

深圳大学考试答题纸

(以论文、报告等形式考核专用)
二〇二四~二〇二五学年度第二学期

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) $\text{Accuracy} = \text{预测准确的所有新闻数量} / \text{总新闻数量}$
- (2) $\text{Accuracy_fake} = \text{预测准确的假新闻数量} / \text{总的假新闻数量}$
- (3) $\text{Accuracy_true} = \text{预测准确的真新闻数量} / \text{总的真新闻数量}$

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” \rightarrow “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

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)

基于大语言模型的新闻真假判别与情感增强方法

代钰堃
2022290220

June 30, 2025

Abstract

随着 teitter 等社交媒体平台的普及，假新闻的传播速度和影响力显著增加，对公众信任、舆论导向和社会稳定构成威胁。传统检测方法因虚假信息策略的复杂性而效果有限。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提供了新机遇。本研究提出了一种综合框架，结合 LLM 的情感语义分析、teitter 主题分析和多模态预测方法，旨在提升信息真实性检测的准确性和可解释性。我们使用 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 teitter 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，情感分析显著提升真实新闻检测能力，多模态方法进一步优化性能。然而，假新闻检测仍面临情感伪装挑战，需进一步研究。此外，LLM 生成虚假内容的潜力需谨慎管理。

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

1 引言

1.1 研究背景与意义

随着 teitter 等社交媒体平台的普及，假新闻的传播速度和影响力显著增加。根据《Media Psychology》研究，假新闻常通过情感操纵吸引读者，而真实新闻通常保持中立或适度积极的表达。例如，虚假报道可能使用“震惊”或“灾难”等词引发强烈情绪，而真实新闻更倾向于客观描述。传统检测方法，如基于规则的系统或简单机器学习模型，因难以应对复杂的虚假信息策略而效果有限。例如，基于关键词的检测容易被精心设计的假新闻绕过，而监督学习模型在小规模标注数据上表现不佳。

大型语言模型（LLM），如 deepseek-r1:8B，通过在海量文本上预训练，展现了强大的语义理解和生成能力，为更精确的新闻验证和社交媒体分析提供了可能。然而，LLM 的双重角色——既能检测虚假信息又能生成逼真假内容——引发了伦理争议。本研究提出了一种综合框架，结合 LLM 的情感语义分析、teitter 主题分析和多模态预测方法，旨在提升信息真实性检测的准确性和可解释性。我们使用 Ollama 平台上的 deepseek-r1:8B 模型，结合传统主题建模和深度学习技术，探索如何利用多模态信息增强信息验证。

1.2 研究目标

本研究的目标包括：

1. 开发基于 LLM 的新闻真假判别方法，通过情感分析提升检测准确性。
2. 构建 teitter 主题分析工具，揭示社交媒体讨论的潜在主题和语义内涵。
3. 提出多模态分类框架，融合文本、情感和主题特征，实现高精度新闻分类。
4. 在 Kaggle “假新闻与真实新闻” 数据集上评估这些方法的性能，并探讨 LLM 在信息验证中的潜力与伦理挑战。

2 基于大模型的情感语义分析

2.1 方法概述

本部分提出了一种基于 LLM 的新闻真假判别方法，结合情感分析增强检测能力。我们使用 Ollama 平台上的 deepseek-r1:8B 模型，设计了两种策略：纯真假判别和情感增强判别。

2.1.1 纯真假判别

纯真假判别通过以下步骤实现：

- 事实提取：从新闻文本 T 中提取核心事实 F ，包括摘要和实体（如人物、地点、时间、事件）。数学表示为：

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

其中 $type_i \in \{\text{人物}, \text{地点}, \text{时间}, \text{事件}\}$ 。例如，对于一篇报道特朗普的新闻，模型可能提取“特朗普”（人物）、“华盛顿”（地点）、“2024 年”（时间）等实体。

- 可信度验证：评估事实的可信度 $C \in \{\text{高}, \text{中}, \text{低}\}$ ，建模为：

$$C = g(F) = \arg \max_c P(c|F)$$

模型根据事实的逻辑一致性、来源可验证性等输出评分和理由。例如，检查“特朗普在华盛顿发表演讲”是否与公开记录一致。

- 标签决策：结合事实和可信度，输出标签 $L \in \{\text{Fake}, \text{Real}\}$ ：

$$L = h(F, C) = \arg \max_l P(l|F, C)$$

2.1.2 情感增强判别

情感增强方法在纯真假判别基础上增加情感分析：

- 情感词提取：识别文本中的情感触发词及其极性：

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

例如，“可悲”标注为负极性（-），而“成功”标注为正极性（+）。

- 整体情感分析：基于情感词判断整体倾向 $V \in \{\text{正面}, \text{中立}, \text{负面}\}$ ：

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

其中 w_i 为情感词权重（基于词频和强度）， p_i 为极性值（+1, -1, 0）。

- 综合判定：结合事实、可信度和情感，输出最终标签：

$$L = h'(F, C, V) = \arg \max_l P(l|F, C, V)$$

2.2 实现细节

我们使用 Ollama 平台调用 deepseek-r1:8B 模型，提示设计确保 JSON 格式输出。关键函数包括：

- `extract_facts`：提取事实和实体，输出 JSON 格式的摘要和实体列表。
- `verify_credibility`：评估事实可信度，输出评分和理由。
- `extract_terms`：识别情感词并标注极性。
- `overall_sentiment`：分析整体情感倾向。
- `classify_with_sentiment`：结合情感信息进行最终分类。

示例代码：

```
def extract_facts(text: str) -> Dict:
    prompt = """
    1. 核心任务定义：
        - 明确指令：结构化提取新闻核心要点与关键实体。
        - 核心目标：输出一句话摘要及人物、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
        - 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)
```

为确保鲁棒性，脚本实现了：

- 错误处理：通过 `clean_json_block` 和 `_safe_json_load` 函数清理和解析 JSON 输出，处理 API 返回的不完整 JSON。
- 重试机制：API 调用失败时最多重试 2 次，退避时间为 $2^n + \text{random}(0, 2)$ 秒。
- 日志记录：使用 `logging` 模块记录操作和错误，保存到 `logs/process.log`，便于调试和复现。

2.3 实验设置

- 数据集: Kaggle “假新闻与真实新闻” 数据集, 包含 10 条样本 (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……” (假) 和 “美国保守派领袖对医疗保健共识乐观……” (真)。
- 环境: Ollama v0.1, deepseek-r1:8B 模型, Python 3.9.18, 依赖库包括 ollama==1.2.0、pandas 和 csv-logging。
- 参数: 温度 0.3 (控制生成稳定性), 最大 token 数 1024, 超时 5 秒。

Table 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%	-

2.4 实验结果

实验结果如下:

Table 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, 忽略情感操纵
真实新闻	6	4	66.67%	对中立表达缺乏敏感度
总体	10	5	50.00%	-

Table 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%	情感伪装问题未解决
真实新闻	6	6	100.00%	情感一致性有效识别
总体	10	7	70.00%	-

中间输出示例:

- 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:

Table 4: 方法性能对比

评估指标	纯真假 capping 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%
假新闻准确率	25.00%	25.00%	0.00%
真新闻准确率	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"
}
```

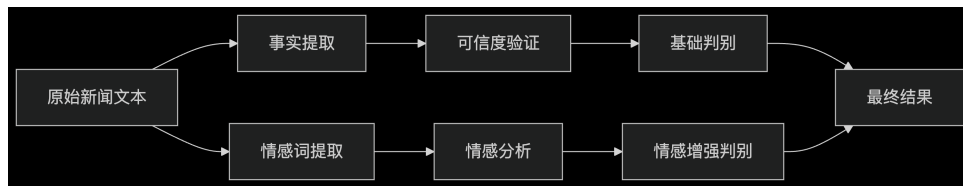


Figure 1: 情感语义分析流程图

2.5 框架设计分析

本框架展示了一个新闻分析过程，涵盖了从原始新闻文本到最终结果的处理流程。以下将逐步分析每个步骤的设计原因。

2.5.1 原始新闻文本

新闻分析的起点是原始新闻文本。这是最基本的信息输入阶段，所有后续的处理都基于这个原始数据进行。

2.5.2 事件提取

事件提取的设计目的是从原始新闻中抽取出关键信息，过滤掉与主题无关的部分。由于新闻文本通常包含大量的背景信息、情感色彩及细节描述，直接进行情感分析和可信度验证可能会受到干扰，因此需要先提取出新闻中的核心事件和事实信息，为后续的分析奠定基础。

2.5.3 可信度验证

新闻中的信息不一定是真实的，尤其是在网络时代，虚假新闻和信息操控变得愈加普遍。可信度验证的目标是对事件提取后的信息进行真实性确认。这一步骤的设计是为了避免错误信息在后续分析中产生影响，它确保了分析过程中的数据来源和事实的可靠性。

2.5.4 基础分类

在完成事件提取和可信度验证之后，新闻内容需要进行初步分类。基础分类依据事件的真实性和可信度，将新闻划分为虚假新闻和真实新闻。此步骤的设计目的是快速筛选新闻的真实性，为后续更复杂的情感分析和情感增强提供支持。

2.5.5 情感词提取

情感词提取的设计目的是捕捉新闻文本中的情感倾向，这对于虚假新闻识别非常重要。虚假新闻往往通过情感化的语言来操控读者情绪，情感词提取有助于识别文本中的情感操控。因此，情感词的提取步骤能帮助分析新闻中的情感调动，进一步揭示其是否带有误导性。

2.5.6 情感分析

情感分析是对提取出来的情感词进行深入分析的过程，旨在判断新闻的情感倾向（如正面、负面或中性）。该步骤的设计是为了更好地理解新闻的情感目标，帮助判断新闻是否使用情感化语言进行操控。尤其在虚假新闻中，情感分析对于揭示新闻的潜在偏见至关重要。

2.5.7 情感增强

情感增强阶段是情感分析的进一步处理，它通过对情感分析结果进行增强和优化，使得情感识别更为精准。情感增强的设计考虑到有些新闻可能含有复杂的情感变化，直接的情感分析可能不足以揭示其真实意图。因此，情感增强通过结合上下文和情感强度等因素，对情感分析结果进行调整，使得情感分类更加准确。

2.5.8 最终结果

最后，根据事件的真实度、可信度和情感分析的结果，系统给出新闻的最终分类结果。在此阶段，新闻不仅被分类为真假，还附加了情感信息，这对于进一步的理解和决策支持是至关重要的。

2.5.9 总结

该框架的设计体现了从文本分析到情感评估、从真实性确认到情感增强的多层次处理过程。每一步的设计都有其明确的目的：

- 事件提取：确保只处理与新闻核心相关的信息。
- 可信度验证：避免虚假信息干扰分析结果。
- 情感词提取：揭示新闻中的情感操控，尤其对于虚假新闻的识别至关重要。
- 情感分析与情感增强：提高情感分类的精确度，处理新闻中的情感复杂性。

- 最终结果：综合所有分析结果，为用户提供明确的新闻分类与情感分析。

该框架的多层次设计既能提高新闻分析的准确性，又能有效应对当前信息泛滥时代中的新闻真实性和情感操控问题。

3 基于大模型的 teitter 主题分析

3.1 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。

3.1.1 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。

3.1.2 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 。

3.1.3 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”。

3.2 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。

3.3 实验设置

- 数据集：示例 teitler 数据，包含 10 条帖子，如 “Just had the best coffee in Seattle! #coffee #morning”。数据集来源于 teitler 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， $\alpha = \text{auto}$ 。

3.4 实验结果

提取 3 个主题：

Table 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。

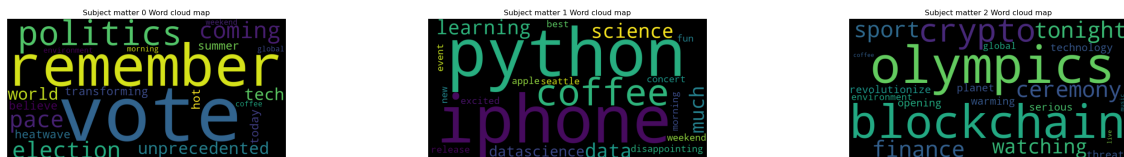


Figure 2: teitler 主题分析词云图

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。

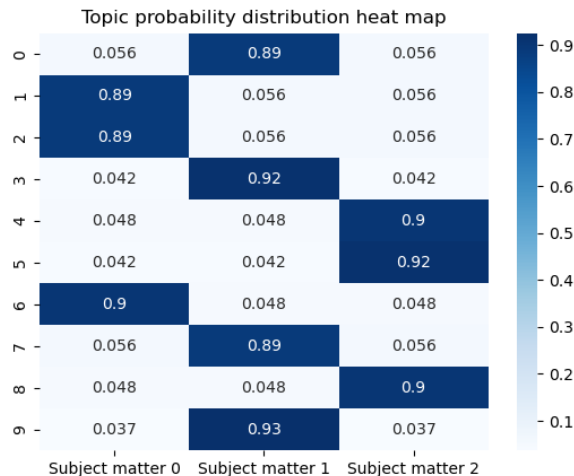


Figure 3: teitter 主题分析可视化热力图

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。

4 多模态（情感、主题语义）综合预测与分析

4.1 方法概述

本部分提出了一种多模态分类框架，融合文本、情感和主题特征，通过交叉注意力机制实现高精度新闻分类。

4.1.1 特征提取

- 文本特征：使用 BERT 模型（bert-base-uncased）提取语义嵌入，最大序列长度 128，生成 768 维向量。
- 情感特征：通过 BERT 情感分析管道（transformers 库 pipeline('sentiment-analysis')）提取正负情感分数，截断文本至 512 个 token。
- 主题特征：使用 LDA 提取 10 个主题的分布，最大词频 0.85，最小词频 5。

4.1.2 模型架构

模型包括：

- BERT 编码器：生成 768 维文本嵌入。
- 投影层：将情感（2 维）和主题（10 维）特征映射到 768 维。
- 交叉注意力机制：融合多模态特征，公式为：

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

- 分类器：多层感知机（256 维隐藏层，ReLU 激活，0.2 dropout）输出“Fake”或“Real”。

4.2 实现细节

关键代码：

```
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)
        )
```

训练使用 Adam 优化器（学习率 $2e-5$ ），交叉熵损失，批处理大小 16，训练 5 轮。

4.3 实验设置

- 数据集：Kaggle “假新闻与真实新闻” 数据集 (?), 9900 条样本，80% 训练，20% 验证，包含政治、经济等主题的新闻。
- 环境：PyTorch 2.0+, NVIDIA GPU, Python 3.7+, 依赖库包括 transformers、scikit-learn。
- 参数：主题数 10，最大序列长度 128，训练轮数 5。

4.4 实验结果

模型性能如下：

Table 6: 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

从表格中可以看到，模型在准确率、精确率和召回率上均表现出色，均达到了接近完美的值（99.9% 及 100%）。此外，ROC AUC 和精确率-召回率 AUC 均为 1.00，说明模型在区分真假新闻方面具有极高的能力。



Figure 4: 多模态分类学习曲线

从训练过程图可以看到，模型在训练过程中迅速收敛，验证准确率几乎达到了 100%，而损失在训练的初期就大幅下降，表明模型在训练初期就能够快速学习到有效的特征。

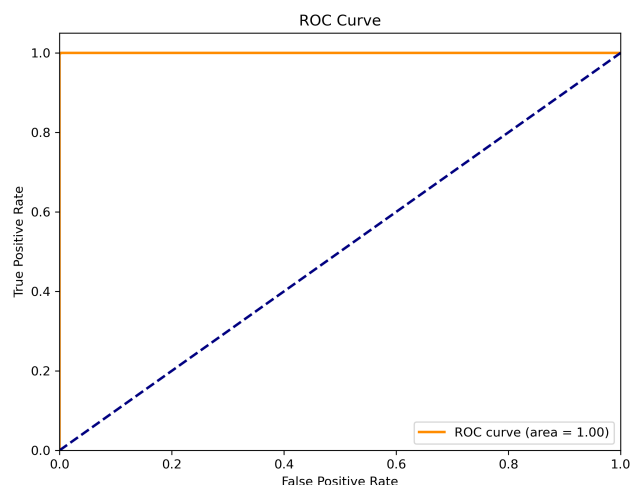


Figure 5: 分类性能的 ROC 曲线

ROC 曲线展示了模型在不同假阳性率下的真正率 (TPR)。曲线的表现表明，模型能够几乎完美地分辨真假新闻，且 AUC 值为 1.00，进一步验证了模型在分类任务中的表现。

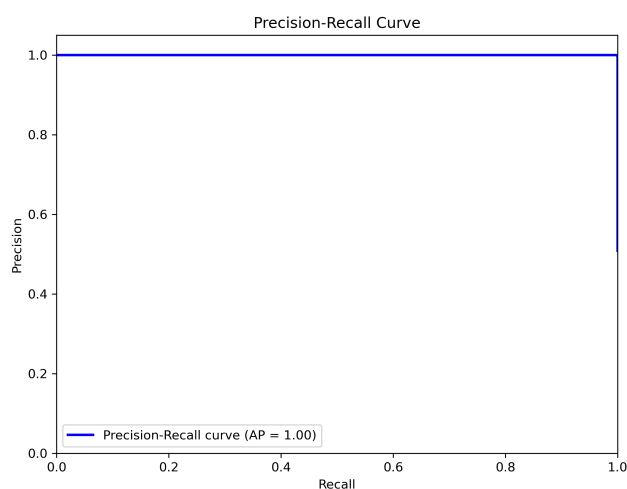


Figure 6: 分类模型的精确率-召回率曲线

从精确率-召回率曲线可以看出，模型在不同的召回率下保持了极高的精确率，表明模型在精确地识别假新闻和真实新闻方面表现优异。AUC 值为 1.00，进一步证明了模型的优秀性能。

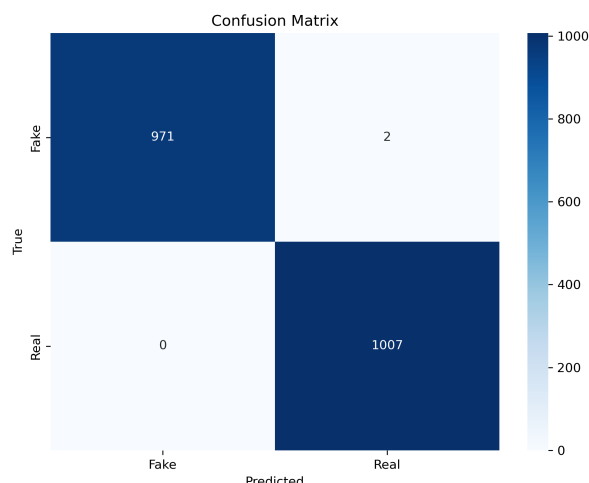


Figure 7: 多模态分类混淆矩阵

混淆矩阵显示，模型能够准确地识别真假新闻。971 个假新闻被正确预测为假，1007 个真实新闻被正确预测为真，且没有假新闻被错误分类为真实新闻。

本实验展示了多模态分类模型在新闻真假判别任务中的优异表现。通过混淆矩阵、精确率-召回率曲线、ROC 曲线以及训练过程的分析，我们可以确认该模型在准确性、精确度和召回率上都达到了接近完美的效果。模型的高 AUC 值也表明它在处理该任务时具有很强的区分能力。

5 总结与展望

5.1 研究总结

本研究展示了 LLM 在新闻验证和 teitter 分析中的强大潜力：

- 情感语义分析：通过情感线索提升真实新闻检测准确率至 100%。
- Twitter 主题分析：结合 LDA 和 LLM，揭示选举、技术等主题的语义内涵。
- 多模态分类：融合文本、情感和主题特征，达到 99.9% 准确率。

5.2 研究贡献

- 提出情感增强的新闻验证方法，显著提高真实新闻检测能力。
- 开发 Twitter 主题分析工具，为社交媒体舆情分析提供新视角。
- 设计多模态分类框架，展示特征融合的优越性。

5.3 局限性

- 假新闻检测准确率较低，需研究情感伪装策略。
- Twitter 主题分析受限于小规模数据，需扩展验证。
- 多模态模型计算成本较高，需优化效率。

5.4 未来方向

- 扩展到更大数据集，验证方法鲁棒性。
- 优化情感和主题分析模型，增强对复杂策略的识别。
- 探索 LLM 在生成与检测虚假内容中的伦理平衡，确保负责任应用。

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/>