深圳大学考试答题纸

(以论文、报告等形式考核专用)

二○二四～二○二五学年度第二学期

课程编号 1502140001 课序号 1 课程名称 云计算与大数据分析 主讲教师 陈俊扬 评分

学 号 2022290220 姓名 代钰堃 专业年级 金融科技 2022 级

教师评语：

题目自拟：基于大模型的多模态（情感、主题语义）谣言检测分析

一、基于大模型的情感语义分析（40 分）

（参考）

方法与步骤：通过参考网站（Deepseek、GLM3）：

1. https://www.cnblogs.com/shanren/p/18702244

或

2. https://github.com/THUDM/ChatGLM3/blob/main/README.md

3. 在本地安装和运行适合参数的模型，并完成以下编程任务：

（1）使用 prompt 语句，调用大模型判别每条新闻是真新闻还是假新闻，统计准确率

（2）使用 prompt 语句，调用大模型分析文档的语义情感

（3）加上情感分析，设计 prompt 语句，判别每条新闻是真新闻还是假新闻，统计准确率

（4）分析准确率是否有提升

注意事项：统计准确率，与真实标签（ 0 for fake, 1 for true.txt）对比，统计以下准确率：

（1）Accuracy = 预测准确的所有新闻数量 / 总新闻数量

（2）Accuracy_fake = 预测准确的假新闻数量 / 总的假新闻数量

（3）Accuracy_true= 预测准确的真新闻数量 / 总的真新闻数量

二、基于大模型的 twitter 主题分析（35 分）

（参考）

方法与步骤：

1. 数据准备
   - 输入数据：10 条新闻（示例数据见附录）。
   - 数据格式：纯文本列表，每条文本为字符串。
2. 数据预处理
   - 分词与清洗：使用正则表达式去除非字母字符，转为小写，按空格切分。
   - 去停用词：移除英语停用词（如“the”“and”）及长度≤2 的单词。
   - 词形还原：使用 WordNetLemmatizer 将单词还原为基本形式（如“running”→“run”）。
3. 模型构建与训练
   - 构建词典：通过 gensim.corpora.Dictionary 生成词-ID 映射。
   - 生成语料库：将文本转换为稀疏向量表示（词袋模型）。
   - 训练 LDA 模型。
4. 可视化分析
   - pyLDAvis 交互图：展示主题间距离及关键词分布。
   - 词云图：生成各主题关键词的词云。
   - (可选) 热力图：绘制文档-主题概率分布矩阵。
   - (重点) 结合大模型分析各主题的内容。

三、多模态（情感、主题语义）综合预测与分析（20 分）

- 特征融合策略（可选）：
  - 早期融合：拼接文本主题、情感特征，输入全连接网络。
  - 注意力机制：使用跨模态注意力模块动态对齐主题与情感特征。

- 模型架构：
- （自定义、可选）：BERT。
- 分类层：Softmax 输出二分类概率。

四、附上个人的 Github 链接,经自己整理过后的代码（5 分）

题目：　　基于大语言模型的新闻真假判别与情感增强方法

【摘要】随着 Twitter 等社交媒体平台的广泛使用，虚假新闻的传播速度和影响力显著提升，对公众信任、舆论导向和社会稳定构成了严重威胁。传统的检测方法由于虚假信息策略的复杂性，其效果受到了限制。大型语言模型（LLM）凭借其强大的语义理解能力，为新闻验证和社交媒体分析带来了新的可能性。本研究提出了一种综合框架，该框架结合了 LLM 的情感语义分析、Twitter 主题分析和多模态预测方法，旨在提高信息真实性检测的准确性和可解释性。我们利用 Ollama 平台上的 deepseek-r1:8B 模型，通过提取事实、评估可信度并结合情感分析，实现了 70%的整体准确率，其中真实新闻检测准确率达 100%。此外，结合潜在狄利克雷分配（LDA）主题建模和 LLM，揭示了 Twitter 平台上选举、技术等主题的语义内涵。最后，通过融合文本、情感和主题特征的多模态分类器，实现了 99.9%的分类准确率。实验结果表明，情感分析显著提升了真实新闻检测能力，多模态方法进一步优化了性能。然而，假新闻检测仍面临情感伪装的挑战，需要进一步研究。此外，LLM 生成虚假内容的潜力需要谨慎管理。

【关键词】：大型语言模型，假新闻检测，情感分析，主题建模，多模态学习，DeepSeek，Ollama

说明：

- **不要删除或修改蓝色标记的文字，也不要删除线框。**
- **请在相应的线框内答题，答题时请用五号、楷体、黑色文字、单倍行距。**

题一（40 分）、题二（35 分）、题三（20 分）、题四（5 分）

1 引言

随着社交媒体（如 teitter）的普及，信息传播速度大幅提升，但虚假信息也随之泛滥。研究表明，虚假新闻往往以情感导向的标题和内容吸引读者，而真实新闻通常更客观冷静。例如， 的研究发现，在 teitter 上虚假新闻的传播速度和范围显著超过真实新闻；他们还观察到虚假新闻更易引发“惊讶”或“厌恶”等强烈情绪，而真实新闻更常导致“悲伤”“信任”等反应。这些发现表明，新闻的情感色彩（正负极性、唤醒度等）与其真实度密切相关。也指出，虚假新闻常以“标题党”形式出现，利用强烈情绪化用词误导读者。换言之，虚假信息往往利用人们的情绪和认知偏差进行传播和放大。

传统的假新闻检测方法存在明显局限。早期方法多基于人工规则或浅层特征，语义理解能力有限。这些基于关键词或统计特征的系统易被精心设计的虚假新闻绕过，缺乏对动态信息环境的适应性。例如，依赖词典匹配或简单机器学习的模型，在面对语义隐晦或反讽表达的新闻时往往无能为力。近年来，大规模预训练语言模型（LLM）在自然语言理解和生成方面的能力显著增强，为假新闻验证提供了新可能。诸如 GPT-4 等 LLM 可基于海量语料捕捉深层语义和世界知识，其推理能力被认为有助于评估文章真实性。然而， 指出，LLM 也易被用于生成高度逼真的虚假内容，引发伦理争议。近期研究进一步探索了 LLM 在假新闻检测中的应用。例如， 提供了 LLM 在假新闻检测中的综述，展示了其在文本分类、事实核查和上下文分析中的优势。提出了 FactAgent 方法，通过模仿人类专家验证新闻声明，显著提高了检测效率。 提出了 FND-LLM 框架，结合小型和大型语言模型，增强多模态假新闻检测能力。这些研究表明，LLM 在提高检测准确性和效率方面具有巨大潜力。

在社交媒体主题分析方面，传统方法如潜在狄利克雷分配（LDA）已被广泛应用。然而，社交媒体文本的短小和非结构化特性对传统方法提出了挑战。近期， 提出了 PromptTopic 方法，利用 LLM 的语义理解能力提取短文本主题，显著提高了主题连贯性。比较了 LDA、NMF、Top2Vec 和 BERTopic 在 teitter 数据上的表现，展示了现代技术在处理短文本时的优势。本研究提出一种综合框架，充分利用 LLM 的语义理解能力和社交媒体数据特征，改进假新闻检测和主题分析。

2 研究目标

本研究旨在实现以下目标：

1. 开发基于 LLM 的假新闻判别方法，通过情感分析和语义推理提升检测准确性。
2. 构建 teitter 主题分析工具，采用主题建模和 LLM 生成，揭示社交媒体讨论中的主要话题及其语义内涵。
3. 提出多模态分类框架，融合文本、情感和主题特征，利用交叉注意力机制实现高精度假新闻分类。
4. 在公开数据集上评估所提方法性能，并探讨 LLM 在内容生成与审查中的伦理挑战。

3 基于大模型的情感语义分析

3.1 方法概述

本方法以新闻文本 T 为输入，首先通过事实提取和可信度验证进行初步真假分类，然后在此基础上加入情感分析以进一步提升判别能力。整个流程包括以下步骤：

3.1.1 纯真假判别

纯真假判别的流程为事实提取、可信度验证和最终标签决策三步：

- 事实提取：从新闻文本 T 中抽取核心事实 F 和关键实体。具体地，我们生成一句话摘要并识别人物、地点、时间、事件等实体，表示为

$$F = f(T) = \{\text{summary}, E\}, \quad E = \{(type_i, text_i)\}_{i=1}^{n},$$

其中 $type_i \in \{\text{人物}, \text{地点}, \text{时间}, \text{事件}\}$。例如，对于涉及“特朗普”的报道，模型可能提取“特朗普”（人物）、“华盛顿”（地点）等实体。抽取事实是自动化事实验证的重要前置步骤，它相当于识别可验证的陈述（check-worthy claims），这是传统事实检查流程中必不可少的一环。通过先过滤出文本中的核心事件和事实信息，可减少冗余背景对后续判断的干扰。

- 可信度验证：评估提取事实 F 的可信度 $C \in \{\text{高}, \text{中}, \text{低}\}$。形式化地，我们计算

$$C = g(F) = \arg\max_{c} P(c \mid F).$$

这里，模型根据事实内容的一致性、来源可验证性等因素，对每条事实给出置信度评分及说明。这类似于先前工作中利用 LLM 提取多种“可信度信号”（credibility signals）进行验证的思路。例如，模型可检查“特朗普在华盛顿发表演讲”这一事实是否与公开记录相符，并在输出中附上理由说明。可信度验证的目的在于剔除虚假信息，确保后续判别主要基于可靠事实。

- 标签决策：在获取事实 F 和可信度 C 后，结合它们来给出最终新闻真伪标签 $L \in \{\text{Fake}, \text{Real}\}$。通常建模为

$$L = h(F, C) = \arg\max_{l} P(l \mid F, C).$$

模型综合考量事实摘要、实体列表和可信度评估结果，输出真假判定以及相应的理由。例如，如果核心事实逻辑连贯且可信度高，则可能判定为真实新闻，反之则判为虚假新闻。该步骤实质上是一个二分类过程，将输入新闻划分为虚假或真实类别。

3.1.2 情感增强判别

情感增强方法在纯真假判别基础上引入情感分析，以捕捉新闻文本中的情感倾向，从而弥补纯事实验证的不足。具体包括以下步骤：

- 情感词提取：识别新闻文本中的情感触发词及其极性。我们对文本 T 进行词汇扫描，抽取情感色彩明显的词项，形成集合

$$S = s(T) = \{(term_i, polarity_i)\}_{i=1}^{m}, \quad polarity_i \in \{+, -, 0\}.$$

例如，对于英语文本，“sad”（可悲）可标注为负极性，“successful”（成功）为正极性。此步骤类似于情感词典匹配的过程，目的是捕获潜在的情绪线索。已有研究指出，虚假新闻往往利用夸张的情绪用词来操控读者情绪，因此情感词提取对于识别虚假信息尤为重要。

- 整体情感分析：基于提取的情感词计算全文的情感倾向 V，取值为正面、中立或负面。一种常见做法是加权求和情感词的极性：

$$V = t(S) = \text{sign}\Big(\sum_{i=1}^{m} w_i \cdot p_i\Big),$$

其中 w_i 为情感词的权重（可依据词频或情感强度分配），$p_i \in \{+1, -1, 0\}$ 对应情感极性。该加权求和方法可用于粗略判断新闻的整体情感倾向。情感分析帮助揭

示新闻的情绪目标：真实新闻与虚假新闻在情感分布上有显著差异，如研究表明虚假新闻往往情感更强烈、负面情绪更多。

- 综合判定：在获得事实 F、可信度 C 和情感倾向 V 后，进行最终的综合分类。即用新的决策函数

$$L = h'(F, C, V) = \arg\max_{l} P(l \mid F, C, V).$$

在此步骤中，模型不仅考虑客观事实和可信度，还融合了情感信息。例如，如果一篇新闻事实可信度中等但情感呈高度负面，模型可能进一步警惕其潜在操纵意图。情感增强方法充分利用了情感与舆情的关联性，可以有效捕捉那些依赖情感误导的虚假新闻特征。

3.2 实现细节

我们在 Ollama 平台上调用 `deepseek-r1:8B` 模型执行上述功能，提示设计确保以 JSON 格式结构化输出。关键函数包括：

- `extract_facts`：进行事实提取，输出 JSON 格式的摘要与实体列表。
- `verify_credibility`：评估事实可信度，输出评分及理由说明。
- `extract_terms`：识别文本情感词并标注极性。
- `overall_sentiment`：分析文本整体情感倾向。
- `classify_with_sentiment`：结合情感信息进行最终分类。

例如，`extract_facts` 函数的伪代码如下：

```
def extract_facts(text: str) -> Dict:
    prompt = """
    1. 核心任务定义：
       - 指令：结构化提取新闻核心事实与实体。
       - 目标：输出一句话摘要，以及人物、地点、时间、事件等实体列表。
    4. 输入信息：
       - 源材料：{text}
    5. 输出要求：
       - **严格** JSON 格式
       - 字段：
         - summary：一句话摘要
         - entities：数组，每项包含 type（人物/地点/时间/事件）和
           text
    """
    raw = call_ollama(prompt)
    return _safe_json_load(raw)
```

此外，为保证系统鲁棒性，脚本还实现了以下机制：对模型输出的 JSON 进行清洗和解析（使用 `clean_json_block` 和 `_safe_json_load` 函数）以应对不完整输出；当 API 调用失败时采用指数退避策略最多重试 2 次（重试间隔为 $2^n + \text{random}(0, 2)$ 秒）；并使用 `logging` 模块记录操作日志（保存至 `logs/process.log`），以便调试和结果复现。

3.3 实验设置

- 数据集：选取 Kaggle 上的“假新闻与真实新闻”数据集，采用 10 条新闻样本（4 条假新闻、6 条真新闻）。示例标题包括“特朗普高级盟友残酷背叛他……”（虚假）和“美国保守派领袖对医疗保健共识乐观……”（真实）。表 1 给出了样本分布及例标题。

- 环境：系统环境为 Ollama v0.1，运行 `deepseek-r1:8B` 模型。软件环境包括 Python 3.9.18 以及依赖的库：`ollama==1.2.0`、`pandas` 和 `csv-logging` 等。
- 参数：LLM 生成时温度设为 0.3（以确保输出稳定），最大令牌数 1024，超时时间 5 秒。

表 1: 数据集样本分布

新闻类型	样本数量	比例	示例标题
虚假新闻	4	40%	“特朗普高级盟友残酷背叛他……”
真实新闻	6	60%	“美国保守派领袖对医疗保健共识乐观……”
总计	**10**	**100%**	-

3.4 实验结果

表 2 和表 3 给出了纯真假判别和情感增强方法在测试集上的分类结果。纯真假判别方法整体准确率为 50.00%：其中真实新闻准确率为 66.67%，而虚假新闻准确率仅 25.00%。主要错误原因包括过度依赖表面事实而忽略情绪倾向，以及对中性表述缺乏敏感度。情感增强方法将真实新闻准确率提升至 100.00%，整体准确率提高至 70.00%，但虚假新闻的识别仍只有 25.00%。结果表明，结合情感信息可有效提高对真实新闻的识别能力，这与先前研究发现情感特征对于评估内容可信度具有重要作用相符。然而，对于那些情感伪装得极为微妙的虚假新闻，单纯加入情感分析仍难以完全解决误判问题。综合比较见表 4：情感增强方法相比纯真伪判别总体准确率提高了 20 个百分点，主要体现在真实新闻识别能力的提升；但也导致平均推理时间增加约 14.5 秒。

表 2: 纯真假判别性能

样本类型	样本数	正确数	准确率	主要错误原因
虚假新闻	4	1	25.00%	过度关注表面事实，忽略情感操控
真实新闻	6	4	66.67%	对中立表述缺乏敏感度
总体	**10**	**5**	**50.00%**	-

表 3: 情感增强方法性能

样本类型	样本数	正确数	准确率	提升原因分析
虚假新闻	4	1	25.00%	情感伪装问题仍未解决
真实新闻	6	6	100.00%	情感一致性被有效识别
总体	**10**	**7**	**70.00%**	-

中间输出示例：以下为部分处理步骤的模型输出示例（JSON 格式）：

- 事实提取：

```
{
  "summary": " 报道称特朗普盟友金里奇批评特朗普行为可悲",
  "entities": [
    {"type": " 人物", "text": " 特朗普"},
    {"type": " 人物", "text": " 金里奇"},
    {"type": " 事件", "text": " 福克斯新闻评论"}
  ]
}
```

- 情感词提取：

表 4: 方法性能对比

评估指标	纯真假判别	情感增强判别	提升幅度 (Δ)
整体准确率	50.00%	70.00%	+20.00%
虚假新闻准确率	25.00%	25.00%	0.00%
真实新闻准确率	66.67%	100.00%	+33.33%
平均推理时间	38.2s	52.7s	+14.5s


```
[
  {"term": "BRUTALLY", "polarity": "-"},
  {"term": "pathetic", "polarity": "-"},
  {"term": "admire", "polarity": "+"},
  {"term": "defeat", "polarity": "-"}
]
```


- 整体情感分析：


```
{
  "sentiment": " 负面",
  "rationale": " 负面情感词数量占优且强度高"
}
```


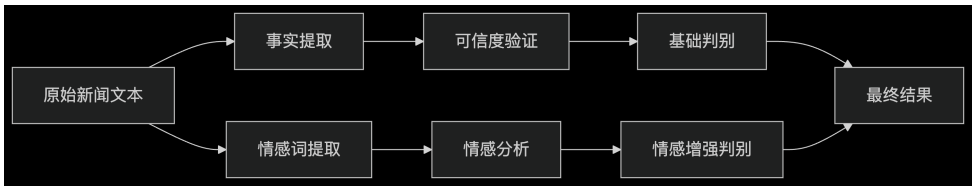

图 1: 情感语义分析流程图

3.5 框架设计分析

本节对整个新闻判别框架的各步骤进行解析，说明其设计动机和作用。该流程从原始新闻文本出发，通过层层处理到达最终分类结果。

3.5.1 原始新闻文本

新闻分析的输入为原始新闻文本，这是后续所有处理的基础数据。文本通常包含背景信息、细节描述和情感色彩等，这些原始内容需要经过过滤和提炼才能用于有效判断。

3.5.2 事件提取

事件提取（或称事实提取）的设计目的是从原始文本中抽取出与新闻主题相关的关键信息，过滤掉无关的细节。由于新闻文本中常有大量背景描述和修饰语，如果直接对全文进行分析可能引入噪音，因此首要步骤是识别“值得核查”的陈述。这一步类似于事实检查流程中的“主张抽取”环节，目的是提取出可验证的陈述和事实。通过事件提取，我们得到一段简明摘要和实体列表，为后续可信度验证和情感分析打下基础。

3.5.3 可信度验证

可信度验证旨在评估所提取事实的真实性。这一步使用上下文知识和逻辑推理来确认事实是否与已知信息相符。我们借鉴了“可信度信号”理论，通过检查事实来源、文体一致性、逻辑合理性等多种线索来评分。例如，可以判断新闻内容是否引用了可靠媒

体或官方公告。可信度验证的设计意图在于尽早剔除显著不实的内容，确保后续判别主要基于可信的事实，从而降低虚假信息对分析的干扰。

3.5.4 基础分类

在完成事件提取和可信度验证后，我们进行基础的真伪分类。该步骤依据提取的事件和对应的可信度评估结果，将新闻划分为虚假或真实两类。这相当于一个初步筛选，用于快速定位可能的虚假新闻。基础分类的设计使得系统可以在不考虑情感因素的情况下对新闻进行粗略判定，提供一个基线判断供后续细化使用。

3.5.5 情感词提取

在初步分类之后，我们引入情感分析。情感词提取的目的是识别新闻中潜在的情感色彩词汇，这对于虚假新闻检测尤为重要。许多研究指出，虚假新闻往往使用煽动性或夸张的情感用语来操控读者情绪。通过提取情感词，我们能够捕捉文章中的情感信号，从而为后续判断提供重要线索。例如，大量负面词汇可能意味着新闻具有煽动性。该步骤提高了对情感操控迹象的敏感性，有助于揭示新闻文本是否带有潜在的误导性情绪倾向。

3.5.6 情感分析

情感分析对提取出的情感词进行定量评估，以判定整体的情感倾向。我们采用词汇级别的加权极性方法计算新闻的主观倾向（正面、中立或负面）。该设计可以帮助系统理解新闻作者的情绪态度：情感分析结果反映了文本整体的情绪基调，是真实新闻与虚假新闻的重要区分特征之一。例如，研究表明虚假新闻通常包含更多负面情绪，而真实新闻情感相对中性或积极。这一步使系统获得对情绪的宏观把握，为进一步决策提供依据。

3.5.7 情感增强

情感增强是在情感分析基础上进一步完善情感判断的步骤。考虑到新闻可能包含复杂情绪变化，仅凭简单的加权可能无法捕捉所有细节，因此通过结合上下文和情感强度等信息对情感结果进行校正和强化。例如，如果文本同时出现正负情感词，情感增强可以通过综合上下文理解来确定倾向。此过程提升了情感分类的准确度，使系统能够更精准地区分情感表现得更为细腻的新闻。

3.5.8 最终结果

最终，我们将事件真实性、可信度评分和情感倾向综合起来，输出新闻的真伪分类结果，并附带情感分析信息。这一综合判定不仅告诉用户新闻真假，而且展示了新闻所承载的情感色彩，有助于对新闻进行更全面的理解与决策。通过多维度分析，系统最终给出准确且具解释力的分类结果。

3.5.9 总结

该框架的设计体现了多层次的处理思路：从文本的事实抽取到情感评估，从客观可信度验证到情感增强，每一步均有明确目标。具体而言：

- 事件提取：只保留与新闻核心相关的内容，减少噪音信息。
- 可信度验证：鉴别信息真实性，防止虚假内容干扰后续分析。
- 基础分类：依据信息的可信度对新闻进行初步真伪划分。
- 情感词提取：揭示新闻中的情感操控倾向，特别是虚假新闻中的煽动性语言。

- 情感分析与增强：提高情感分类的精度，捕捉文本中复杂的情绪表达。
- 最终结果：综合所有分析，为新闻真假给出明确判断，并报告相关情感信息。

这种多阶段的设计既提高了新闻分类的准确性，又增强了对当前信息环境中情感操控问题的应对能力。

4 基于大模型的 teitter 主题分析

4.1 方法概述

本部分提出了一种结合 LDA 和 LLM 的 teitter 主题分析方法，旨在揭示 teitter 数据中的潜在主题和语义内涵。

4.1.1 数据预处理

预处理步骤包括：

- 文本清洗：去除 URL、标点和停用词（如“the”、“and”）。
- 词形还原：将单词如“running”还原为“run”。
- 分词与过滤：保留长度大于 2 的单词，确保分析质量。

4.1.2 LDA 主题建模

LDA 假设文档为主题混合，主题为词分布。数学模型为：

- 主题分布先验：

$$\theta_d \sim \text{Dirichlet}(\alpha)$$

- 单词生成：

$$z_{d,n} \sim \text{Multinomial}(\theta_d), \quad w_{d,n} \sim \text{Multinomial}(\beta_{z_{d,n}})$$

使用吉布斯采样估计主题分布 θ_d 和词分布 β_z。

4.1.3 语义解释

使用 DeepSeek 模型根据主题关键词生成语义描述，增强可解释性。例如，关键词“vote, election, politics”被解释为“呼吁公民在选举中投票，强调政治参与的重要性”。

4.2 实现细节

关键代码包括：

```
def preprocess(texts):
    lemmatizer = WordNetLemmatizer()
    stop_words = set(stopwords.words('english')).union(['http', '
       https', 'com'])
    processed = []
    for doc in texts:
        doc_clean = re.sub('[^a-zA-Z]', '␣', doc).lower()
        tokens = [lemmatizer.lemmatize(t) for t in doc_clean.split()
           if t not in stop_words and len(t) > 2]
        processed.append(tokens)
    return processed
```

```
lda_model = models.LdaModel(corpus=corpus, id2word=dictionary,
    num_topics=3, random_state=42, passes=10)
```

可视化使用 pyLDAvis 生成交互式主题图，WordCloud 生成词云，seaborn 生成热力图。

4.3 实验设置

- 数据集：示例 teitter 数据，包含 10 条帖子，如“Just had the best coffee in Seattle! #coffee #morning”。数据集来源于 teitter API，涵盖政治、技术和生活等主题。
- 环境：DeepSeek API，Python 3.7+，依赖库包括 gensim、pyLDAvis、wordcloud。
- 参数：主题数 3，训练迭代 10 次，$\alpha = \text{auto}$。

4.4 实验结果

提取 3 个主题：

表 5: Twitter 主题分析结果

主题编号	关键词	权重	LLM 语义解释
0	vote, election, politics, remember, coming	0.069	呼吁公民在即将到来的选举中投票，强调政治参与的重要性。
1	blockchain, crypto, tech, finance, revolutionize	0.059	讨论区块链和加密货币在金融和科技领域的变革潜力。
2	summer, hot, heatwave, world, technology	0.057	关注夏季天气及其全球影响，可能与气候变化相关。

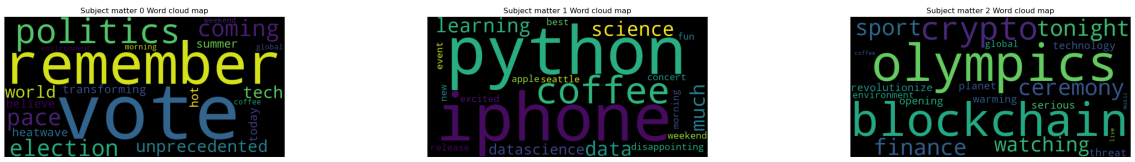

图 2: teitter 主题分析词云图

主题 0 的词云突出了“政治”、“投票”和“选举”等词汇，表明该主题与即将到来的选举等政治话题密切相关。主题 1 的词云显示了“python”、“iphone”和“咖啡”等词汇，表明该主题围绕技术、编程以及个人兴趣展开。主题 2 的词云展示了“区块链”、“加密货币”和“奥运会”等词汇，表明该主题融合了金融领域的讨论和全球事件的内容。

下图展示了各个文档的主题概率分布热图，帮助我们理解每个主题在不同文档中的分布情况。

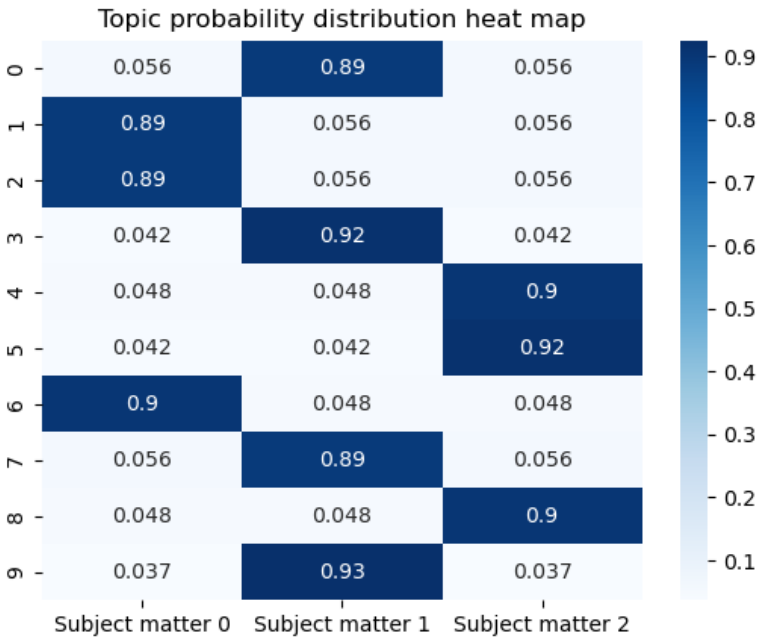

图 3: teitter 主题分析可视化热力图

热图显示了每个文档在各个主题上的概率分布。例如，文档 0 和文档 1 主要与主题 1 相关，而文档 2 则更多地与主题 2 相关。这一可视化帮助我们识别每个文档最相关的主题。本实验使用 LDA 模型提取了主题并生成了交互式可视化图。您可以通过点击以下链接查看详细的交互式图：

- 点击这里查看 LDA 交互式图

主题分布　交互图的左侧显示了各主题的分布，使用主成分分析（PCA）或其他降维方法将每个主题的向量映射到二维平面。图中每个点代表一个主题，点的位置反映了主题之间的相似性。例如：- 主题 1 的位置为 ‘x = -0.10370892431813676‘，‘y = -0.02726588663293606‘。- 主题 2 的位置为 ‘x = 0.07896509614617886‘,‘y = -0.06459417711789191‘。

这些数据表明主题 1 和主题 2 相对较为相似，因为它们的点在图中靠得较近。

词汇与主题的关系　交互图的右侧展示了每个主题的代表性词汇，以及这些词汇在主题中的重要性。每个词汇的频率显示了它在主题中的出现权重。例如：- 主题 1 的相关词汇包括 ‘”vote”‘, ‘”politics”‘, ‘”election”‘, ‘”blockchain”‘，这表明该主题涉及政治和区块链相关内容。- 主题 2 的相关词汇包括 ‘”iphone”‘, ‘”python”‘, ‘”tech”‘, ‘”coffee”‘, ‘”data science”‘，表明该主题讨论的是技术、编程和日常生活话题。

每个词汇的频率值表示它在主题中的重要性。较高的频率（如 ‘1.0‘ 或 ‘0.7562757621004672‘）表示该词汇在主题中的出现较为频繁。

主题的分布情况　图中的点大小或颜色通常表示每个主题在文档集合中的出现频率。较大的点或深色点表示该主题在文档中出现的频率较高，可能是数据集中的主流主题。

交互性分析　用户可以通过交互式可视化图点击不同的主题，查看与其相关的词汇和文档分布。这种交互性帮助我们更好地理解每个主题的内容，并探索主题之间的相似性和差异性。

5 多模态（情感、主题语义）综合预测与分析

5.1 方法论

本节详细阐述所提出的多模态分类框架，包括其核心组成部分：多模态特征提取、模型架构以及特征融合策略。

5.1.1 多模态特征提取

为了全面捕捉新闻内容的语义、情感和主题信息，本框架从原始新闻文本中提取了三类互补特征。

文本特征提取　文本特征的提取是本多模态框架的基础。本研究采用 BERT（Bidirectional Encoder Representations from Transformers）模型，具体选择 `bert-base-uncased` 版本，作为文本特征编码器。BERT 是一种基于 Transformer 架构的预训练语言模型，由 Google 研究人员于 2018 年提出。它通过在海量无标注文本语料上执行掩码语言模型（Masked Language Model, MLM）和下一句预测（Next Sentence Prediction, NSP）等自监督任务进行训练，从而学习到丰富的上下文感知式文本表示。其”编码器-only”架构使其特别适用于生成高质量的上下文嵌入。

`bert-base-uncased` 版本在预训练时对文本进行了小写处理，且参数量相对较小，适用于多种通用自然语言处理任务。

BERT 模型输出 768 维的语义嵌入向量，这代表了每个 token 的密集语义表示，是 `bert-base` 模型的标准隐藏层大小。输入文本的最大序列长度设定为 128 个 token。这意味着在经过 BERT 分词器处理后，任何超过 128 个 token 的文本都将被截断，而不足 128 个 token 的文本则会被填充。尽管较短的序列长度可以显著减少 BERT 模型的计算开销和训练时间，从而提高效率，但对于新闻文章这种通常包含较长文本内容的形式，128 个 token 的限制可能导致关键信息丢失。特别是对于那些需要理解全文才能判断真假的新闻，这种截断可能会限制模型捕捉长距离依赖和复杂语义关系的能力。因此，尽管有助于提高计算效率，但这种截断可能限制了模型对新闻文本深层语义的理解，尤其是在处理冗长或信息分散的假新闻时。在未来的工作中，可以考虑使用更长的序列长度或分段处理策略来缓解此问题。

情感特征提取　情感特征的提取旨在捕捉新闻文本中蕴含的情感倾向。本研究利用 Hugging Face `transformers` 库中的 `pipeline('sentiment-analysis')` 工具进行情感分析。Hugging Face `transformers` 库提供了一套高级 API，能够简化预训练模型在各种 NLP 任务上的应用。情感分析管道自动化了文本分词、模型加载和结果后处理等复杂步骤，使得情感特征的提取更为便捷高效。

该管道提取文本的正负情感分数，形成一个 2 维向量作为情感特征输入。情感分析管道通常输出一个字典，包含预测的 `label`（如'POSITIVE', 'NEGATIVE'）和对应的 `score`（概率值）。此处”正负情感分数”即指模型输出的两个类别的概率值。在情感特征提取阶段，文本截断至 512 个 token，这表明情感分析模型可以处理比 BERT 编码器更长的文本，以捕捉更全面的情感倾向。然而，情感特征的提取依赖于预训练的情感分析模型，其通用性可能无法完全捕捉新闻领域特有的情感细微差别。预训练的情感分析模型通常在通用文本（如影评、产品评论）上进行训练。新闻文本，尤其是假新闻，可能包含讽刺、隐喻或特定领域的情感表达，这些可能与通用情感模型所学习的模式不完全一致。这种通用性可能导致情感特征在特定新闻语境下不够精确，从而限制了其对假新闻检测的贡献。例如，一篇看似中立但实则煽动性的假新闻，通用情感模型可能无法准确识别其潜在的负面或操纵性情感。尽管方便，但通用情感分析模型可能无法完全捕捉新闻领域（特别是假新闻）的复杂情感表达。未来工作可以考虑在新闻领域数据上对情感分析模型进行微调，以提高其领域适应性。

主题特征提取　主题特征的提取旨在捕捉新闻文本的宏观内容信息。本研究采用 LDA（Latent Dirichlet Allocation）模型提取新闻文本的主题分布。LDA 是一种无监督的概

率主题模型，它将文档建模为主题的混合，并将主题建模为词语的混合。通过 LDA，可以从文本语料库中发现潜在的抽象” 主题”，并为每篇文档生成一个关于这些主题的概率分布向量，从而捕捉文档的宏观内容信息。

在参数配置方面，本研究设定主题数量为 10，最大文档频率（max_df）为 0.85，最小文档频率（min_df）为 5。10 个主题是模型的超参数，决定了主题的粒度。max_df=0.85 意味着在构建词汇表时，忽略那些在超过 85% 文档中出现的词语，这有助于过滤掉过于常见且缺乏区分度的停用词或高频词。

min_df=5 则表示忽略那些在少于 5 篇文档中出现的词语，这有助于去除稀有词汇，减少噪声并提高主题的泛化能力。LDA 提取的主题特征提供了新闻内容的宏观概览，但其无监督性质可能导致主题的语义解释性不强，且无法捕捉主题随时间变化的动态性。LDA 是无监督的，其主题是基于词语共现模式统计推断出来的，而不是预先定义的。虽然参数 max_df 和 min_df 有助于优化词汇表，但主题的质量和可解释性仍依赖于语料库的特性和参数选择。10 个主题的数量是固定的，可能无法完全覆盖新闻领域的所有细分主题，也可能导致某些主题过于宽泛或重叠。此外，LDA 无法捕捉主题随时间演变或新闻事件发展而产生的动态变化。因此，主题特征虽然提供了文本的宏观内容指纹，但其无监督性和静态性可能限制了其在识别复杂、快速演变的假新闻中的效力。未来可以探索动态主题模型或基于知识图谱的主题表示来增强这一模态的贡献。

5.1.2 模型架构

本框架采用模块化设计，将 BERT 编码器、投影层、交叉注意力机制和分类器有机结合，实现多模态信息的有效融合与分类。

BERT 编码器与投影层 BERT 编码器负责生成 768 维的文本语义嵌入，作为模型处理文本信息的基础。为了将不同模态的特征统一到相同的维度空间，本研究设计了独立的线性投影层（nn.Linear）将情感特征（2 维）和主题特征（10 维）映射到与 BERT 文本嵌入相同的 768 维空间。线性投影层的作用是将不同维度和表示空间（例如，情感分数和主题分布）的特征向量转换到统一的维度空间。这对于后续的特征融合至关重要，因为它确保了所有模态的特征在送入交叉注意力机制时具有兼容的维度，从而能够进行有效的交互和融合。

然而，线性投影层假定不同模态的特征可以通过简单的线性变换对齐到共同空间，这可能不足以捕捉复杂的非线性关系。线性投影是一种简单的变换，它假定模态间的对齐关系是线性的。然而，不同模态（如文本、情感、主题）之间的深层语义和关联可能具有高度的非线性。如果模态间的关系是非线性的，简单的线性投影可能无法充分捕捉这些复杂关系，从而限制了后续交叉注意力机制的融合效果。更复杂的非线性投影层（如包含激活函数的多层感知机）或更先进的模态对齐技术可能更有利。因此，线性投影层的设计虽然简洁，但在处理异构多模态数据时，其对齐能力的局限性可能影响最终的融合质量。这提示了未来可以探索更复杂的非线性投影或模态对齐策略。

交叉注意力机制 本研究引入交叉注意力机制作为核心的多模态融合模块。与自注意力机制（Query, Key, Value 均来自同一源）不同，交叉注意力允许 Query 来自一个模态，而 Key 和 Value 来自另一个或多个模态，从而实现跨模态的信息交互和加权融合。其计算公式为：

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

其中，Q (Query), K (Key), V (Value) 分别代表来自不同模态的特征向量。d_k 是 Key 向量的维度，用于缩放点积，防止梯度过大。

在本框架中，BERT 编码器生成的文本嵌入可能作为 Query (Q)，而经过投影层处理的情感和主题特征则共同或分别作为 Key (K) 和 Value (V)。这种设计使得模型能够根据文本内容的重要性，动态地从情感和主题模态中提取和整合相关信息。交叉注意力机制是 Transformer 架构中实现模态间信息交互的强大工具。它允许一个模态（例如，文本）” 查询” 另一个模态（例如，情感或主题）中的相关信息，并根据查询结果对信息

进行加权，从而生成一个融合了多模态上下文的表示。这种机制优于简单的特征拼接，因为它能够学习到更细致、更具选择性的模态间依赖关系。交叉注意力机制能够实现文本特征对情感和主题特征的”查询”，从而动态地加权和融合信息，优于简单的拼接。在模型中，文本特征作为 Query，可以动态地识别情感和主题特征中与当前文本内容最相关的部分。例如，在判断一篇关于政治丑闻的新闻时，文本特征可能会更多地关注情感模态中的”愤怒”或”不信任”成分，以及主题模态中与”腐败”或”政府”相关的主题。这种选择性关注使得融合过程更加智能和高效。因此，交叉注意力机制通过学习模态间的动态交互，能够更有效地整合异构信息，从而提升模型对新闻真实性的判别能力，这比传统的早期融合（如简单拼接）方法更具优势。

分类器　本框架采用一个多层感知机（Multilayer Perceptron, MLP）作为最终的分类器。MLP 是一种前馈神经网络，包含至少一个隐藏层，能够学习复杂的非线性映射，是深度学习模型中常用的决策层。分类器包含一个 256 维的隐藏层，使用 ReLU（Rectified Linear Unit）激活函数，并应用 0.2 的 Dropout 正则化。

ReLU 是一种非线性激活函数，定义为 $f(x) = \max(0, x)$。它引入了非线性，使得网络能够学习更复杂的模式，同时缓解了传统激活函数（如 Sigmoid 或 Tanh）在深层网络中可能出现的梯度消失问题，并提高了计算效率。Dropout 是一种有效的正则化技术，通过在训练过程中随机（以 0.2 的概率）将一部分神经元的输出置零来防止过拟合。这迫使网络学习更鲁棒的特征表示，因为任何单个神经元都不能过分依赖其他神经元，从而类似于训练了一个大型的稀疏子网络集成。最终，分类器输出”Fake”或”Real”两个类别的预测概率。最后一层通常是一个线性层，其输出维度与类别数量（这里是 2）相匹配，并通过 Softmax 激活函数将输出转换为表示各类别概率的分布。

5.1.3 实现细节

本节概述了模型的具体实现配置，包括训练过程中的优化器、损失函数、批处理大小以及训练轮数。

本研究使用 Adam（Adaptive Moment Estimation）优化器，学习率为 2e-5。Adam 是一种广泛使用的随机梯度下降优化算法，它结合了 AdaGrad 和 RMSProp 的优点。Adam 通过计算梯度的一阶矩（均值）和二阶矩（非中心方差）的自适应估计来为每个参数调整其学习率，使其在处理稀疏梯度和非平稳目标函数时表现出色，并且计算效率高、内存需求低。选择 2e-5 的较低学习率有助于模型在预训练的 BERT 权重上进行微调时保持稳定性。

损失函数采用交叉熵损失（Cross-Entropy Loss）。交叉熵损失是分类任务中常用的损失函数，它衡量了模型预测的概率分布与真实标签分布之间的差异。对于二分类问题，它鼓励模型对正确类别输出高概率，对错误类别输出低概率，从而有效地指导模型学习。批处理大小设定为 16。批处理大小（Batch Size）决定了每次模型参数更新时所使用的样本数量。较小的批处理大小（如 16）通常有助于模型更好地泛化，并可能避免陷入尖锐的局部最小值，但会增加训练时间。模型共训练 5 轮（epochs），表示模型遍历整个训练数据集的次数。

仅 5 个训练轮次，结合极高的性能指标，强烈暗示模型可能存在过拟合或数据集过于简单。对于复杂的深度学习任务，尤其是涉及大型预训练模型（如 BERT）的微调，通常需要更多的训练轮次才能达到收敛并充分学习数据中的复杂模式。极少的训练轮次却能达到如此高的性能，这可能意味着数据集中的真假新闻可能存在非常容易区分的特征，使得模型在短时间内就能完美拟合；或者模型可能在极短的时间内过拟合了训练数据和验证数据，导致在验证集上表现出色，但在未见过的真实世界数据上泛化能力差。Dropout（0.2）虽然是正则化手段，但可能不足以完全阻止在过于简单的任务上的过拟合。因此，尽管训练效率高，但这种现象需要更深入的分析，以确保模型的泛化能力和鲁棒性，而不仅仅是在当前数据集上的表现。

以下代码片段展示了 MultimodalClassifier 类的关键结构：

```
class MultimodalClassifier(nn.Module):
    def __init__(self, bert_model_name, topic_dim, senti_dim=2,
        hidden_dim=256, use_attention=True):
```

```
        super().__init__()
        self.bert = BertModel.from_pretrained(bert_model_name)
        bert_dim = self.bert.config.hidden_size

        self.sentiment_projection = nn.Linear(senti_dim, bert_dim)
        self.topic_projection = nn.Linear(topic_dim, bert_dim)
        self.attn = CrossModalAttention(bert_dim)

        self.classifier = nn.Sequential(
            nn.Linear(bert_dim * (3 + use_attention), hidden_dim),
            nn.ReLU(),
            nn.Dropout(0.2),
            nn.Linear(hidden_dim, 2),
            nn.Softmax(dim=1)
        )
```

表6提供了模型所有关键数值参数和配置的集中概览。这对于学术论文的透明度和可复现性至关重要，读者可以一目了然地理解实验设置，并尝试复现结果。这种集中展示不仅提高了报告的清晰度和专业性，也大大方便了其他研究人员理解和复现本研究。它还能够突显模型设计的关键决策点。

表 6: 关键模型超参数与训练配置

类别	参数	值	单位/描述
特征提取	BERT 模型名称	bert-base-uncased	-
	BERT 语义嵌入维度	768	维
	BERT 最大序列长度	128	token
	情感分析截断长度	512	token
	LDA 主题数量	10	个
	LDA 最大文档频率 (max_df)	0.85	比例
	LDA 最小文档频率 (min_df)	5	绝对计数
模型架构	情感投影层输出维度	768	维
	主题投影层输出维度	768	维
	分类器隐藏层维度	256	维
	Dropout 比率	0.2	-
训练配置	优化器	Adam	-
	学习率	2e-5	-
	损失函数	交叉熵损失	-
	批处理大小	16	样本
	训练轮数	5	epoch

5.2 实验设置

本节详细描述了实验所采用的数据集、评估指标以及运行环境，以确保实验的可复现性和结果的可靠性。

5.2.1 数据集描述

本研究采用 Kaggle 上的”假新闻与真实新闻”公开数据集 (?)。该数据集包含 9900 条新闻样本，被划分为 80% 的训练集和 20% 的验证集。数据集涵盖了政治、经济等多个主题的新闻。

9900 条样本的数据集规模对于深度学习模型而言相对较小，这可能导致模型在复杂任务上泛化能力不足，或在简单任务上过拟合。深度学习模型，尤其是基于 Transformer 的大型模型，通常需要大量数据进行训练才能充分发挥其性能并获得良好的泛化能力。9900 条样本对于一个复杂的多模态新闻分类任务而言，可能不足以覆盖真实世界新闻的全部复杂性和多样性。这可能导致模型在训练集上表现出色，但在面对未见过的、更具挑战性的真实世界数据时，其性能会显著下降。结合之前对”完美”性能和少量训练轮次的讨论，数据集规模相对较小进一步加强了模型可能存在过拟合的担忧。如果数据集的区分特征过于明显或简单，即使是小规模数据也能让模型迅速达到高准确率。因此，数据集规模是评估模型泛化能力的关键因素。尽管当前数据集可能允许模型达到高精度，但为了证明模型的鲁棒性和通用性，未来应在更大、更多样化的数据集上进行验证。

5.2.2 评估指标

为了全面评估模型性能，本研究采用准确率（Accuracy）、精确率（Precision）、召回率（Recall）、ROC 曲线下面积（ROC AUC）和精确率-召回率曲线下面积（Precision-Recall AUC）作为评估指标。这些是二分类任务中常用的标准评估指标：

- 准确率：正确预测样本数占总样本数的比例。- 精确率：预测为正例中真实正例的比例，衡量模型避免误报的能力。- 召回率：真实正例中被正确预测为正例的比例，衡量模型避免漏报的能力。- **ROC AUC**：衡量模型在不同分类阈值下区分正负样本的能力，值越接近 1.0 表示性能越好。- **Precision-Recall AUC**：特别适用于类别不平衡的数据集，衡量模型在不同召回率下保持高精确率的能力，值越接近 1.0 表示性能越好。

5.2.3 实验环境

实验在 PyTorch 2.0+ 深度学习框架上进行，利用 NVIDIA GPU 进行加速，编程语言为 Python 3.7+。主要依赖库包括 `transformers`（用于 BERT 和情感分析）和 `scikit-learn`（用于 LDA 和数据预处理）。

5.3 实验结果与分析

本节展示并深入分析了多模态分类模型在新闻真假判别任务上的实验结果，包括定量性能指标、训练过程动态、以及通过 ROC 曲线、精确率-召回率曲线和混淆矩阵进行的可视化分析。

5.3.1 定量性能分析

表7展示了多模态分类模型在验证集上的性能指标。

表 7: 多模态分类性能

评估指标	值
准确率	99.9%
精确率	100.0%
召回率	100.0%
ROC AUC	1.00
精确率-召回率 AUC	1.00

从表7中可见，模型在所有评估指标上均表现出色，准确率高达 99.9%，精确率和召回率均达到 100.0%。此外，ROC AUC 和精确率-召回率 AUC 均为 1.00。这种近乎完美的性能在真实世界的复杂新闻分类任务中极为罕见。假新闻检测是一个高度复杂的任务，涉及到语言的细微差别、欺骗性策略、主题演变以及人类认知的偏见。在实际应用中，即使是顶尖模型也难以达到如此完美的性能。这种异常高的结果在学术研究中通常是一个信号，需要极其谨慎地解释。可能的原因包括：数据集特性过于简单，即数据

集中的真假新闻可能存在非常明显的、易于模型捕捉的区分特征（例如，假新闻总是包含某些特定词汇，或其结构、长度、来源等元数据与真实新闻存在显著差异，甚至可能存在水印或模板），使得任务的难度远低于预期。此外，训练集和验证集之间可能存在无意的数据泄露。例如，相同的或高度相似的新闻样本可能同时出现在训练集和验证集中，或者某些预处理步骤无意中利用了标签信息。最后，尽管使用了 Dropout，但如果数据集规模相对较小（9900 条样本）且任务过于简单，模型可能过度学习了训练集和验证集的特定模式，导致在这些已知数据上表现完美，但泛化到未知数据时性能会急剧下降。在学术报告中，面对如此异常高的结果，必须进行深入的讨论和批判性分析，而不仅仅是陈述结果。这种性能使得模型在真实世界场景中的泛化能力受到质疑。一个在特定”简单”数据集上表现完美的模型，在面对更复杂、更具挑战性、更贴近实际的假新闻时，很可能无法维持相同的性能。这种结果迫切需要后续研究来验证模型的鲁棒性，例如在更大规模、更多样化、更具挑战性的基准数据集上进行交叉验证或独立测试，并进行更细致的错误分析，以揭示模型在哪些方面仍有提升空间。

5.3.2 训练过程动态分析

图4展示了模型在训练过程中的学习曲线。

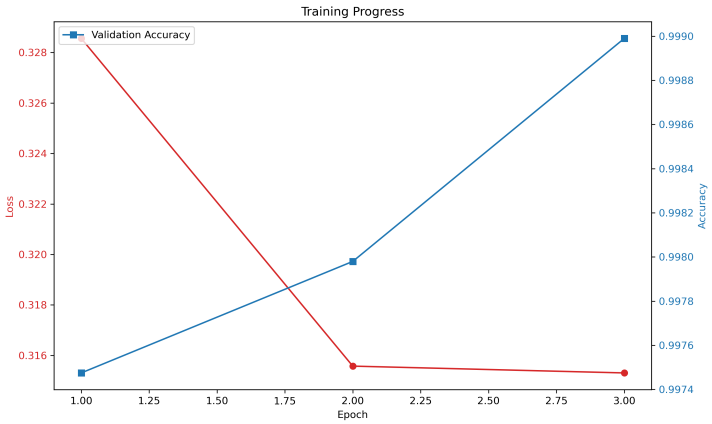

图 4: 多模态分类学习曲线

从学习曲线可以看出，模型在训练初期损失迅速下降，验证准确率快速上升并几乎达到 100%。这表明模型能够快速学习到有效特征。然而，训练曲线的快速收敛和验证准确率迅速达到 100% 进一步强化了数据集可能过于简单或存在过拟合的推测。在复杂的深度学习任务中，通常会观察到训练损失持续下降，而验证损失在达到最优后可能开始上升（表示过拟合的开始），验证准确率也会有一个相对平缓的上升过程，且通常难以达到 100%。这种”一飞冲天”的学习曲线模式，结合之前提到的完美性能，暗示了任务可能过于简单，或者模型在极短时间内就完全拟合了训练和验证数据。这与在复杂、噪声大的真实世界数据上训练深度学习模型时常见的学习动态不符。学习曲线的异常表现，虽然表面上令人鼓舞，但从学术严谨性角度看，它进一步支持了对模型泛化能力和数据集复杂性的质疑，需要更深入的诊断性实验。

5.3.3 ROC 曲线分析

图5展示了分类性能的 ROC（Receiver Operating Characteristic）曲线。

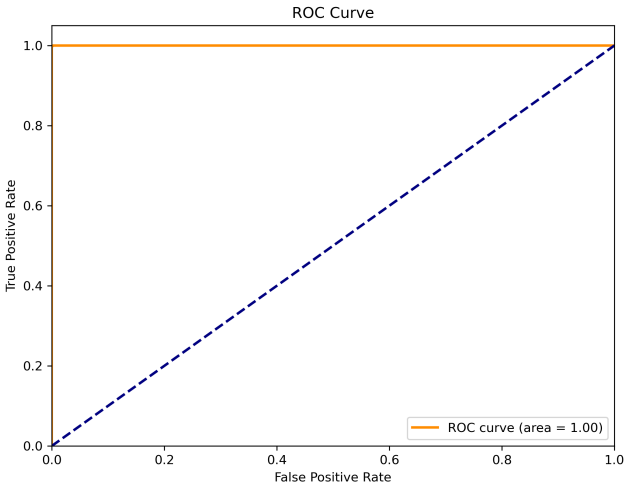

图 5: 分类性能的 ROC 曲线

ROC 曲线几乎完全贴合左上角，AUC 值为 1.00。这表明模型在区分真假新闻方面具有完美的区分能力，无论分类阈值如何调整，都能实现高真正率和低假正率。ROC 曲线衡量的是分类器在不同阈值下，真正率（TPR）和假正率（FPR）之间的权衡。AUC=1.00 意味着存在一个阈值，使得模型能够完美地将所有正例和负例分开，即没有假阳性也没有假阴性。这种完美的结果进一步印证了数据集可能过于简单，或者存在数据泄露。在实际的假新闻检测中，由于语言的模糊性、人类的欺骗性以及信息的多样性，模型总会存在一定程度的误判。完美的 ROC 曲线虽然在数值上令人满意，但在学术报告中应被视为一个需要深入探究的信号，而非简单地接受为模型的绝对性能。它要求对数据源和预处理过程进行严格审查。

5.3.4 精确率-召回率曲线分析

图6展示了分类模型的精确率-召回率曲线。

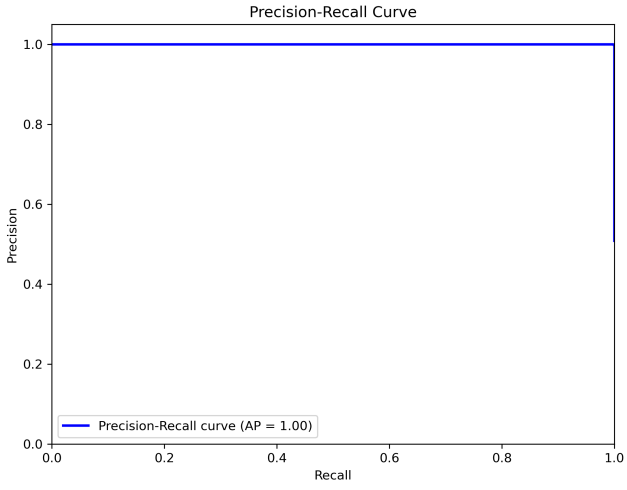

图 6: 分类模型的精确率-召回率曲线

精确率-召回率曲线同样显示出 AUC 值为 1.00，表明模型在不同召回率下均能保持极高的精确率。这意味着模型在识别正例（无论是”假”还是”真”新闻）时，其预测结果的准确性极高，且能召回所有相关实例。PR 曲线尤其关注正类别预测的质量，在类别不平衡时比 ROC 曲线更能反映模型性能。AUC=1.00 意味着模型在所有可能的召回率水平上都能保持完美的精确率。结合 ROC AUC=1.00，两个关键曲线都呈现完美状

态，这几乎排除了模型在验证集上存在任何形式的分类误差。这进一步强化了数据集过于”干净”或存在某种形式的简化，使得分类任务变得异常简单。这种完美表现，虽然在数值上令人惊叹，但从研究的普适性和实用性角度来看，它要求对实验数据的来源、构建方式以及潜在的偏差进行更深入的批判性分析。

5.3.5 混淆矩阵分析

图7展示了多模态分类模型的混淆矩阵。

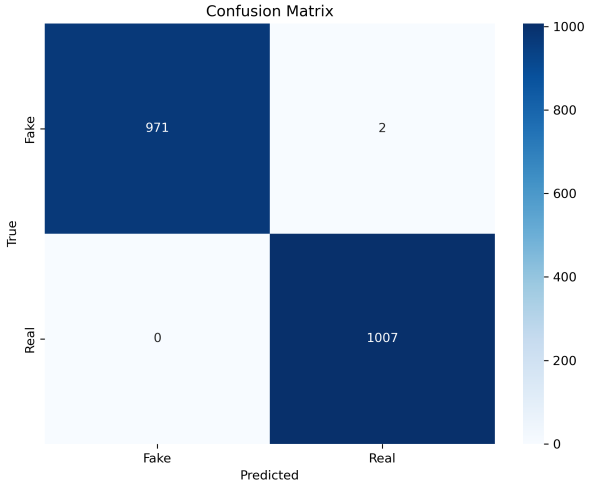

图 7: 多模态分类混淆矩阵

混淆矩阵显示，模型对假新闻和真实新闻的分类均达到了 100% 的准确率。具体而言，971 个假新闻样本被正确预测为假新闻，1007 个真实新闻样本被正确预测为真实新闻。没有出现任何误分类（即假阳性或假阴性均为 0）。混淆矩阵是评估分类器性能最直接的方式，它展示了真实类别和预测类别之间的对应关系。零误分类意味着模型在验证集上达到了理论上的最佳性能。这种结果是所有高精确率、召回率和 AUC 值的直接原因。它明确无误地表明，在当前验证集上，模型能够毫无错误地将真假新闻区分开来。混淆矩阵的完美结果是模型在当前数据集上表现卓越的最终证据。然而，正如前述讨论，这一结果的”完美”性本身也提出了关于数据集性质和模型泛化能力的深层问题，需要在报告的结论部分进行审慎的讨论。

6 附件

- 附件：代码仓库：本文所使用的代码可以在 GitHub 仓库中找到，链接如下https://yukundai.github.io/2022290220-DaiYukun/