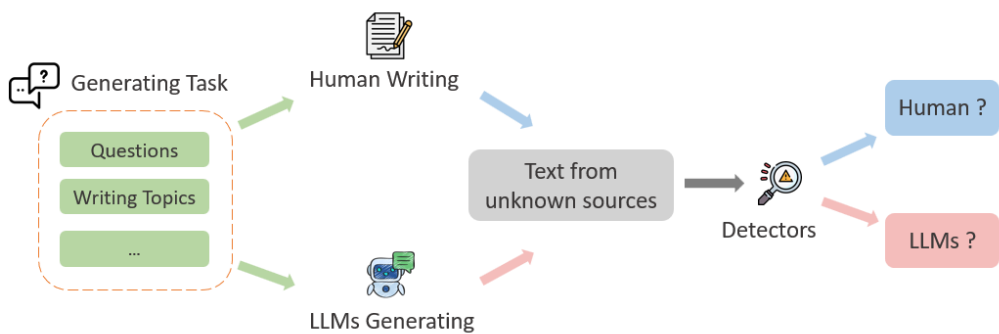


图 1-1 AIGC 文本检测任务  
Figure 1-1 AIGC text detection task

随着大模型技术的不断进步，对 AIGC 文本进行检测的需求日益迫切。这不仅是一项技术挑战，更涉及法律、伦理以及社会影响等多重维度。因此，研究和开发精准有效的检测工具变得至关重要，以确保人工智能技术的负责任使用。

## 1.2 研究意义

AIGC 文本不仅在语法、词汇和句法层面表现出流畅性，还在逻辑推理、数学运算等功能性语言方面展示出挑战。当前的检测方法在形式语言文本上取得了较好的效果，但在涉及功能性语言的检测中，尤其是在数学和逻辑推理领域，依然存在局限。通过研究形式语言和功能性语言的差异，探索形式语言文本检测以及功能性语言文本检测方法，包括数学运算以及逻辑推理文本。在此过程中，如何设计算法来有效应对复杂的自然语言文本，尤其是能够识别大模型与人类之间不同的语言风格，逻辑错误、数学推导过程中的错误等，将是推动 AIGC 检测技术进步的关键。研究这一课题还可以推动相关理论的发展，比如文本聚类算

法、结构信息熵以及无监督学习方法等，为未来的无监督检测方法提供理论基础。

随着 AIGC 技术在各个领域的广泛应用，如何在实践中准确识别和筛选 AIGC 生成的内容，成为确保内容真实性和可信度的重要手段。从学术研究到实际应用，AIGC 文本检测技术将为政策监管、学术诚信、内容安全等提供坚实的技术支持，帮助各行业有效辨别人工智能生成的文本，降低虚假信息和误导性内容的风险。

此外，大模型推理和理解能力不断增强，但仍然面临着复杂任务和高阶认知能力的挑战。AIGC 文本检测不仅仅局限于识别生成的内容，更深入地探讨了模型如何处理逻辑推理、数学运算等复杂问题，这为人工智能系统的推理能力提供了进一步的提升方向。

### 1.3 主要工作

本文围绕 AIGC 文本检测中的关键问题展开研究，提出了三种创新方法，分别针对现有技术在处理复杂文本、数学运算文本和逻辑推理文本时的局限性，具体内容如下所述。

基于结构信息熵的 AIGC 文本检测方法：大多 AIGC 文本检测方法依赖于神经网络分类模型。然而，这类方法在处理复杂文本时存在显著局限性，尤其在面对分布不均匀或特征边界模糊的文本时，神经网络分类方法往往需要大量标注数据进行训练，且对未见数据的泛化能力有限。此外，神经网络分类方法更倾向于捕捉文本的局部特征模式，缺乏有效的跨文本全局关系和相似性的建模能力，难以处理文本间语言连贯性以及用词风格的潜在关联性。

尽管监督检测方法通过在标注数据集上微调分类器取得了良好的检测效果，但它们面临两大关键挑战：（1）依赖昂贵且高质量的标注数据，而该数据难以跟上 LLMs 的快速发展；（2）由于对特定生成模式的过拟合，这些方法在不同 LLMs 之间的泛化能力有限。无监督方法避免了上述问题，但通常仅基于独立的文本特征（如对数似然或 token 分布）进行分析，未能捕捉文本之间的更深层次统计关系。

针对上述问题，本研究点提出基于结构熵的 AIGC 文本检测方法，通过构建文本多关系图，综合考虑文本之间语言连贯性的相似性和用词风格的相似性进行聚类分析建模。该方法首先将文本表示为图中的节点，并通过边编码多维特征相似性。这种结构能够更丰富地表征文本在生成统计空间中的相互关系。接着，将多关系图转换为编码树，并应用二维结构熵最小化算法，以挖掘层次化的文本聚类结构。提出一种结合 MultiRank 平稳概率分布和编码树熵的标签推理方法，该方法利用文本之间的潜在关联性和聚类质量来判断聚类簇是否由 LLM 生成。

基于数学运算一致性验证的 AIGC 文本检测方法：AIGC 生成文本的真实性和逻辑性问题比较普遍，特别是在涉及数学运算和逻辑推理的文本中，现有检测方法往往难以精准识别其中的数学推理错误或逻辑矛盾。传统 AIGC 文本检测方法主要依赖统计特征匹配或语言模型的概率分布，在普通自然语言文本检测中表现良好，但面对具有数学运算特征的文本时，常常难以有效判断其逻辑合理性。例如，在涉及数学运算的文本中，即便语言表达符合自然语言模式，仍可能存在数学运算错误或逻辑悖论，给现有检测方法带来挑战。

为了解决这一问题，本研究提出了一种基于数学运算一致性验证的 AIGC 文本检测方法，该方法与现有的针对形式语言文本的 AIGC 文本检测方法不同，该方法通过构建一种端到端的数学公式解析模型，将文本转换为数学表达式，并利用数学运算规则进行推理验证，从而有效识别文本中的逻辑错误。该方法不仅限于检测简单的等式关系，还能够处理更复杂的数学运算，涵盖嵌套的加减乘除以及数之间的大小关系。通过引入对比学习技术，构建语义相似但对应的数学表达式不同的对比样本，增强模型对数学表达式中细微差异的学习能力。创建了一个新的数学表达式数据集，其中引入了包含“ $>$ ”和“ $<$ ”不等关系的数学表达式。

融合逻辑推理一致性验证的 AIGC 文本检测方法：AIGC 生成的文本在为众多任务提供支持的同时，也暴露出了一系列质量问题，包括主题偏离、逻辑错误以及幻觉现象等。这些问题不仅影响了生成文本的可靠性，还可能对依赖这些文本的任务带来严重的负面影响。现有的 AIGC 文本检测方法大多集中于单一维度的评估，如基于语言模型的文本语言流畅度或通过扰动前后文本语言特征之间的变化进行检测。然而，这些方法往往忽视了文本深层次的一致性问题的，尤其是主题偏离和逻辑推理错误。

为了解决这一问题，本研究提出了一种融合逻辑推理一致性验证的 AIGC 文本检测方法，综合考虑了主题相关性和逻辑推一致性，能够深入分析文本的语义和逻辑结构，从而提供更精准的检测结果。首先，设计了一种端到端的自然语言到一阶谓词逻辑模块，将自然语言文本转换为一阶谓词逻辑表达式，表达复杂的逻辑关系和推理规则。其次，利用成熟的 FOL 求解器对转换后的逻辑表达式进行验证，检查是否存在逻辑错误。提出了一种错误判断策略，能够缓解自然语言转一阶谓词逻辑过程中潜在的错误传播问题，从而提升文本检测的精度。

#### 1.4 论文组织架构

本文共分为六个章节，本章为绪论，旨在介绍本文的研究背景及研究意义、主要工作和文章组织架构。后续章节内容如下：

第二章从基于水印的 AIGC 文本检测方法、基于统计分析的 AIGC 文本检测方法、基于神经网络的 AIGC 文本检测方法以及 AIGC 文本检测研究趋势与挑战

角度，介绍了本文的相关工作和国内外研究现状。第三章介绍了当前方法在检测形式语言文本上的不足和局限，提出了基于结构信息熵的 AIGC 文本检测方法，通过构建多关系文本图有效捕获文本之间的相似性，并利用结构熵最小化算法对多关系文本图进行高质量的聚类，最后提出一种标签推断的方法有效实现 AIGC 文本的检测。第四章介绍了当前方法在检测功能形语言文本上的不足，提出了基于数学运算一致性验证的 AIGC 文本检测方法，通过构建 NL2Math 模块将自然语言文本转换为数学表达式，并利用外部工具验证其准确性，从而辨别出是由大模型生成还是人写的。第四章介绍了逻辑推理过程的一致性检测的困难以及现有方法的不足，提出了融合逻辑推理一致性验证的 AIGC 文本检测方法，通过将主题和逻辑推理一致性相结合，从而能够深入分析文本的语义和逻辑结构，提供更精准的检测结果。



## 第 2 章 相关工作与研究现状

### 2.1 基于水印的 AIGC 文本检测方法

最初在计算机视觉领域，水印技术一直是检测人工智能生成图像的关键，作为视觉艺术知识产权的保护措施。随着大语言模型的出现和普及，水印技术的应用已经扩展到文本领域。这项技术不仅保护了重要模型免受未经授权的获取，例如通过序列蒸馏，还减轻了与复制和滥用 AIGC 文本相关的风险。

#### 2.1.1 数据驱动水印

数据驱动方法通过在大模型的训练数据集中嵌入特定的模式或标签，实现了数据所有权的验证或非法复制和滥用的追踪。这些方法通常依赖于后门插入技术，即在数据集中添加少量带有水印的样本，使模型能够隐式学习防御者设置的加密函数。当特定的触发器被激活时，后门水印被触发，这通常在黑盒设置中实现<sup>[36]</sup>。这种机制通过在模型训练的基础和多任务学习框架阶段嵌入后门，保护模型免受未经授权的微调或超出许可证条款的使用。即使模型针对多个下游任务进行了微调，水印仍然难以消除。

然而，后续研究发现了这种技术的漏洞，表明它相对容易被攻破。Lucas 等人<sup>[37]</sup>通过分析自回归模型生成的内容，详细描述了一种针对这种水印策略的攻击方法，以精确定位后门水印的触发词或短语。研究指出，由随机组合的常用词汇组成的触发器比由独特和罕见标记组成的触发器更容易被检测。此外，研究提到，访问模型的权重是检测后门水印的唯一先决条件。最近，Tang 等人<sup>[38]</sup>提出了一个清洁标签后门水印框架，该框架使用对抗性扰动来标记和触发样本。这种方法在最小化对原始任务性能影响的同时有效地保护了数据集。结果表明，仅添加 1% 的水印样本就可以注入可追踪的水印特征。

数据驱动方法最初是为了保护数据集的版权而设计的，因此通常缺乏大量的有效载荷容量和泛化能力。此外，在 AIGC 文本检测领域应用这些技术需要大量的资源投入，包括在大量数据中嵌入水印和对大模型进行重新训练。

#### 2.1.2 模型驱动水印

模型驱动方法通过在推理过程中操纵 logits 输出分布或字符采样，将水印直接嵌入到大模型中。因此，大模型生成的回应会携带嵌入的水印。

##### 2.1.2.1 基于 Logits 的方法

Kirchenbauer 等人<sup>[39]</sup>首次为大模型设计了一个基于 logits 的水印框架，其特点是对文本质量影响最小，无需访问大模型的接口或参数。在文本生成之前，

该方法随机选择一组“绿色”字符，将其余定义为“红色”，然后引导模型在采样过程中从“绿色”集合中选择字符。此外，他们还提出了一种基于可解释  $p$  值的水印检测方法，通过在文本中对红色和绿色字符进行统计分析来计算  $p$  值的显著性，以此识别水印。之后，Lee 等人<sup>[40]</sup>提出了一种名为 SWEET 的新水印方法，该方法仅在字符分布熵高的位置上提升“绿色”字符，从而保持水印的隐蔽性和完整性。它使用基于熵的统计测试和  $Z$  分数来检测加水印的代码。

尽管 Kirchenbauer 等人<sup>[39]</sup>提出的方法表现优异，但其鲁棒性仍然存在争议。他们的最新研究<sup>[41]</sup>探讨了水印文本对手动重写、使用未加水印的大模型重写或整合到大量手写文档中的攻击的抵抗力。这项研究引入了一种称为“WinMax”的窗口测试方法，以评估在大量文档中准确检测水印区域的有效性。为了解决同义词替换和文本改写带来的挑战，Liu 等人<sup>[42]</sup>提出了一种语义不变的鲁棒水印方法。这种方法为所有前面的字符生成语义嵌入，并用它们来确定水印逻辑，展现出对同义词替换和文本改写的鲁棒性。此外，当前的水印检测算法在生成过程中需要一个密钥，这可能导致安全漏洞和公共检测过程中的伪造问题。为了解决这个问题，Liu 等人<sup>[43]</sup>提出了第一个私有水印算法，用于水印生成和检测，每个阶段部署两个不同的神经网络。在两个阶段都避免使用密钥，这种方法扩展了现有的文本水印算法。此外，它在水印生成和检测网络之间共享某些参数，从而提高了检测网络的效率和准确性，同时最小化对生成和检测速度的影响。

#### 2.1.2.2 基于字符采样的方法

在正常的模型推理过程中，字符采样通常由采样策略决定，往往是随机的，这有助于引导大模型产生更不可预测的文本。基于字符采样的方法通过影响字符采样过程来实现水印，主要通过设置随机种子，或者为字符采样指定特定模式。Kuditipudi 等人<sup>[44]</sup>使用一系列随机数字作为秘密水印密钥来干预并决定字符采样，然后将这些采样映射到大模型中以生成加水印的文本。在检测阶段，利用秘密密钥将文本与随机数序列对齐，以完成检测任务。这种方法在对抗改写攻击方面显示出强大的鲁棒性，即使大约 40-50% 的字符被修改，也能保持鲁棒性。

SemStamp<sup>[45]</sup>，它是一种基于局部敏感哈希（LSH）的鲁棒句子级语义水印算法。该算法首先对大模型生成的候选句子进行编码和 LSH 哈希处理，将语义嵌入空间划分为加水印和未加水印的区域。然后，它持续执行句子级的拒绝采样，直到采样的句子落入语义嵌入空间的加水印部分。实验结果表明，这种方法不仅在防御常见和更有效的二元组改写攻击方面比之前的 SOTA 方法更鲁棒，而且在保持文本生成质量方面也更优越。

SynthID-Text<sup>[1]</sup> 是一种生成性水印方案，它结合了一个新颖的采样算法——锦标赛采样，如图2-1。该方法提供了两种配置：无失真（保持文本质量）和有失真（通过牺牲文本质量来提高水印可检测性）。该方案旨在检测水印，同时根

据所选配置保留或优化文本质量。此水印方法已经应用于实际的生产系统，如 Gemini 和 Gemini Advanced<sup>[20]</sup>，并且证明了它的可扩展性和实际应用能力。

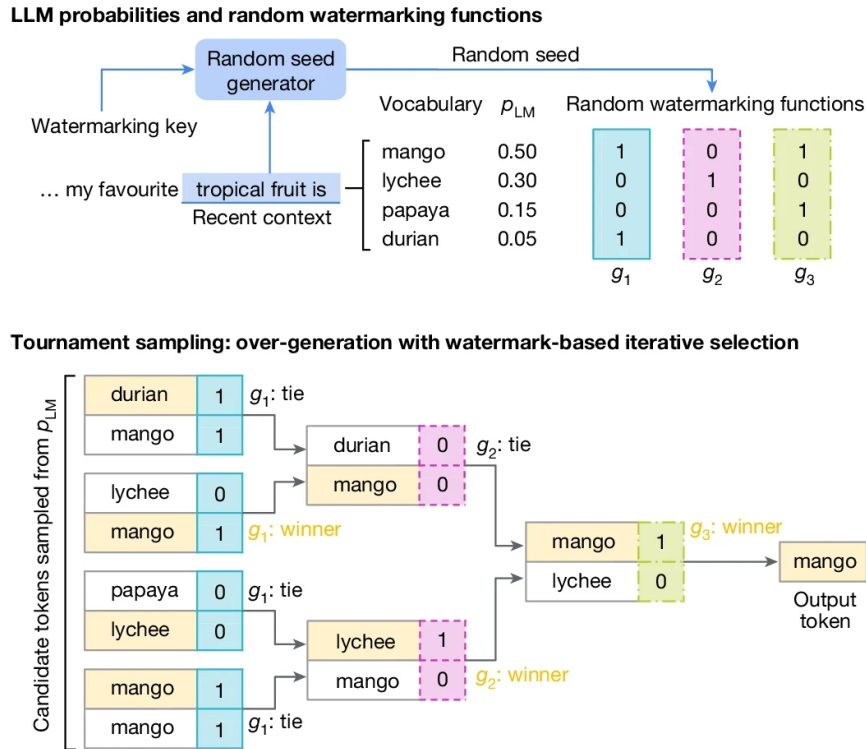


图 2-1 基于字符采样的 AIGC 检测方法<sup>[1]</sup>

Figure 2-1 AIGC detection method based on character sampling<sup>[1]</sup>

总的来说，模型驱动的水印是一种即插即用的方法，不需要对模型参数进行任何更改，对文本质量的影响最小，使其成为一种可靠且实用的水印方法。然而，其鲁棒性仍有显著的改进空间，具体适用性需要通过更多的实验和实际应用进一步探索。

### 2.1.3 后处理水印

这种方法通常作为一个独立的模块，作为处理流程的一部分，对输出文本进行额外处理，以添加水印信息。

#### 2.1.3.1 字符嵌入方法

早期的后处理水印技术依赖于在文本中插入或替换特殊的 Unicode 字符。这些字符肉眼难以识别，但携带了明确的编码信息<sup>[46,47]</sup>提出了 Easymark 方法，该方法巧妙地利用了 Unicode 中许多具有相同或相似外观的代码点这一事实。具体来说，Easymark 通过替换常规空格字符（U+0020）与另一个空白代码点（例如 U+2004），使用 Unicode 的变化选择器，替换子字符串，或使用长度略有不同的空格和同形异义字，同时确保文本的外观几乎不变。结果表明，Easymark 嵌入的

水印可以在不降低 BLEU 分数或增加文本困惑度的情况下可靠地被检测到，提高了在质量和水印可靠性方面的水平。

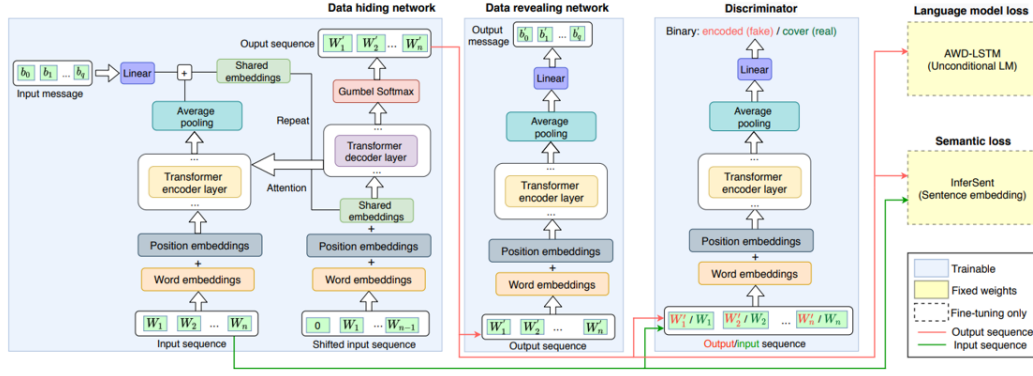
### 2.1.3.2 同义词替换方法

鉴于字符级方法容易受到针对性攻击的影响，一些研究转向在单词级别嵌入水印，主要是通过同义词替换。早期的水印嵌入方案涉及不断用同义词替换单词，直到文本携带预期的水印内容。为了解决这个问题，Topkara 等人<sup>[48]</sup>使用 Wordnet<sup>[49]</sup>提出了一种可量化且具有弹性的水印技术。在此基础上，Yang 等人<sup>[50]</sup>、Munyer 等人<sup>[51]</sup>、Yoo 等人<sup>[52]</sup>采用了微调预训练模型的方法进行单词替换和检测任务，从而更好地保持原始句子的语义完整性。此外，Yang 等人<sup>[53]</sup>定义了一个二进制编码函数来计算对应于单词的随机二进制代码，并选择性地用代表二进制“1”的上下文相关同义词替换代表二进制“0”的单词，有效地嵌入了水印。实验表明，这种方法确保了水印在重译、文本润色、单词删除和同义词替换等攻击下的鲁棒性，同时不损害原始文本的语义。

### 2.1.3.3 端到端方法

最近的研究探索了端到端的水印加密技术，目标是提高灵活性并减少水印引入的人工痕迹。例如，Abdelnabi 和 Fritz<sup>[2]</sup>提出了对抗性水印变换器 (AWT)，这是第一个自动化学习单词替换及其内容以嵌入水印的框架，如图2-2。这种方法结合了端到端和对抗训练，能够在编码层将二进制消息注入指定的输入文本，产生不易察觉且最小程度改变输入语义和正确性的输出文本。该方法使用 Transformer 编码层来提取嵌入在文本中的加密信息。同样，Zhang 等人<sup>[54]</sup>提出了 REMARK-LLM 框架，包括三个组成部分：(i) 一个消息编码模块，它将二进制签名注入由大模型生成的文本中；(ii) 一个重新参数化模块，将消息编码的密集分布转换为生成加水印文本标记的稀疏分布；(iii) 一个专门用于提取签名的解码模块。实验表明，与 AWT 相比，REMARK-LLM 能够在保持语义完整性的同时嵌入更多的签名位，并显示出对各种水印移除和检测攻击的增强抵抗力。

与模型驱动的水印相比，后处理水印可能更多地依赖于特定规则，使其更容易受到利用可见线索的复杂攻击的影响。尽管存在这种风险，后处理水印在各种应用中仍具有显著的潜力。许多现有的水印技术通常需要在白盒模型中进行训练，使它们不适用于黑盒大模型设置。例如，由于 GPT-4 的专有和闭源性质，嵌入水印几乎是不可能的。尽管如此，后处理水印为向黑盒大模型生成的文本添加水印提供了解决方案，使第三方能够独立嵌入水印。

图 2-2 基于 AWT 的 AIGC 文本检测<sup>[2]</sup>Figure 2-2 AIGC detection method based on AWT<sup>[2]</sup>

## 2.2 基于统计分析的 AIGC 文本检测方法

在基于统计分析的 AIGC 文本检测方法中，无需通过监督信号进行额外的训练。这种方法假设能够获取对 LLM 的访问权限或从文本中提取相关特征，并依赖于这些独特特征与统计数据来推导统计规律，例如计算阈值。

### 2.2.1 白盒统计

目前，用于检测 LLM 生成文本的白盒方法需要直接访问源模型才能实现。现有的白盒检测技术主要采用零样本方法，涉及获取模型的输出并计算特定指标。然后，这些指标与通过统计方法确定的预定阈值进行比较，以识别 LLM 生成的文本。

#### 2.2.1.1 基于 Logits 的统计方法

Logits 是大模型在文本生成过程中产生的原始输出，特别来自于模型的最后一个线性层，通常位于 softmax 函数之前。这些输出表明了模型对于生成每个潜在后续词的信心水平。Log-Likelihood<sup>[55]</sup> 是一种直接从 logits 派生的指标，它通过计算原始 LLM 来测量所提供文本中每个词元的平均 token-wise 对数概率。这种测量有助于确定文本是由 LLM 生成的可能性。目前，Log-Likelihood 被认为是 LLM 生成文本检测任务中最受欢迎的基线指标之一。

同样，Rank<sup>[55]</sup> 是另一个从 logits 计算的基线。Rank 指标计算样本中每个词在模型输出概率分布中的排名。这个排名是通过将词的 logit 分数与所有其他可能词的 logit 分数进行比较来确定的。如果样本中每个词的平均排名较高，这表明样本可能是由 LLM 生成的。Log-Rank 则进一步通过应用对数函数处理每个 token 的排名值，并引起了越来越多的关注。另一个直观方法是 GLTR<sup>[56]</sup>，它被设计为一种视觉取证工具，以促进比较和判断。该工具根据字符的采样频率级别划分不同的标记颜色，并通过对提供文本的字符进行不同颜色的标记，突

出大模型在分析文本中倾向于使用的词的比例。Su 等人<sup>[57]</sup>提出的 LRR 结合了 Log-Likelihood 和 Log-Rank，通过取两个指标的比值。这种方法通过有效地将 Log Likelihood 评估与 Log Rank 分析相结合，提供了更全面的评估，从而提高了性能。

熵是另一种用于评估 LLM 生成文本的早期零样本方法。它通常被用来测量文本或模型输出中的不确定性或信息量，也是通过词的概率分布来计算的。高熵表明样本文本的内容不清晰或高度多样化，意味着许多词被选择的概率相似。在这种情况下，样本可能是由 LLM 生成的。Lavergne 等人<sup>[58]</sup>采用了 KL 散度来给 n-gram 打分，考虑它们初始和最终词之间的语义关系。这种方法识别了初始和终端词之间存在显著依赖关系的 n-gram，从而有助于检测虚假内容，并提高了检测过程的整体性能。

最近的一些工作，如 HowkGPT<sup>[59]</sup>通过计算和比较学生撰写和 ChatGPT 生成文本的困惑度分数来识别大模型生成的文本，特别是课后作业。通过这种比较，建立了识别提交作业来源的阈值。此外，广受认可的 GPTZero<sup>1</sup>基于文本的困惑度和突发性指标，估计了评论文本由大模型生成的可能性。在最近的一项研究中，Wu 等人<sup>[60]</sup>提出了 LLMDet，用于量化和记录不同模型对选定 n-gram 的困惑度分数，具体通过计算下一个 token 的概率实现。LLMDet 利用了文本的内在自水印特性，如代理困惑度，来追踪文本来源并进行有效检测。此外，Venkatraman 等人<sup>[61]</sup>通过分析文章的字符概率提取基于 UID 的特征，然后训练一个逻辑回归分类器来适应不同大模型生成文本的 UID 特征，以识别文本的来源。GHOSTBUSTER<sup>[62]</sup>将大模型生成的文本输入到一系列较弱的语言模型中以获得字符概率，然后对这些模型输出的组合进行结构化搜索，以训练一个线性分类器来区分大模型生成的文本。

### 2.2.1.2 基于扰动的统计方法

一些零样本方法通过比较统计扰动后性能指标的差异来检测大模型生成的文本。Mitchell 等人<sup>[3]</sup>提出了一种方法，通过分析大模型概率函数的结构特征，尤其是负曲率区域，来识别由大模型生成的文本。该方法的核心假设是，大模型生成的文本通常会集中在局部对数概率的最大值附近。检测过程利用预训练的掩码填充模型（如 T5）生成语义相似的文本扰动，并比较这些文本的对数概率与原文本的对数概率，从而实现文本的识别。

虽然这种方法创新且有时比有监督方法更有效，但 DetectGPT 存在局限性，包括如果重写未能充分代表有意义的替代空间，则可能导致性能下降，以及高计算需求，因为它需要对许多文本扰动进行评分。为了应对这一挑战，Miao 等人<sup>[63]</sup>提出了一种使用贝叶斯替代模型来选择少量典型样本进行评分的方法。通

<sup>1</sup><https://gptzero.me/>

过将典型样本的评分插值到其他样本来提高查询效率，同时保持性能。Bao 等人<sup>[4]</sup>使用更高效的采样步骤替换了 DetectGPT 中的扰动步骤，显著提高了检测精度，并将检测速度提高了 340 倍，如图2-3。DetectLLM<sup>[57]</sup>与 DetectGPT 的概念框架相似。它采用标准化扰动对数排名来检测大模型生成的文本，对扰动模型和扰动次数的敏感性更低。

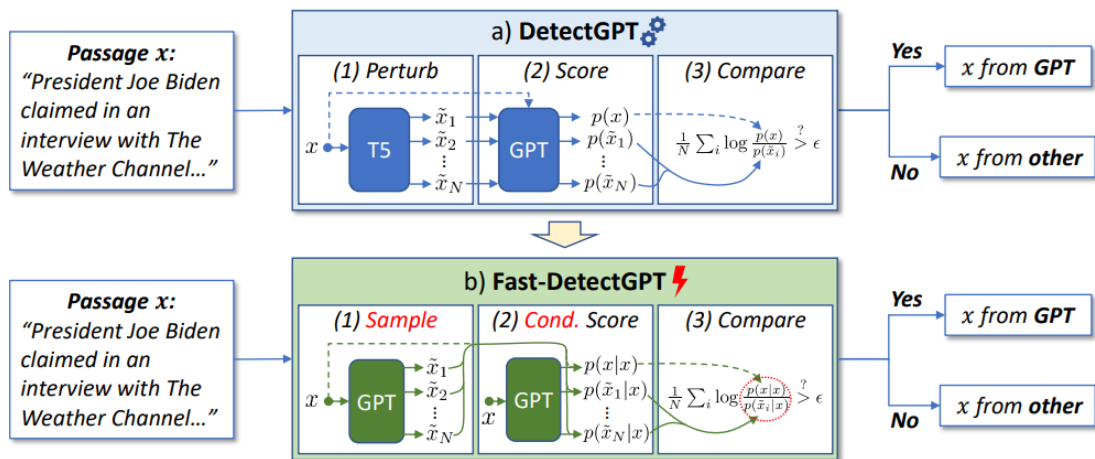


图 2-3 基于扰动的 AIGC 文本检测<sup>[3,4]</sup>

Figure 2-3 AIGC detection method based on disturbance<sup>[3,4]</sup>

### 2.2.1.3 基于内在维度估计的统计方法

Tulchinskii 等人<sup>[64]</sup>进行的研究提出了人类和大模型在各自文本领域中表现出能力的不变性。该研究利用给定文本样本的嵌入集所支撑的流形的内在维度来构建检测器，即计算目标自然语言中流畅的人类写作文本和大模型生成文本的平均内在维度值。这两组数据之间的统计分离有助于为目标语言建立一个分离阈值，从而实现对大模型生成文本的检测。这种方法在多种场景下具有鲁棒性，包括跨领域挑战、模型变化和对抗性攻击。然而，当面对次优或高温生成器时，其可靠性会下降。

### 2.2.2 黑盒统计

与需要了解模型内部细节（如模型参数或输出 logits）的白盒统计方法不同，黑盒统计方法在检测或分析文本时无需直接访问源模型或替代模型的内部数据（例如 logits）。相反，黑盒方法将模型视为一个“不可见的盒子”，通过输入文本并观察其输出结果（如预测分数或概率），来计算文本的某些特征评分。这种方法更灵活，因为它不依赖于对模型内部机制的了解，而是通过与模型交互获取必要的信息。DNA-GPT<sup>[65]</sup>利用大模型续写被截断的文本，它通过计算原始文本和续写文本之间的概率散度来分析差异。Zhu 等人<sup>[66]</sup>和 Mao 等人<sup>[67]</sup>提出了一种通过计算原始文本与其重写或修订版本之间相似度来识别大模型生成文本的

方法。这些方法基于这样一个假设：相比于大模型生成的文本，人类编写的文本在被要求重写或编辑时往往会引发更多的修订。Yu 等人<sup>[68]</sup>引入了一种检测机制，也利用原始文本和再生文本之间的相似性。与其他方法不同，该方法首先识别提示生成文本的原始问题，并根据该推断问题重新生成文本。此外，Quidwai 等人<sup>[69]</sup>分析了 LLM 生成文本及其改写的句子，通过计算余弦相似性来区分它们与人类编写的文本。Guo 等人<sup>[70]</sup>提出了一种基于去噪的黑盒零样本统计方法，该方法使用 LLM 对输入文本中人为添加的噪声进行去噪。然后，将去噪后的文本与原始文本进行语义比较。然而，黑盒统计方法访问 LLM 耗费大量的开销和长时间的响应时间。

## 2.3 基于神经网络的 AIGC 文本检测方法

### 2.3.1 基于特征的 AIGC 文本检测方法

#### 2.3.1.1 基于语言特征的 AIGC 文本检测方法

众多语言学特征的差异为基于特征的 AIGC 文本检测方法提供了有力的区分依据。这类方法的工作流程通常从提取关键的统计语言特征开始，然后应用机器学习技术来训练分类模型。该方法已广泛用于假新闻的识别。例如，Aich 等人<sup>[71]</sup>通过提取 21 个文本特征并使用 KNN 分类器。借鉴于检测假新闻和 LLM 生成文本的任务，文本的语言学特征可以被广泛分类为风格特征、复杂性特征、语义特征、心理特征和基于知识的特征。这些特征主要是通过统计方法获得的。

风格特征主要关注可以突出文本风格元素的词频，包括大写词、专有名词、动词、过去时态词、停用词、技术术语、引用和标点符号的频率<sup>[72]</sup>。复杂性特征被提取以展示文本的复杂性，如类型-标记比和文本词汇多样性<sup>[73]</sup>。语义特征包括高级语义、词汇语义以及语义依存标记的统计数据等语义层面特征。这些特征可以使用 LingFeat<sup>[74]</sup>等工具提取。心理特征通常与情感分析相关，可以基于 SentiWordNet<sup>[75]</sup>等工具来计算情感分数或使用情感分类器提取。信息特征包括命名实体、观点和实体关系提取，可以使用 UIE<sup>[76]</sup>和 CogIE<sup>[77]</sup>等工具提取。

Shah 等人<sup>[78]</sup>基于音节数、词长、句子结构、功能词使用频率和标点符号比例等风格特征构建了一个分类器，有效地展示了风格特征对 LLM 生成文本检测的重要性。其他工作通过数据融合技术将文本建模与各种语言学特征相结合<sup>[79]</sup>，包括不同类型的标点符号的使用、段落结构、平均句子长度、高频词的重复性以及情感分数。Mindner 等人<sup>[80]</sup>进一步采用了多维方法来增强分类器的辨别能力，包括复杂性度量、语义分析、列表搜索、基于错误的特征、可读性评估、人工智能反馈和文本向量特征。

尽管基于语言学特征的 AIGC 文本检测方法在区分人类和 AIGC 文本方面具有优势，但其不足之处也不容忽视。Schaaff 等人<sup>[81]</sup>的研究结果表明，这类特

征分类器对模糊语义的鲁棒性较差,且通常表现不如神经网络。此外,基于风格特征的分类方法虽然能够区分人类编写的文本和 LLM 生成的文本,但其检测 LLM 生成的虚假信息的能力有限。这一局限性在 Schuster 等人<sup>[82]</sup>的研究中得到了突出,该研究表明语言模型倾向于产生风格一致的文本。然而, Crothers 等人<sup>[83]</sup>提出,统计特征可以提供额外的对抗鲁棒性,并且可以用于构建集成检测模型。

### 2.3.1.2 基于模型特征的 AIGC 文本检测方法

除了语言学特征,基于模型特征的分类器也引起了研究者的广泛关注。这些分类器不仅能检测 LLM 生成的文本,还可以用于文本来源追踪。当比较一个模型  $\theta_i$  和另一个模型  $\theta_j$  时, Sniffer<sup>[84]</sup> 提取对齐的字符级困惑度和对比特征,计算具有较低困惑度的单词的百分比。SeqXGPT<sup>[85]</sup> 在文本来源追踪领域进行了进一步探索,基于提出的特征设计了一个上下文网络,该网络结合了 CNN 和两层 transformer 对文本进行编码,并通过序列标注任务进行文本检测。然而,这些方法的一个共同缺点是它们都需要访问源模型的 logits。对于那些 logits 不可见的强大闭源模型,这些方法可能难以发挥作用。

### 2.3.2 基于预训练模型的 AIGC 文本检测方法

随着与训练技术的进步,与训练模型被广泛运用于 AIGC 文本检测领域,有效克服了传统有监督学习标注数据缺乏和泛化能力不足的问题。

#### 2.3.2.1 基于领域内模型微调的 AIGC 文本检测方法

这一小节主要涉及微调基于 Transformer 的语言模型的检测方法,该方法需要成对的样本对模型进行有监督的微调。预训练的语言模型在自然语言理解方面已被证明是强大的,这对于增强 NLP 中的各种任务至关重要,尤其是在文本分类方面尤为显著。

微调 Roberta 模型为 AIGC 文本检测提供了一个鲁棒的基线。最近的工作<sup>[86-89]</sup>发现微调的 BERT 系列模型,如 RoBERTa,在识别 AIGC 文本方面的优越性能。平均而言,这些微调模型在各自领域内的准确率超越了零样本和水印方法,并在领域内对各种攻击技术表现出一定的韧性。尽管如此,基于编码器的微调方法缺乏一定的鲁棒性,因为它们容易过度拟合训练数据或源模型的训练分布,导致在面对跨领域或未见数据时性能下降。此外,微调后的模型在面对不同模型生成的数据时也是有限的。尽管如此,基于 RoBERTa 的检测器在鲁棒性方面展现出显著潜力,只需要几百个标注数据进行微调就能取得不错的结果。mBERT<sup>[90]</sup> 在文档级的 AIGC 文本检测中表现出一致的鲁棒性能,特别是在英语和西班牙语任务中性能最佳。相比之下,像 XLM-RoBERTa<sup>[91]</sup> 和 TinyBERT<sup>[92]</sup> 这样的编码器

模型在相同的文档级任务中显示出显著的性能差异，表明这两个任务需要模型具备不同的能力。

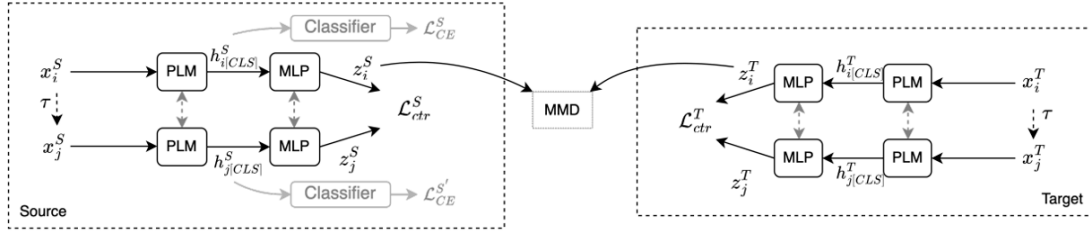


图 2-4 基于对比学习的 AIGC 文本检测<sup>[5]</sup>

Figure 2-4 AIGC detection method based on contrast learning<sup>[5]</sup>

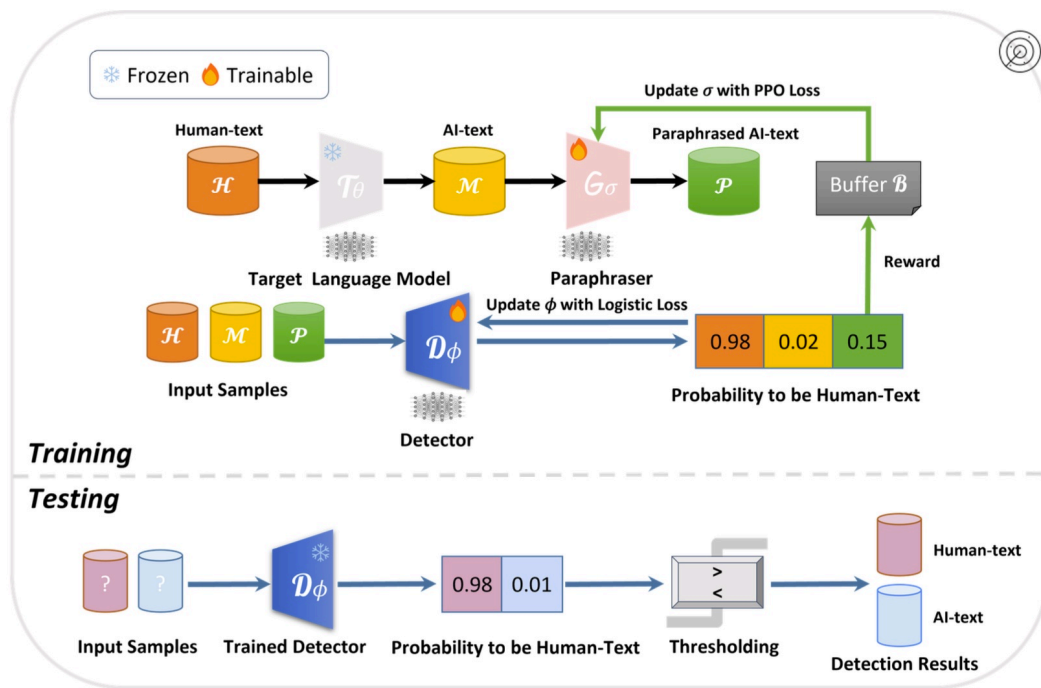
### 2.3.2.2 基于对比学习的 AIGC 文本检测方法

数据稀缺性推动了对比学习在 AIGC 文本检测中的应用，该方法的核心是自监督学习。这种策略通过空间变换，旨在最小化锚点和正样本之间的距离，同时最大化与负样本的距离。Liu 等人<sup>[93]</sup>提出了一种增强的对比损失，它给困难负样本分配更大的权重，从而优化模型的效果并促进在资源匮乏环境中的性能提升。这种方法充分考虑了语言特征和句子结构，将文本表示为一致性图，以封装其固有的实体一致性。研究结果证实了结合信息事实结构以提高基于语言模型的检测器效能的有效性。随后，Bhattacharjee 等人<sup>[5]</sup>提出了 ConDA，这是一个对比域适应框架（图2-4），它结合了标准的域适应技术和对比学习的表征能力，极大地提高了对未知模型的检测能力。

### 2.3.2.3 基于对抗学习的 AIGC 文本检测方法

鉴于检测器对不同攻击的脆弱性和鲁棒性问题，大量研究利用对抗性学习作为一种缓解策略。主要来说，对抗性学习方法与微调语言模型方法相关。Koike 等人<sup>[8]</sup>的最新研究揭示了在不微调模型的情况下进行对抗性训练是可行的，通过冻结模型参数，并利用上下文信息来引导模型，能够有效地进行对抗性训练。基于对抗学习的 AIGC 文本检测研究主要分为两类：基于样本增强的对抗训练和双人游戏。

在基于样本增强的对抗训练中，一个方法是部署基于样本增强的对抗性攻击，其总体目标是制造欺骗性输入，从而增强模型处理具有潜在欺骗性的更广泛场景的能力。具体来说，这种方法强调样本增强的重要性，并通过注入预定的对抗性攻击来实现。这种增强过程对于加强检测器的鲁棒性至关重要，为检测器提供了更广泛的对抗样本池。Yang<sup>[94]</sup>，Shi 等人<sup>[95]</sup>以及 He 等人<sup>[96]</sup>对 LLM 生成的文本进行了对抗性数据增强处理，发现在增强的数据上训练的模型对潜在攻击表现出了可观的鲁棒性。

图 2-5 基于对抗学习的 AIGC 文本检测<sup>[6]</sup>Figure 2-5 AIGC detection method based on adversarial learning<sup>[6]</sup>

双人游戏的方法基本上与生成对抗网络 (GANs)<sup>[97]</sup> 和 Break-It-Fix-It 策略<sup>[98]</sup> 的原则一致, 通常涉及攻击模型和检测模型, 两者之间的迭代对抗最终增强检测能力。Hu 等人<sup>[6]</sup> 提出 RADAR 框架 (图2-5), 该框架引入了一个改写模型, 负责生成能够逃避检测的真实内容, 同时还包括一个检测器, 旨在提高识别 AIGC 文本的能力。这两个组件通过互动, 利用强化学习中的 PPO 算法以及来自检测器的反馈, 逐步改进改写模型。

尽管 RADAR 在对抗改写攻击方面表现出色, 但它并未提供对其他类型攻击的防御机制的全面分析。同样, Koike 等人<sup>[8]</sup> 提出了一种基于攻击者和检测器之间持续互动的检测器训练方法。与 RADAR 不同, OUTFOX 更加强调检测器利用上下文学习 (ICL) 对攻击者进行识别。具体来说, OUTFOX 框架中的攻击者利用检测器预测的标签作为 ICL 样本来生成构成检测挑战的文本。相反, 检测器使用对抗性生成的内容作为 ICL 样本来增强其对抗强大攻击者的检测能力。这种相互考虑对方输出的做法促进了 LLM 生成文本检测器的鲁棒性的提高。

#### 2.3.2.4 基于特征增强的 AIGC 文本检测方法

除了在训练方法上的改进, Tu 等人<sup>[99]</sup> 还证明了提取语言学特征能够有效提高基于 RoBERTa 的检测器的鲁棒性, 并且在多种相关模型中取得了显著效果。Cowap 等人<sup>[7]</sup> 基于预训练语言模型开发了一个情感感知检测器, 从而增强了情感特征在识别合成文本中的潜力, 如图2-6。他们通过针对情感分类进一步微调

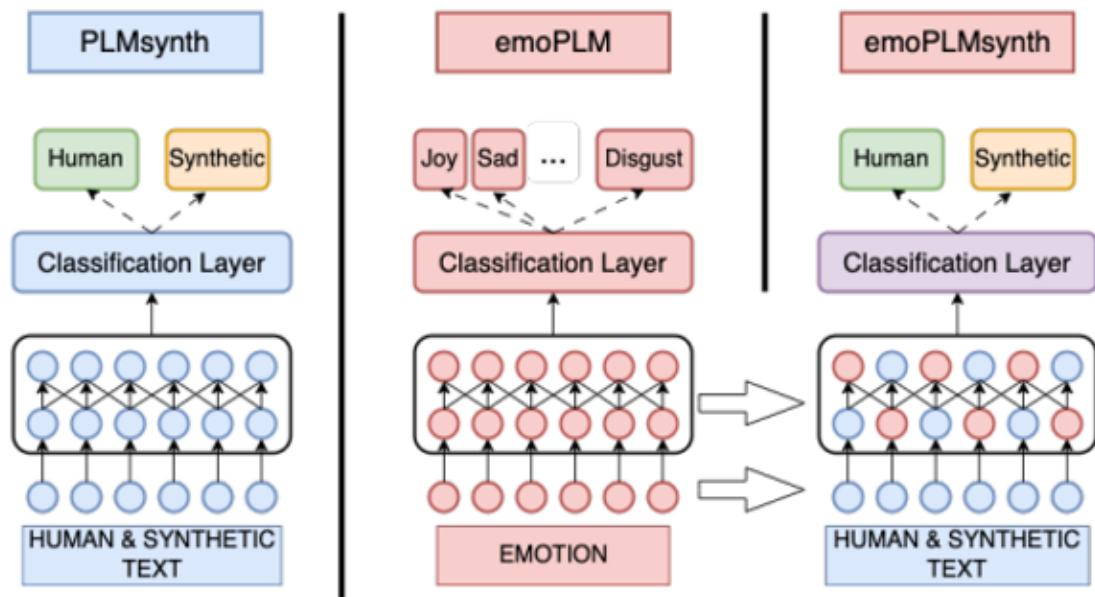


图 2-6 基于特征增强的 AIGC 文本检测<sup>[7]</sup>

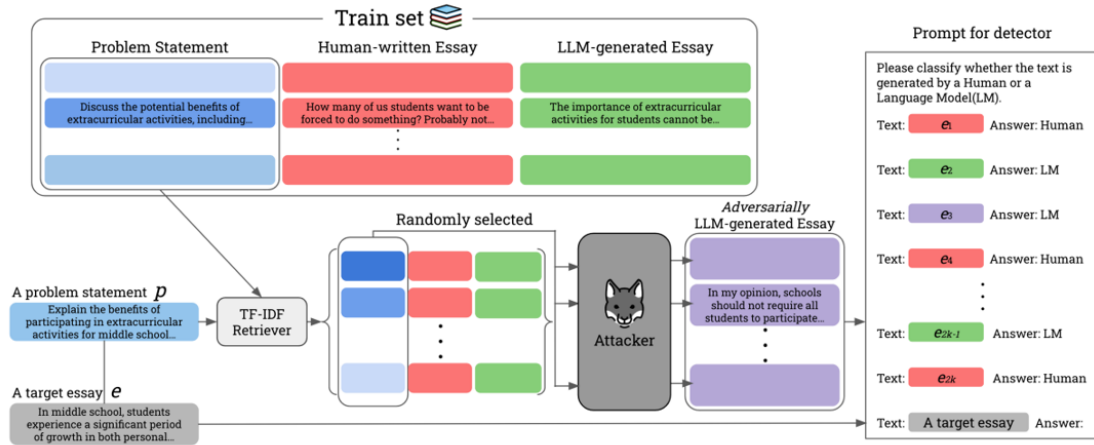
Figure 2-6 AIGC detection method based on feature enhancement<sup>[7]</sup>

BERT，实现了 F1 值提升高达 9.03%。

Uchendu 等人<sup>[100]</sup> 则使用 RoBERTa 捕捉上下文表示，如语义和句法特征，并结合拓扑数据分析方法，分析数据的形状和结构，包括语言结构。这种方法在 SynSciPass 和 M4 数据集上超越了单独使用 RoBERTa 的表现。J-Guard 框架<sup>[101]</sup> 通过提取新闻特征，引导有监督检测器识别 AI 生成的新闻内容，这些特征帮助检测器识别 LLM 生成的假新闻文本。该框架展现出强大的鲁棒性，在面对对抗性攻击时，平均性能仅下降 7%。

### 2.3.3 基于大模型的 AIGC 文本检测方法

尽管使用大模型直接检测 AIGC 文本存在不可靠问题，但最近的实证研究强调了上下文学习在增强大模型检测能力方面的潜在有效性。上下文学习是一种特殊的提示工程形式，它将示例整合到提供给模型的提示中，从而促进大模型学习新任务。通过上下文学习，现有的大模型能够熟练处理不同任务，而无需额外的微调。如图2-7，OUTFOX<sup>[8]</sup> 采用了上下文学习的方法，不断为大模型提供示例样本以进行文本生成检测任务。实验结果表明，该方法表现优于传统的零样本方法和基于 RoBERTa 的检测器。

图 2-7 基于大模型的 AIGC 文本检测<sup>[8]</sup>Figure 2-7 AIGC detection method based on LLM<sup>[8]</sup>

## 2.4 AIGC 文本检测研究趋势与挑战

近年来，AIGC 文本检测领域的研究确实呈现出一些显著的发展趋势。(1) 为了应对跨领域和跨模型泛化能力的挑战，检测方法正逐渐从传统的特征工程方法转向基于预训练模型的微调方法。因其强大的泛化能力和适应性，成为研究的热点。(2) 研究者们也在积极探索如何提升模型的鲁棒性，特别是在资源受限的环境中的性能。这包括对比学习、对抗学习等技术，它们通过增强模型对多样化数据的理解和处理能力，提高了检测器的稳定性和准确性。(3) 模型规模的增长也是一个明显的趋势。从小模型到大模型的转变，不仅增强了模型的处理能力，也提高了其在复杂任务中的表现。随着提示工程的兴起，研究者们开始更多地关注上下文学习，这种方法通过将示例整合到模型的提示中，使得大型语言模型能够在无需额外微调的情况下处理各种不同的任务。

尽管 AIGC 文本检测领域取得显著的研究进展，但也面临着一些挑战。(1) 检测方法的泛化能力挑战：传统检测方法通常依赖单一特征或特征的简单组合，难以适应不同文本类型和复杂特征分布的变化，导致对多样化生成模型输出的检测效果较差。尤其在跨领域、跨语言和跨模型的场景中。(2) 功能性语言文本检测挑战：尽管当前的检测方法在形式语言文本的识别上表现出色，能够有效捕捉语法、词汇和句法方面的流畅性，但在功能性语言文本的检测上仍然存在显著局限，尤其是在逻辑推理和数学运算的领域。



## 第3章 基于结构信息熵的 AIGC 文本检测方法

### 3.1 引言

自然语言处理(NLP)的快速发展催生了 ChatGPT<sup>[102]</sup>、GPT-4<sup>[10]</sup> 和 LLaMA3<sup>[103]</sup> 等大型语言模型 (LLMs)。这些技术进步显著提升了 LLMs 生成连贯且高度拟人化文本的能力。因此, 在多个领域中, 检测 LLM 生成文本已成为一项关键任务。这些模型在问答、文章撰写和代码生成等任务上表现出色。然而, LLMs 的滥用可能会带来不良的社会影响, 例如学术不端<sup>[104]</sup>、虚假信息传播<sup>[105–107]</sup>、网络钓鱼<sup>[108,109]</sup>、虚假产品评论<sup>[106,110]</sup>, 以及有害垃圾信息的泛滥<sup>[111]</sup>。鉴于这些潜在的负面影响, 研究如何检测文本是由机器生成还是人工撰写至关重要。

尽管监督检测方法<sup>[55,86]</sup> 通过在标注数据集上微调分类器取得了良好的检测效果, 但它们面临两大关键挑战: (1) 依赖昂贵且高质量的标注数据, 而该数据难以跟上 LLMs 的快速发展; (2) 由于对特定生成模式的过拟合, 这些方法在不同 LLMs 之间的泛化能力有限。无监督方法<sup>[3,56]</sup> 避免了上述问题, 但通常仅基于独立的文本特征 (如对数似然或 token 分布) 进行分析, 未能捕捉文本之间的更深层次统计关系。

一个重要的观察是, LLM 生成的文本不仅在独立的统计特性上存在相似性, 还在其关联结构上具有内在共性。相同或相似 LLMs 生成的文本由于共享的生成模式, 在统计空间中往往会聚集在一起。然而, 现有检测方法大多聚焦于独立分析单个文本, 未能充分利用这些高阶结构依赖关系。因此, 如何建模和利用文本间的复杂关系, 以增强 LLM 生成文本的检测能力, 特别是在跨模型和跨数据集的场景下, 仍是一个重大挑战。

基于图的检测方法在捕捉全局结构关系方面展现出强大能力, 但目前 LLM 生成文本的检测研究尚未充分利用图结构。同时, 结构信息理论<sup>[112]</sup> 和二维结构熵最小化<sup>[113]</sup> 在无监督聚类任务中取得了显著成效。受这些研究成果的启发, 该方法提出利用多关系图和结构熵最小化方法来弥补这一研究空白。

为了解决这一挑战, 本章提出了基于结构信息熵的无监督检测方法 **Structural Information for LLM-generated Text Detection**, 记作 SILTD。与依赖单一特征分析的传统方法不同, SILTD 通过构建多关系文本图来建模文本间的结构依赖关系。该方法的方法包括以下三个关键步骤:

**多关系图构建:** 该方法将文本表示为图中的节点, 并通过边编码多维特征相似性。这种结构能够更丰富地表征文本在生成统计空间中的相互关系。

**基于结构熵最小化的层次聚类:** 该方法将多关系图转换为编码树, 并应用二维结构熵最小化算法, 以挖掘层次化的文本聚类结构。该过程能够发现文本数据

的潜在组织结构，并提高聚类质量。

**聚类内标签推理：**该方法提出一种结合 MultiRank 站稳概率分布和编码树熵的标签推理方法，以判断聚类是否由人工文本或 LLM 生成文本组成。

本研究的主要贡献如下：

- 提出了一种基于结构信息熵的无监督 LLM 生成文本检测方法。
- 提供了一种新的特征融合视角，通过构建多关系文本图，更好地捕捉文本间的相关性。
- 提出了一种二维结构熵最小化聚类算法，并基于 MultiRank 框架和聚类内熵进行标签推理。该方法能够对多关系图进行层次划分，从而实现准确高效的聚类分类。
- 在 WP、Essay 和 Reuters 等公开数据集上进行了广泛实验，这些数据集涵盖了六种主流大型语言模型生成的文本。实验结果表明，该方法达到了最先进的检测性能，并展现出良好的泛化能力。

## 3.2 相关工作

本节介绍了大模型文本生成与检测的相关研究工作，以及本文所采用的结构信息熵。

### 大模型文本生成

早期的语言模型基于统计方法<sup>[114–116]</sup>，利用大规模文本数据集计算单词或短语的出现概率来进行文本生成。然而，这些模型在处理长距离依赖方面存在局限性。随着深度学习的发展，语言模型开始使用神经网络来预测序列中下一个单词的概率，有效地解决了长距离依赖问题<sup>[117–119]</sup>。2017 年，Transformer 结构<sup>[120]</sup>通过自注意力机制和并行计算显著提升了模型性能和训练效率，使其能够更有效地处理长距离依赖。

预训练语言模型 (PLMs) 的出现，例如 GPT-2<sup>[121]</sup> 和 BERT<sup>[90]</sup>，确立了“预训练-微调”范式成为主流方法。随后，OpenAI 的 GPT 系列模型通过大规模无监督预训练和监督微调，在文本生成任务中展现了卓越的性能。GPT-3<sup>[122]</sup> 拥有 1750 亿参数，实现了更流畅、自然的文本生成。ChatGPT 通过引入人类反馈强化学习 (RLHF)<sup>[102]</sup> 进一步提升了文本质量和相关性。GPT-4<sup>[10]</sup> 进一步增强了文本生成和上下文理解能力。此外，许多开源大语言模型也陆续涌现，如 LLama<sup>[123]</sup>、ChatGLM<sup>1</sup> 和 Dolly<sup>2</sup>。

<sup>1</sup><https://github.com/THUDM/ChatGLM3>

<sup>2</sup><https://github.com/databricks/dolly>

## 大模型生成文本检测

大模型生成文本检测被定义为一个二元分类问题，其目标是区分人类生成的文本和机器生成的文本<sup>[124]</sup>。给定一个可能的文本表示集合  $S$ ，人类生成文本的分布  $h(s)$ ，以及机器生成文本的分布  $m(s)$ ，检测器  $D(s)$  将样本  $s \in S$  映射到一个实数。通过将  $D(s)$  与阈值  $\beta$  进行比较来执行分类：如果  $D(s) \geq \beta$ ，则该文本被分类为 AI 生成，否则被分类为人类生成。

早期的文本检测研究采用特征工程方法<sup>[125]</sup>。随后，GROVER<sup>[107]</sup> 被提出，该模型既能生成可控新闻文本，也能识别这些文本。为了使机器生成文本可检测，GLTR<sup>[56]</sup> 计算了每个 token 的对数似然值直方图。DetectGPT<sup>[3]</sup> 通过对文本进行扰动，并比较扰动前后模型输出的对数概率差异来检测文本是否为机器生成。Guo 等开发了基于 RoBERTa<sup>[126]</sup> 模型的 ChatGPT 文本检测器。

COCO<sup>[93]</sup> 是一种对比学习模型，通过图结构编码连贯性信息并改进对比损失，以解决低资源环境下的机器生成文本检测问题。DetectLLM<sup>[57]</sup> 提出了对数似然对数排序比 (LRR) 和归一化对数排序扰动 (NPR) 方法，利用对数排序信息联合分析对数似然和扰动效应，以更准确地捕捉生成文本和人类文本之间的细微差异。ConDA<sup>[5]</sup> 是一种对比领域自适应框架，将领域自适应技术与对比学习相结合，以实现无监督检测任务的领域不变表示学习。SSSE<sup>[127]</sup> 提出了一种新的框架，旨在解决半监督聚类在大规模数据集集中的可扩展性问题，以及在整合异构约束时的灵活性不足问题。

## 结构信息熵

结构熵理论最初由 Li 和 Pan 提出<sup>[112]</sup>，用于衡量在分层划分策略下同质图的不确定性。与以往研究类似<sup>[128]</sup>，分层划分可以用一种称为编码树的树结构来表示。本节中主要介绍了编码树和结构熵的概念及其应用。

**编码树 (Encoding Tree)** 图  $G$  的编码树定义为一个根树  $\mathcal{T}$ ，其满足以下性质：

- (1) 对于编码树  $\mathcal{T}$  中的每个节点  $\alpha$ ，在图  $G$  中存在一个对应的顶点子集  $T_\alpha$ ，其中  $T_\alpha \subseteq V$ ， $V$  表示图的所有顶点集合。
- (2) 对于编码树  $\mathcal{T}$  的根节点  $\lambda$ ，满足  $T_\lambda = V$ 。
- (3) 对于任意节点  $\alpha$ ，其子节点记作  $\alpha_i$ ，按照从左到右的顺序排列，索引  $i$  递增，且  $\alpha_i^- = \alpha$  表示  $\alpha_i$  的父节点。
- (4) 对于任意节点  $\alpha$ ，如果它有  $L$  个子节点，则顶点子集  $T_{\alpha_i}$  彼此不相交，且  $T_\alpha = \cup T_{\alpha_i}$ 。
- (5) 树中的每个叶子节点  $v$  对应于图  $G$  中的一个单独顶点，即  $T_v = v$ 。

**一维结构熵 (One-dimensional Structural Entropy)** 一维结构熵用于衡量图  $G$  在没有进行分层划分时的不确定性。此时，对应的编码树仅由一个根节点和叶子节

点组成，根节点代表整个顶点集  $V$ ，而每个叶子节点代表一个单独的顶点。一维结构熵的定义如下：

$$H^1(G) = - \sum_{v \in V} \frac{d_v}{\text{vol}(G)} \log_2 \frac{d_v}{\text{vol}(G)} \quad (3-1)$$

其中， $\text{vol}(G) = \sum_{v \in V} d_v$ ，即图  $G$  的体积 (volume)，是所有顶点的度数之和。

**K 维结构熵 (K-dimensional Structural Entropy)** 对于具有最大高度为  $K$  的编码树  $\mathcal{T}$ ，图  $G$  的  $K$  维结构熵定义如下：

$$H^{\mathcal{T}}(G) = - \sum_{\alpha \in \mathcal{T}, \alpha \neq \lambda} \frac{g_{\alpha}}{\text{vol}(G)} \log_2 \frac{\text{vol}(\alpha)}{\text{vol}(\alpha^-)} \quad (3-2)$$

$$H^K(G) = \min H^{\mathcal{T}}(G) \quad (3-3)$$

其中， $g_{\alpha}$  表示从顶点子集  $T_{\alpha}$  内的顶点到  $T_{\alpha}$  以外的顶点的所有边的总权重； $\text{vol}(\alpha)$  表示  $T_{\alpha}$  内所有顶点的度数总和； $\frac{g_{\alpha}}{\text{vol}(G)}$  表示通过随机游走到达  $T_{\alpha}$  的概率。

SEP<sup>[129]</sup> 是一种受结构熵启发的分层池化方法，采用全局优化算法来避免固定压缩配额和局部结构损伤，在图分类和节点分类任务中表现良好。结构熵可以分层引导状态抽象，提高强化学习算法的效率和效果<sup>[130,131]</sup>。在多智能体系统中，结构熵最小化有助于定义稳定且有效的角色，提高智能体间的协作能力<sup>[132]</sup>。

HISEvent<sup>[113]</sup> 是一种用于社交事件检测的无监督框架，通过 1D 和 2D 结构熵最小化来增强消息图结构，从而在无需先验事件数量的情况下检测事件。在对抗性社交机器人建模方面，SIASM<sup>[128]</sup> 通过最小化异构图的结构熵，提高了对抗行为建模的准确性和有效性。这些应用表明，结构熵在多个领域中的检测能力和性能提升方面具有广泛的适用性。

UnDBot<sup>[133]</sup> 提出了一种无监督、可解释的社交机器人检测框架，旨在解决当前基于神经网络的方法存在的黑箱问题。SEPC<sup>[134]</sup> 是一种新的概率嵌入模型，解决了现有基于信息瓶颈 (IB) 方法的两个关键限制。HAGK<sup>[135]</sup> 是一种新颖的图核框架，解决了传统 R-卷积图核忽略分层结构信息的关键问题。Yang 等提出了 Incre-2dSE，一种新的增量框架，用于解决动态图结构熵测量的关键问题，其中现有方法无法增量更新编码树或高效计算图快照间的结构熵。

### 3.3 方法介绍

本节介绍了一种基于结构信息熵的 LLM 生成文本检测方法 SILTD，其整体架构如图 3-1 所示。首先，基于文本特征在生成统计空间中的相似性构建文本多关系图。接下来，提出了一种基于结构信息熵的无监督图聚类方法。最后，结合 MultiRank 算法和簇内熵进行簇内标签推理，以获得最终的分类结果。

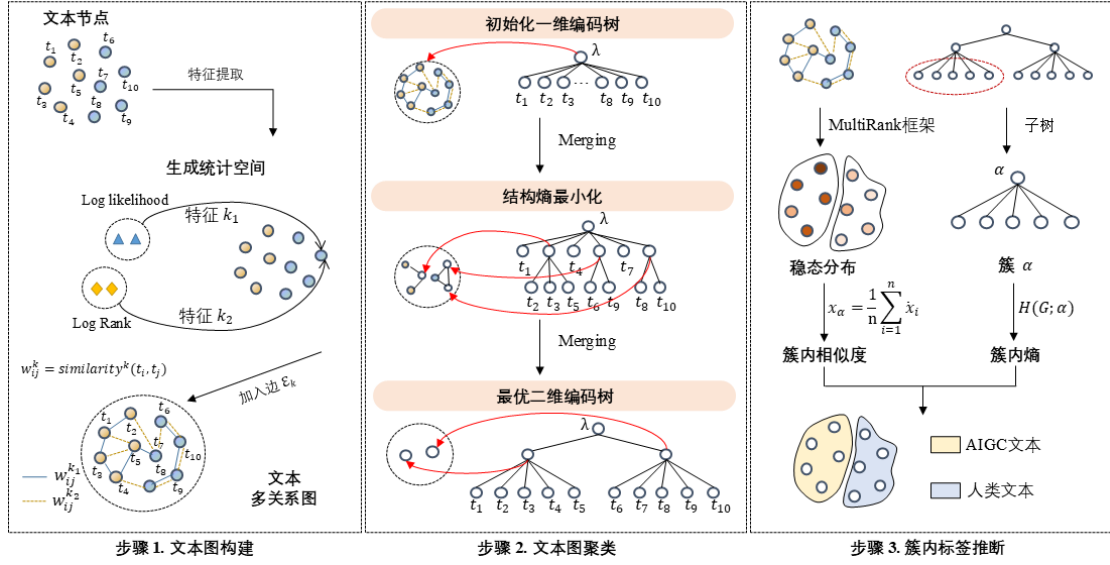


图 3-1 基于结构信息熵的 AIGC 文本检测方法示意图

Figure 3-1 AIGC text detection based on structure entropy

### 3.3.1 文本图构建

对于给定的文本  $x = x_1, x_2, \dots, x_N$ ，其中  $x_i$  是第  $i$  个 token， $N$  为序列长度，将无向图  $G = (\mathcal{V}, \mathcal{E}_k)$  定义为一个多关系图。其中， $\mathcal{V}$  表示文本节点集合， $\mathcal{E}_k$  表示边集合。节点  $v_i$  和  $v_j$  之间在关系  $k$  下的边记为  $ei_j^k$ 。

根据对数似然 (Log Likelihood, LL) 和对数排序 (Log Rank, LR) 方法在对数似然对数排序比例 (Log-Likelihood Log-Rank Ratio, LRR)<sup>[57]</sup> 中取得的良好效果，如公式 3-6 所示，在构建文本多关系异构图时优先使用这两个特征。LL 特征的获取步骤如下：(1) 将预处理后的文本输入 GPT-2 模型，计算每个单词的条件概率  $p_\theta(x_i|x_{<i})$ ，其中  $x_{<i}$  表示第  $i$  个单词之前的所有单词， $\theta$  为模型参数。(2) 计算条件概率的对数，得到对数概率  $\log p_\theta(x_i|x_{<i})$ 。(3) 计算所有单词对数概率的平均值，以获得整个文本的对数似然值：

$$\text{Log Likelihood}(x) = \frac{1}{N} \sum_{i=1}^N \log p_\theta(x_i|x_{<i}) \quad (3-4)$$

LR 的公式如下：

$$\text{Log Rank}(x) = \frac{1}{N} \sum_{i=1}^N \log r_\theta(x_i|x_{<i}) \quad (3-5)$$

其中  $r_\theta(x_i|x_{<i}) \geq 1$  表示标记  $x_i$  的排名。

LRR 结合了 LL 和 LR，公式如下：

$$\begin{aligned} \text{LRR} &= \left| \frac{\frac{1}{t} \sum_{i=1}^t \log p_{\theta}(x_i | x_{<i})}{\frac{1}{t} \sum_{i=1}^t \log r_{\theta}(x_i | x_{<i})} \right| \\ &= - \frac{\sum_{i=1}^t \log p_{\theta}(x_i | x_{<i})}{\sum_{i=1}^t \log r_{\theta}(x_i | x_{<i})}, \end{aligned} \quad (3-6)$$

此外，对于 Dolly 和 StableLM 大语言模型，基于 LL 和 LR 特征的区分能力不足。因此，该研究使用它们的变体作为特征。具体而言，使用 DetectGPT 和归一化对数排序扰动（Normalized Log-Rank Perturbation, NPR）特征来构建多关系图。

DetectGPT 特征通过对原始文本添加微小扰动来衡量模型对数概率函数的变化。可以定义扰动差异  $d(x, p_{\theta}, q)$  如下：

$$d(x, p_{\theta}, q) = \log p_{\theta}(x) - \mathbb{E}_{\tilde{x} \sim q(\cdot | x)} \log p_{\theta}(\tilde{x}) \quad (3-7)$$

其中， $q(\cdot | x)$  是扰动函数， $\tilde{x}$  是在应用扰动后生成的文本。

类似地，NPR 特征通过向原始文本添加小的扰动来衡量模型对数排序的变化：

$$\text{NPR} = \frac{\frac{1}{N} \sum_{i=1}^N \log r_{\theta}(\tilde{x}_i)}{\log r_{\theta}(x)} \quad (3-8)$$

使用上述特征，构建了一个多关系图，如图3-1的步骤 1 所示。在该图中，节点表示文本，边的权重基于文本特征的相似度。

$$e_{ij}^k = \text{similarity}^k(t_i, t_j) \quad (3-9)$$

其中  $t_i$  和  $t_j$  是两个文本节点的特征表示。相似度是通过特征之间的曼哈顿距离计算得出的。

总体而言，该方法基于对数似然<sup>[55]</sup> 和对数排名<sup>[56]</sup> 特征以及文本特征之间的相似度构建了文本多关系异质图。具体而言，对于由 Dolly 和 StableLM 大型语言模型生成的异质文本图，该方法使用 NPR 和 DetectGPT 特征。

### 3.3.2 基于结构熵最小化的图聚类

结构熵本质上通过计算随机游走的不确定性来衡量聚类树的质量<sup>[137]</sup>。基于结构信息理论，可以通过最小化结构熵将文本节点划分为不同的聚类。这种方法能够减少噪声和离群点的影响，从而得到更稳定的聚类。

在文档集合中，如果一组文本表现出一致的模式或倾向于使用特定的词汇，这表明它们可能是由大型语言模型生成的。通过模拟随机游走在过程，这些文本由于结构相似性，可能会被紧密地聚集在一起。

利用文本多关系图，本方法构建了一个初始的单维编码树，其中图中的所有节点被视为叶节点，形成一个高度为 2 的树，如图3-1所示。优化过程涉及反复执行合并操作，以最小化编码树  $\mathcal{T}$  的二维结构熵。合并操作首次由 Li 等提出。具体步骤如下：(1) 定义  $T_\alpha = x_1, x_2, \dots, x_M, y_1, y_2, \dots, y_N$ ，其高度为  $h(\alpha)$ ；(2) 对于每个  $i \in 1, 2, \dots, M$ ，定义  $T_{\alpha_i} = x_i$ ，并设置  $h(\alpha_i) \leftarrow h(\alpha) + 1$ ；(3) 对于每个  $j \in M + 1, M + 2, \dots, M + N$ ，定义  $T_{\alpha_j} = y_{j-M}$ ，并设置  $h(\alpha_j) = h(\alpha) + 1$ ；(4) 删除子树节点  $\beta$ ；

总结来说，合并操作将编码树中的两个子树  $\alpha$  和  $\beta$  合并成一个新的子树  $\gamma$ 。每次合并操作后，编码树的结构熵可能发生变化，目的是使每次操作都能够减少熵。二维编码树结构熵的变化可以通过以下公式表示：

$$\begin{aligned} \Delta_G^M SE_{\alpha, \beta} &= SE_{new} - SE_{old} \\ &= \left( H^{\mathcal{T}'}(G; \alpha) + \sum_{\gamma^- = \alpha} H^{\mathcal{T}'}(G; \gamma) \right) \\ &\quad - \left( H^{\mathcal{T}(G; \alpha) + H^{\mathcal{T}(G; \beta) + \sum_{\gamma^- = \alpha, \beta} H^{\mathcal{T}(G; \gamma)}} \right) \end{aligned} \quad (3-10)$$

其中  $H^{\mathcal{T}}(G; \gamma)$  是合并前子树  $\alpha$  和  $\beta$  的结构熵， $H^{\mathcal{T}'}(G; \gamma)$  是合并后子树  $\gamma$  的结构熵。

通过比较合并操作前后编码树的二维结构熵变化，评估该操作是否有利于编码树的优化。较小的  $\Delta_G^M SE$  值表示二维编码树的结构熵减少得更多。通过不断应用合并操作，逐步减少结构熵，最终获得最小结构熵树，即最优编码树。较低的熵值表示较高的类内相似度、较大的类间差异以及更合理的树结构。因此，二维编码树的熵值越小，聚类质量越好。

算法 1 描述了最小化二维结构熵的过程。首先，定义并初始化变量，建立编码树，每个文本作为一个聚类（第 1-4 行）；遍历图的所有边，筛选出符合合并操作条件的边（第 6-13 行）；如果边的数量超过  $k$ ，则令  $k + 1$  为当前的聚类数量，并保留那些导致结构熵减少最大的  $k$  条边（第 17-21 行）；更新聚类和边的状态，直到找不到合适的候选边（第 12-15 行）。

### 3.3.3 聚类内标签推断

通过在 MultiRank<sup>[138]</sup> 中使用共同排名机制，基于概率分布确定多关系图  $G$  中文本节点和特征的重要性。这使得排名不仅能反映单个文本节点的重要性，还能反映文本之间的相对关系。

MultiRank 算法可以得到多关系图  $G$  的两个平稳概率分布： $\mathbf{x} = [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_N]$  和  $\mathbf{y} = [\bar{y}_1, \bar{y}_2, \dots, \bar{y}_k]$ 。其中， $N$  是文本的数量， $k$  是特征或关系的数量。 $\mathbf{x}$  反映

---

**算法 1** 二维结构熵最小化算法

---

**Require:** 图  $G = (\mathcal{V}, \mathcal{E}_k)$

```

1: 初始化编码树，其中每个文本代表一个聚类
2:  $n\_cluster \leftarrow$  文本数量
3:  $mg\_cands \leftarrow$  待合并的候选边
4:  $r \leftarrow$  最大合并边的比例
5: while True do
6:    $k \leftarrow (n\_cluster - 1) \times r$ 
7:   for  $(u, v) \in edges$  do
8:      $dH(u, v) \leftarrow \Delta_G^M SE$  根据公式3-10
9:      $m \leftarrow median(dH > 0)$  ▷ 所有正值  $dH$  的中位数
10:    if  $dH(u, v) > m$  then
11:      将边  $(u, v)$  添加到  $mg\_cands$ 
12:    end if
13:  end for
14:  if  $len(mg\_cands) > k$  then
15:    从  $mg\_cands$  中选择前  $k$  个边
16:  end if
17:  if  $mg\_cands = \emptyset$  then
18:    break ▷ 结束循环
19:  end if
20:   $n\_cluster \leftarrow n\_cluster - len(mg\_cands)$ 
21:  更新边和聚类簇
22: end while

```

---

了每个文本与其他文本之间的相似性。在前面的聚类步骤中，假设图中的节点被划分为  $m$  个聚类，每个聚类的表示可以用其子节点  $v_i$  的平均值表示，记作  $x_\alpha = \frac{1}{n} \sum_{i=1}^n \bar{x}_i$ ，其中  $n$  是聚类  $\alpha$  中节点的数量。 $x_\alpha$  反映了聚类内部文本之间的相似性。

如果文本是由 LLM 生成的，它们将遵循一定的模式，例如 LLM 使用的各种采样方法。因此，聚类内的文本之间将有更多的边连接，这反映在较高的聚类内熵中。

公式 3-11 将这两个指标结合起来，用于推断聚类标签，即评估当前聚类中的文本是更可能由人类写作还是由 LLM 生成的：

$$Label(\alpha) = (1 - p) \frac{H(G; \alpha)}{\sum_{\beta \in \lambda} H(G; \beta)} + p \frac{x_\alpha}{\sum_{\beta \in \lambda} x_\beta} \quad (3-11)$$

其中,  $H(G; \alpha)$  由公式 3-2 得到。如果从上述公式得到的值超过某个阈值  $\theta$ , 则聚类  $\alpha$  中的文本被分类为大模型生成的文本; 否则, 它们被分类为人类写作的文本。

### 3.3.4 时间复杂度分析

本节介绍对本方法的时间复杂度进行了分析, 该方法包括三个阶段: 多关系文本图构建、聚类 and 分类。构建多关系文本图的复杂度取决于文本节点及其关系的数量。给定总共有  $N$  个文本样本, 图构建过程需要  $O(N^2)$  次操作。对于聚类过程, 结构熵编码树的构建复杂度为  $O(|E| \log N)$ , 其中  $|E|$  表示图中的边数。每次迭代计算合并聚类之间的结构熵差异, 复杂度为  $O(|E_i|)$ , 其中  $E_i$  是第  $i$  次迭代中聚类之间的边数。由于总的迭代次数最多为  $\log_{1-r} \frac{1-r}{N-1}$ , 其中  $r$  表示合并比率, 决定了每次迭代中选择合并的边的比例, 它是一个常数。根据经验, 通常将其值设为 0.15。因此, 整体复杂度为  $O(|E| \log N)$ 。最后, 分类过程, 即为文本聚类分配标签, 采用基于排名的方法, 可以简化为  $O(N)$ 。因此, 该方法的总时间复杂度为  $O(N^2 + |E| \log N)$ 。

## 3.4 实验结果与分析

### 3.4.1 数据集

为了评估检测 LLM 生成文本的方法的有效性, 采用了五个公开数据集: Essay、WP、Reuters、NarrativeQA 和 SQuAD1。这些数据集最初由 Verma 等提供, 随后 He 等增加了四个 LLM 生成的额外文本。为了保证实验的有效性, 本方法排除了数据集中无意义的部分, 例如 LLM 生成的空文本。表 3-1 展示了数据集的各种统计信息, 以及在本章图构建方法下, 不同特征的连接边数。此外, 它还包括了五个数据集上与文本节点相关的平均特征相似度的计算结果。因此, 每个数据集包括了人类编写的文本和由不同 LLM 生成的文本。

**WP.** 这是一个基于 subreddit r/WritingPrompts 的创意写作数据集, 用户分享创意写作提示并根据这些提示创作故事。数据集包含 1,000 个样本。对于每个样本, 使用提示生成相应的故事, 提示为: “用  $K$  个词写一个关于  $\langle prompt \rangle$  的故事。”

**Essay.** 这是一个基于高中和大学水平论文的学生论文数据集, 来自 Ivy-Panda。数据集包含 1,000 个样本。对于每个样本, 使用提示生成一篇文章, 提示为: “用  $K$  个词写一篇关于  $\langle prompt \rangle$  的文章。”

表 3-1 常用数据集数据统计

Table 3-1 Data statistics of common datasets

| 数据集       | 大模型 | 人   | 文本  | 边数量        | 特征相似度  |
|-----------|-----|-----|-----|------------|--------|
| WP        | 200 | 200 | 400 | LL: 16,884 | 0.5748 |
|           |     |     |     | LR: 102    | 0.2778 |
| Essay     | 168 | 168 | 336 | LL: 8,089  | 0.6997 |
|           |     |     |     | LR: 68     | 0.5867 |
| Reuters   | 200 | 200 | 400 | LL: 16,139 | 0.6721 |
|           |     |     |     | LR: 101    | 0.4458 |
| Narrative | 152 | 152 | 304 | LL: 4,534  | 0.4523 |
|           |     |     |     | LR: 29     | 0.1911 |
| SQuAD1    | 138 | 138 | 276 | LL: 2,787  | 0.4422 |
|           |     |     |     | LR: 11     | 0.1376 |

**Reuters.** 这是一个基于 Reuters 50-50 作者识别数据集<sup>[139]</sup>的新闻数据集，包含 1,000 篇由 50 位记者撰写的新闻文章。由于缺乏真实的文章标题或摘要，作者首先指导 ChatGPT 为每篇文章生成标题，然后根据这些生成的标题撰写文章。使用的提示为：“写一篇以以下标题为题的 K 字数新闻文章：〈*headline*〉。”

**NarrativeQA.**<sup>1</sup> 这是一个包含 1,000 个样本的电影和电子书问答数据集。LLM 基于摘要生成对给定问题的答案。

**SQuAD1.**<sup>2</sup> 这是一个包含 1,000 个样本的问答数据集，涵盖了广泛的问题范围。LLM 基于标题和上下文生成对给定问题的答案。

### 3.4.2 相关研究工作

为了验证所提出方法的有效性，将其与现有方法进行比较，包括无监督方法和经过监督微调的模型。

**Rank.** 计算每个单词在其前文上下文中的绝对排名。然后将每个单词的排名值进行平均，以计算文本的得分。<sup>[56]</sup>

**GLTR.** 使用语言模型，GLTR<sup>[56]</sup> 评估了文本中所有标记的平均对数似然、排名、对数排名和预测熵。遵循 MGTBench 中的基线<sup>[96]</sup>，并采用了 Guo 等提出的 Test-2 特征，包括排名在 10、100、1000 及其他范围内的单词比例。

**DetectGPT.** 通过对原始文本进行轻微扰动，测量模型的对数概率函数的变化<sup>[3]</sup>。由 LLMs 生成的文本往往位于模型对数概率函数的局部最优点。因此，对模型生成的文本应用小的扰动通常会降低其在模型下的对数概率。

<sup>1</sup>[https://huggingface.co/datasets/huyen89/NarrativeQA\\_LLMs](https://huggingface.co/datasets/huyen89/NarrativeQA_LLMs)

<sup>2</sup>[https://huggingface.co/datasets/huyen89/SQuAD1\\_LLMs](https://huggingface.co/datasets/huyen89/SQuAD1_LLMs)

**LRR.** Su 等提出了 LRR 方法，结合了对数似然和对数排名。

**NPR.** 类似于 DetectGPT，向原始文本引入了扰动。LLMs 在扰动后的对数排名得分变化更为显著。

**ConDA.** ConDA<sup>[5]</sup> 使用无监督领域自适应和自监督对比学习来学习更好的表示，以用于分类任务。

**ChatGPT-D.** 通过对 RoBERTa<sup>[126]</sup> 模型进行微调，增强了识别文本来源的能力，为文本检测领域提供了有效的解决方案<sup>[10]</sup>。

评估指标主要使用 F1 分数和 AUC-ROC 分数。第5.4.2节中的比较实验侧重于 F1 分数，而第5.4.2.1节则涉及 AUC-ROC 分数的比较。本节还考虑了归一化互信息（NMI）和调整兰德指数（ARI）作为评估聚类方法性能的指标。

F1 分数是分类任务中广泛使用的评估指标。它将精确率和召回率结合成一个单一的值，提供了模型性能的平衡衡量。精确率定义为真正例预测数与总预测为正的比率，反映了正类预测的准确性：

$$\text{精确率} = \frac{\text{真正例}}{\text{真正例} + \text{假正例}} \quad (3-12)$$

其中，真正例是正确预测为正例的数量，假正例是错误预测为正例的数量。召回率定义为真正例预测数与实际正例总数的比率，衡量了模型识别所有相关实例的能力：

$$\text{召回率} = \frac{\text{真正例}}{\text{真正例} + \text{假负例}} \quad (3-13)$$

F1 分数定义为精确率和召回率的调和平均数，如下所示：

$$F1 = 2 \cdot \frac{\text{精确率} \cdot \text{召回率}}{\text{精确率} + \text{召回率}} \quad (3-14)$$

这个公式确保了 F1 分数考虑了假阳性和假阴性。在需要平衡这两个指标时特别有用，尤其是在假阳性和假阴性的成本同样高，或者数据集不平衡的情况下。F1 分数的范围从 0 到 1，其中 1 代表最佳性能（完美的精确度和召回率），0 代表最差的性能。

归一化互信息用于评估两个聚类结果的相似性，其公式如下：

$$\text{NMI}(U, V) = \frac{I(U, V)}{\sqrt{H(U)H(V)}} \quad (3-15)$$

其中，其中： $I(U, V)$  是  $U$  和  $V$  之间的互信息，定义为：

$$I(U, V) = \sum_{u \in U} \sum_{v \in V} p(u, v) \log \frac{p(u, v)}{p(u)p(v)} \quad (3-16)$$

$H(U)$  和  $H(V)$  分别是聚类结果  $U$  和  $V$  的熵，定义为：

$$H(U) = - \sum_{u \in U} p(u) \log p(u) \quad (3-17)$$

$$H(V) = - \sum_{v \in V} p(v) \log p(v) \quad (3-18)$$

其中,  $p(u, v)$  是联合概率分布,  $p(u)$  和  $p(v)$  分别是单个聚类的边际概率分布。调整兰德指数 (ARI) 用于衡量两个聚类结果的相似性, 并且考虑了偶然的随机一致性。其公式为:

$$ARI = \frac{RI - \mathbb{E}[RI]}{\max(RI) - \mathbb{E}[RI]} \quad (3-19)$$

其中,  $RI$  是兰德指数, 表示两个聚类的匹配程度, 公式为:

$$RI = \frac{a + d}{\binom{n}{2}} \quad (3-20)$$

其中,  $a$  是同类点对的数量,  $d$  是异类点对的数量,  $\binom{n}{2}$  是所有点对的总数。 $\mathbb{E}[RI]$  是对偶然一致性的期望值,  $\max(RI)$  是兰德指数的最大值。

这两个指标可以用来评估聚类的质量, 特别是在多聚类比较的情境下, 能够量化不同聚类结果的相似度与差异。

### 3.4.3 实验结果

五个数据集的结果如表 3-2 至 3-3 所示。实验结果表明, 本方法在大多数基准模型中表现更好。值得注意的是, ConDA 模型并未公开, 因此在 NarrativeQA 和 SQuAD1 数据集上的 ConDA 结果未包含在内。然而, ConDA 在其他数据集上的表现较差。

首先, 在 WP 数据集上, SILTD 方法在检测 Dolly、ChatGPT 和 Claude 模型生成的文本方面表现最佳。如表 3-2 所示, 这三种模型在 WP 数据集上的 F1 分数分别为 0.822、0.932 和 0.774, 分别比其他基准方法提高了至少 0.005、0.091 和 0.001。当检测 ChatGLM、GPT4All 和 StableLM 生成的文本时, 本方法不如 GLTR 有效, 后者考虑了在 10、100、1,000 等范围内排名的单词的比例。因为这些模型在 WP 数据集中生成的文本在单词选择上更加多样化, 例如, 倾向于选择那些稀有但在语境中合适的单词。本方法取整个文本的平均排名, 对于这种变化不够敏感。但总体而言, SILTD 能更有效地检测 LLM 生成的文本, 并在不同 LLM 上展示出显著的平均性能提升。因为 SILTD 利用结构信息理论有效地利用了多关系图的结构信息, 这种方法导致了层次化的聚类划分, 从而更有效地对 LLM 生成的文本进行分类和识别。

其次, 其他方法在不同模型上的表现差异显著, 而本方法表现出更强的普适性。根据表 3-2 至表 3-3, 检测六种大型语言模型生成文本的平均 F1 分数分别提高了至少 0.029、0.013、0.05、0.008 和 0.047。因为许多方法依赖于单一特征, 如 NPR 和 DetectGPT, 它们不足以处理不同类型的 LLM。此外, 基于预训练模型

表 3-2 不同检测方法在创作数据集上的 F1 分数

Table 3-2 The F1-score of different detection methods on the writing datasets

| 方法          | ChatGLM      | Dolly        | ChatGPT-turbo | GPT4All      | StableLM     | Claude       | 平均 F1 值      |
|-------------|--------------|--------------|---------------|--------------|--------------|--------------|--------------|
| WP 数据集      |              |              |               |              |              |              |              |
| Rank        | 0.840        | 0.760        | 0.797         | 0.891        | 0.781        | 0.709        | 0.796        |
| GLTR        | <b>0.983</b> | 0.766        | <u>0.800</u>  | <b>0.935</b> | <u>0.861</u> | <u>0.733</u> | <u>0.846</u> |
| Entropy     | 0.800        | 0.662        | 0.770         | 0.766        | 0.644        | 0.731        | 0.729        |
| LRR         | <u>0.980</u> | 0.774        | 0.728         | 0.911        | <b>0.875</b> | 0.649        | 0.820        |
| NPR         | 0.970        | <u>0.801</u> | 0.352         | 0.905        | 0.764        | 0.521        | 0.719        |
| DetectGPT   | 0.812        | 0.719        | 0.608         | 0.808        | 0.695        | 0.517        | 0.693        |
| ConDA       | 0.585        | 0.039        | 0.075         | 0.674        | 0.667        | 0.000        | 0.340        |
| ChatGPT-D   | 0.880        | 0.528        | 0.352         | 0.795        | 0.616        | 0.044        | 0.536        |
| SILTD       | 0.971        | <b>0.822</b> | <b>0.932</b>  | <u>0.917</u> | 0.799        | <b>0.774</b> | <b>0.875</b> |
| Essay 数据集   |              |              |               |              |              |              |              |
| Rank        | 0.740        | 0.737        | 0.915         | 0.843        | 0.667        | 0.772        | 0.779        |
| GLTR        | <b>0.988</b> | 0.848        | <u>0.954</u>  | <u>0.925</u> | <u>0.756</u> | 0.806        | <u>0.880</u> |
| Entropy     | 0.806        | 0.683        | 0.874         | 0.699        | 0.566        | 0.771        | 0.733        |
| LRR         | 0.982        | 0.810        | 0.925         | 0.904        | 0.748        | 0.746        | 0.853        |
| NPR         | 0.956        | 0.865        | 0.218         | <b>0.927</b> | 0.740        | 0.238        | 0.657        |
| DetectGPT   | 0.891        | 0.844        | 0.227         | 0.908        | 0.704        | 0.236        | 0.635        |
| ConDA       | 0.668        | 0.069        | 0.000         | 0.260        | 0.663        | 0.664        | 0.387        |
| ChatGPT-D   | 0.923        | 0.630        | 0.742         | 0.815        | 0.491        | 0.057        | 0.610        |
| SILTD       | <u>0.985</u> | <b>0.876</b> | <b>0.956</b>  | <b>0.927</b> | <b>0.761</b> | <b>0.851</b> | <b>0.893</b> |
| Reuters 数据集 |              |              |               |              |              |              |              |
| Rank        | 0.650        | 0.413        | 0.847         | 0.665        | 0.635        | 0.648        | 0.643        |
| GLTR        | <u>0.987</u> | 0.556        | <u>0.946</u>  | 0.742        | 0.750        | 0.772        | 0.792        |
| Entropy     | 0.477        | 0.553        | 0.703         | 0.668        | 0.620        | 0.694        | 0.619        |
| LRR         | <b>0.992</b> | 0.590        | <b>0.948</b>  | 0.796        | <u>0.766</u> | 0.715        | <u>0.801</u> |
| NPR         | 0.950        | <u>0.790</u> | 0.284         | <u>0.843</u> | 0.751        | 0.560        | 0.696        |
| DetectGPT   | 0.866        | 0.782        | 0.270         | 0.821        | 0.756        | 0.558        | 0.676        |
| ConDA       | 0.664        | 0.137        | 0.000         | 0.667        | 0.000        | 0.667        | 0.356        |
| ChatGPT-D   | 0.968        | 0.650        | 0.931         | <b>0.898</b> | 0.617        | 0.019        | 0.681        |
| SILTD       | 0.977        | <b>0.872</b> | 0.932         | 0.709        | <b>0.866</b> | <b>0.801</b> | <b>0.851</b> |

的方法，如 ConDA 和 ChatGPT-D，在低资源环境中仅在特定 LLM 上表现稳定。然而，当检测具有不同能力的 LLM 时，这些方法可能会变得不那么有效。

总体而言，SILTD 方法在多个数据集上的表现更好，达到了更高的 F1 分数。这表明本方法具有更好的普适性和更强的检测能力，使其能够适应不同文本生成任务中检测各种类型的大模型。

如图 3-2 所示，WP 和 Essay 数据集的 NMI 和 ARI 指标相对接近。本方法在检测 GPT4All、ChatGPT 和 ChatGLM 时表现良好，ARI 分数均高于 0.7，最高可达 0.94。然而，在检测 Claude、StableLM 和 Dolly 时，表现不如预期。这可能是因为 WP 数据集侧重于创意写作，而 Essay 数据集则专注于学术论文写作。Claude 和 StableLM 在这些领域可能具有更强的创作能力，生成的内容在风格上

表 3-3 不同检测方法在问答数据集上的 F1 分数

Table 3-3 The F1-score of different detection methods on the question-answering datasets

| 方法              | ChatGLM      | Dolly        | ChatGPT-turbo | GPT4         | StableLM     | 平均 F1 值      |
|-----------------|--------------|--------------|---------------|--------------|--------------|--------------|
| NarrativeQA 数据集 |              |              |               |              |              |              |
| Rank            | 0.861        | 0.766        | 0.878         | <b>0.846</b> | 0.737        | 0.818        |
| GLTR            | <u>0.927</u> | <u>0.869</u> | <u>0.907</u>  | <u>0.811</u> | <u>0.846</u> | <u>0.872</u> |
| Entropy         | 0.734        | 0.673        | 0.730         | 0.636        | 0.653        | 0.685        |
| LRR             | 0.878        | 0.829        | 0.890         | 0.715        | 0.705        | 0.803        |
| NPR             | 0.495        | 0.553        | 0.489         | 0.728        | 0.708        | 0.595        |
| DetectGPT       | 0.495        | 0.597        | 0.568         | 0.784        | 0.790        | 0.647        |
| ConDA           | -            | -            | -             | -            | -            | -            |
| ChatGPT-D       | 0.761        | 0.661        | 0.821         | 0.733        | 0.779        | 0.751        |
| SILTD           | <b>0.932</b> | <b>0.884</b> | <b>0.930</b>  | 0.794        | <b>0.862</b> | <b>0.880</b> |
| SQuAD1 数据集      |              |              |               |              |              |              |
| Rank            | 0.841        | 0.773        | 0.827         | 0.771        | 0.713        | 0.785        |
| GLTR            | 0.873        | <u>0.820</u> | 0.862         | <u>0.825</u> | 0.733        | <u>0.823</u> |
| Entropy         | 0.628        | 0.555        | 0.640         | 0.593        | 0.563        | 0.596        |
| LRR             | <u>0.876</u> | 0.815        | <u>0.876</u>  | 0.775        | 0.701        | 0.809        |
| NPR             | 0.422        | 0.542        | 0.454         | 0.632        | 0.756        | 0.561        |
| DetectGPT       | 0.458        | 0.565        | 0.497         | 0.653        | <u>0.825</u> | 0.600        |
| ConDA           | -            | -            | -             | -            | -            | -            |
| ChatGPT-D       | 0.626        | 0.533        | 0.693         | 0.646        | 0.635        | 0.627        |
| SILTD           | <b>0.882</b> | <b>0.845</b> | <b>0.897</b>  | <b>0.856</b> | <b>0.869</b> | <b>0.870</b> |

更接近人类作品，导致聚类效果不如预期明显。值得注意的是，在新闻数据集中，GPT4All 的 NMI 和 ARI 指标均低于 0.2，表明其在新闻写作中的特征与人类高度相似。相反，NarrativeQA 和 SQuAD1 是问答数据集，倾向于将回答和词汇选择限制在相对狭窄的范围内。这导致人类回答和大语言模型生成的回答相比其他数据集显示出更大的相似性。这一特征在图 3-2 中有所体现，可以看到这两个数据集的 NMI 和 ARI 指标普遍呈下降趋势。这些发现不仅揭示了本方法的优势，还突出了不同类型数据集和语言模型在内容生成特征上的差异。

3.4.3.1 消融实验

本节进行了消融实验以证明本方法的有效性。对于 ChatGLM、ChatGPT、Claude 和 GPT4All 等大语言模型，比较了以下方法的平均 F1 分数和平均 AU-

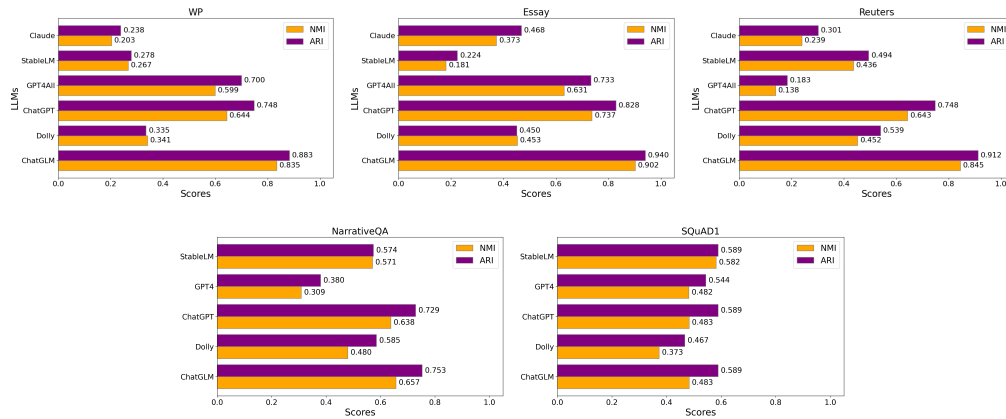


图 3-2 不同数据集下 NMI 和 ARI 指标

Figure 3-2 NMI and ARI metrics in various datasets

CROC 分数: Log Likelihood、Log Rank、LRR、SILTD-LL (仅使用 Log Likelihood 构建文本图)、SILTD-LR (仅使用 Log Rank 构建文本图) 和 SILTD。对于 Dolly 和 StableLM, 比较了以下方法: SILTD-DetectGPT (仅使用 DetectGPT 特征构建多关系图)、SILTD-NPR (仅使用 NPR 特征构建多关系图) 和 SILTD。

表 3-4 显示, 对于大语言模型 ChatGLM、ChatGPT、Claude 和 GPT4ALL, SILTD-LL 方法在所有数据集上表现良好。它比 LL 方法有显著改进, 甚至超越了 LRR 方法。然而, 它仍然稍逊于完整的 SILTD 方法。这表明, 结合两种特征比仅使用单一特征更有效地检测生成文本。

表 3-4 使用 LL 和 LR 特征的消融实验结果

Table 3-4 Results of ablation experiments using LL and LR features

| Method          | WP           |              | Essay        |              | Reuters      |              | NarrativeQA  |              | SQuAD1       |              |
|-----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                 | F1           | AUROC        | F1           | AUROC        | F1           | AUROC        | F1           | AUROC        | F1           | AUROC        |
| <b>LL</b>       | 0.86         | 0.868        | 0.925        | 0.925        | 0.813        | 0.798        | <b>0.889</b> | 0.890        | 0.866        | 0.874        |
| <b>LR</b>       | 0.838        | 0.838        | 0.908        | 0.905        | 0.774        | 0.741        | 0.885        | 0.891        | 0.870        | 0.876        |
| <b>LRR</b>      | 0.817        | 0.819        | 0.895        | 0.894        | 0.851        | 0.834        | 0.831        | 0.844        | 0.836        | <b>0.914</b> |
| <b>SILTD-LL</b> | 0.896        | 0.893        | 0.929        | 0.928        | 0.851        | 0.854        | 0.884        | 0.887        | 0.865        | 0.872        |
| <b>SILTD-LR</b> | 0.498        | 0.717        | 0.603        | 0.73         | 0.473        | 0.671        | 0.341        | 0.572        | 0.390        | 0.550        |
| <b>SILTD</b>    | <b>0.899</b> | <b>0.907</b> | <b>0.930</b> | <b>0.937</b> | <b>0.855</b> | <b>0.870</b> | 0.885        | <b>0.891</b> | <b>0.870</b> | 0.880        |
| Improvement(%)  | 0.31-80.70↑  | 1.62-26.54↑  | 0.08-54.29↑  | 1-28.36↑     | 0.53-80.86↑  | 1.84-29.75↑  | 0.56↓        | 0-55.70↑     | 0-128.46↑    | 3.7↓         |

SILTD 方法在五个数据集上的 F1 和 AUCROC 分数上显著优于 LRR 方法。尽管 LRR 方法结合了 LL 和 LR 特征, 但它仅在 Reuters 数据集上超越了 LL 和 LR 方法, 在其他数据集上表现较差。这是因为异质图能够有效融合文本中不同关系的信息, 从而生成更丰富、更具判别性的节点表示。这为高级聚类分析奠定了坚实的基础, 进而导致了更好的结果。

表 3-5 展示了使用 NPR 和 DetectGPT 作为特征的不同检测方法在 LLMs 上的平均 F1 分数和 AUCROC 值。SILTD-LLR 方法使用 LL 和 LR 特征构建多关系图, 而 SILTD 方法则使用 NPR 和 DetectGPT 特征。结果表明, SILTD 方法优

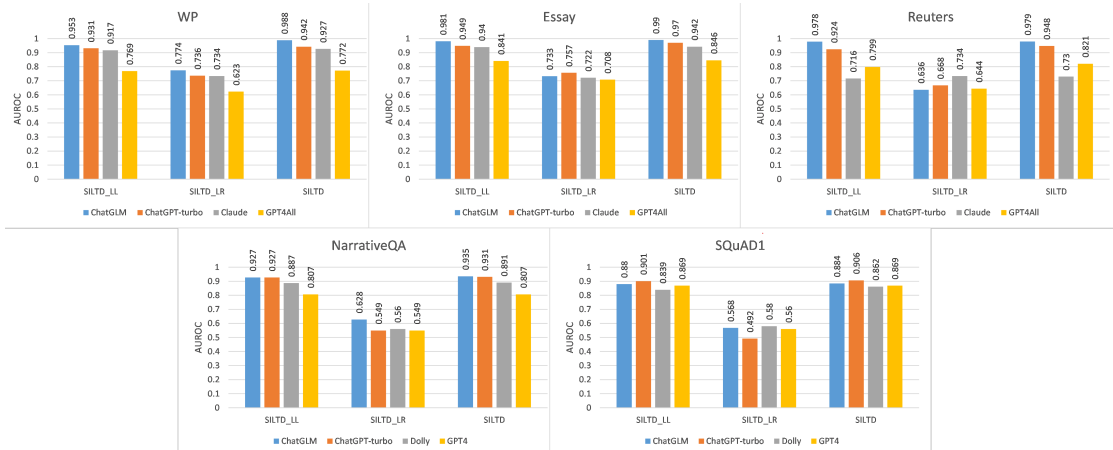


图 3-3 SILTD 方法在检测不同模型时的 AUROC 指标

Figure 3-3 AUROC results of the SILTD method across various models

于 SILTD-LLR 方法，特别是在 Reuters 数据集上，F1 和 AUCROC 分别提高了 39.93% 和 43.09%。NPR 和 DetectGPT 特征分别监控文本中 LL 和 LR 值的波动，在对文本施加扰动后，丰富了文本的表示。结果表明，本方法在一定程度上依赖于文本的表示，单独使用 LL 和 LR 特征可能不足以有效检测所有类型的 LLM。因此，未来的工作中有必要探索更多普适性强的特征来表示文本。

### 3.4.4 分析实验

表 3-5 使用 NPR 和 DG 特征的消融实验结果

Table 3-5 Results of ablation experiments using NPR and DG features

| Method         | WP           |              | Essay        |              | Reuters      |              | NarrativeQA  |              | SQuAD1        |              |
|----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|---------------|--------------|
|                | F1           | AUROC        | F1           | AUROC        | F1           | AUROC        | F1           | AUROC        | F1            | AUROC        |
| SILTD-LL       | 0.805        | 0.808        | 0.772        | <b>0.794</b> | 0.605        | <b>0.632</b> | 0.721        | 0.752        | 0.606         | <b>0.692</b> |
| SILTD-LR       | 0.569        | 0.677        | 0.480        | 0.621        | 0.287        | 0.479        | 0.389        | 0.549        | 0.332         | 0.547        |
| SILTD-LLR      | <b>0.811</b> | <b>0.839</b> | <b>0.782</b> | 0.788        | <b>0.621</b> | 0.601        | <b>0.764</b> | <b>0.795</b> | <b>0.691</b>  | 0.632        |
| Improvement(%) | 0.81-42.53↑  | 3.84-24.02↑  | 1.3-63.09↑   | 0.76↓        | 2.65-116.75↑ | 4.83↓        | 5.96-96.40↑  | 5.71-44.81↑  | 14.03-108.13↑ | 8.67↓        |
| SILTD-NPR      | 0.802        | 0.742        | <b>0.836</b> | 0.809        | 0.861        | 0.844        | 0.818        | 0.846        | 0.864         | 0.880        |
| SILTD-DG       | 0.805        | 0.759        | 0.806        | 0.816        | 0.866        | 0.859        | 0.858        | 0.875        | 0.864         | 0.880        |
| SILTD          | <b>0.811</b> | <b>0.765</b> | 0.819        | <b>0.839</b> | <b>0.869</b> | <b>0.860</b> | <b>0.862</b> | <b>0.879</b> | <b>0.869</b>  | <b>0.884</b> |
| Improvement(%) | 0.75-1.19↑   | 0.86-3.10↑   | 1.97↓        | 2.88-3.71↑   | 0.35-0.93↑   | 0.12-1.96↑   | 0.47-2.13↑   | 0.46-7.46↑   | 0.58↑         | 0.45↑        |

在图 3-3 中，展示了 SILTD 方法在不同模型上的 AUROC 结果，并与其变体（SILTD-LL 和 SILTD-LR）进行了比较。从图中可以明显看出，SILTD-LL 的结果显著优于 SILTD-LR 的结果。SILTD 方法相较于 SILTD-LL 方法最高提升了 3.67%。

然而，在 Reuters 数据集上，SILTD-LR 方法在 Claude 模型上的 AUROC 得分最高。其潜在原因是，在这种情况下，LL 已经足以捕捉文本之间的差异，而 LR 引入了冗余或噪声，导致整体模型性能下降。尽管如此，SILTD 方法相比于其他两种方法仍然显示出显著的改进，最高改进达到 4%。

总体而言，仅使用 LL 或 LR 等单一特征来区分 LLM 生成文本和人工写作

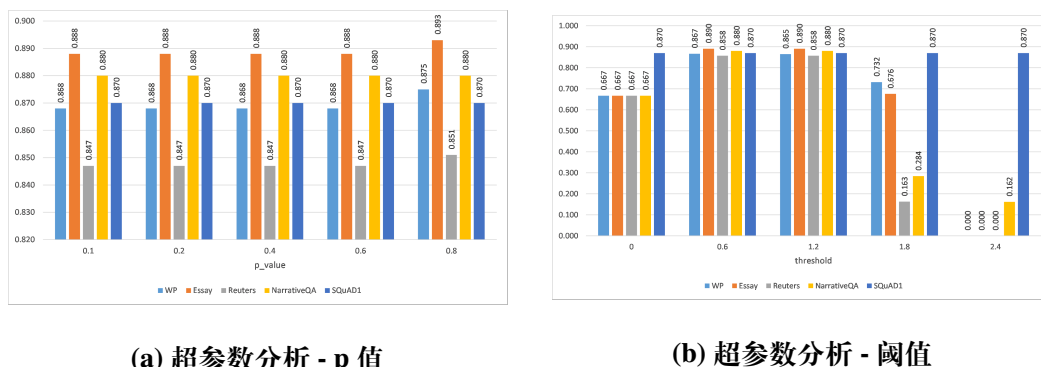


图 3-4 超参数分析

Figure 3-4 Hyperparameter sensitivity

文本通常是具有挑战性的。两种特征的有效结合弥补了单一特征使用中的不足。AUROC 结果也表明，SILTD 方法可以显著提升 LLM 生成文本检测的性能。

#### 3.4.4.1 超参数分析

对于 SILTD 中标签推断的超参数  $p$  和阈值  $\theta$ ，首先通过应用不同的  $p$  值来进行超参数实验，调整图 3-1 中第 3 步的簇内文本相似度和熵。实验结果如图 3-4 所示，表明  $p$  值对 WP、Essay 和 Reuters 数据集中的 LLM 生成文本检测的 F1 分数有影响，而对 NarrativeQA 和 SQuAD1 的检测没有影响。

接下来，为了确定分类阈值，进行了不同阈值的实验，并分析它们对检测性能的影响。实验结果显示，SILTD 方法的性能表现存在显著的阈值依赖性，如图 3-4 的第二张图所示。在阈值为 0 时，所有数据集的 F1 分数保持在较为平稳的水平 (0.667-0.870)。当阈值增加到 0.6 时，四个数据集达到了最佳性能 (0.858-0.890)，而 SQuAD1 保持稳定 (0.870)。进一步增加阈值超过 1.2 后，所有数据集的性能明显下降，尤其在阈值为 1.8 时，Reuters 和 NarrativeQA 的 F1 分数分别暴跌至 0.163 和 0.284。这表明，较低的阈值 (0.6) 在大多数数据集中能最佳平衡分类的灵敏度和特异性，提供了更优的性能。因此，0.6 是区分 LLM 生成文本和人工编写内容的最佳分类阈值。

#### 3.4.4.2 案例分析

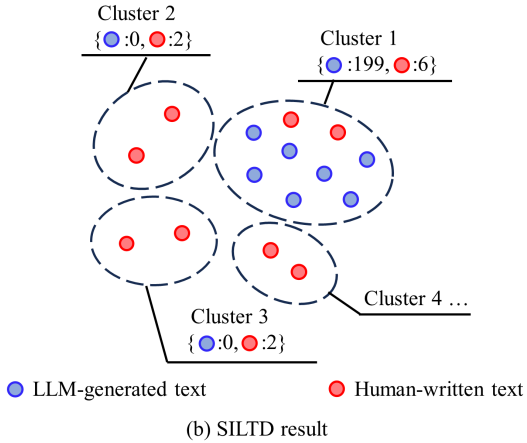
图 3-5 展示了在 Essay 数据集上检测 ChatGLM 的结果。人工编写的文本表现出多样性，特别是在不同的提示下，它们被分配到不同的簇中。相比之下，LLM 生成的文本，如图 3-5(b) 所示，能够准确地聚类在一起。然而，仔细观察后，可以发现一些错误分类的情况，这突显了 SILTD 在区分人工文本和 LLM 生成文本时面临的挑战：人工文本被误分类为 LLM 生成文本。详细分析表明，这些文本通常表现出一致的文本连贯性或词汇风格，使其在统计上与 LLM 生成的内容相似。由于本方法是基于所有字符的平均特征值来推导文本表示，因此它主要捕捉了人工文本和 LLM 生成文本之间的全局统计相似性。然而，本方法主要在文档

Prompt: Write an essay in 300 words to the prompt "Discuss the importance of effective change leadership in adopting..."

Human: Charles Schwab is a for-profit Corporation whose main business is savings and loans. The company provides...In the corporation, shareholders are free to transfer ownership, which means that in critical situations such as bankruptcy...

ChatGLM: Effective change leadership is crucial in adopting social media ... Effective change leadership involves a range of skills, including communication, motivation, and organization. It requires the ability to clearly...

(a) sample texts



(b) SILTD result

图 3-5 在论文数据集中的检测示例

Figure 3-5 An example of detection in a Essay dataset

级别操作，可能忽略了在句子或标记级别上更微妙和复杂的局部相似性。这一局限性减少了区分在局部模式上相似而在整体上有所不同的文本的能力，从而导致分类错误。为了解决这个问题，未来的工作应重点改进聚类过程，通过引入局部特征表示来提高准确性。

### 3.5 本章小结

本章提出了一种创新的无监督方法，用于检测由大型语言模型（LLMs）生成的文本，利用结构信息理论的原理。首先通过构建多关系文本图，捕捉文本之间复杂的关联关系，从而提供了一个新的特征集成视角。接着，提出了一种二维结构熵最小化聚类算法，并结合了簇内标签推断方法。这一策略通过分层划分多关系图，从而实现精确且高效的簇分类。在五个公共数据集上的实验结果表明，在考虑 F1 和 AUROC 分数时，SILTD 方法优于包括无监督和监督方法在内的基线方法。本章展示了使用聚类方法进行 LLM 生成文本检测的潜力，并可能为 LLM 生成文本检测的研究开辟新的方向。未来，可以研究更多具有区分性的文本特征以及提升编码树优化算法的方法。

## 第4章 基于数学运算一致性验证的 AIGC 文本检测方法

### 4.1 引言

AIGC 生成文本的真实性和逻辑性问题比较普遍，特别是在涉及数学运算和逻辑推理的文本中，现有检测方法往往难以精准识别其中的数学推理错误或逻辑矛盾。传统 AIGC 文本检测方法主要依赖统计特征匹配或语言模型的概率分布，在普通自然语言文本检测中表现良好，但面对具有数学运算特征的文本时，常常难以有效判断其逻辑合理性。例如，在涉及数学运算的文本中，即便语言表达符合自然语言模式，仍可能存在数学运算错误或逻辑悖论，给现有检测方法带来挑战。

已有研究表明，大模型在处理简单任务时仍然存在较高的错误率<sup>[9]</sup>。例如，在经过人工验证的高质量基准测试上，研究人员统计了各大模型的错误数量，并发现即使在清理了测试数据中的错误和歧义后，几乎所有模型仍然会犯错（如图4-1）。其中，在阅读理解、数学运算以及常识推理等任务上，大模型的错误率远高于人类水平。基于这一观察，本文方法通过比较大模型生成文本与人类文本之间的错误率，从而实现对 AIGC 文本的检测。

因此，本节提出了一种基于数学运算一致性验证的 AIGC 文本检测方法 **Mathematical Calculation Consistency Verification for LLM-generated Text Detection**，记作 **MCCV-LTD**，该方法结合数学运算验证与对比学习技术，以提升对 AIGC 文本的数学推理能力和逻辑一致性检测能力。该方法的核心思想是将自然语言文本转换为数学表达式，并利用数学运算规则验证其逻辑合理性，从而判断文本的数学运算是否正确。

具体而言，本方法包括以下三个创新点：

(1) 与现有的针对形式语言文本的 AIGC 文本检测方法不同，该方法通过构建一种端到端的数学公式解析模型，将文本转换为数学表达式，并利用数学运算规则进行推理验证，从而有效识别文本中的数学运算错误。

(2) 本方法不仅限于检测简单的等式关系，还能够处理更复杂的数学运算，涵盖嵌套的加减乘除以及数之间的大小关系。这种方法扩展了 AIGC 文本检测的适用范围，使其能够更精准地识别文本中的数学推理缺陷。为了提升模型在数学运算文本上的泛化能力，引入了对比学习技术，通过构建语义相似但对应的数学表达式不同的对比样本，增强模型的表达能力。具体而言，本方法采用数据增强的方法生成正负样本对，引导模型学习数学表达式中的细微差异，从而确保能够准确区分运算正确与错误的文本。

(3) 本文创建了一个新的数学表达式数据集，与现有公开数据集有所区别，

|                           | SingleOp |     | SingleEq |     | MultiArith |     | SVAMP |     | GSM8K |     | MMLU HS Math |     | Logic Ded. 3-Obj |     | Object Counting |        | Navigate |  | TabFact |     | HotpotQA |  | SQuAD2.0 |    | DROP |       | Winograd WSC |  | VQA v2.0 |  |
|---------------------------|----------|-----|----------|-----|------------|-----|-------|-----|-------|-----|--------------|-----|------------------|-----|-----------------|--------|----------|--|---------|-----|----------|--|----------|----|------|-------|--------------|--|----------|--|
| Type                      | Math     |     |          |     |            |     |       |     |       |     |              |     |                  |     |                 |        | Logic    |  |         | Tab | RC       |  |          | CR | Vis  | Score |              |  |          |  |
| # Platinum Questions      | 150      | 100 | 171      | 273 | 274        | 268 | 200   | 191 | 200   | 173 | 184          | 164 | 209              | 195 | 248             |        |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| o1-2024-12-17 (high)      | 0        | 0   | 0        | 0   | 2          | 1   | 0     | 0   | 0     | 0   | 0            | 5   | 0                | 5   | 10              | 0.75%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| Claude 3.5 Sonnet (Oct)   | 0        | 0   | 0        | 1   | 3          | 21  | 0     | 0   | 1     | 1   | 0            | 3   | 6                | 3   | 17              | 1.08%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| o1-2024-12-17 (med)       | 0        | 0   | 0        | 1   | 2          | 2   | 0     | 0   | 0     | 0   | 0            | 8   | 2                | 8   | 6               | 1.27%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| DeepSeek-R1               | 0        | 0   | 1        | 1   | 1          | 2   | 0     | 0   | 1     | 3   | 1            | 6   | 6                | 7   |                 | 1.64%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| Claude 3.5 Sonnet (June)  | 0        | 0   | 0        | 2   | 5          | 29  | 0     | 0   | 0     | 3   | 0            | 3   | 3                | 8   | 21              | 1.83%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| Llama 3.1 405B Inst       | 0        | 0   | 0        | 3   | 2          | 28  | 0     | 1   | 5     | 1   | 2            | 4   | 4                | 8   |                 | 1.91%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| GPT-4o (Aug)              | 0        | 0   | 0        | 7   | 4          | 22  | 0     | 0   | 4     | 2   | 3            | 9   | 4                | 10  | 8               | 2.40%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| GPT-4o (Nov)              | 0        | 0   | 0        | 6   | 7          | 20  | 0     | 4   | 4     | 1   | 2            | 7   | 3                | 12  | 6               | 2.48%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| o1-preview                | 0        | 0   | 0        | 1   | 2          | 3   | 0     | 15  | 3     | 4   | 3            | 11  | 5                | 6   |                 | 2.49%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| DeepSeek-V3               | 1        | 0   | 0        | 3   | 3          | 12  | 0     | 10  | 0     | 1   | 1            | 9   | 3                | 16  |                 | 2.85%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| o1-mini                   | 1        | 0   | 1        | 1   | 2          | 5   | 1     | 2   | 1     | 2   | 3            | 9   | 3                | 19  |                 | 3.03%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| Gemini Thinking (12/19)   | 0        | 0   | 1        | 0   | 4          | 3   | 4     | 5   | 3     | 2   | 3            | 6   | 6                | 17  | 10              | 3.03%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| Qwen 2.5 72B Inst         | 0        | 0   | 0        | 4   | 7          | 16  | 3     | 10  | 3     | 3   | 4            | 8   | 5                | 12  |                 | 3.09%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| o3-mini-2025-01-31 (high) | 0        | 0   | 2        | 1   | 1          | 2   | 1     | 1   | 0     | 0   | 2            | 35  | 3                | 14  |                 | 3.18%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| Grok 2                    | 1        | 0   | 0        | 4   | 3          | 15  | 1     | 2   | 5     | 5   | 1            | 8   | 5                | 15  | 27              | 3.20%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| Mistral Large             | 0        | 0   | 0        | 7   | 3          | 34  | 4     | 1   | 11    | 3   | 4            | 11  | 11               | 11  |                 | 3.5%   |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| Gemini 2.0 Flash          | 0        | 0   | 1        | 4   | 8          | 6   | 0     | 0   | 7     | 2   | 3            | 7   | 6                | 22  | 7               | 3.55%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| Llama 3.3 70B Inst        | 0        | 0   | 0        | 7   | 7          | 44  | 0     | 1   | 10    | 4   | 2            | 8   | 7                | 14  |                 | 3.61%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| Llama 3.1 70B Inst        | 2        | 0   | 0        | 7   | 7          | 39  | 4     | 2   | 9     | 4   | 1            | 7   | 9                | 17  |                 | 4.02%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| Gemini 1.5 Pro            | 0        | 0   | 1        | 6   | 6          | 13  | 2     | 9   | 9     | 5   | 1            | 14  | 7                | 17  | 9               | 4.16%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| GPT-4o mini               | 0        | 1   | 1        | 6   | 6          | 24  | 2     | 14  | 7     | 14  | 4            | 16  | 13               | 27  | 29              | 6.88%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| Claude 3.5 Haiku          | 0        | 0   | 1        | 8   | 10         | 43  | 4     | 5   | 11    | 13  | 3            | 10  | 10               | 31  |                 | 6.88%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| Gemini 1.5 Flash          | 0        | 1   | 0        | 13  | 11         | 23  | 5     | 15  | 18    | 17  | 3            | 13  | 12               | 21  | 11              | 6.96%  |          |  |         |     |          |  |          |    |      |       |              |  |          |  |
| Mistral Small             | 1        | 0   | 0        | 11  | 19         | 62  | 27    | 18  | 30    | 12  | 10           | 21  | 25               | 39  |                 | 11.09% |          |  |         |     |          |  |          |    |      |       |              |  |          |  |

图 4-1 大模型在简单数学问题上的出错率<sup>[9]</sup>

Figure 4-1 Errorness of LLMs in simple math problems<sup>[9]</sup>

数据集中引入了包含“>”和“<”等不等关系的数学表达式。实验结果表明，在本文构建数据集上进行评估后，MCCV-LTD 方法有效缓解了现有检测方法在数学运算文本上的不足，并显著提升了对包含数学推理内容文本的检测准确率。

4.2 相关工作

本节介绍了自然语言文本转换为数学表达式过程中的数学单词问题以及对比学习的相关研究工作。

数学单词问题

数学单词问题（Math Word Problem，MWP）求解的研究最早可追溯至 20 世纪 60 年代。最早的方法主要是基于规则的，依靠人工手工编写的规则来分析文本中的数学关系。这类方法虽然可以捕捉某些特定语言模式，但在处理复杂问题时具有较大局限性。统计学习方法利用机器学习分类器来提取数学问题中的实体、数值和运算符，并结合简单的逻辑推理得出答案。这类方法提升了模型的泛化能力，但受限于特定的模板，难以应对多样化的数学表达方式。基于树的求解方法（Koncel-Kedziorski et al., 2015; Roy and Roth, 2016, 2017）充分利用数学表达式的二叉树结构进行求解。这些方法通过树形结构来构建数学公式，使得计算

过程更加直观。

序列到序列 (SEQ2SEQ) 方法是最早引入深度学习进行 MWP 求解的方法。随后, 研究者尝试使用强化学习 (Wang et al., 2018b; Huang et al., 2018)、密集问题表示 (Mishra et al., 2018)、模板匹配 (Wang et al., 2019) 以及组注意力机制 (Li et al., 2019) 来提升 SEQ2SEQ 方法的表现。序列到树 (SEQ2TREE) 方法 (Xie and Sun, 2019) 通过树状解码器生成数学表达式, 以模仿人类的目标驱动问题求解方式。结合 BERT、RoBERTa 等预训练模型的 SEQ2TREE 方法 (Xiong et al., 2022) 进一步提升了求解能力。图到树 (GRAPH2TREE) 方法 (Zhang et al., 2020b; Li et al., 2020) 融合图神经网络和树状解码器, 充分利用数学问题中的结构信息。复杂关系抽取方法 (Jie et al., 2022) 将 MWP 求解建模为复杂关系抽取任务, 结合可解释的演绎推理技术来构建目标方程。

基于大模型的提示方法利用大模型的小样本学习能力进行数学推理。例如, Cobbe et al. (2021) 使用 GPT-3 进行数学推理, 但实验表明, 虽然大模型具备一定的自然语言理解和推理能力, 但在复杂推理任务上存在明显局限性。有研究指出, 相较于直接使用 LLM 进行提示学习, 微调小型模型的效果更好 (Ho et al., 2022)。

## 对比学习

对比学习 (Contrastive Learning) 是一种常用于表征学习的技术, 旨在通过构造样本之间的相似性约束, 使模型学习到更加判别性的特征。根据是否利用标注信息, 对比学习可以分为无监督对比学习 (Unsupervised Contrastive Learning, UCL) 和监督对比学习 (Supervised Contrastive Learning, SCL)。

无监督对比学习是一种不依赖人工标注的表征学习方法, 通常采用自监督学习 (Self-Supervised Learning, SSL) 的方式, 通过数据增强或其他手段构造正负样本对, 进而优化模型的表征能力。在这种方法中, 模型主要通过实例区分任务进行训练, 即: 相似样本 (正样本对) 应该在特征空间中靠近, 不相似样本 (负样本对) 应该在特征空间中远离。该方法广泛应用于计算机视觉和自然语言处理等领域, 尤其在无监督学习任务中表现优异。在自然语言处理 (NLP) 中, 对比学习被广泛应用于句子表征任务, 例如 Dropout 机制被用作数据增强手段, 以构造正负样本对。

监督对比学习 (Supervised Contrastive Learning, SCL) 最早在计算机视觉领域被提出, 它通过对比损失优化, 使同类别样本的表示更加紧密, 而不同类别样本的表示更加分散。因此, 模型能够利用类别标签信息进行优化, 从而提升判别能力。在 NLP 领域, 已有研究引入自然语言推理 (NLI) 数据集作为监督信号, 以增强对比学习的效果<sup>[140]</sup>。对于数学文本理解任务, 其数学表达式的相似性可以被视为一种自然语言推理关系。因此, 本节所采用的监督对比学习方法可以看

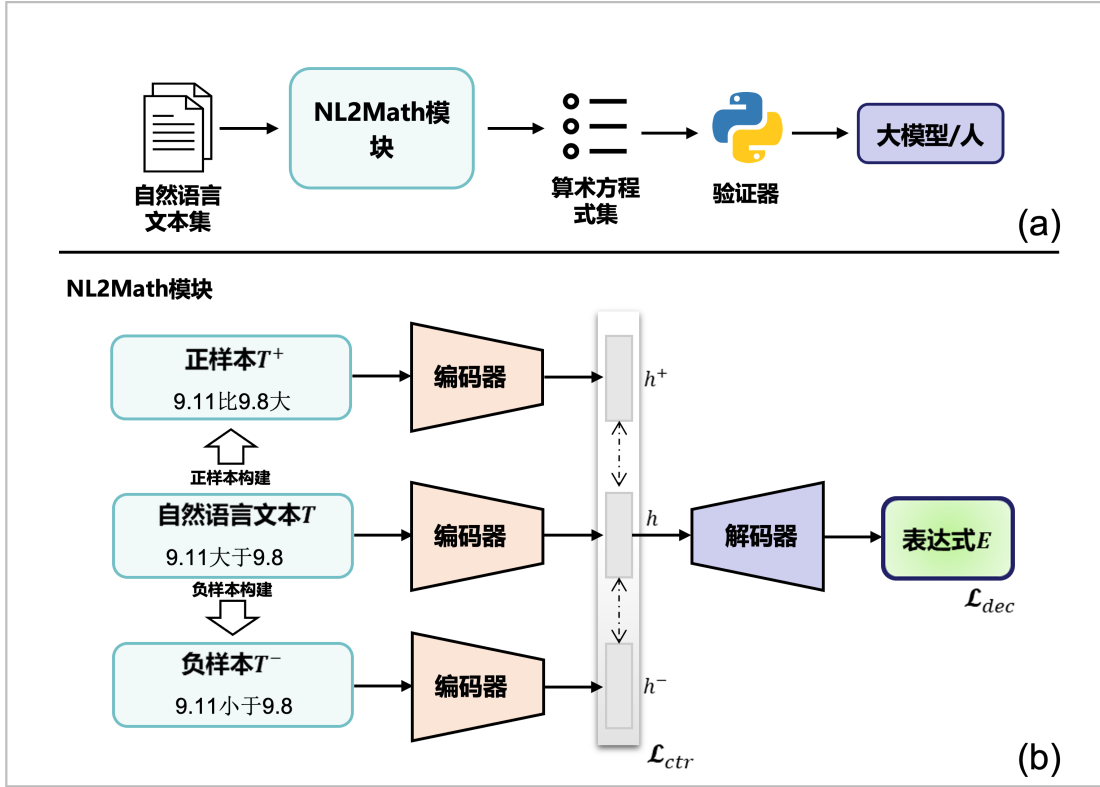


图 4-2 基于数学运算一致性验证的 AIGC 文本检测方法示意图

Figure 4-2 AIGC text detection based on mathematical operation consistency verification

作是对这些技术的拓展，使其更适用于数学文本场景。

### 4.3 方法介绍

本章介绍了 MCCV-LTD 方法，该方法的整体架构如图4-2所示。首先是通过 NL2Math 模块，将自然语言文本集合中的数学关系转换为数学表达式构造数学表达式集（如图4-2（b）），然后使用外部工具比如 Python 等验证器判断数学运算结果的错误率（如图4-2（a））。

#### 4.3.1 问题定义

假设自然语言文本集合为  $D = d_1, d_2, \dots, d_N$ ，其中每个文本  $d_i$  包含潜在的数学逻辑关系。目标是通过数学表达式转换以及运算一致性验证两阶段流程检测  $D$  中的数学运算错误：

**数学表达式转换** 数学表达式转换过程将自然语言文本  $d_i$  转换为数学表达式  $e_i$ ，定义转换函数：

$$F : D \rightarrow E \quad (4-1)$$

该过程包括两个关键步骤：

(1) 数学实体提取：从  $d_i$  中提取数学相关的实体，包括数值、变量和运算符

$$\text{Entity}(d_i) = \{(v_j, \tau_j)\}_{j=1}^K, \quad \tau_j \in \{\text{数值}, \text{变量}, \text{运算符}\} \quad (4-2)$$

(2) 数学关系建模：根据文本上下文推导数学表达式

$$e_i = \bigcup_{j=1}^K R(\tau_j, v_j, \text{Context}(d_i)) \quad (4-3)$$

其中， $R$  是关系映射函数，将自然语言逻辑（如“比... 少”）转换为数学运算（如减法“-”）。

在本节中，使用一种端到端的编解码器模型来直接学习自然语言到数学表达式的转换过程  $F$ ，从而自动建模数学实体关系。该方法能够更高效地解析数学文本，并自动生成数学表达式，提高数学计算错误检测的准确性。

**运算一致性验证** 验证  $e_i$  的逻辑一致性，输出检测结果  $Y = y_1, y_2, \dots, y_N$ ，其中

$$y_i = \begin{cases} 1 & (\text{存在错误}) \\ 0 & (\text{无错误}) \end{cases} \quad (4-4)$$

定义验证函数  $V: E \rightarrow Y$ ，通过符号计算验证逻辑一致性：

$$\mathcal{V}(e_i) = R(e_i^{\text{LHS}}, e_i^{\text{RHS}}) \quad (4-5)$$

其中， $I = R(\cdot)$  表示是否满足数学表达式中的关系， $LHS/RHS$  表示表达式左右两侧，最终错误检测结果区分出 AIGC 文本。

#### 4.3.2 NL2MATH 模块设计

NL2Math 模块的核心功能是将自然语言转化为数学等式或不等式，例如，给定一段自然语言文本“小张摘了 4 个苹果，小李摘了 3 个苹果，小张和小李摘的苹果总数是 7 个。”，上述问题对应的表达式为“ $4 + 3 = 7$ ”。在这一过程中，如何有效消除具体数字的干扰，使模型关注数量之间的关系而非具体数值的计算，是关键的一步。本小节将详细介绍 NL2MATH 模块的设计，主要从编码器、数字掩码和解码器三个部分展开。

##### 4.3.2.1 编码器

在本节中，选择 DeBERTa (Decoding-enhanced BERT with Disentangled Attention) 作为预训练语言模型，以提升对数学运算文本的理解和推理能力。DeBERTa 由 He 等提出，相较于 BERT 和 RoBERTa，该模型在结构设计上进行了多项优化，主要包括解耦注意力机制和增强掩码解码器 (Enhanced Mask Decoder, EMD)。

**解耦注意力机制** 传统的 Transformer 结构，在处理文本信息时，通常将词向量与位置信息简单相加后再输入模型。而 DeBERTa 采用了一种解耦的策略，将词汇内容信息和位置信息分别编码，从而提高模型在不同上下文环境下的泛化能力。具体而言，每个词元在模型中被表示为内容向量和相对位置向量的组合：

$$x_i = (H_i, P_{i|j}) \quad (4-6)$$

其中， $H_i$  代表词元本身的语义信息，而  $P_{i|j}$  则描述了该词元相对于其他词元的相对位置信息。相较于传统基于绝对位置编码的方法，DeBERTa 采用的相对位置编码方式可以更好地建模数学文本的逻辑关系，增强对数值之间相对位置的理解。

在计算词元之间的注意力权重时，将其分解为四个部分，以分别建模内容和位置之间的交互关系：

$$A_{i,j} = \underbrace{H_i H_j^T}_{\text{内容-内容交互}} + \underbrace{H_i P_{j|i}^T}_{\text{内容-位置交互}} + \underbrace{P_{i|j} H_j^T}_{\text{位置-内容交互}} + \underbrace{P_{i|j} H_j^T}_{\text{位置-位置交互}} \quad (4-7)$$

这四个部分分别表示：(1) 内容-内容交互：仅基于词元内容信息的相似性计算；(2) 内容-位置交互：词元内容与相对位置信息之间的交互；(3) 位置-内容交互：相对位置信息对词汇内容的影响；(4) 位置-位置交互：仅基于位置信息的交互，这一部分对结果的影响很小，所以一般不包含这一项。

为了实现更精细的上下文建模和相对位置感知，自注意力机制采用了参数投影和相对位置编码策略。对于内容向量的投影，通过三个不同的权重矩阵  $W_Q^c$ 、 $W_K^c$  和  $W_V^c$  将原始嵌入向量  $H$  投影到查询空间、键空间和值空间，分别得到投影后的查询向量  $Q_c$ 、键向量  $K_c$  和值向量  $V_c$ 。这些投影操作使得模型能够在不同的语义空间中对词进行表征和交互，增强了模型对语义信息的捕捉能力。同时，为了引入相对位置信息，相对位置编码通过权重矩阵  $W_Q^r$  和  $W_K^r$  对位置信息进行投影，得到相对位置查询向量  $Q_r$  和相对位置键向量  $K_r$ 。相对位置的计算采用了一种特定的度量方式，即对于两个词  $x_i$  和  $x_j$ ，其相对距离  $\delta(i, j)$  的取值范围被限制在  $[0, 2k)$ 。词元  $x_i$  和词元  $x_j$  之间的相对距离定义为：

$$\delta(i, j) = \begin{cases} 0, & \text{if } i - j \leq -k \\ 2k - 1, & \text{if } i - j \geq k \\ i - j + k, & \text{otherwise} \end{cases} \quad (4-8)$$

注意力分数矩阵  $A$  的计算综合考虑了内容信息和相对位置信息，以实现词之间关联的全面建模。对于注意力矩阵中的每个元素  $A_{ij}$ ，它表示词  $x_i$  对词  $x_j$  的注意力分数，其计算公式如下：

$$\bar{A}_{ij} = \underbrace{Q_i^c K_j^{c\top}}_{\text{内容-内容交互}} + \underbrace{Q_i^c K_{\delta(i,j)}^{r\top}}_{\text{内容-位置交互}} + \underbrace{K_j^c Q_{\delta(j,i)}^{r\top}}_{\text{位置-内容交互}} \quad (4-9)$$

这一设计使得模型能够有效地感知词之间的相对位置关系，并将这种位置信息融入到注意力计算过程中，从而生成更具语义和结构信息的上下文嵌入表示。

#### 4.3.2.2 数字掩码

基于之前的研究，对生成的句子进行了数字掩码处理<sup>[142]</sup>。具体做法是将文本中的数值替换为掩码字符，如 [NUM1]、[NUM2]、[NUM3] 等。例如，将原表达式“4 + 3 = 7”转换为：[NUM1]+[NUM2]=[NUM3]。通过掩码标记，模型可以更清晰地捕捉数学表达式中的关系模式，而无需处理具体数值带来的复杂性。

#### 4.3.2.3 解码器

在解码阶段，采用了 DeBERTa 改进的掩码解码策略，即增强掩码解码器。这一机制旨在优化掩码语言建模任务，使得模型能够更充分地利用相对位置信息，提高学习效果。

在 Transformer 结构中，由于绝对位置信息的过早引入，可能会影响模型对相对位置信息的建模能力。而 DeBERTa 仅在最后一层，即接近 softmax 预测层，才引入绝对位置信息，从而保证 Transformer 的自注意力机制主要依赖相对位置信息进行计算，而不会受到绝对位置信息的干扰。解码过程的核心计算如下：

$$H_{\text{output}} = \text{softmax} \left( \frac{\bar{A}}{\sqrt{3d}} \right) V_c \quad (4-10)$$

其中， $\bar{A}$  是解耦注意力机制得到的加权矩阵，而  $\sqrt{3d}$  是归一化因子，旨在稳定梯度并提高训练的稳定性。

数学文本的推理任务往往涉及复杂的数值运算和逻辑推理，因此通过 DeBERTa 的 EMD 机制，使模型在训练过程中能够更加依赖相对位置信息，减少对绝对位置信息的依赖。增强了模型对数学问题中数值关系的理解能力，进而提高在数学运算一致性检测任务上的准确性。

#### 4.3.3 训练过程

##### 4.3.3.1 正负样本构建

在训练过程中，构建正负样本有助于后续的对比学习。正样本的生成主要有利用提示模板让大模型生成语义不变的回答。对于多个生成的文本，计算它们与

原文本之间的语义相似度，并选取语义最为接近的句子作为正样本。语义相似度计算如下：

$$Sim(T_i, T_j) = \cos(Emb(T_i), Emb(T_j)) \quad (4-11)$$

其中嵌入向量  $Emb(T_i)$  和  $Emb(T_j)$  是通过预训练的语言模型（如 SBERT<sup>[143]</sup> 等）得到的向量表示。

负样本则通过改变数学表达式中的表述以及对应的自然语言文本内容来构建。例如，通过调整表达式中的等式关系为不等式（比如大于或小于关系），对应的自然语言文本中表示相等关系的表述修改为大于或小于关系；或是改变运算符号，对应的自然语言文本可以替换为样本集合中与改变后的运算符相匹配的自然语言文本内容。这样的负样本能够帮助模型更好地学习区分不同的数学关系，尤其是等式和不等式之间的细微差别。

#### 4.3.3.2 对比学习

在模型训练过程中，引入对比学习框架，以捕捉正负样本之间的语义差异。假设原始文本为  $T_i$ ，正样本为  $T_i^+$ ，负样本为  $T_i^-$ 。 $h_i, h_i^+, h_i^-$  分别为它们输入到编码器之后隐层的表示，损失函数采用对比学习损失：

$$\mathcal{L}_{ctr} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(sim(h_i, h_i^+)/\tau)}{\sum_{j=1}^N (\exp(sim(h_i, h_j^+)/\tau) + \exp(sim(h_i, h_j^-)/\tau))} \quad (4-12)$$

#### 4.3.3.3 损失函数

在模型设计上，使用 DeBERTa 模型<sup>[141]</sup> 作为编码器和解码器。DeBERTa 具有强大的上下文建模能力，能够有效捕捉语义关系和逻辑结构。其增强的注意力机制和相对位置编码方法，使模型能够理解复杂的依赖关系和数学表达式中的数量关联。

在训练过程中，损失函数包括对比学习损失  $\mathcal{L}_{ctr}$  和有监督损失  $\mathcal{L}_{dec}$ ：

$$\mathcal{L}_{ctr} = \alpha \cdot \mathcal{L}_{ctr} + \beta \cdot \mathcal{L}_{dec} \quad (4-13)$$

其中， $\alpha$  和  $\beta$  是两个可调超参数，用于控制两种损失的权重平衡。

#### 4.3.4 数学运算一致性验证

通过 NL2Math 模块将自然语言文本集合中的文本转换为数学表达式，构成表达式集合。使用外部验证器如 Python，依次验证数学运算结果是否正确。由于具备一定数学知识的人类在简单的数学表达式的运算中可以做到不出错，而当前一些主流的大模型可能会出现错误。因此，根据运算过程和结果是否一致，则可判断文本是否由 AIGC 生成。

提示模板:

下面给出几个示例, 任务是将疑问句改为陈述句, 注意给出问题答案; 修改相应的方程式格式, 注意括号要有; 保持索引顺序, 确保 lEquations 以 X= 开始, 最后也要保留运算结果。注意 sQuestion 和 lEquation 中的数字的严格对齐, 如 48.0 对应 48.0 而不是 48。

Example 1:

Input:

Description: "There are 397 butterflies. Each butterfly has 12 black dots and 17 yellow dots. How many black dots in all?"

Equation: "397.0\*12.0"

Solution: "4764.0"

Output:

Description: "There are 397 butterflies. Each butterfly has 12 black dots and 17 yellow dots. There are 4764.0 black dots in all."

Equation: "X=(397.0\*12.0)=4764.0"

Example 2:

Input:

Description: "Spencer walked 0.3 of a mile from his house to the library, 0.1 of a mile from the library to the post office, and 0.4 of a mile from the post office back home. How many miles did Spencer walk in all ?"

Equation: "0.3+0.1+0.4"

Solution: "0.8"

Output:

Description: "Spencer walked 0.3 of a mile from his house to the library, 0.1 of a mile from the library to the post office, and 0.4 of a mile from the post office back home. The total miles Spencer walked was 0.8."

Equation: "X=(0.3+0.1+0.4)=0.8"

Input:

图 4-3 构建数据集的提示模板

Figure 4-3 Prompt template for building dataset

## 4.4 实验结果与分析

### 4.4.1 数据集

**MAWPS 数据集**<sup>[144]</sup> MAWPS 数据集是一个在线数学应用题 (Math Word Problems) 存储库。该数据集支持自动构建具有特定特征的数据集, 并提供工具来调

表 4-1 数据集数据统计

Table 4-1 Data statistics of datasets

| 数据集        | MAWPS | SVAMP | PARAMAWPS | MATH_NEW |
|------------|-------|-------|-----------|----------|
| 文本数量       | 2,373 | 1,000 | 16,278    | 1,463    |
| 操作符种类      | 4     | 4     | 4         | 7        |
| 平均操作符数量    | 1.606 | 1.236 | 1.68      | 2.596    |
| 表达式中平均数字数量 | 2.59  | 2.53  | 2.67      | 3.49     |

整词汇和模板重叠度，同时能够过滤来源于网络的不合语法的问题。MAWPS 具有开放性，支持社区贡献。该数据集已包含 3,320 道数学应用题，其中包括多个重要研究使用的完整数据集。

**SVAMP 数据集<sup>[145]</sup>** SVAMP 数据集是一个专门用于挑战现有数学应用题(MWP)求解模型的测试集，主要包含一元算术应用题，每道题最多涉及两个运算符。与 MAWPS 相比，SVAMP 在算术表达式的复杂度上相似，并未刻意增加解题难度。然而，SVAMP 通过特殊设计，避免了模型仅依赖表层模式匹配来求解问题。

**PARAMAWPS 数据集<sup>[146]</sup>** PARAMAWPS 数据集是一个大规模数学应用题数据集，共包含 16,278 道单一方程应用题。它以 MAWPS 数据集为基础，通过复述和手动构造逆问题生成。该数据集保留了 MAWPS 的全部问题，并对其中较具挑战性的问题生成多种复述版本。每道原始问题最多可生成 16 个变体，涵盖 4 种不同类型的变化。所有生成样本均由 3 名本科生人工审核，确保正确性。

**MATH\_NEW 数据集** 本研究基于 WAPS 数据集的模板，构建了一个包含数学问题描述及其对应答案的大规模数据集。通过构造示例模型作为上下文信息提示大模型，将自然语言描述的数学问题转换为包含最终答案的陈述句，模板设计如图4-3。例如，给定问题“Melanie picked 4 plums, and Sally picked 3 plums. How many plums were picked in total?”，转换后为“Melanie picked 4 plums, and Sally picked 3 plums. The total number of plums picked by Melanie and Sally is 7.”。此外，MATH\_NEW 数据集进一步引入包含数字比较的不等式表达式，以扩展数学表达式的覆盖范围。这些文本主要由大模型（如 GPT-4o、Claude 3.5 和 Claude 3.7）生成，数据集的统计信息如表4-1所示。由于当前大模型在处理简单数学和逻辑推理任务时仍存在较高错误率，本研究在测试集中引入了包含数学运算错误的样例，基于之前的研究<sup>[9]</sup>。其中，大模型生成的错误样例约占 5%，而人类错误

表 4-2 大模型生成文本检测指标

Table 4-2 Metrics of LLM-generated Text detection

| Method                | Accuracy     | Precision    | Recall       | F1           |
|-----------------------|--------------|--------------|--------------|--------------|
| <b>Log-Likelihood</b> | <u>0.571</u> | <u>0.564</u> | 0.625        | 0.593        |
| <b>Rank</b>           | 0.429        | 0.454        | 0.714        | 0.556        |
| <b>GLTR</b>           | 0.500        | 0.500        | 0.571        | 0.533        |
| <b>Entropy</b>        | <u>0.654</u> | <u>0.654</u> | 0.654        | <u>0.654</u> |
| <b>LRR</b>            | 0.464        | 0.471        | 0.571        | 0.516        |
| <b>NPR</b>            | 0.500        | 0.500        | 0.563        | 0.530        |
| <b>DetectGPT</b>      | 0.467        | 0.462        | 0.400        | 0.429        |
| <b>ChatGPT-D</b>      | 0.542        | 0.526        | <b>0.833</b> | <u>0.645</u> |
| <b>MCCV-LTD</b>       | <b>0.714</b> | <b>0.714</b> | <u>0.789</u> | <b>0.749</b> |

率仅约为 1%。最终在该测试集上进行 AIGC 检测评估，以衡量本文方法与现有方法的识别能力。

#### 4.4.2 实验结果

对比 MCCV-LTD 和其他模型方法进行实验，得出的实验结果如表4-2。Log-Likelihood 方法（0.571 准确率）能够捕捉模型生成的低概率文本，但无法判断数学运算逻辑是否正确。Rank、GLTR、Entropy 等方法依赖文本的统计特征，而 AIGC 生成的数学文本在形式上与人类文本接近，因此这些方法在数学运算的检测上表现不佳。ChatGPT-D 方法的召回率较高，达到了 0.833，但精确率较低，说明难以准确区分 AIGC 生成的数学运算文本。对比的方法主要依赖于语言流畅度或生成概率进行 AIGC 文本检测。这些方法在检测大模型生成的通用文本方面可能有效，但数学推理错误往往符合语言模型的流畅性，因此传统方法难以区分 AIGC 生成的错误数学文本与真实人类书写的正确数学文本。基于数学运算规则直接验证数学运算结果的正确性，突破仅依赖语言流畅性的检测方法的局限性，提高对数学推理错误的检测能力。实验结果表明 MCCV-LTD 具有明显优势，准确率和 F1 值都显著高于其他方法，分别提升了约 6% 和 9%，体现了更强的数学推理错误检测能力。

**结论** 由实验结果，可以得出以下结论：

- 现有方法依赖语言流畅性，在数学推理错误检测任务上存在局限性，导致准确率和 F1 分数较低。
- MCCV-LTD 通过数学运算一致性验证，在 AIGC 生成的数学文本检测任

表 4-3 自然语言文本转数学表达式模块的准确率

Table 4-3 Accuracy of NL2MATH module

| Method         | MAWPS(%) | SVAMP(%) | PARAMAWPS(%) | MATH_NEW(%) |
|----------------|----------|----------|--------------|-------------|
| RNNVAE         | 77.9     | 23.0     | 65.8         | 62.7        |
| LSTM           | 76.4     | 27.0     | 66.1         | 65.3        |
| Transformer    | 72.9     | 22.0     | 58.6         | 53.3        |
| BERT           | 77.4     | 26.0     | 65.1         | 64.4        |
| RoBERTa        | 77.9     | 28.0     | 66.5         | 65.3        |
| DeBERTa w/o CL | 78.4     | 63.4     | 68.1         | 67.1        |
| w/ CL          | -        | -        | -            | 69.2        |

务上有明显的性能提升，提升了约 9%。

- 实验结果验证了该方法的有效性，后续可进一步扩展到更复杂的数学推理任务，以提升对高阶数学运算错误的检测能力。

4.4.2.1 消融实验

为了验证对比学习的有效性以及 NL2MATH 模块对 AIGC 文本检测性能的影响，本节进行了消融实验。实验使用了四个数据集，包括三个公开数据集 (MAWPS、SVAMP、PARAMAWPS) 以及本节构建的 MATH\_NEW 数据集，实验结果如表4-3所示，并以准确率作为评价指标。

从实验结果来看，不同模型在 NL2MATH 任务上的表现存在明显差异：传统序列模型 (RNNVAE、LSTM) 在 MAWPS 和 PARAMAWPS 数据集上表现尚可，但在更具挑战性的 SVAMP 数据集上表现较差 (仅 23.0% 和 27.0%)。BERT 和 RoBERTa 在 SVAMP 上的准确率略高 (26.0%、28.0%)，但仍不足以处理复杂数学推理问题。Transformer 在 SVAMP (22.0%) 和 PARAMAWPS (58.6%) 这两个数据集上的表现较差，表明其难以有效建模数学推理关系。DeBERTa 在所有数据集上的表现最佳，特别是在 SVAMP 和 MATH\_NEW 上取得了明显提升。DeBERTa w/CL 在 MATH\_NEW 数据集上进一步提升至%，说明对比学习能够增强模型对数学表达式的区分能力，提高数学推理能力。因此可以得出下面的结论：

- 标准 Transformer 结构的数学推理能力有限，而预训练语言模型具有更好的数学推理泛化能力。

- 对比学习显著提升了模型对数学推理关系的理解能力，特别是在 MATH\_NEW 数据集上的表现进一步验证了其有效性。

- NL2MATH 模块结合对比学习后，能够更精准地检测数学推理错误，相较

于基线方法提升了约 2.1%。

#### 4.5 本章小结

本章提出了一种基于数学运算一致性验证的检测方法——MCCV-LTD。该方法通过将文本转换为数学表达式，并利用数学运算规则验证其逻辑合理性，从而有效识别文本中的数学推理错误和逻辑矛盾。相比于仅依赖统计特征或语言模型概率分布的方法，MCCV-LTD 能够更精准地捕捉文本中的数学推理缺陷。本方法不仅适用于简单等式，还能处理更复杂的数学运算，如嵌套的加减乘除运算以及数值大小关系。为提升模型在数学运算文本上的泛化能力，MCCV-LTD 引入了对比学习技术，通过构建正负样本对，引导模型关注数学表达式中的细微差异，从而增强对运算关系的区分能力。实验结果表明，在新构建的 MATH\_NEW 数据集上，MCCV-LTD 相较于其他检测方法取得了 9% 的提升，验证了其在检测包含数学运算这种功能性语言文本方面的有效性。同时也存在不足，目前只是从简单的数学公式出发进行检测，当面临更加复杂的数学运算或推理的时候，检测效果可能会下降，同时面临比较复杂的表达式时 NL2MATH 的过程中可能会出错，导致检测准确率下降。



## 第5章 融合逻辑推理一致性验证的 AIGC 文本检测方法

### 5.1 引言

AIGC 生成的文本在为众多任务提供支持的同时，也暴露出了一系列质量问题。这些问题主要包括主题偏离、逻辑错误以及幻觉现象等。例如，在自动问答系统中，主题偏离可能导致回答与问题不相关；逻辑错误则可能使回答缺乏可信度；而幻觉现象则可能生成虚假或误导性的信息。这些问题不仅影响了生成文本的可靠性，还可能对依赖这些文本的任务带来严重的负面影响。因此，如何有效检测 AIGC 文本中的错误，保障其质量和一致性，成为亟待解决的问题。

现有的 AIGC 文本检测方法大多集中于单一维度的评估，如基于语言模型的文本语言流畅度或通过扰动前后文本语言特征之间的变化进行检测。然而，这些方法往往忽视了文本深层次的一致性问题的，尤其是主题偏离和逻辑推理错误。这些问题通常表现在生成文本的内容与其所讨论的主题不完全对齐，或者推理过程中的前提和结论不一致，导致生成文本出现不符合常理的“幻觉”现象。

本节方法主要包括以下三个创新点：

(1) 提出了综合考虑主题相关性和逻辑推理能力的多维度检测框架 ILCV-TLD (Integrating Logical Consistency Verification for LLM-generated Text Detection)，克服了现有方法单一维度评估的局限性。通过将主题和逻辑一致性相结合，本方法能够深入分析文本的语义和逻辑结构，从而提供更精准的检测结果。

(2) 提出了两阶段的逻辑一致性检测方法。首先，设计了一种端到端的自然语言到一阶谓词逻辑 (First-Order Logic, FOL) 模块，将自然语言文本转换为一阶谓词逻辑表达式，表达复杂的逻辑关系和推理规则。其次，利用成熟的 FOL 求解器对转换后的逻辑表达式进行验证，检查是否存在逻辑错误。

(3) 提出了一种纠正策略，能够针对自然语言转一阶谓词逻辑过程中可能出现的错误进行有针对性的检测，一定程度上缓解了这一过程中潜在的错误传播问题，从而提升生成文本的整体质量。

### 5.2 相关工作

本节介绍了自然语言-一阶谓词逻辑翻译以及 FOL 求解器的相关研究工作。

#### 5.2.1 自然语言-一阶谓词逻辑翻译

早期的自然语言到一阶谓词逻辑逻辑 (NL-FOL) 转换工作主要依赖于基于语法的处理方法，这些方法在处理结构化良好的文本时较为有效，但在面对复杂

的语言构造时效果不佳。由于它们难以处理无固定结构的句子，而自然语言固有的歧义性使得这一问题更加复杂。神经网络、深度学习和大型语言模型的出现为 NL-FOL 转换引入了基于数据的方法，使得线性时序逻辑<sup>[147]</sup>和第一阶逻辑<sup>[148]</sup>等形式逻辑的自然语言转换取得了进展。然而，语言的歧义性仍然使得这些方法无法准确地将复杂句子转化为逻辑形式。Yang 等通过从 GPT-4 收集 NL-FOL 对进行微调，提升了像 LLaMA-7B 这样的模型在 NL-FOL 转换任务中的表现。但是对于参数量比较小的模型，在处理复杂逻辑推理方面的能力可能会受到限制。Aly 等则提出了一种用于问答的推理过程，通过生成自然逻辑证明来识别断言和证据文本片段之间的关系。而Lalwani 等则结合事实和前一步的证明信息，这种方法处理的任务更加复杂，适用于涉及更复杂推理链的任务。但是，如果推理链中的任何一步出现错误，都会影响最终的转换结果。针对上述问题，本节设计一种端到端的自然语言到第一阶逻辑（NL2FOL）模块，将自然语言文本转换为一阶谓词逻辑表达式，并提出一种纠正策略对转换过程中出错的结果进行重新检测。

### 5.2.2 FOL 求解器

近年来，求解器如 Z3<sup>[152]</sup>和 CVC<sup>[153]</sup>已被广泛用于验证逻辑公式的可满足性和有效性。这些工具通过应用多种理论和推理规则，对一组逻辑公式进行分析，以判断其是否可满足。在验证公式的有效性时，通常采用将公式取反并测试其不可满足性的方法。Olausson 等的研究表明，理论求解器可以应用于自然语言的逻辑推理任务。为了优化在面对模糊不清的前提和结论情况下的性能，Lalwani 等改进了该解析器，使其能够将自然语言转换为一阶谓词逻辑。

## 5.3 方法介绍

本章介绍了 ILCV-LTD 方法，该方法的整体架构如图5-1 所示。

### 5.3.1 问题定义

给定提示（问题）集合  $P = \{p_1, p_2, \dots, p_N\}$  和对应的生成文本（回答）集合  $D = \{d_1, d_2, \dots, d_N\}$ ，本方法旨在结合主题相关性检测和逻辑推理验证，评估每个文本的 AIGC 生成可能性。首先，通过计算每个文本与提示的语义相似度  $x_i$  来衡量主题相关性；然后，通过逻辑推理验证，输出  $y_i$ ，表示推理是否正确。最终，结合主题和逻辑一致性  $r_i$  得到每个文本由 AIGC 生成的可能性，若  $r_i$  大于阈值  $\tau$ ，则认为该文本可能是 AIGC 生成。

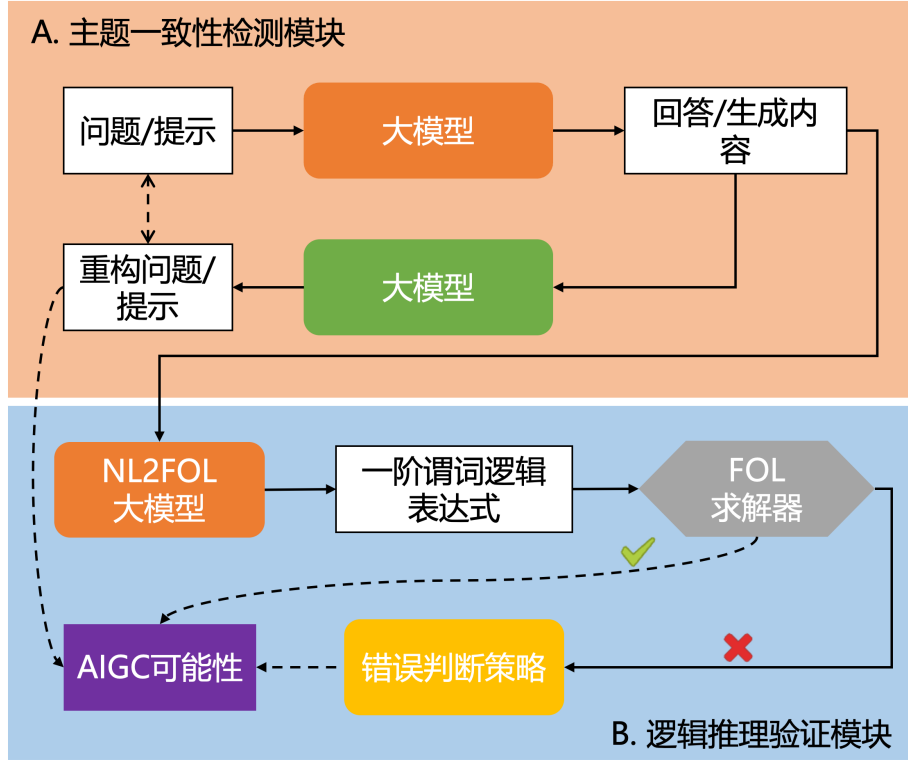


图 5-1 融合逻辑推理一致性验证的 AIGC 文本检测方法示意图

Figure 5-1 AIGC text detection integrating logical consistency verification

### 5.3.2 主题相关性检测模块

本小节提出了一种基于语义重构的主题相关性检测 (TD) 模块, 主要方法如图5-1.A 所示, 其中橙色和绿色的部分表示对同一大模型使用不同的提示词。主要实现步骤如下:

首先, 引入逆向生成过程, 用于从生成内容 (回答)  $d_i$  反推出可能的原始问题  $q_i$ 。具体做法是将提示词组织成“内容-问题”对的形式, 表示为:

$$T = d_1, p_1, \dots, d_n, p_n \quad (5-1)$$

其中,  $T$  作为小样本示例提供给大模型, 让它了解到任务目标, 进行上下文的学习。然后, 将待检测的文本  $d_i$  添加到示例  $T$  的末尾, 便可以得到重构后的问题  $q_i$ 。本方法遵循Yehuda 等的做法, 设置不同温度, 得到不同的大模型的重构结果。原来重构出的单个问题便扩充为  $k$  个, 得到  $Q_i = q_{i1}, q_{i2}, \dots, q_{ik}$ 。这里  $k$  的值也就是大模型的数量, 可以表示为:

$$Q_i = LLM^{-1}(X), \quad X = [T; D] \quad (5-2)$$

其中,  $LLM(X)$  表示大模型根据提示 (问题) 生成内容, 那么  $LLM^{-1}(X)$  表示大模型根据回答 (生成内容) 生成的提示 (问题)。

接下来,使用文本词向量之间的相似度来衡量重构问题(提示)与原始问题(提示)之间的相关性。具体而言,首先使用词嵌入模型  $Embedding: \text{text} \rightarrow \mathbb{R}^m$  将文本映射到  $m$  维向量空间:

$$\begin{aligned} P_i^{\text{vec}} &= Embedding(P_i) \\ Q_i^{\text{vec}} &= \{Embedding(Q_{ij}) \mid j \in \{1, 2, \dots, k\}\} \end{aligned} \quad (5-3)$$

然后,计算  $P_i^{\text{vec}}$  与  $Q_i^{\text{vec}}$  之间的平均余弦相似度:

$$\text{sim}(P, Q) = \text{Average} \left[ \cos(P_i^{\text{vec}}, Q_{ij}^{\text{vec}}) \right], i \in \{1, 2, \dots, N\}, j \in \{1, 2, \dots, k\} \quad (5-4)$$

最后得到的相似度  $\text{sim}(P, Q)$  就表示文本内容与主题之间的一致性。若重构问题与原始问题存在较大偏差,则  $D_i$  更可能是模型生成的内容。

### 5.3.3 逻辑推理验证模块

本小节主要介绍 ILCV-LTD 方法中的推理验证 (LCV) 模块,详细步骤如下:

**基于小样本的 NL2FOL 方法** 本小节提出了一种基于小样本学习的自然语言到一阶逻辑 (NL2FOL) 转换方法,用于将自然语言文本转换为一阶谓词逻辑表达式,从而支持后续的逻辑推理验证。该方法的核心思想是利用小样本学习,指导大模型掌握从自然语言到 FOL 表达的映射关系。本方法的主要流程如下:

设计多组示例,包括自然语言句子及其对应的 FOL 表达式。这些示例作为输入,帮助大模型学习如何将自然语言逻辑结构形式化。示例形式如下:

$$S = \{(s_1, F_1), (s_2, F_2), \dots, (s_n, F_n)\} \quad (5-5)$$

其中,  $s_i$  表示自然语言句子,  $F_i$  表示其对应的一阶逻辑表达式。

将小样本示例  $S$  作为上下文输入到千问推理模型 QwQ,生成其对应的 FOL 表达式  $F$ :

$$F = LLM(s|S) \quad (5-6)$$

将生成的 FOL 表达式  $F$  传递给一阶逻辑求解器 (如 Z3 或 CVC),通过逻辑推理判断文本的正确性。求解器返回一个二值结果:

$$y = \begin{cases} 0, & \text{推理错误} \\ 1, & \text{推理正确} \end{cases} \quad (5-7)$$

其中,  $y = 0$  表示文本存在逻辑推理错误,  $y = 1$  表示推理正确或文本未涉及推理。该方法的优势在于能够形式化推理规则, 提高检测的解释性。然而, 由于语言的复杂性, NL2FOL 转换可能会引入错误, 因此需要进一步的错误判断策略。

**基于提示学习的错误判断策略** 上述过程中, 有一些文本被检测为错误, 可能存在两种情况。一种是因为文本本身推理出错, 在检测过程准确将它识别出来, 这是最为理想的情况。另一种则是文本本身没有出现推理错误, 而是在 NL2FOL 转换的过程中发生了错误, 导致结果出错, 这也是此类方法存在的问题。即在 NL2FOL 转换过程中, 由于自然语言的复杂性和模糊性, 可能会出现逻辑表达式生成错误的情况。为了缓解这一问题, 本文提出了一种基于提示学习的错误判断策略  $LLM_j$ , 利用大模型的推理能力进行辅助判别, 提高逻辑一致性检测的准确性。具体步骤如下:

将检测错误的自然语言文本, 通过思维链的方式输入给大模型, 判断其中是否存在逻辑错误, 得到最终的逻辑验证结果。

最终, 综合考虑大模型可能出现的幻觉和逻辑推理问题, 给出最终结果:

$$\begin{aligned} x &= (1 - \text{sim}(P, Q)) \\ r &= (x > \theta) \vee (1 - y) \end{aligned} \quad (5-8)$$

其中,  $x$  为文本与主题的相关性评估值, 表示文本偏离主题的程度,  $\theta$  是阈值。  $y$  为逻辑推理验证模块的输出, 1 表示推理正确, 0 表示推理错误。最终, 通过评估值  $r$ , 可以判断文本的生成来源。

## 5.4 实验结果与分析

### 5.4.1 数据集

**LOGIC 数据集<sup>[155]</sup>** LOGIC 数据集是一套专门用于研究逻辑谬误的语料库, 包含 2,449 个常见的逻辑谬误实例。其中, LOGICCLIMATE 数据集是 LOGIC 的一个特定子集, 包含 1,079 个与气候变化相关的逻辑谬误实例。这些实例均来自 Climate Feedback 网站上的气候变化新闻报道, 为逻辑谬误研究提供了高质量的真实世界数据。

**斯坦福自然语言推理 (SNLI) 数据集<sup>[156]</sup>** SNLI 数据集由超过 170,000 组句子对构成, 每组句子对均来源于蕴含关系数据, 并被标注为蕴含 (entailment)、矛盾 (contradiction) 或中性 (neutral) 关系。在逻辑推理与谬误检测的研究中, LOGIC

和 LOGICCLIMATE 数据集常与 SNLI 结合使用，研究者通常会从 SNLI 中随机抽取与逻辑谬误数量相等的有效陈述，以构建平衡数据集。

本研究参考了相关工作<sup>[151]</sup>，将 LOGIC 和 LOGICCLIMATE 数据集与 SNLI 结合，并利用大模型在数据上添加了句子描述的主题信息。由于大模型在文本生成过程中可能出现内容偏离原始主题的情况，该数据集包含了一定比例的主题偏离文本。本节使用的数据集包含四种文本类型：T+L+、T+L-、T-L+ 和 T-L-，以评估使用的方法在识别逻辑谬误和主题一致性上的能力。

#### 5.4.2 实验结果

实验结果表明，ILCV-LTD 在三类错误类型（T+L-、T-L+、T-L-）的 AIGC 文本检测任务中均显著优于现有方法，体现了其在主题一致性分析与逻辑推理验证方面的有效性。

在 T+L-（主题一致但逻辑错误）类别中，ILCV-LTD 在所有指标上均远超其他方法 F1-score 高达 95.26%，表明该方法能够精准识别逻辑推理错误。相比之下，传统方法在此类错误的检测上表现较弱，例如 DetectGPT 在此类别下的 F1-score 仅为 59.09%，而 Log-Likelihood 和 GLTR 的 F1-score 分别为 55.81% 和 53.66%，说明基于统计特征的方法难以有效捕捉逻辑错误。

其次，在 T-L+（逻辑正确但主题偏离）类别中，ILCV-LTD 依然保持领先，准确率达到 86.22%，F1-score 为 86.22%，在所有基准方法中均排名最高。这表明 ILCV-LTD 的主题相关性检测模块在识别文本主题偏离方面具有极高的可靠性。相比之下，现有方法在该类别下表现不稳定，如 DetectGPT 的 F1-score 仅为 0.00%，表明其在主题一致性检测方面几乎失效，而 Log-Likelihood 的 F1-score 也仅为 68.18%，表明基于语言模型的概率分布方法难以准确刻画主题相关性。

最后，在 T-L-（两种错误同时存在）类别中，ILCV-LTD 进一步展现了其强大的检测能力，F1-score 高达 98.48%，远超其他方法。

**结论** 由实验结果，可以得到以下结论：

- ILCV-LTD 在所有错误类别下均表现出显著优势，证明了其在 AIGC 文本检测中的有效性。
- 依赖统计特征或概率分布的传统检测方法难以精准识别文本的逻辑推理错误和主题偏离问题。
- ILCV-LTD 通过融合主题一致性检测与逻辑推理验证，实现了更高的检测准确率和稳健性，为 AIGC 文本检测提供了更可靠的解决方案。

##### 5.4.2.1 消融实验

本小节进行了消融实验，为了验证 ILCV-LTD 方法中每个模块的有效性，实验结果如表5-2所示。

表 5-1 融合逻辑推理一致性验证的 AIGC 文本检测实验结果  
Table 5-1 Metrics of LLM-generated Text detection

| Method              | T+L-  |       |       |       | T-L+  |       |       |       | T-L-  |       |       |       |
|---------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                     | ACC   | P     | R     | F1    | ACC   | P     | R     | F1    | ACC   | P     | R     | F1    |
| Log-Likelihood      | 56.81 | 57.14 | 54.54 | 55.81 | 68.18 | 68.18 | 68.18 | 68.18 | 60.23 | 62.16 | 52.27 | 56.79 |
| Rank                | 47.73 | 48.21 | 61.36 | 54.00 | 55.68 | 57.14 | 45.45 | 50.63 | 57.95 | 59.46 | 50.00 | 54.32 |
| GLTR                | 56.81 | 57.89 | 50.00 | 53.66 | 63.63 | 63.63 | 63.63 | 63.63 | 57.95 | 59.46 | 50.00 | 54.32 |
| Entropy             | 42.05 | 41.46 | 38.63 | 40.00 | 56.82 | 56.82 | 56.82 | 56.82 | 48.86 | 48.94 | 52.27 | 50.55 |
| LRR                 | 48.86 | 48.83 | 47.72 | 48.28 | 64.77 | 63.27 | 70.45 | 66.67 | 56.82 | 57.89 | 50.00 | 53.66 |
| NPR                 | 50.00 | 0.00  | 0.00  | 0.00  | 63.15 | 72.72 | 42.11 | 53.33 | 52.63 | 51.72 | 78.95 | 62.50 |
| DetectGPT           | 52.63 | 52.00 | 68.42 | 59.09 | 0.50  | 0.00  | 0.00  | 0.00  | 63.16 | 73.73 | 42.11 | 53.33 |
| ChatGPT-D           | 50.00 | 50.00 | 100   | 66.67 | 50.00 | 50.00 | 100   | 66.67 | 50.00 | 50.00 | 100   | 66.67 |
| Qwen-Max(zero-shot) |       |       |       |       |       |       |       |       |       |       |       |       |
| Qwen-Max(few-shot)  |       |       |       |       |       |       |       |       |       |       |       |       |
| ILCV-LTD            | 90.95 | 100   | 90.95 | 95.26 | 86.22 | 100   | 92.60 | 86.22 | 97.01 | 100   | 97.01 | 98.48 |

表 5-2 消融实验结果

Table 5-2 Ablation experiment results

| Method                | $\theta = 0.05$ |              | $\theta = 0.2$ |              | $\theta = 0.4$ |               |
|-----------------------|-----------------|--------------|----------------|--------------|----------------|---------------|
| Method                | ACC             | F1           | ACC            | F1           | ACC            | F1            |
| ILCV-LTD w/o LCV      | 71.60           | 82.21        | 61.59          | 70.06        | 41.46          | 38.86         |
| ILCV-LTD w/o TD       | 52.56           | 57.59        | 52.56          | 57.59        | 52.56          | 57.59         |
| ILCV-LTD w/ TD w/ LCV | <b>74.65</b>    | <b>84.56</b> | <b>70.40</b>   | <b>79.17</b> | <b>66.16</b>   | <b>74.23</b>  |
| Improv.               | <b>3.05↑</b>    | <b>2.35↑</b> | <b>8.81↑</b>   | <b>9.11↑</b> | <b>13.6↑</b>   | <b>16.64↑</b> |

• **ILCV-LTD w/o LCV**: 表示使用本小节方法中的主题相关性检测模块, 而不使用逻辑推理验证模块;

• **ILCV-LTD w/o TD**: 表示使用本小节方法中的逻辑推理验证模块, 而不使用主题相关性检测模块;

• **ILCV-LTD w/ TD w/ LCV**: 表示同时使用本小节方法中的主题相关性检测模块和逻辑推理验证模块。

实验对比了 ILCV-LTD w/o LCV 与完整模型的性能差异。结果显示, 在所有评估阈值  $\theta$  设置下, 完整模型在准确率和 F1 值两项指标上均表现出显著优势。表明 LCV 模块对于提升检测精度具有重要作用。此外, ILCV-LTD w/o LCV 也就是 TD 模块, 随着阈值的增加, 性能呈下降趋势。由于, 该阈值主要对 TD 模块产生影响, LCV 模块则不受影响。阈值小于 0.4 的时候, TD 模块的准确率和 F1 值总是优于 LCV 模块, 而当阈值到 0.4 之后, TD 模块性能明显不如 LCV 模块。融合两个模块的方法 ILCV-LTD 仍然有 66.16% 的准确率, 表明两种模块在不同阈值下的有效结合, 提升了检测性能。

实验结果表明, 移除主题相关性检测 (TD) 和逻辑推理验证 (LCV) 模块均会导致模型性能的显著下降, 这充分验证了两者在 AIGC 文本检测中的关键作用。

#### 5.4.2.2 分析实验

**超参数实验** 本小节进行了超参数 (阈值) 的实验, 结果如图5-2所示。可以看到, 不同阈值对精确率 (P)、召回率 (R) 和 F1 值 (F1) 产生了显著影响。随着阈值的增加, P 整体呈上升趋势, 在阈值为 0.3 时达到峰值 0.897, 随后在 0.4 时略有下降至 0.878, 说明较高阈值有助于提高精确率, 减少误报 (错误分类的正例)。R 则呈下降趋势, 特别是在阈值从 0.2 提升到 0.3 时急剧下降, 表明较高阈值会导致更多的假阴性, 但在适当范围内可能有所回升。F1 值在较低阈值时保持较高值, 0.05 时达到 0.846。总体来看, 低阈值 (0.05-0.1) 适用于对错误容忍

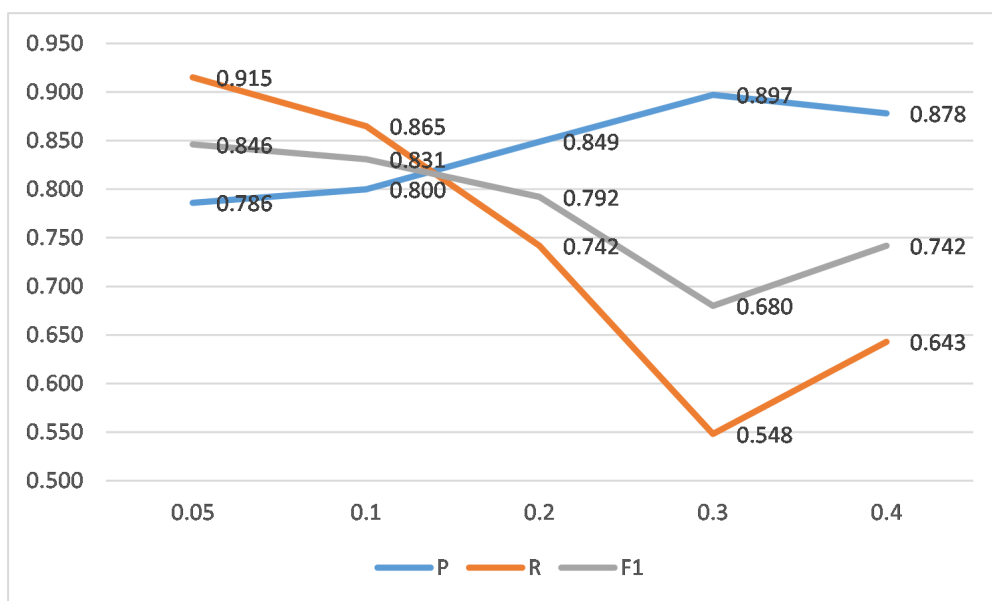


图 5-2 超参数实验

Figure 5-2 Hyperparameter sensitivity

度较高的场景，中等阈值（0.2）在精确率和召回率之间相对均衡，而较高阈值（0.3-0.4）适用于对 Precision 要求更高的任务。实验结果表明，阈值选择取决于具体任务需求。若更关注 Precision，可选择较高阈值，而若希望 Recall 更高，则需降低阈值以减少漏检。

## 5.5 本章小结

本节提出了一种基于主题相关性与逻辑推理的检测方法——ILCV-LTD，通过构建多维度检测框架，将文本的主题一致性与逻辑合理性相结合，以提升 AIGC 文本检测的精准度和鲁棒性。该方法首先利用自然语言到一阶谓词逻辑（FOL）的转换模块，将文本形式化为可验证的逻辑表达式，随后借助 FOL 求解器进行推理验证，从而识别文本中的逻辑错误。此外，为了减少转换过程中的误差累积，ILCV-LTD 引入了针对性的错误判断策略，以优化逻辑验证过程并提高检测的可靠性。实验结果表明，该方法在 AIGC 文本检测任务中表现出较高的准确性和稳定性，尤其在涉及主题相关性分析和逻辑推理验证的场景下，相较于现有方法具有显著优势。尽管错误判断策略在一定程度上减少了错误传播，但无法彻底消除 NL2FOL 转换过程的准确性，且 FOL 求解器在实际求解过程中的效果并不好。因此未来可探索更优化的 FOL 转换方法，例如结合强化学习技术，以进一步提升 NL2FOL 转换的可靠性和检测能力，以及研发出更加高效的 FOL 求解方法。



## 第6章 总结与展望

### 6.1 本文工作总结

自然语言处理领域中，形式语言和功能性语言在认知机制上的差异显著。形式语言区域主要负责语法和语义的处理，而功能性语言区域则涉及更高阶的认知功能，如逻辑推理、数学运算和抽象思维。然而，当前主流的大语言模型在形式语言的处理上已表现优异，但在数学推理和逻辑推理方面仍然存在明显局限性。本文围绕对形式语言和功能性语言文本检测这两个方面，研究了基于分类和一致性验证的 AIGC 文本检测方法。主要关注如何准确地识别和区分 AIGC 生成文本与人类文本在形式语言文本和逻辑语言文本上的差异。一个重要的观察是，LLM 生成的文本不仅在独立的统计特性上存在相似性，还在其关联结构上具有内在共性。相同或相似 LLMs 生成的文本由于共享的生成模式，在统计空间中往往会聚集在一起。然而，现有检测方法大多聚焦于独立分析单个文本，未能充分利用这些高阶结构依赖关系。针对这个问题，本文提出了基于结构信息熵的方法，通过建模文本间的相似关系，增强 AIGC 文本检测的能力。AIGC 生成文本的真实性和逻辑性问题比较普遍，特别是在涉及数学运算和逻辑推理的文本中，现有检测方法往往难以精准识别其中的数学推理错误或逻辑矛盾。针对这一问题，本文提出了基于数学运算一致性验证的检测方法以及融合逻辑推理一致性验证的检测方法，关注 AIGC 内容和人类文本在数学运算和逻辑推理上的出错率差异。通过这些方法，本文不仅解决了现有技术在处理复杂文本时的局限性，还为 AIGC 文本检测领域提供了新的思路和解决方案。

本文围绕“基于分类和一致性验证的 AIGC 文本检测方法”展开研究，探讨了现有 AIGC 检测方法在形式性语言文本和功能性语言文本检测方面的挑战，特别是在数学运算和逻辑推理文本上的检测挑战。在研究中，提出了三个关键研究点：

- (1) 基于结构信息熵的 AIGC 文本检测方法；
- (2) 基于数学运算一致性验证的 AIGC 文本检测方法；
- (3) 融合逻辑推理一致性验证的 AIGC 文本检测方法。

这些方法旨在解决 AIGC 生成文本的多样性和复杂性带来的检测挑战，为该领域的研究提供了新思路。

以下是本文取得的主要成果：

- **基于结构信息熵的 AIGC 文本检测方法** 提出了 SILTD，一种基于结构信息熵的分类方法，该方法通过构建一个文本多关系图，将文本的特征表示转换为图中的节点，然后利用结构信息理论，通过最小化结构熵的方法，可以将文本

节点划分成不同的簇。最小化结构熵的过程有助于减少噪声和异常值对聚类结果的影响，从而得到更稳定的簇。最后，通过聚类结果的质量和簇内文本之间的相似度，进行簇内标签推断得到最终的分类结果。在创作和问答数据集上，对 6 个大模型的平均检测性能相较于其他方法最高提升了 5%，证明了算法对非形式语言文本检测的有效性。

- **基于数学运算一致性验证的 AIGC 文本检测方法** 提出了 MCCV-LTD，一种基于数学运算一致性验证的检测方法，该方法通过将文本转换为数学表达式，并利用数学运算规则验证其逻辑合理性，有效识别文本中的数学运算错误。本方法不仅限于简单等式，还能处理更为复杂的数学运算，如嵌套的加减乘除和数值大小关系。为提升模型在数学运算文本上的泛化能力，引入对比学习技术，通过构建正负样本对，增强模型对细微差异的区分能力。在新建数据集上相比于基线方法提升了 XXX，证明了算法在检测功能性语言文本——数学运算时的有效性。

- **融合逻辑推理一致性验证的 AIGC 文本检测方法** 提出了 ILCV-LTD，一种基于主题相关性与逻辑推理的检测方法。该方法构建多维度检测框架，将文本的主题相关性与逻辑合理性相结合。方法包括两阶段的逻辑检测：（1）设计自然语言到一阶谓词逻辑（FOL）转换模块，将文本形式化为可验证的逻辑表达式；（2）利用 FOL 求解器验证推理正确性，识别逻辑错误。此外，提出针对转换过程的错误判断策略，以减少错误传播影响，提升检测性能。实验结果表明，该方法在 AIGC 文本检测任务中表现出较高的准确性和稳定性，尤其在涉及主题相关性分析和逻辑推理验证的场景下，显著优于现有检测方法。

## 6.2 未来工作展望

- **更高效地文本表示方法**：当前基于结构信息熵的 AIGC 文本检测方法主要依赖于文本的全局特征，难以有效捕捉文本局部的相似性。未来可以探索更高效的文本表示方法，例如结合局部特征的层次化表示或引入图神经网络（GNN）来更好地捕捉文本的局部和全局结构信息。此外，可以研究如何利用预训练语言模型的上下文嵌入来增强文本特征表示，从而提高检测方法对局部相似性的敏感度。

- **更全面的评估指标**：基于数学运算一致性验证的 AIGC 文本检测方法目前只是从较为简单的数学公式出发进行检测，当面临更加复杂的数学运算或推理的时候，检测效果可能会下降，同时面临比较复杂的表达式时 NL2MATH 的过程中可能会出错，导致检测准确率下降。此外，NL2FOL 转换过程中的错误也可能导致检测结果不准确。未来可以设计更全面的评估指标，例如引入文本多样性指标、逻辑复杂度指标以及跨数据集的泛化能力评估。同时，可以探索基于置

信度的评估方法，为检测结果提供更可靠的量化指标。

- 更准确快捷的 NL2FOL 方法：融合逻辑推理一致性验证的 AIGC 文本检测方法虽然提出了错误判断策略以减少错误传播，但无法彻底消除 NL2FOL 转换过程的准确性，且 FOL 求解器在实际求解过程中的效果并不好。因此未来可探索更优化的 FOL 转换方法，例如结合强化学习技术，以进一步提升 NL2FOL 转换的可靠性和检测能力，以及研发出更加高效的 FOL 求解方法。



## 参考文献

- [1] Dathathri S, See A, Ghaisas S, et al. Scalable watermarking for identifying large language model outputs [J]. *Nature*, 2024, 634(8035): 818-823.
- [2] Abdelnabi S, Fritz M. Adversarial watermarking transformer: Towards tracing text provenance with data hiding [C]//42nd IEEE Symposium on Security and Privacy. 2021.
- [3] Mitchell E, Lee Y, Khazatsky A, et al. Detectgpt: Zero-shot machine-generated text detection using probability curvature [C]//International Conference on Machine Learning. PMLR, 2023: 24950-24962.
- [4] Bao G, Zhao Y, Teng Z, et al. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature [J]. *arXiv preprint arXiv:2310.05130*, 2023.
- [5] Bhattacharjee A, Kumarage T, Moraffah R, et al. Conda: Contrastive domain adaptation for ai-generated text detection [J]. *arXiv preprint arXiv:2309.03992*, 2023.
- [6] Hu X, Chen P Y, Ho T Y. Radar: Robust ai-text detection via adversarial learning [J]. *Advances in Neural Information Processing Systems*, 2023, 36: 15077-15095.
- [7] Cowap A, Graham Y, Foster J. Do stochastic parrots have feelings too? improving neural detection of synthetic text via emotion recognition [J]. *arXiv preprint arXiv:2310.15904*, 2023.
- [8] Koike R, Kaneko M, Okazaki N. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples [C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 38. 2024: 21258-21266.
- [9] Vendrow J, Vendrow E, Beery S, et al. Do large language model benchmarks test reliability? [EB/OL]. 2025. <https://arxiv.org/abs/2502.03461>.
- [10] Achiam J, Adler S, Agarwal S, et al. Gpt-4 technical report [J]. *arXiv preprint arXiv:2303.08774*, 2023.
- [11] Anthropic. Model card and evaluations for claude models [J]. 2023.
- [12] Chowdhery A, Narang S, Devlin J, et al. Palm: Scaling language modeling with pathways [J]. *Journal of Machine Learning Research*, 2023, 24(240): 1-113.
- [13] Hanley H W, Durumeric Z. Machine-made media: Monitoring the mobilization of machine-generated articles on misinformation and mainstream news websites [C]//Proceedings of the International AAAI Conference on Web and Social Media: volume 18. 2024: 542-556.
- [14] Lin S, Hilton J, Evans O. Truthfulqa: Measuring how models mimic human falsehoods [J]. *arXiv preprint arXiv:2109.07958*, 2021.
- [15] Ayooobi N, Shahriar S, Mukherjee A. The looming threat of fake and llm-generated linkedin profiles: Challenges and opportunities for detection and prevention [C]//Proceedings of the 34th ACM Conference on Hypertext and Social Media. 2023: 1-10.
- [16] Mirsky Y, Demontis A, Kotak J, et al. The threat of offensive ai to organizations [J]. *Computers & Security*, 2023, 124: 103006.

- [17] Kasneci E, Seßler K, Küchemann S, et al. Chatgpt for good? on opportunities and challenges of large language models for education [J]. *Learning and individual differences*, 2023, 103: 102274.
- [18] Li X, Yu P, Zhou C, et al. Self-alignment with instruction backtranslation [J]. *arXiv preprint arXiv:2308.06259*, 2023.
- [19] Taori R, Gulrajani I, Zhang T, et al. Stanford alpaca: an instruction-following llama model (2023) [J]. URL [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca), 2023, 1(9).
- [20] Alemohammad S, Casco-Rodriguez J, Luzi L, et al. Self-consuming generative models go mad [J]. *arXiv preprint arXiv:2307.01850*, 2023.
- [21] Epstein Z, Hertzmann A, of Human Creativity I, et al. Art and the science of generative ai [J]. *Science*, 2023, 380(6650): 1110-1111.
- [22] Price G, Sakellarios M D. The effectiveness of free software for detecting ai-generated writing [J]. *International Journal of Teaching, Learning and Education*, 2023, 2(6).
- [23] Walters W H. The effectiveness of software designed to detect ai-generated writing: A comparison of 16 ai text detectors [J]. *Open Information Science*, 2023, 7(1): 20220158.
- [24] Weber-Wulff D, Anohina-Naumeca A, Bjelobaba S, et al. Testing of detection tools for ai-generated text [J]. *International Journal for Educational Integrity*, 2023, 19(1): 26.
- [25] Uchendu A, Ma Z, Le T, et al. Turingbench: A benchmark environment for turing test in the age of neural text generation [J]. *arXiv preprint arXiv:2109.13296*, 2021.
- [26] Clark E, August T, Serrano S, et al. All that's' human'is not gold: Evaluating human evaluation of generated text [J]. *arXiv preprint arXiv:2107.00061*, 2021.
- [27] Dou Y, Forbes M, Koncel-Kedziorski R, et al. Is gpt-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text [J]. *arXiv preprint arXiv:2107.01294*, 2021.
- [28] Soni M, Wade V. Comparing abstractive summaries generated by chatgpt to real summaries through blinded reviewers and text classification algorithms [J]. *arXiv preprint arXiv:2303.17650*, 2023.
- [29] Ippolito D, Duckworth D, Callison-Burch C, et al. Automatic detection of generated text is easiest when humans are fooled [J]. *arXiv preprint arXiv:1911.00650*, 2019.
- [30] Stokel-Walker C, Van Noorden R. What chatgpt and generative ai mean for science [J]. *Nature*, 2023, 614(7947): 214-216.
- [31] Porsdam Mann S, Earp B D, Nyholm S, et al. Generative ai entails a credit-blame asymmetry [J]. *Nature Machine Intelligence*, 2023, 5(5): 472-475.
- [32] Shevlane T, Farquhar S, Garfinkel B, et al. Model evaluation for extreme risks [J]. *arXiv preprint arXiv:2305.15324*, 2023.
- [33] Pu J, Sarwar Z, Abdullah S M, et al. Deepfake text detection: Limitations and opportunities [C]//2023 IEEE symposium on security and privacy (SP). IEEE, 2023: 1613-1630.
- [34] Jawahar G, Abdul-Mageed M, Lakshmanan L V. Automatic detection of machine generated text: A critical survey [J]. *arXiv preprint arXiv:2011.01314*, 2020.
- [35] Uchendu A, Le T, Lee D. Attribution and obfuscation of neural text authorship: A data mining perspective [J]. *ACM SIGKDD Explorations Newsletter*, 2023, 25(1): 1-18.

- [36] Gu C, Huang C, Zheng X, et al. Watermarking pre-trained language models with backdooring [J]. arXiv preprint arXiv:2210.07543, 2022.
- [37] Lucas E, Havens T. Gpts don' t keep secrets: Searching for backdoor watermark triggers in autoregressive language models [C]//Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023). 2023: 242-248.
- [38] Tang R, Feng Q, Liu N, et al. Did you train on my dataset? towards public dataset protection with cleanlabel backdoor watermarking [J]. ACM SIGKDD Explorations Newsletter, 2023, 25(1): 43-53.
- [39] Kirchenbauer J, Geiping J, Wen Y, et al. A watermark for large language models [C]//International Conference on Machine Learning. PMLR, 2023: 17061-17084.
- [40] Lee T, Hong S, Ahn J, et al. Who wrote this code? watermarking for code generation [J]. arXiv preprint arXiv:2305.15060, 2023.
- [41] Kirchenbauer J, Geiping J, Wen Y, et al. On the reliability of watermarks for large language models [J]. arXiv preprint arXiv:2306.04634, 2023.
- [42] Liu A, Pan L, Hu X, et al. A semantic invariant robust watermark for large language models [J/OL]. ArXiv, 2023, abs/2310.06356. <https://api.semanticscholar.org/CorpusID:263830310>.
- [43] Liu A, Pan L, Xuming H, et al. A private watermark for large language models [EB/OL]. 2023. DOI: [10.48550/arXiv.2307.16230](https://doi.org/10.48550/arXiv.2307.16230).
- [44] Kuditipudi R, Thickstun J, Hashimoto T, et al. Robust distortion-free watermarks for language models [EB/OL]. 2024. <https://arxiv.org/abs/2307.15593>.
- [45] Hou A B, Zhang J, He T, et al. Semstamp: A semantic watermark with paraphrastic robustness for text generation [EB/OL]. 2024. <https://arxiv.org/abs/2310.03991>.
- [46] Por L Y, Wong K, Chee K O. Unispach: A text-based data hiding method using unicode space characters [J/OL]. Journal of Systems and Software, 2012, 85(5): 1075-1082. <https://www.sciencedirect.com/science/article/pii/S0164121211003177>. DOI: <https://doi.org/10.1016/j.jss.2011.12.023>.
- [47] Rizzo S G, Bertini F, Montesi D. Content-preserving text watermarking through unicode homoglyph substitution [C/OL]//IDEAS '16: Proceedings of the 20th International Database Engineering & Applications Symposium. New York, NY, USA: Association for Computing Machinery, 2016: 97–104. <https://doi.org/10.1145/2938503.2938510>.
- [48] Topkara U, Topkara M, Atallah M J. The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions [C/OL]//MMSec '06: Proceedings of the 8th Workshop on Multimedia and Security. New York, NY, USA: Association for Computing Machinery, 2006: 164–174. <https://doi.org/10.1145/1161366.1161397>.
- [49] Fellbaum C. WordNet: An Electronic Lexical Database [M/OL]. The MIT Press, 1998. <https://doi.org/10.7551/mitpress/7287.001.0001>.
- [50] Yang X, Zhang J, Chen K, et al. Tracing text provenance via context-aware lexical substitution [EB/OL]. 2021. <https://arxiv.org/abs/2112.07873>.
- [51] Munyer T, Tanvir A, Das A, et al. Deeptextmark: A deep learning-driven text watermarking approach for identifying large language model generated text [J]. IEEE Access, 2024.
- [52] Yoo K, Ahn W, Jang J, et al. Robust multi-bit natural language watermarking through invariant features [EB/OL]. 2023. <https://arxiv.org/abs/2305.01904>.

- [53] Yang X, Chen K, Zhang W, et al. Watermarking text generated by black-box language models [EB/OL]. 2023. <https://arxiv.org/abs/2305.08883>.
- [54] Zhang R, Hussain S S, Neekhara P, et al. Remark-llm: A robust and efficient watermarking framework for generative large language models [EB/OL]. 2024. <https://arxiv.org/abs/2310.12362>.
- [55] Solaiman I, Brundage M, Clark J, et al. Release strategies and the social impacts of language models [J]. arXiv preprint arXiv:1908.09203, 2019.
- [56] Gehrmann S, Strobel H, Rush A M. Gltr: Statistical detection and visualization of generated text [J]. arXiv preprint arXiv:1906.04043, 2019.
- [57] Su J, Zhuo T Y, Wang D, et al. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text [J]. arXiv preprint arXiv:2306.05540, 2023.
- [58] Lavergne T, Urvoy T, Yvon F. Detecting fake content with relative entropy scoring. [C]// 2008.
- [59] Vasilatos C, Alam M, Rahwan T, et al. Howkgpt: Investigating the detection of chatgpt-generated university student homework through context-aware perplexity analysis [EB/OL]. 2023. <https://arxiv.org/abs/2305.18226>.
- [60] Wu K, Pang L, Shen H, et al. Llm-det: A third party large language models generated text detection tool [J]. arXiv preprint arXiv:2305.15004, 2023.
- [61] Venkatraman S, Uchendu A, Lee D. Gpt-who: An information density-based machine-generated text detector [J]. arXiv preprint arXiv:2310.06202, 2023.
- [62] Verma V, Fleisig E, Tomlin N, et al. Ghostbuster: Detecting text ghostwritten by large language models [J]. arXiv preprint arXiv:2305.15047, 2023.
- [63] Miao Y, Gao H, Zhang H, et al. Efficient detection of llm-generated texts with a bayesian surrogate model [J]. arXiv preprint arXiv:2305.16617, 2023.
- [64] Tulchinskii E, Kuznetsov K, Kushnareva L, et al. Intrinsic dimension estimation for robust detection of ai-generated texts [J]. Advances in Neural Information Processing Systems, 2024, 36.
- [65] Yang X, Cheng W, Wu Y, et al. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text [J]. arXiv preprint arXiv:2305.17359, 2023.
- [66] Zhu B, Yuan L, Cui G, et al. Beat llms at their own game: Zero-shot llm-generated text detection via querying chatgpt [C]//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. 2023: 7470-7483.
- [67] Mao C, Vondrick C, Wang H, et al. Raidar: generative ai detection via rewriting [J]. arXiv preprint arXiv:2401.12970, 2024.
- [68] Yu X, Qi Y, Chen K, et al. Gpt paternity test: Gpt generated text detection with gpt genetic inheritance [J]. CoRR, 2023.
- [69] Quidwai M A, Li C, Dube P. Beyond black box ai-generated plagiarism detection: From sentence to document level [J]. arXiv preprint arXiv:2306.08122, 2023.
- [70] Guo Z, Yu S. Authentigpt: Detecting machine-generated text via black-box language models denoising [J]. arXiv preprint arXiv:2311.07700, 2023.

- [71] Aich A, Bhattacharya S, Parde N. Demystifying neural fake news via linguistic feature-based interpretation [C]//Proceedings of the 29th International Conference on Computational Linguistics. 2022: 6586-6599.
- [72] Horne B, Adali S. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news [C]//Proceedings of the international AAAI conference on web and social media: volume 11. 2017: 759-766.
- [73] McCarthy P M. An assessment of the range and usefulness of lexical diversity measures and the potential of the measure of textual, lexical diversity (mtld) [D]. The University of Memphis, 2005.
- [74] Lee B W, Jang Y S, Lee J H J. Pushing on text readability assessment: A transformer meets handcrafted linguistic features [J]. arXiv preprint arXiv:2109.12258, 2021.
- [75] Baccianella S, Esuli A, Sebastiani F, et al. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. [C]//Lrec: volume 10. Valletta, 2010: 2200-2204.
- [76] Lu Y, Liu Q, Dai D, et al. Unified structure generation for universal information extraction [J]. arXiv preprint arXiv:2203.12277, 2022.
- [77] Jin Z, Chen Y, Sui D, et al. Cogie: An information extraction toolkit for bridging texts and cognet [C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations. 2021: 92-98.
- [78] Shah A, Ranka P, Dedhia U, et al. Detecting and unmasking ai-generated texts through explainable artificial intelligence using stylistic features [J]. International Journal of Advanced Computer Science and Applications, 2023, 14(10).
- [79] Corizzo R, Leal-Arenas S. A deep fusion model for human *vs.* machine-generated essay classification [C]//2023 International Joint Conference on Neural Networks (IJCNN). IEEE, 2023: 1-10.
- [80] Mindner L, Schlippe T, Schaaff K. Classification of human-and ai-generated texts: Investigating features for chatgpt [C]//International Conference on Artificial Intelligence in Education Technology. Springer, 2023: 152-170.
- [81] Schaaff K, Schlippe T, Mindner L. Classification of human-and ai-generated texts for english, french, german, and spanish [J]. arXiv preprint arXiv:2312.04882, 2023.
- [82] Schuster T, Schuster R, Shah D J, et al. The limitations of stylometry for detecting machine-generated fake news [J]. Computational Linguistics, 2020, 46(2): 499-510.
- [83] Crothers E, Japkowicz N, Viktor H, et al. Adversarial robustness of neural-statistical features in detection of generative transformers [C]//2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022: 1-8.
- [84] Li L, Wang P, Ren K, et al. Origin tracing and detecting of llms [J]. arXiv preprint arXiv:2304.14072, 2023.
- [85] Wang P, Li L, Ren K, et al. Seqxgpt: Sentence-level ai-generated text detection [J]. arXiv preprint arXiv:2310.08903, 2023.
- [86] Guo B, Zhang X, Wang Z, et al. How close is chatgpt to human experts? comparison corpus, evaluation, and detection [J]. arXiv preprint arXiv:2301.07597, 2023.

- [87] Liu Y, Zhang Z, Zhang W, et al. Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models [J]. arXiv preprint arXiv:2304.07666, 2023.
- [88] Chen Y, Kang H, Zhai V, et al. Gpt-sentinel: Distinguishing human and chatgpt generated content [J]. arXiv preprint arXiv:2305.07969, 2023.
- [89] Wang Z, Cheng J, Cui C, et al. Implementing bert and fine-tuned roberta to detect ai generated news by chatgpt [J]. arXiv preprint arXiv:2306.07401, 2023.
- [90] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding [J]. arXiv preprint arXiv:1810.04805, 2018.
- [91] Conneau A. Unsupervised cross-lingual representation learning at scale [J]. arXiv preprint arXiv:1911.02116, 2019.
- [92] Jiao X, Yin Y, Shang L, et al. Tinybert: Distilling bert for natural language understanding [J]. arXiv preprint arXiv:1909.10351, 2019.
- [93] Liu X, Zhang Z, Wang Y, et al. Coco: Coherence-enhanced machine-generated text detection under data limitation with contrastive learning [J]. arXiv preprint arXiv:2212.10341, 2022.
- [94] Yang L, Jiang F, Li H, et al. Is chatgpt involved in texts? measure the polish ratio to detect chatgpt-generated text [J]. APSIPA Transactions on Signal and Information Processing, 2023, 13(2).
- [95] Shi Z, Wang Y, Yin F, et al. Red teaming language model detectors with language models [J]. Transactions of the Association for Computational Linguistics, 2024, 12: 174-189.
- [96] He X, Shen X, Chen Z, et al. Mgtbench: Benchmarking machine-generated text detection [J]. arXiv preprint arXiv:2303.14822, 2023.
- [97] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks [J]. Communications of the ACM, 2020, 63(11): 139-144.
- [98] Yasunaga M, Liang P. Break-it-fix-it: Unsupervised learning for program repair [C]// International conference on machine learning. PMLR, 2021: 11941-11952.
- [99] Tu S, Li C, Yu J, et al. Chatlog: Recording and analyzing chatgpt across time [J]. arXiv preprint arXiv:2304.14106, 2023.
- [100] Uchendu A, Le T, Lee D. Toproberta: Topology-aware authorship attribution of deepfake texts [J]. arXiv preprint arXiv:2309.12934, 2023.
- [101] Kumarage T, Bhattacharjee A, Padejski D, et al. J-guard: Journalism guided adversarially robust detection of ai-generated news [J]. arXiv preprint arXiv:2309.03164, 2023.
- [102] Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback [J]. Advances in neural information processing systems, 2022, 35: 27730-27744.
- [103] Meta. The llama 3 herd of models [J/OL]. 2024. <https://ai.meta.com>.
- [104] Dehouche N. Plagiarism in the age of massive generative pre-trained transformers (gpt-3) [J]. Ethics in Science and Environmental Politics, 2021, 21: 17-23.
- [105] Shu K, Wang S, Lee D, et al. Mining disinformation and fake news: Concepts, methods, and recent advancements [J]. Disinformation, misinformation, and fake news in social media: Emerging research challenges and opportunities, 2020: 1-19.
- [106] Stiff H, Johansson F. Detecting computer-generated disinformation [J]. International Journal of Data Science and Analytics, 2022, 13(4): 363-383.

- [107] Zellers R, Holtzman A, Rashkin H, et al. Defending against neural fake news [J]. *Advances in neural information processing systems*, 2019, 32.
- [108] Baki S, Verma R, Mukherjee A, et al. Scaling and effectiveness of email masquerade attacks: Exploiting natural language generation [C]//*Proceedings of the 2017 ACM on Asia conference on computer and communications security*. 2017: 469-482.
- [109] Giaretta A, Dragoni N. Community targeted phishing: A middle ground between massive and spear phishing through natural language generation [C]//*Proceedings of 6th International Conference in Software Engineering for Defence Applications: SEDA 2018* 6. Springer, 2020: 86-93.
- [110] Adelani D I, Mai H, Fang F, et al. Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection [C]//*Advanced information networking and applications: Proceedings of the 34th international conference on advanced information networking and applications (AINA-2020)*. Springer, 2020: 1341-1354.
- [111] Kurenkov A. Lessons from the gpt-4chan controversy [J/OL]. *The Gradient*, 2022. <https://thegradient.pub/gpt-4chan-lessons>.
- [112] Li A, Pan Y. Structural information and dynamical complexity of networks [J]. *IEEE Transactions on Information Theory*, 2016, 62(6): 3290-3339.
- [113] Cao Y, Peng H, Yu Z, et al. Hierarchical and incremental structural entropy minimization for unsupervised social event detection [C]//*Proceedings of the AAAI Conference on Artificial Intelligence: volume 38*. 2024: 8255-8264.
- [114] Brown P F, Della Pietra V J, Desouza P V, et al. Class-based n-gram models of natural language [J]. *Computational linguistics*, 1992, 18(4): 467-480.
- [115] Rosenfeld R. Two decades of statistical language modeling: Where do we go from here? [J]. *Proceedings of the IEEE*, 2000, 88(8): 1270-1278.
- [116] Stolcke A, et al. Srilm-an extensible language modeling toolkit. [C]//*Interspeech: volume 2002*. 2002: 2002.
- [117] Bengio Y, Ducharme R, Vincent P. A neural probabilistic language model [J]. *Advances in neural information processing systems*, 2000, 13.
- [118] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model. [C]//*Interspeech: volume 2*. Makuhari, 2010: 1045-1048.
- [119] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks [J]. *Advances in neural information processing systems*, 2014, 27.
- [120] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [J]. *Advances in neural information processing systems*, 2017, 30.
- [121] Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners [J]. *OpenAI blog*, 2019, 1(8): 9.
- [122] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners [J]. *Advances in neural information processing systems*, 2020, 33: 1877-1901.
- [123] Touvron H, Lavril T, Izacard G, et al. Llama: Open and efficient foundation language models (2023) [J]. *arXiv preprint arXiv:2302.13971*, 2023.

- [124] Chakraborty S, Bedi A S, Zhu S, et al. On the possibilities of ai-generated text detection [J]. arXiv preprint arXiv:2304.04736, 2023.
- [125] Badaskar S, Agarwal S, Arora S. Identifying real or fake articles: Towards better language modeling [C]//Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II. 2008.
- [126] Liu Y, Ott M, Goyal N, et al. Roberta: A robustly optimized bert pretraining approach [J]. arXiv preprint arXiv:1907.11692, 2019.
- [127] Zeng G, Peng H, Li A, et al. Scalable semi-supervised clustering via structural entropy with different constraints [J]. IEEE Transactions on Knowledge and Data Engineering, 2024.
- [128] Zeng X, Peng H, Li A. Adversarial socialbots modeling based on structural information principles [C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 38. 2024: 392-400.
- [129] Wu J, Chen X, Xu K, et al. Structural entropy guided graph hierarchical pooling [C]//International conference on machine learning. PMLR, 2022: 24017-24030.
- [130] Zeng X, Peng H, Li A, et al. Hierarchical state abstraction based on structural information principles [J]. arXiv preprint arXiv:2304.12000, 2023.
- [131] Zeng X, Peng H, Su D, et al. Effective reinforcement learning based on structural information principles [J]. arXiv preprint arXiv:2404.09760, 2024.
- [132] Zeng X, Peng H, Li A. Effective and stable role-based multi-agent collaboration by structural information principles [C]//Proceedings of the AAAI conference on artificial intelligence: volume 37. 2023: 11772-11780.
- [133] Peng H, Zhang J, Huang X, et al. Unsupervised social bot detection via structural information theory [J]. ACM Transactions on Information Systems, 2024, 42(6): 1-42.
- [134] Huang X, Peng H, Sun L, et al. Structural entropy guided probabilistic coding [J]. arXiv preprint arXiv:2412.08841, 2024.
- [135] Yang R, Peng H, Li A, et al. Hierarchical abstracting graph kernel [J]. IEEE Transactions on Knowledge and Data Engineering, 2024.
- [136] Yang R, Peng H, Liu C, et al. Incremental measurement of structural entropy for dynamic graphs [J]. Artificial Intelligence, 2024, 334: 104175.
- [137] Wang S, Wang P, Wu B, et al. Structural entropy minimization combining graph representation for money laundering identification [J]. International Journal of Machine Learning and Cybernetics, 2024: 1-18.
- [138] Ng M K P, Li X, Ye Y. Multirank: co-ranking for objects and relations in multi-relational data [C]//Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. 2011: 1217-1225.
- [139] Houvardas J, Stamatatos E. N-gram feature selection for authorship identification [C]//International conference on artificial intelligence: Methodology, systems, and applications. Springer, 2006: 77-86.
- [140] Gao T, Yao X, Chen D. Simcse: Simple contrastive learning of sentence embeddings [J]. arXiv preprint arXiv:2104.08821, 2021.
- [141] He P, Liu X, Gao J, et al. Deberta: Decoding-enhanced bert with disentangled attention [J]. arXiv preprint arXiv:2006.03654, 2020.

- [142] Raiyan S R, Faiyaz M N, Kabir S M J, et al. Math word problem solving by generating linguistic variants of problem statements [J]. arXiv preprint arXiv:2306.13899, 2023.
- [143] Reimers N. Sentence-bert: Sentence embeddings using siamese bert-networks [J]. arXiv preprint arXiv:1908.10084, 2019.
- [144] Koncel-Kedziorski R, Roy S, Amini A, et al. MAWPS: A math word problem repository [C/OL]//Knight K, Nenkova A, Rambow O. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, California: Association for Computational Linguistics, 2016: 1152-1157. <https://aclanthology.org/N16-1136/>. DOI: 10.18653/v1/N16-1136.
- [145] Patel A, Bhattamishra S, Goyal N. Are nlp models really able to solve simple math word problems? [EB/OL]. 2021. <https://arxiv.org/abs/2103.07191>.
- [146] Raiyan S R, Faiyaz M N, Kabir S M J, et al. Math word problem solving by generating linguistic variants of problem statements [EB/OL]. 2023. <https://arxiv.org/abs/2306.13899>.
- [147] Fuggitti F, Chakraborti T. NI2ltl—a python package for converting natural language (nl) instructions to linear temporal logic (ltl) formulas [C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 37. 2023: 16428-16430.
- [148] Olausson T X, Gu A, Lipkin B, et al. Linc: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers [J]. arXiv preprint arXiv:2310.15164, 2023.
- [149] Yang Y, Xiong S, Payani A, et al. Harnessing the power of large language models for natural language to first-order logic translation [J]. arXiv preprint arXiv:2305.15541, 2023.
- [150] Aly R, Strong M, Vlachos A. Qa-natver: Question answering for natural logic-based fact verification [J]. arXiv preprint arXiv:2310.14198, 2023.
- [151] Lalwani A, Chopra L, Hahn C, et al. NI2fol: Translating natural language to first-order logic for logical fallacy detection [J]. arXiv preprint arXiv:2405.02318, 2024.
- [152] De Moura L, Bjørner N. Z3: An efficient smt solver [C]//International conference on Tools and Algorithms for the Construction and Analysis of Systems. Springer, 2008: 337-340.
- [153] Barbosa H, Barrett C, Brain M, et al. cvc5: A versatile and industrial-strength smt solver [C]//International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Springer, 2022: 415-442.
- [154] Yehuda Y, Malkiel I, Barkan O, et al. Interrogatellm: Zero-resource hallucination detection in llm-generated answers [C]//Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2024: 9333-9347.
- [155] Jin Z, Lalwani A, Vaidhya T, et al. Logical fallacy detection [J]. arXiv preprint arXiv:2202.13758, 2022.
- [156] Bowman S R, Angeli G, Potts C, et al. A large annotated corpus for learning natural language inference [J]. arXiv preprint arXiv:1508.05326, 2015.



## 致 谢

此处填写致谢。

2025 年 6 月



## 作者简历及攻读学位期间发表的学术论文与其他相关学术成果

### 作者简历：

2017 年 09 月——2021 年 06 月，在苏州大学计算机科学与技术院获得学士学位。

2022 年 09 月——2025 年 06 月，在中国科学院计算技术研究所攻读硕士学位。

### 已发表（或正式接受）的学术论文：

- (1) 2025, Jing Yang, Shi Wang, Kangli Zi, et al. SILTD: Structural Information for LLM-generated Text Detection. International journal of Machine Learning and Cybernetics.

### 参加的研究项目及获奖情况：

