Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心

从生成机制探索机生文本检测新方法

硕士研究生 邢倚康

2026年1月3日

问题回溯

- 总结反思
  - 所选算法创新性不足，启发性不足
  - 讲解流畅性需要提升
- 相关内容
  - 2025.01.05 刘佳《人工智能生成内容检测》
  - 2023.08.20 杨宗源《文本生成中的幻觉》

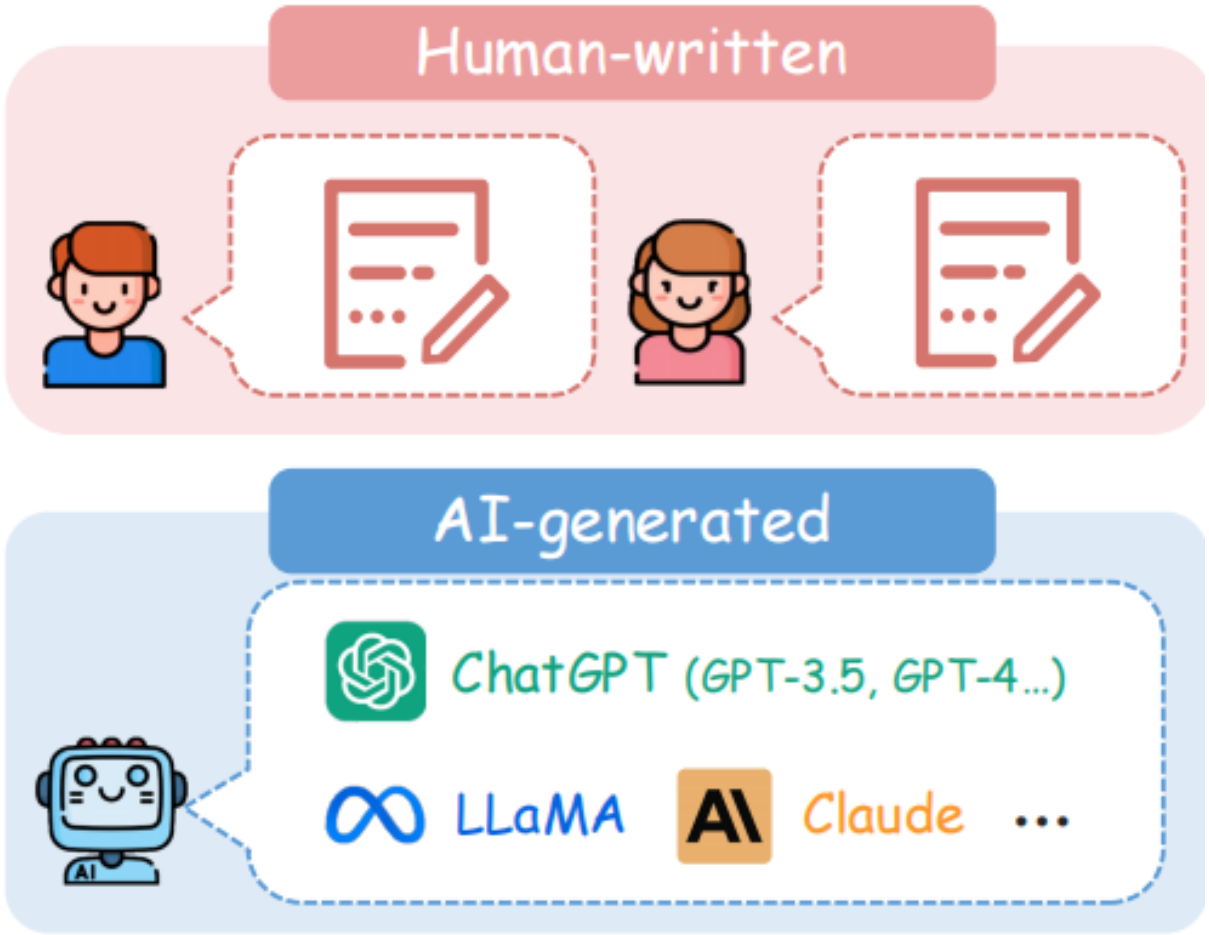
内容提要

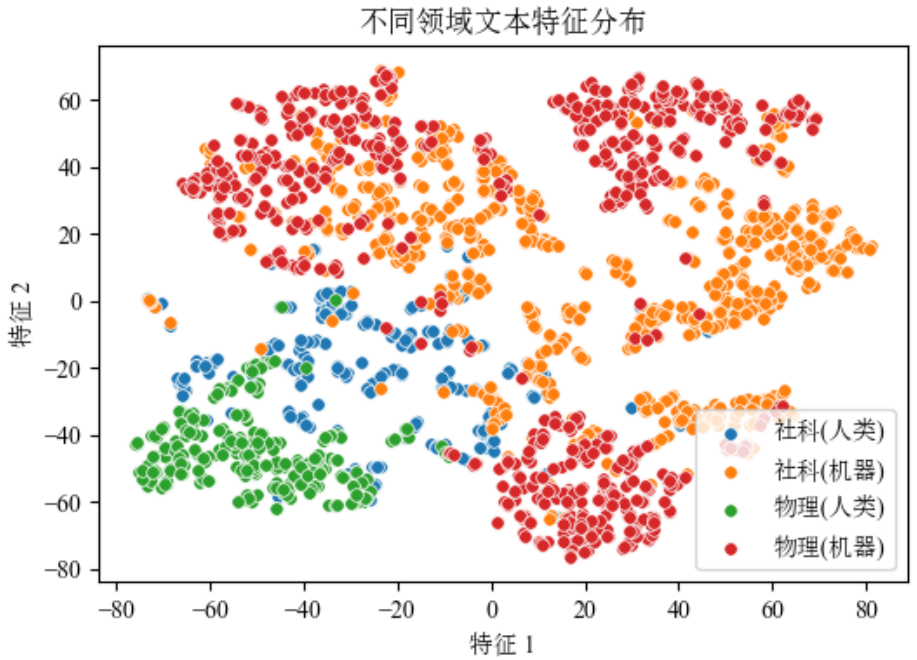
- 预期收获
- 内容引入
- 内涵解析与研究目标
- 研究背景与意义
- 研究历史与现状
- 知识基础
- 算法原理
  - BISCOPE
  - M-RangeDetector
- 特点总结与未来展望
- 参考文献

预期收获

- 预期收获
  - 掌握机器生成文本检测的基本概念与特点
  - 理解机器生成文本检测的常见方法及原理
  - 了解机器生成文本检测的未来发展方向

内容引入





跨模型、跨领域泛化能力差

学习到的特征往往与领域词汇、模型特征强相关

如何破局?

内涵解析与研究目标

- 内涵解析
  - 机生文本检测：对大模型生成的文本进行识别，区分人类撰写文本与机器生成文本
    - 跨领域：检测方法在训练域和检测域不一致时仍能有效工作
    - 跨模型：检测方法对未见过/未知来源大模型生成文本仍具备识别能力
  - 生成机制：以Transformer为核心架构的大语言模型在文本生成过程中的内在机制，基于自回归生成，通过条件概率逐步预测下一个token

纯人类文本

纯机器文本
机器补全文本
机器润色文本
……

- 研究目标
  - 面向大模型时代的机器生成文本检测任务
  - 结合深度学习、自然语言处理等技术
  - 在跨领域、跨模型场景下精准实现机生文本检测，为学术诚信、内容审核、媒体平台治理等提供检测能力，降低虚假信息带来的风险

研究背景与意义

- 研究背景
  - 大语言模型广泛应用，机生文本已大量出现在新闻写作、社交媒体、学术写作、客服对话等场景，生成内容规模持续增长
  - 机生文本的低成本与高流畅性降低了内容生成门槛，同时也带来了学术不端、虚假信息传播等风险，迫切需要可靠的识别与治理手段
  - 现有机生文本检测方法多依赖特定模型或特定数据分布，面对模型迭代、提示词变化、领域变化时，检测性能显著下降
- 研究意义
  - 提升机生文本检测的泛化性与鲁棒性，面对真实开放场景，构建在跨模型、跨领域下仍稳定有效的检测方法
  - 支撑可信内容生态与安全治理，为学术诚信、内容审核、媒体平台治理等提供检测能力，降低虚假信息带来的风险

研究历史与现状

2017 Vaswani等提出Transformer，成为大语言模型的核心架构基础；次年，Radford等提出GPT，推动了机器生成文本快速发展

2019 Liu等提出RoBERTa，优化BERT预训练，并建立“编码器表征+二分类微调”框架，成为机器生成文本检测的常用基础框架

2023 Mitchell等提出DetectGPT，对文本做随机扰动并比较曲率特征，实现无需训练的零样本机器文本检测

2023 Verma等提出Ghostbuster，将文本输入一系列语言模型，抽取Token-level概率等特征，再对候选特征组合进行结构化搜索并训练分类器

2023 Bao等提出Fast-DetectGPT，在DetectGPT的曲率思想上采用更高效的采样策略，在保持效果的同时显著加速零样本检测

2024 Guo等提出DeTeCtive，构建多任务辅助学习+多层对比学习框架，学习细粒度风格差异，并强调对OOD与新模型的适应能力

2024 Hans等提出Binoculars，通过两种相关语言模型，以跨模型对照的方式抑制提示词等干扰，实现无需训练的机生文本检测

2024 Guo等提出BISCOPE，从生成机制出发，提出双向交叉熵度量并以其统计特征训练分类器

2025 Jiao等提出M-RangeDetector，提出“人类与LLM写作策略差异”特征，学习不同上下文范围的策略表示

- 零样本检测：
  - 在不使用标注数据、不额外训练检测器的情况下，仅基于文本本身与通用语言模型的打分/统计规律，对文本是否为机生文本进行判别
  - 常见思路：困惑度/似然得分、Token Rank分布、词频信息统计等
    - 对数似然、平均负对数似然（NLL）、困惑度（PPL）

$$\log p(x) = \sum_{t=1}^{N} \log p(x_t | x_{<t})$$

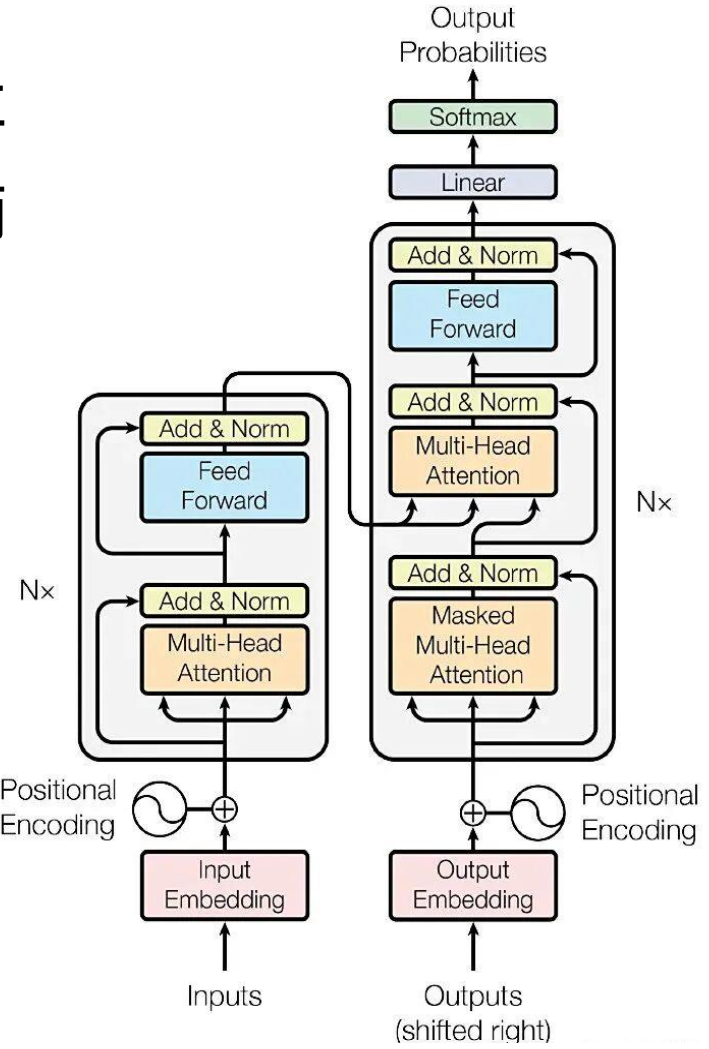
$$NLL = -\frac{1}{N} \sum_{t=1}^{N} \log p(x_t | x_{<t}) \qquad PPL = \exp(NLL)$$

  - DetectGPT（生成扰动，计算扰动样本与原始样本的对数似然差异）
  - Fast-DetectGPT、Binoculars等
  - 部署简单、成本低，但对领域迁移、模型迭代较为敏感

- 监督训练判别器：
  - 利用已标注的人类文本、机生文本数据，训练一个分类模型学习区分边界，从而对新文本进行检测
  - 常见思路
    - 利用BERT/RoBERTa 等预训练语言模型提取文本特征
    - 利用代理大语言模型（如GPT-Neo-2.7B，自回归模型，可输出token 级概率）提取特征
  - RoBERTa、DeTeCtive、Ghostbuster等
  - 检测准确率更高，但无论采用 BERT/RoBERTa 等预训练编码器提取语义/风格特征，还是采用代理大语言模型提供概率/表示特征，本质上都是在学习“由某个代理模型刻画出来的文本分布差异”
  - 检测效果高度依赖代理模型的质量与匹配程度

从生成机制探索机生文本检测新方法

- 以Transformer为核心架构
- Encoder-Decoder：两阶段，显式对齐；先理解输入，再条件生成，和标准Transformer架构一致；适合翻译、摘要等输入输出映射明确的任务，如T5大模型等
- Decoder-only：单阶段，单向累积；在生成中完成理解，适合对话、写作、推理、代码等，广泛应用于现有大模型，如GPT、LLaMA3等
  - 不再区分输入输出，完全统一为一个序列
  - 子层 1：掩码多头自注意力
  - 子层 2：前馈神经网络
- Encoder-only：单阶段，全可见；在完整上下文中学习表示，无生成阶段；适合表征学习等任务，如BERT模型



BISCOPE: AI-generated Text Detection by Checking Memorization of Preceding Tokens

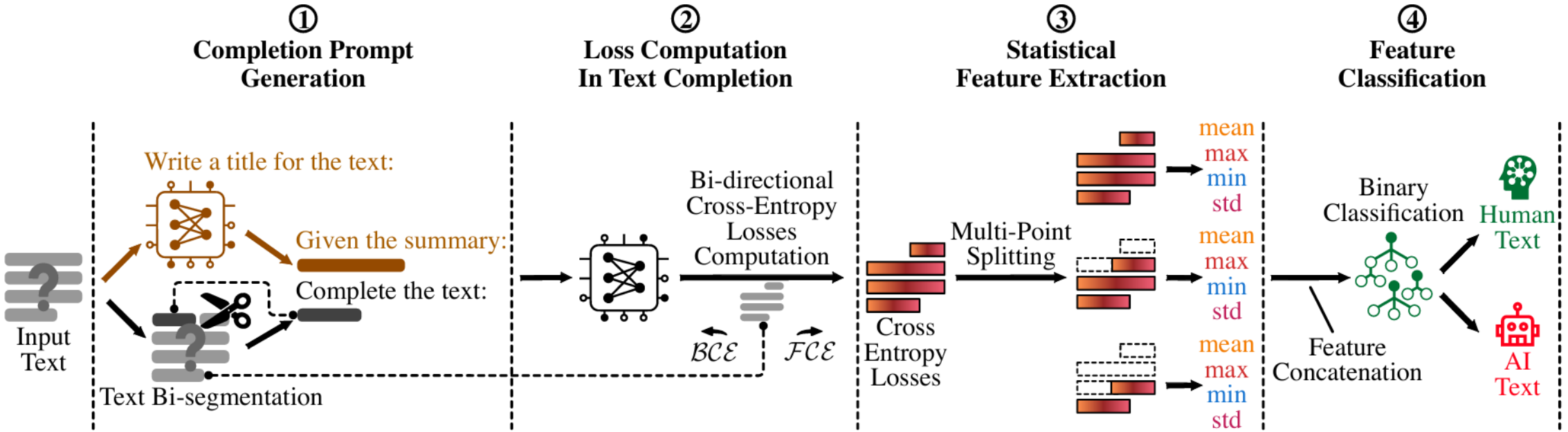
BISCOPE TIPO

T	目标	区分人类撰写文本与机器生成文本，二分类任务
I	输入	待检测文本、代理大模型（自回归模型，可输出token 级概率） 数据集：Arxiv（2100，1400）、Yelp（11740，8000）、 Creative（5840，4000）、Essay（5897，3999）、Code（ 983，656 ）
P	处理	1. 摘要生成，构建“引导续写任务” 2. 计算FCE、BCE双向交叉熵信号作为特征 3. 分段提取均值、最大、最小、标准差特征 4. 训练二分类器输出标签
O	输出	待测文本是否属于机器生成文本

P	问题	1.现有方法忽略了大模型输出时的“前文记忆”特点 2.现有方法在文本长度不一致时检测效果下降
C	条件	需要代理大模型（自回归模型，可输出token 级概率）
D	难点	1.如何体现大模型输出时的前文记忆特点 2.如何处理长度不一致的输入文本，提取特征
L	水平	2024 CCF A类

BISCOPE 算法原理

- 算法原理图
  - 摘要生成，构建“引导续写任务”
  - 计算FCE、BCE双向交叉熵信号作为特征
  - 分段提取均值、最大、最小、标准差特征
  - 训练二分类器输出标签

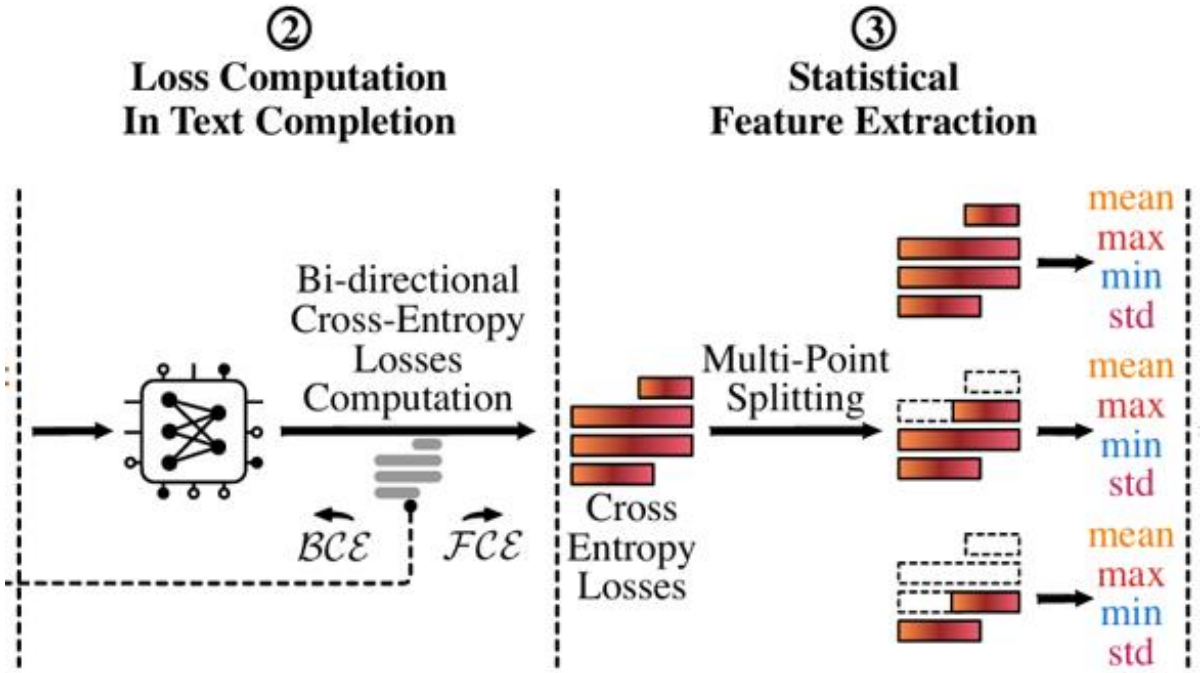


BISCOPE 创新点分析 FCE、BCE双向交叉熵信号

- 现有方法存在问题
  - 忽略了大模型输出时的“前文记忆”特点
  - 因果语言模型的训练目标是下一词预测，在每个位置输出的概率被显式优化去提升该任务，前一词的概率分布更小
  - 人类文本对大模型来说更难预测下一词，预测的概率分布更分散
- 解决方法
  - 在同一个位置的概率分布上，计算两类交叉熵数值
    - FCE：用下一个Token来计算交叉熵，捕捉预测信息
    - BCE：用紧邻的前一个Token来计算交叉熵，捕捉记忆程度
  - 原始文本$[t_1, t_2, t_3, \cdots, t_{20}]$，生成摘要$S$，续写前缀$[t_1, t_2\ ]$
    - 输入S，$[t_1, t_2\ ]$，得到预测的t_3Token分布，输出概率分布中t_2, t_3的概率
    - 输入S，$[t_1, t_2, t_3]$，得到预测的t_4Token分布，输出概率分布中t_3, t_4的概率
    - ……

BISCOPE 创新点分析 分段提取特征

- 现有方法存在问题
  - 文本长度不同导致特征维度不一致
- 解决方法
  - 固定切成n段，论文中n为10
  - 对每段内所有位置的FCE、BCE数值，分别取均值、最大值、最小值、标准差作为特征
  - 不仅能对齐长度，还能让后续分类器通过学习找到最有效的分区位置
    - 原始文本$[t_1, t_2, t_3, \cdots, t_{20}]$，生成摘要$S$，续写前缀$[t_1, t_2\ ]$
    - 分段：$[t_3, t_4\ ]$，$[t_5, t_6\ ]$，……，$[t_{19}, t_{20}\ ]$，共9段



实验设计　数据资源

- 数据集
  - 短文本自然语言：Arxiv (2100，1400)、Yelp (11740，8000)
  - 长文本自然语言：Creative (5840，4000) 、Essay (5897，3999)
  - 代码文本：Code (983，656)
- 生成文本来源
  - GPT-3.5-Turbo、GPT-4-Turbo、Claude-3-Sonnet、Claude-3-Opus、Gemini-1.0-Pro
- 对比方法

  Zero-shot Query (2023)、LogRank (2019)、 LRR (2023)、DetectGPT (2023)、RADAR (2023)、Raidar (2024)、OpenAI Detector (2019)、Binoculars (2024)、GhostBuster (2023)
- 评价指标
  - 5折交叉验证平均F1值

实验设计　对比实验

- 评估BISCOPE在不同数据集上的表现

	Method	Normal Dataset					Paraphrased Dataset				Normal	Normal	Paraphrased
		GPT-3.5 Turbo	GPT-4 Turbo	Claude-3 Sonnet	Claude-3 Opus	Gemini 1.0-pro	GPT-3.5 Turbo	GPT-4 Turbo	Claude-3 Sonnet	Claude-3 Opus	OOD Avg.-CM	OOD Avg.-CD	OOD Avg.
Arxiv	Zero-shot Query	0.5768	0.5835	0.6764	0.6667	0.6666	0.5587	0.6116	0.6916	0.6935	-	-	-
	Log Rank	0.6572	0.7006	0.8015	0.8809	0.8560	0.6628	0.6660	0.6634	0.6747	0.6913	0.6219	0.3655
	LRR	0.6602	0.7031	0.8116	0.8596	0.8544	0.6654	0.6654	0.6654	0.6654	0.7319	0.6130	0.2353
	DetectGPT	0.6654	0.6634	0.6673	0.6673	0.6673	0.6641	0.6628	0.6654	0.6654	0.6642	0.7091	0.6352
	RADAR	0.9566	0.7858	0.7034	0.7754	0.7868	0.9203	0.6970	0.6884	0.7202	0.7404	0.8035	0.7388
	Raidar	0.8316	0.8157	0.8029	0.8289	0.7366	0.9004	0.8851	0.8052	0.8303	0.6984	0.6524	0.7270
	OpenAI Detector	0.7889	0.6660	0.6673	0.6673	0.6976	0.7062	0.6654	0.6673	0.6673	0.6249	0.6569	0.6705
	Binoculars	0.9097	0.9135	0.9256	0.9699	0.9560	0.6617	0.6971	0.8112	0.8672	0.9163	**0.8199**	0.5835
	GhostBuster	0.9716	0.9886	0.9815	0.9813	0.9571	0.9700	**0.9943**	**0.9814**	0.9856	0.9187	0.6811	**0.9672**
	BiScope	0.9870	0.9928	0.9796	0.9885	0.9708	0.9769	0.9800	0.9625	0.9870	0.9517	0.7131	0.8534
	BiScope*	**0.9928**	**0.9943**	**0.9869**	**0.9913**	**0.9797**	**0.9870**	0.9859	0.9593	**0.9884**	**0.9775**	0.7767	0.8740
Yelp	Zero-shot Query	0.0020	0.0010	0.0110	0.0168	0.0080	0.0000	0.0009	0.0188	0.0188	-	-	-
	Log Rank	0.6776	0.6721	0.7120	0.6946	0.6439	0.6754	0.6660	0.6745	0.6743	0.6574	0.6695	0.6258
	LRR	0.6671	0.6678	0.6733	0.6678	0.6358	0.6681	0.6662	0.6674	0.6666	0.6589	0.6615	0.6508
	DetectGPT	0.6945	0.6737	0.7252	0.7477	0.6626	0.6702	0.6669	0.7009	0.7166	0.6738	0.7187	0.6710
	RADAR	0.7618	0.7090	0.7310	0.7590	0.7497	0.7485	0.7030	0.7117	0.7345	0.7148	0.7370	0.7033
	Raidar	**0.9023**	0.8985	0.9180	0.8876	0.8915	0.8948	0.9124	0.9344	0.9128	0.8572	0.6850	0.7817
	OpenAI Detector	0.7286	0.6668	0.6668	0.6616	0.6798	0.7240	0.6668	0.6668	0.6668	0.6348	0.6308	0.6563
	Binoculars	0.7295	0.6665	0.7583	0.8260	0.6885	0.6683	0.6655	0.6908	0.7284	0.6930	0.8474	0.6681
	GhostBuster	0.8193	0.8369	0.8746	0.8644	0.8625	0.8174	0.8649	0.9271	0.9145	0.7975	0.5859	0.8452
	BiScope	**0.9023**	0.9405	0.9652	0.9532	0.9486	0.9064	0.9473	0.9814	**0.9789**	0.9063	**0.8608**	**0.9523**
	BiScope*	0.9010	**0.9452**	**0.9658**	**0.9570**	**0.9545**	**0.9102**	**0.9530**	**0.9830**	0.9757	**0.9128**	0.8455	0.9505
Creative	Zero-shot Query	0.2730	0.1502	0.2691	0.3186	0.2719	0.2398	0.1694	0.1897	0.2948	-	-	-
	Log Rank	0.9673	0.7341	0.8779	0.9269	0.8044	0.7673	0.6685	0.6823	0.7701	0.7993	0.5824	0.5213
	LRR	0.9512	0.6732	0.8062	0.8884	0.7209	0.6638	0.6662	0.6649	0.6662	0.7242	0.5772	0.3846
	DetectGPT	0.8305	0.7090	0.7922	0.8166	0.7580	0.6850	0.6715	0.7364	0.7066	0.7573	0.5415	0.6209
	RADAR	0.9543	0.8869	0.9131	0.9345	0.9382	0.9298	0.8744	0.9160	0.9145	0.8934	0.7699	0.8937
	Raidar	0.8933	0.8303	0.8481	0.8661	0.8588	0.8271	0.8217	0.8120	0.7978	0.8151	0.7580	0.5621
	OpenAI Detector	0.6666	0.6671	0.6669	0.6671	0.6271	0.6671	0.6671	0.6671	0.6671	0.6103	0.6708	0.5475
	Binoculars	0.9945	0.9681	0.9814	0.9866	0.9880	0.9627	0.8381	0.9348	0.9540	0.9711	0.7599	0.8738
	GhostBuster	0.9965	0.9821	0.9834	0.9834	0.9920	0.9861	0.9786	0.9871	0.9865	0.9501	**0.8206**	0.9012
	BiScope	**0.9985**	0.9950	**0.9960**	0.9930	0.9964	0.9955	0.9945	**0.9955**	0.9940	**0.9846**	0.7980	**0.9707**
	BiScope*	0.9975	**0.9955**	0.9955	**0.9945**	**0.9970**	**0.9955**	**0.9955**	0.9950	**0.9945**	0.9780	0.8154	0.9513

实验设计　对比实验

- 评估BISCOPE在不同数据集上的表现
  - OOD Avg.-CM：跨模型测试； OOD Avg.-CD：跨领域测试
  - BISCOPE：不使用摘要生成步骤； BISCOPE*：完整步骤

		Normal Dataset					Paraphrased Dataset				Normal	Normal	Paraphrased
	Method	GPT-3.5 Turbo	GPT-4 Turbo	Claude-3 Sonnet	Claude-3 Opus	Gemini 1.0-pro	GPT-3.5 Turbo	GPT-4 Turbo	Claude-3 Sonnet	Claude-3 Opus	OOD Avg.-CM	OOD Avg.-CD	OOD Avg.
Essay	Zero-shot Query	0.0156	0.0098	0.0175	0.0059	0.0322	0.0078	0.0214	0.0423	0.0078	-	-	-
	Log Rank	0.9936	0.9065	0.9771	0.9811	0.9774	0.9004	0.7067	0.8170	0.9313	0.9025	0.4518	0.6074
	LRR	0.9945	0.8416	0.9673	0.9836	0.9685	0.8121	0.6658	0.7243	0.8546	0.8911	0.4724	0.5395
	DetectGPT	0.9344	0.8709	0.9302	0.9378	0.9311	0.7910	0.7622	0.8609	0.8429	0.9140	0.5769	0.7513
	RADAR	0.9812	0.8978	0.9648	0.9555	0.9650	0.9509	0.8211	0.9471	0.9118	0.9386	0.7665	0.8883
	Raidar	0.9786	0.9424	0.9672	0.9710	0.9574	0.9448	0.9186	0.9146	0.9216	0.9416	0.7136	0.8000
	OpenAI Detector	0.7069	0.6664	0.6667	0.6669	0.6426	0.6669	0.6662	0.6667	0.6664	0.5870	0.6411	0.5300
	Binoculars	0.9995	0.9970	0.9945	0.9960	0.9978	0.9920	0.9607	0.9787	0.9955	0.9967	0.7383	0.9429
	GhostBuster	0.9995	0.9950	0.9960	0.9965	0.9967	0.9916	0.9861	0.9880	0.9930	0.9804	**0.7740**	**0.9435**
	BISCOPE	**1.0000**	**0.9990**	**0.9985**	0.9970	**0.9994**	0.9965	**0.9990**	**0.9990**	0.9980	**0.9946**	0.5456	**0.9435**
	BISCOPE*	**1.0000**	**0.9990**	**0.9985**	**0.9975**	0.9989	**0.9975**	**0.9990**	0.9985	**0.9985**	0.9914	0.5669	0.9292
Code	Zero-shot Query	0.6300	0.5833	0.4351	0.3524	0.1854	0.6690	0.6784	0.6400	0.4545	-	-	-
	Log Rank	0.6581	0.6610	0.6611	0.6569	0.6583	0.6612	0.6611	0.6556	0.6581	0.6521	0.5306	0.5539
	LRR	0.6639	0.6639	0.6639	0.6639	0.6542	0.6639	0.6639	0.6639	0.6639	0.6613	0.6475	0.6591
	DetectGPT	0.6361	0.6474	0.6583	0.6612	0.6682	0.6612	0.6639	0.6639	0.6612	0.6445	0.5936	0.6282
	RADAR	0.6680	0.6653	0.6652	0.6597	0.6626	0.6598	0.6653	0.7322	0.6653	0.6652	**0.8114**	0.6660
	Raidar	0.9368	0.8220	0.6121	0.6156	0.4858	0.9325	0.8744	0.8250	0.6197	**0.8878**	0.1378	0.6521
	OpenAI Detector	0.7213	0.6977	0.6916	0.6542	0.6666	0.7514	0.6639	0.6639	0.6695	0.6567	0.4083	0.5767
	Binoculars	0.7073	0.6512	0.6612	0.6653	0.6624	0.7101	0.6338	0.8041	0.7179	0.6273	0.7181	0.6771
	GhostBuster	0.8524	0.7942	0.6556	**0.6749**	0.3860	0.8662	0.7729	0.7757	0.5390	0.6232	0.5091	0.6790
	BISCOPE	0.9665	**0.9655**	**0.8528**	0.6069	**0.7809**	**0.9659**	**0.9464**	**0.9691**	**0.9250**	0.7974	0.5895	0.8999
	BISCOPE*	**0.9692**	0.9586	0.8526	0.6620	0.7741	0.9597	0.9435	0.9600	0.9222	0.7898	0.5855	**0.9024**

- 评估FCE、BCE双向交叉熵信号模块的功能
  - BCE 在多数情形比 FCE 更具判别性，同时 BCE+FCE 组合通常优于单独使用

Table 5: Detailed contribution comparison between $\mathcal{FCE}$ and $\mathcal{BCE}$.

	Llama-2-7b w/o summary	Generative AI Model				
Dataset	Method	GPT-3.5-Turbo	GPT-4-Turbo	Claude-3-Sonnet	Claude-3-Opus	Gemini-1.0-pro
Arxiv	$\mathcal{FCE}$ Only	0.9281	0.9698	0.9407	0.9827	0.9647
	$\mathcal{BCE}$ Only	0.9524	0.9685	0.9668	0.9740	0.9429
	$\mathcal{BCE}$+$\mathcal{FCE}$	**0.9827**	**0.9957**	**0.9766**	**0.9855**	**0.9708**
Yelp	$\mathcal{FCE}$ Only	0.7934	0.7865	0.8834	0.8626	0.8342
	$\mathcal{BCE}$ Only	0.8435	**0.9005**	0.9396	0.9198	0.9151
	$\mathcal{BCE}$+$\mathcal{FCE}$	**0.8566**	0.9002	**0.9446**	**0.9289**	**0.9209**
Creative	$\mathcal{FCE}$ Only	0.9965	0.9498	0.9799	0.9855	0.9791
	$\mathcal{BCE}$ Only	0.9955	**0.9945**	**0.9950**	**0.9935**	**0.9940**
	$\mathcal{BCE}$+$\mathcal{FCE}$	**0.9985**	0.9930	0.9940	0.9915	0.9934
Essay	$\mathcal{FCE}$ Only	0.9980	0.9860	0.9929	0.9930	0.9989
	$\mathcal{BCE}$ Only	**0.9995**	0.9970	**0.9975**	0.9965	0.9994
	$\mathcal{BCE}$+$\mathcal{FCE}$	**0.9995**	**0.9970**	0.9965	**0.9965**	**0.9995**
Code	$\mathcal{FCE}$ Only	0.7849	0.6439	0.4938	0.4628	0.3817
	$\mathcal{BCE}$ Only	**0.9496**	**0.9466**	0.8173	**0.5747**	0.7819
	$\mathcal{BCE}$+$\mathcal{FCE}$	0.9479	0.9301	**0.8275**	0.5677	**0.7878**

- 分段划分比例对实验结果的影响

Table 6: Ablation results of different segmentation methods in multi-point splitting in BISCOPE.

		Normal Dataset					Paraphrased Dataset				Normal	Paraphrased
	Method	GPT-3.5 Turbo	GPT-4 Turbo	Claude-3 Sonnet	Claude-3 Opus	Gemini 1.0-pro	GPT-3.5 Turbo	GPT-4 Turbo	Claude-3 Sonnet	Claude-3 Opus	Avg.	Avg.
Arxiv	Every 50% Text	0.9754	0.9871	0.9635	0.9855	0.9637	0.9587	0.9756	0.9486	0.9767	0.9750	0.9649
	Every 25% Text	0.9813	0.9914	0.9662	0.9884	0.9690	0.9675	0.9769	0.9622	0.9797	0.9792	0.9716
	Every 10% Text (In Paper)	**0.9870**	**0.9928**	**0.9796**	**0.9885**	**0.9708**	**0.9769**	**0.9800**	**0.9625**	**0.9870**	**0.9837**	**0.9766**
Yelp	Every 50% Text	0.8922	0.9314	0.9584	0.9471	0.9337	0.8921	0.9359	0.9779	0.9704	0.9326	0.9441
	Every 25% Text	0.9002	0.9381	0.9651	0.9527	0.9466	0.9041	0.9452	**0.9817**	0.9757	0.9405	0.9517
	Every 10% Text (In Paper)	**0.9023**	**0.9405**	**0.9652**	**0.9532**	**0.9486**	**0.9064**	**0.9473**	**0.9814**	**0.9789**	**0.9420**	**0.9535**
Creative	Every 50% Text	**0.9985**	**0.9960**	0.9940	**0.9955**	0.9958	0.9950	**0.9955**	0.9935	0.9930	**0.9960**	0.9943
	Every 25% Text	0.9980	0.9955	**0.9960**	0.9930	**0.9970**	**0.9960**	**0.9955**	0.9950	0.9935	0.9959	**0.9950**
	Every 10% Text (In Paper)	**0.9985**	0.9950	**0.9960**	0.9930	0.9964	0.9955	0.9945	**0.9955**	**0.9940**	0.9958	0.9949
Essay	Every 50% Text	**1.0000**	**0.9990**	0.9965	0.9970	**0.9994**	**0.9975**	**0.9995**	0.9975	0.9975	0.9984	0.9980
	Every 25% Text	**1.0000**	**0.9990**	**0.9985**	**0.9980**	**0.9994**	0.9965	0.9990	0.9985	**0.9980**	**0.9990**	0.9980
	Every 10% Text (In Paper)	**1.0000**	**0.9990**	**0.9985**	0.9970	**0.9994**	0.9965	0.9990	**0.9990**	**0.9980**	0.9988	**0.9981**
Code	Every 50% Text	0.8564	0.8790	0.7706	0.5933	0.6479	0.8798	0.8752	0.9211	0.8427	0.7495	0.8797
	Every 25% Text	0.9532	0.9333	0.8115	**0.6184**	0.7088	0.9363	0.9322	0.9470	0.8856	0.8050	0.9252
	Every 10% Text (In Paper)	**0.9665**	**0.9655**	**0.8528**	0.6069	**0.7809**	**0.9659**	**0.9464**	**0.9691**	**0.9250**	**0.8345**	**0.9516**

算法总结 BISCOPE

- 算法贡献
  - 计算FCE、BCE双向交叉熵信号作为特征
    - 明确提出并验证“当因果语言模型遇到人类文本时，更偏向记忆前一Token、较少体现下一Token预测信息”的可检测差异
  - 分段提取特征
    - 不仅能对齐长度，还能让后续分类器通过学习找到最有效的分区位置
- 算法不足
  - 整体花费时间较长，摘要生成、双向计算等操作加大了计算量
    - 当使用摘要引导版本时，单样本耗时会从0.14s提升到1.35s

M-RangeDetector: Enhancing Generalization in Machine-Generated Text Detection through Multi-Range Attention Masks

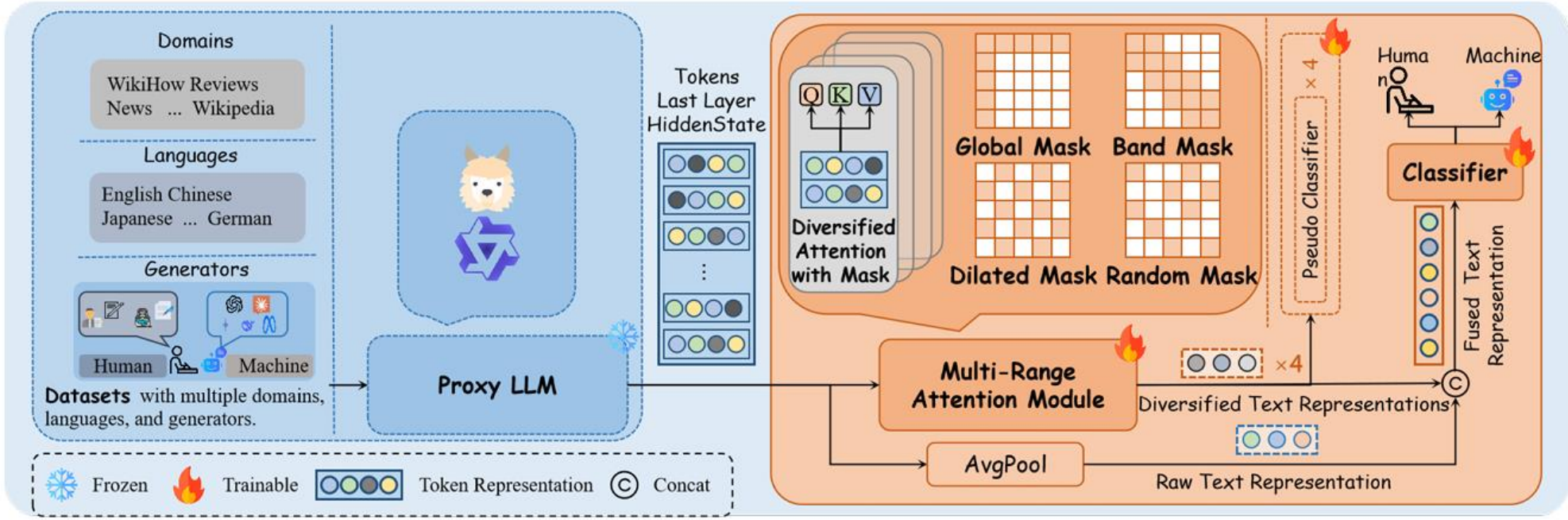
M-RangeDetector TIPO

T	目标	区分人类撰写文本与机器生成文本，二分类任务
I	输入	待检测文本、代理大模型（自回归模型） 数据集：Ghostbuster（2.1万）、M4（29.6万）、 OUTFOX（1.54万）、TuringBench（20万）
P	处理	1. 利用代理大模型提取输入文本的隐藏层表示向量H，并平均池化得到r_S 2. 对向量H采用4种不同的掩码注意力机制，分别计算得到r_1, r_2, r_3, r_4 3. 将r_1, r_2, r_3, r_4分别接入伪分类器产生辅助损失，形成差异化学习约束 4. 拼接r_S, r_1, r_2, r_3, r_4作为特征，训练二分类器输出标签
O	输出	待测文本是否属于机器生成文本

P	问题	1.现有方法学到的特征往往与领域词汇、模型特征强相关 2.现有方法在训练时各分支可能学习到相似的表征
C	条件	需要代理大模型（自回归模型）
D	难点	1.如何提取领域无关特征，即大模型与人类区别的本质特征 2.如何在训练时避免各分支学习到相似的表征
L	水平	2025 CCF A类

M-RangeDetector 算法原理

- 算法原理图
  - 利用代理大模型提取输入文本的隐藏层表示向量H，并平均池化得到r_S
  - 对向量H采用4种不同的掩码注意力机制，分别计算得到r_1, r_2, r_3, r_4
  - 将r_1, r_2, r_3, r_4分别接入伪分类器产生辅助损失，形成差异化学习约束
  - 拼接r_S, r_1, r_2, r_3, r_4作为特征，训练二分类器输出标签



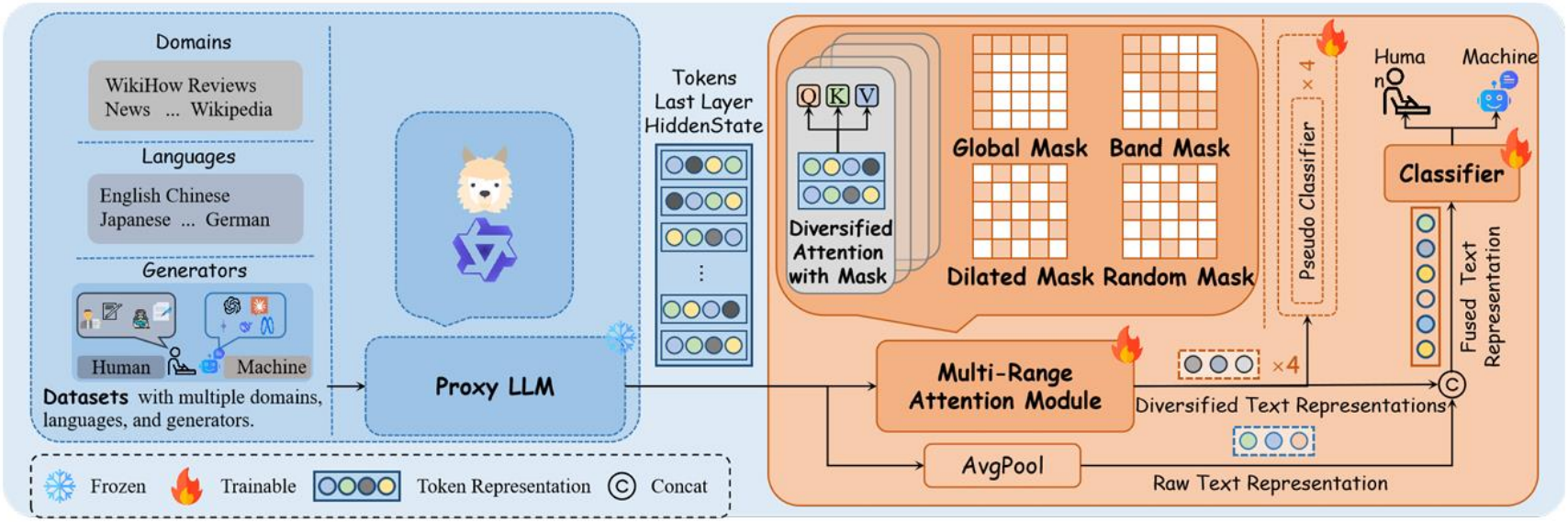
M-RangeDetector 创新点分析 掩码注意力建模写作差异

- 现有方法存在问题
  - 检测器学到的特征往往与领域词汇、模型特征强相关，而不是与“生成机制”相关
  - 大模型生成时只能看前文范围；人类写作时可能考虑全局/局部/双向等多种范围
- 解决方法
  - 输入：代理大模型提取输入文本的隐藏层表示向量H
  - 利用掩码注意力机制建模写作策略差异
    - **Global mask**：选一组全局Token G，任意Token都可以与G中Token交互
    - **Band mask**：设定窗口宽度ω，Token仅能与距离不超过ω的邻近Token交互
    - **Dilated mask**：设定间隔d， Token仅与满足“间隔为d的倍数”的Token交互
    - **Random mask**：设定稀疏比r，每个Token随机选择一部分Token进行交互
  - 分别训练，提取特征并平均池化得到r_1, r_2, r_3, r_4

全局锚点；局部连贯；稀疏长距离关联；不规则信息抽取 {1,2,3,4,5,6,7,8,9,10}

M-RangeDetector 创新点分析 差异化学习约束

- 现有方法存在问题
  - 并行计算多种掩码注意力特征向量时，如果融合后只用一个分类损失监督训练，各分支可能学习到相似的表征，无法形成真正差异
- 解决方法
  - 给每个分支r_1, r_2, r_3, r_4分别接一个辅助分类头（伪分类器），让每个分支都必须“单独完成分类任务”，产生辅助损失



实验设计　数据资源

- 数据集
  - Ghostbuster： 面向三领域，包含人类文本与两类商用模型生成文本，样本约2.1万
  - M4： 覆盖多领域、多语言、多生成器，样本约29.6万
  - OUTFOX： 以“作文场景 + 对抗/鲁棒性评测”为核心，样本约1.54万
  - TuringBench： 人类、多生成器来源识别的新闻体裁基准，样本约20万，对应 20 个标签，即1个人类文本与19个生成器
- 对比方法

  Binoculars (2024) 、 DetectGPT (2023) 、 FastDetectGPT (2023)、GPTZero (2023)、RoBERTa (2019)、T5-Sentinel (2023)、GhostBuster (2023)、DeTeCtive (2024)、OUTFOX detector (2024)
- 评价指标
  - Acc、F1、AvgRec(平均召回率)

实验设计　对比实验

- 评估M-RangeDetector在不同数据集上的表现
  - 在多语言、跨领域场景中，表现优异

Method	M4-monolingual		M4-multilingual	
	AvgRec	F1	AvgRec	F1
Binoculars (Hans et al., 2024)	89.89	89.89	80.63	82.43
RoBERTa (Guo et al., 2023)	88.70	88.44	80.01	84.44
T5 − Sentinel (Chen et al., 2023)	84.01	81.08	76.21	68.99
DeTeCTive (Guo et al., 2024)	**98.44**	98.38	93.42	93.05
M − RangeDetector	98.42	**98.41**	**97.06**	**96.98**

Model	**In-Domain**				**Out-of-Domain**			
	All Domains	News	Creative Writing	Student Essays	News	Creative Writing	Student Essays	**Average**
Binoculars (Hans et al., 2024)	92.7	97.4	92.4	87.9	97.4	92.4	87.9	92.6
DetectGPT (Mitchell et al., 2023)	57.4	56.6	48.2	67.3	56.6	48.2	67.3	57.4
FastDetectGPT (Bao et al., 2023)	90.8	92.5	88.5	91.2	92.5	88.5	91.2	90.7
GPTZero (Tian, 2023)	93.1	91.5	93.1	83.9	91.5	93.1	83.9	89.5
RoBERTa (Guo et al., 2023)	98.1	99.4	97.6	97.4	88.3	95.7	71.4	85.1
T5 − Sentinel (Chen et al., 2023)	96.6	97.8	95.6	96.2	89.6	95.6	87.9	91.0
Ghostbuster (Verma et al., 2023)	99.0	99.5	98.4	99.5	97.9	95.3	**97.7**	97.0
M − RangeDetector (Ours)	**99.8**	**100.0**	**99.5**	**100.0**	**98.6**	**99.0**	**97.7**	**98.4**

实验设计　对比实验

- 评估M-RangeDetector在不同数据集上的表现
  - 在多样化提示策略、跨模型、非母语英语撰写文本、改写攻击场景中，表现优异

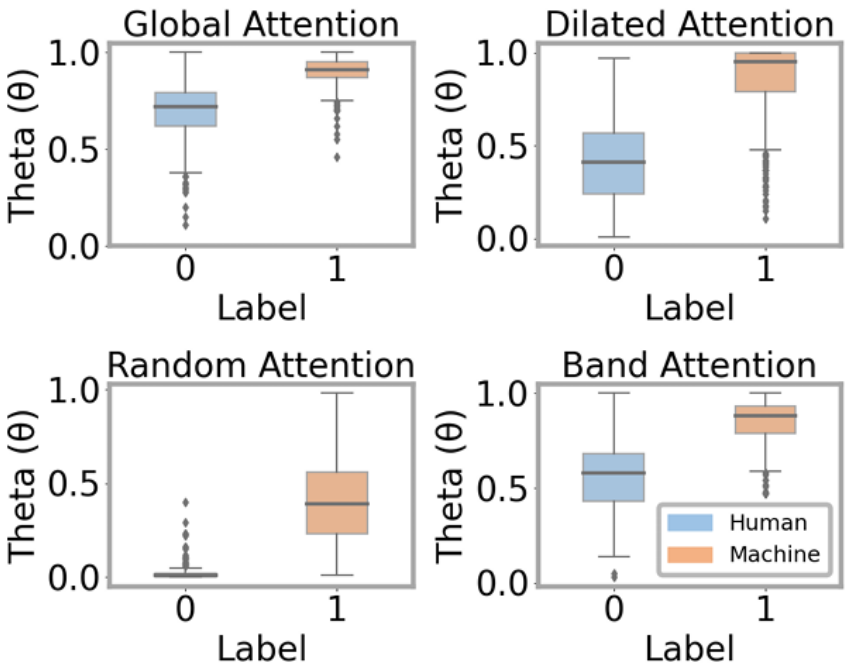
Model	Prompts (F1)	Claude (F1)	Lang8 (Acc.)	TOEFL 11 (Acc.)	TOEFL 91 (Acc.)
Binoculars	48.4	46.0	94.5	100.0	64.8
DetectGPT	70.8	64.2	98.6	**100.0**	63.7
FastDetectGPT	94.6	84.0	90.2	90.9	64.8
GPTZero	96.1	75.6	99.2	**100.0**	92.3
RoBERTa	97.4	87.8	98.6	98.1	96.7
T5 – Sentinel	94.6	84.1	98.9	99.6	97.8
Ghostuster	99.5	92.2	95.5	99.9	74.7
M – RangeDetector	**99.7**	**96.5**	**99.9**	**100.0**	**100.0**

Attacker Detector	Non-attacked		DIPPER		OUTFOX	
	AvgRec	F1	AvgRec	F1	AvgRec	F1
Binoculars	49.3	33.0	55.4	45.2	89.1	89.0
FastDetectGPT	75.1	74.6	88.2	88.2	94.9	94.9
RoBERTa	90.8	90.7	94.3	94.4	73.9	68.3
T5 – Sentinel	99.0	98.9	96.1	96.1	94.8	94.8
OUTFOX	96.5	96.4	82.4	79.0	61.8	39.4
DeTeCTive	99.1	99.1	97.7	97.5	97.0	96.9
M – RangeDetector	**99.4**	**99.4**	**99.2**	**99.2**	**99.1**	**99.1**

实验设计　消融实验与其它实验

- 评估M-RangeDetector的不同模块在不同数据集上的表现
  - 掩码注意力建模写作策略差异效果较好
- 比较人类撰写文本与机器生成文本在四种注意力机制下的θ值分布
  - 差异较大，且在不同掩码注意力机制下分布不同

Method	M4-monolingual	
	AvgRec	F1
$M-RangeDetector$	**98.42**	**98.41**
$w/o\ All\ Attentions$	92.14	92.20
$w/o\ AvgPool$	97.82	97.85
$w/o\ Global\ Attention$	96.75	96.83
$w/o\ Band\ Attention$	96.15	96.24
$w/o\ Dilated\ Attention$	96.66	96.74
$w/o\ Random\ Attention$	96.25	96.28
$w/o\ Pseudo\ Classifier$	97.63	97.68



算法总结　M-RangeDetector

- 算法贡献
  - 利用掩码注意力机制建模写作策略差异：让模型在同一个文本上产生“多范围表征”，把“写作时依赖的信息范围差异”显式化为特征
    - 捕获“写作策略、注意力范围差异”特征，提升泛化性
  - 差异化学习约束：给每个分支分别接一个辅助分类头（伪分类器），让每个分支都必须“单独完成分类任务”，产生辅助损失
    - 确保分支差异真实存在，避免分支退化
- 算法不足
  - 掩码注意力机制中窗口宽度ω、间隔d、稀疏比r等超参数未明确进行实验
  - 差异化学习约束设计较为简单，并不保证各分支特征在信息论意义上独立、互补

特点总结与未来展望

特点总结与未来展望

- 特点总结
  - 从生成机制探索机生文本检测新方法
  - BISCOPE
    - 计算FCE、BCE双向交叉熵信号作为特征
    - 明确提出并验证“当因果语言模型遇到人类文本时，更偏向记忆前一Token、较少体现下一Token预测信息”的可检测差异
  - M-RangeDetector
    - 利用掩码注意力机制建模写作策略差异
    - 捕获人类与大模型写作策略的本质特征，提升泛化性
- 未来发展
  - 细化机生文本的分类，如机器润色、机器补全等
  - 在短文本上存在性能下降的问题，短文本提供的特征更少，且句子层面的扰动相比长文本更敏感

参考文献

[1] Cheng S, Guo H, Jin X, et al. BiScope: AI-generated Text Detection by Checking Memorization of Preceding Tokens. Advances in Neural Information Processing Systems 37[C]. Vancouver, BC, Canada: Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024: 104065-104090.

[2] Jiao K, Wang Q, Zhang L, et al. M-RangeDetector: Enhancing Generalization in Machine-Generated Text Detection through Multi-Range Attention Masks. Findings of the Association for Computational Linguistics: ACL 2025[C]. Vienna, Austria: Association for Computational Linguistics, 2025: 8971-8983.

知人者智，自知者明。胜人者有力，自胜者强。知足者富。强行者有志。不失其所者久。死而不亡者，寿。

谢谢！