**Modeling and Simulation** 建模与仿真, 2025, 14(5), 1051-1063
Published Online May 2025 in Hans. https://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2025.145456

Hans汉斯

# 面向风格转变的文本抄袭检测方法

**罗楚淋，周元鼎，韩彦芳**

上海理工大学光电信息与计算机工程学院，上海

收稿日期：2025年4月30日；录用日期：2025年5月23日；发布日期：2025年5月30日

## 摘　要

**近年来，随着大语言模型(LLM)的飞速发展，抄袭检测任务面临着前所未有的挑战。针对这一问题，文章提出了一种面向风格转变的检测模型。所提出模型通过结合BERT与图注意力网络，能够有效学习文本的风格特征并实现风格分类。同时，还巧妙地引入对比学习机制，进一步增强了文本的风格特征表示能力，从而显著提升了模型对写作风格改变的检测性能。实验结果表明，在PAN 2022写作风格改变检测数据集上，本文提出的模型相较于现有代表性方法取得了更优秀的检测效果。此外，通过消融实验验证了风格增强机制的有效性，并证明了图注意力网络在捕捉文本写作风格特征方面的优势。本文提出的方法不仅提高了风格转变检测的准确性，还为后续抄袭检测任务提供了前置条件。**

## 关键词

**写作风格改变，内在抄袭检测，风格增强，图神经网络，大语言模型**

# Style Change-Oriented Text Plagiarism Detection Method

**Chulin Luo, Yuanding Zhou, Yanfang Han**

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Apr. 30th, 2025; accepted: May 23rd, 2025; published: May 30th, 2025

## Abstract

**In recent years, with the rapid development of large language models (LLM), plagiarism detection is facing unprecedented challenges. To solve this problem, this paper proposes a style change oriented detection model. By combining BERT and graph attention networks, the proposed model can effectively learn text style features and realize style classification. At the same time, it also cleverly introduces a contrastive learning mechanism to further enhance the representation ability of text**

文章引用: 罗楚淋, 周元鼎, 韩彦芳. 面向风格转变的文本抄袭检测方法[J]. 建模与仿真, 2025, 14(5): 1051-1063.
DOI: 10.12677/mos.2025.145456

**style features, thus significantly improving the model's detection performance of writing style changes. The experimental results show that the model proposed in this paper achieves better detection results than the existing representative methods on the PAN 2022 writing style change detection dataset. In addition, the effectiveness of the style enhancement mechanism was verified through ablation experiments, and the advantage of the graph attention network in capturing stylistic features of text writing was demonstrated. The method proposed in this paper not only improves the accuracy of style change detection, but also provides preconditions for subsequent plagiarism detection tasks.**

## Keywords

**Writing Style Change, Intrinsic Plagiarism Detection, Style Enhancement, Graph Attention Network, Large Language Model**



## 1. 引言

近年来，随着深度学习的飞速发展，大语言模型(LLM)在文本生成领域取得了突破性进展[1]，能够生成高质量、流畅自然的文本。这些模型在辅助人们处理日常事务，如撰写邮件、生成报告等方面展现了巨大潜力。然而，大语言模型的快速发展也为学术不端行为提供了新的工具[2]。一些人利用大语言模型对已有学术成果进行改写、润色甚至重新组织[3]，生成看似原创的文本，从而规避传统的抄袭检测方法[4]。这种行为本质上可以视为一种“对抗性攻击”[5]，不仅破坏了学术诚信，也对现有的抄袭检测技术提出了新的挑战。针对大语言模型辅助抄袭这一新兴问题，写作风格检测技术显得尤为重要。写作风格是作者在长期写作实践中形成的独特语言表达习惯，具体体现在词汇选择、句式结构、修辞手法等多个方面[6]。在传统方法中，风格特征多依赖于手工设计的语言学特征，如词汇使用、句法结构、语气、情感等，这些特征通过简单的统计学方法或规则来提取，虽有一定的可解释性，但在处理复杂的风格差异时可能显得不足。而深度学习中的风格特征则是通过大规模预训练模型自动学习的，能够在更复杂的语言层面上捕捉风格差异，通过词嵌入、上下文依赖的语法结构分析、情感和语气分析等，能够提取出更加细粒度的风格特征，深度学习方法相比于传统方法在风格分析中具有更强的表现和灵活性。写作风格如同指纹一般，具有高度的辨识度，可以用于作者身份识别、文本风格分析等任务。通过分析文本中的写作风格变化，可以有效识别出大语言模型生成的文本片段，从而为后续的抄袭检测提供重要线索。

抄袭检测任务从是否有对照文本，可以分为外部抄袭检测[7]和内部抄袭检测[8]两类。外部抄袭检测是通过将可疑文本与可疑源文本进行对比评估；内部抄袭检测则是根据可疑文本自身的特征来评估是否存在抄袭行为。抄袭检测技术随着科技的不断进步和发展，主要经历了传统方法、基于机器学习的方法以及基于深度学习的方法三个发展阶段。传统方法完全依赖人工设计的规则和特征，其中较为有代表性的方法是最长公共子序列方法[9]，需要人为定义字符串匹配规则，或者需要人工基于词频统计的方法选择关键词或短语，这些特征的质量直接影响检测效果，它的优点是规则清晰，可解释性强，但缺点就是特征设计复杂，难以应对多样化的抄袭行为，并且对领域知识和人工经验有较强的依赖性。基于机器学习的方法虽然也需要人为设计特征，但是之后的特征处理和分类则是交给机器完成的，其中具有代表性的词频－逆文档频率(TF-IDF)方法[10]首先人工选取文本的统计特征，然后使用支持向量机(SVM) [11]等

模型对文本特征进行分类，其优点是能够识别部分改写抄袭行为，性能优于传统方法，且减少了对人工的依赖程度，模型可以自动学习特征之间的关系，缺点则是特征设计仍然依赖人工，特征的质量直接影响模型性能，且难以捕捉文本的深层语义信息。对于风格改变检测任务来说，其泛化能力差，对于未见过的风格变化表现较差，难以适应多样化的文本数据[12]，而且上下文建模能力有限，传统方法难以捕捉文本中的长距离依赖关系和上下文信息。

基于深度学习的方法，则是极大地减少了对人工的依赖，它无需人为设计特征，且模型能够自动从文本中学习特征表示，捕捉文本的深层语义信息，从而显著提升抄袭检测能力，除此之外，由于无需人工设计特征，也减少了对领域知识的依赖。但这类方法的缺点是需要消耗大量计算资源，需要海量标注数据来提升相关表现，且模型的可解释性较差。深度学习方法发展也分为两个阶段，第一个阶段主要以卷积神经网络(CNN) [13]和循环神经网络(RNN) [14]为代表，CNN 通过卷积操作提取局部特征，用于处理文本中的局部信息，RNN 则是通过循环结构捕捉序列数据中的时间依赖关系，两者都是通过多层神经网络逐步提取低层次到高层次的特征，但是存在的问题是两者难以捕捉文本中的长距离依赖关系，且特征提取能力受限于模型结构和数据规模，难以提取风格变化等细粒度特征；第二个阶段则是以 Transformer 为基础的各种模型，例如 BERT [15]、RoBERTa [16]、SBERT [17]以及 GPT-2 [18]等。BERT 能够同时考虑上下文信息，捕捉词语的双向关系。RoBERTa 则是 BERT 的改进版本，通过更大的数据集、更长的训练时间和动态掩码策略进行优化，去除下一句预测(NSP)任务，专注于掩码语言模型(MLM)任务。SBERT 也是基于 BERT 改进的模型，专门用于句子级的语义表示，通过孪生网络结构和三元组损失来优化特征表示。GPT-2 则是一种单向 Transformer 模型，通过自回归语言模型进行预训练。然而这些模型均只能对传统的文本任务进行处理，例如语义理解、普通文本分类等。现有的抄袭检测算法中，使用深度学习模型进行内在抄袭检测的方法还比较少。本文提出了一种基于预训练模型 BERT 和图注意力网络(GAT) [19]的面向风格转变的检测模型。通过增强 BERT 模型对写作风格的特征表示，再佐以图神经网络捕捉复杂结构的优势，在增强风格特征的同时，还能够对风格特征进行更全面的捕捉，从而提升风格变化检测的性能。本文的主要贡献如下：

1) 提出了一种新的内部抄袭检测模型。通过结合预训练模型 BERT 和图注意力网络(GAT)，能更好地提取特征，并体现特征表示之间的关系，从而提高模型对风格转变检测的性能；

2) 提出了一种风格特征增强机制，通过更新与风格特征相关的参数比重，使模型更注重写作风格方面的特征，而弱化其他特征，进一步提高模型的检测性能。

## 2. 模型应用场景

随着大语言模型(LLM)的快速发展，其应用范围不断扩大，包括日常事务处理(如写邮件、安排日程)以及学术领域的文本润色等。然而，这种技术也可能被滥用，例如将已有学术成果通过大语言模型转换为自身成果，从而规避传统抄袭检测机制。针对这一潜在风险，本文提出了一种面向细微风格改变的检测模型。与传统方法主要检测显著风格差异(如文学写作风格与学术写作风格)的任务不同，该模型通过分析文本内部的写作风格一致性，为防范抄袭行为提供技术解决方案。

本文模型采用内部抄袭检测方法，无需依赖外部比对文本，而是通过分析目标文本内部的写作风格特征进行判断。具体而言，模型通过检测文本是否存在多个作者的写作风格特征，为后续抄袭检测提供预判依据。当检测到文本存在多种写作风格时，系统将标记为疑似抄袭文本，并移交至后续检测环节。尽管模型在 PAN 2022 数据集(仅包含真人撰写的文本) [20]上进行训练和测试，但其学习到的风格特征识别能力可迁移至检测大语言模型改写文本的场景。当用户利用大语言模型对已有文本进行改写，导致改写后的文本与自身写作风格不一致时，本模型能够有效识别这种风格差异，从而为防范潜在的抄袭行为

提供技术支持，显著提高抄袭检测的整体效率。图 1 展示了模型的具体应用场景。

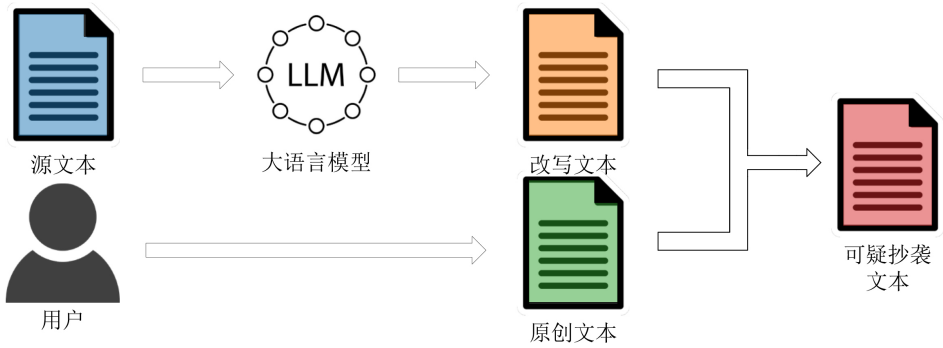



**Figure 1.** Model application scenarios
**图 1.** 模型应用场景

## 3. 所提方法

如图 2 所示，本文所提出的面向风格转变的文本抄袭检测模型主要由文本预处理、文本编码、写作风格增强机制以及图神经网络处理四部分组成。

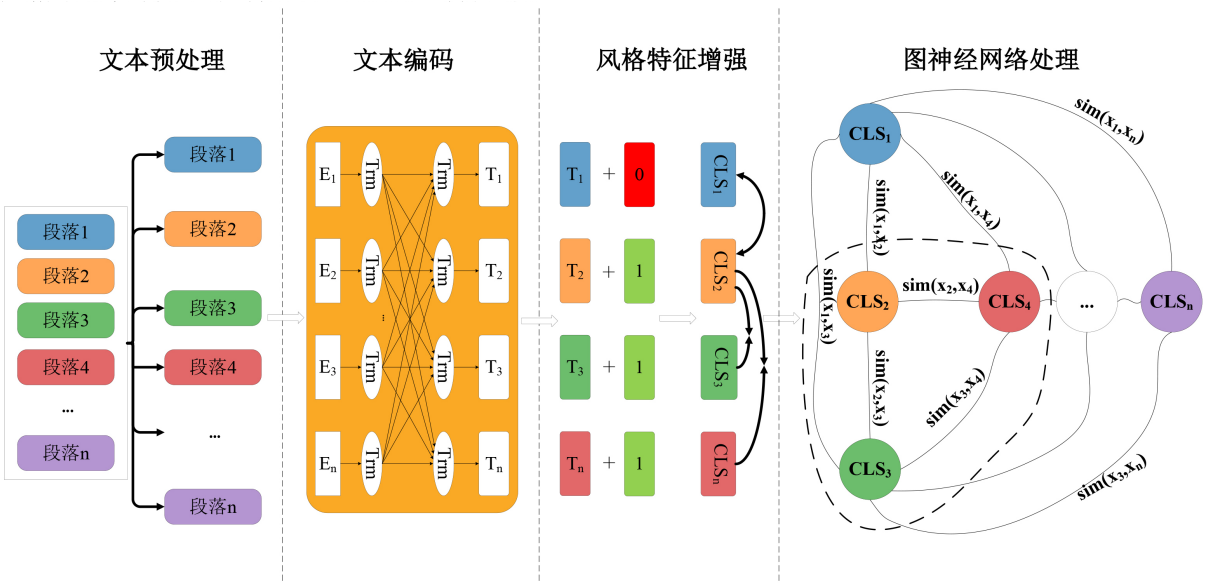



**Figure 2.** Overall framework diagram of our model
**图 2.** 模型总体框架图

首先，在文本预处理阶段，根据任务需求对输入文本进行处理：对于段落级任务，将文本按段落进行两两匹配；对于句子级任务，则按句子进行配对。随后，在文本编码阶段，将预处理后的文本输入预训练的 BERT 模型，转换为高维语义特征表示，该表示能够有效捕捉文本的深层语义信息和结构特征。在获得了文本的特征表示之后，使用风格增强机制对特征表示进行风格特征增强，以获得风格方面的特征增强表示。最后，在图神经网络处理阶段，将 BERT 提取的特征作为节点输入，构建图结构并引入图注意力网络，通过节点间的信息传递和聚合机制，进一步提取文本中的细粒度风格特征。在模型训练过程中，采用对比学习策略，通过设计对比损失函数引导模型聚焦于文本写作风格的学习，从而显著提升

模型对风格特征的区分能力。

### 3.1. 数据处理与文本编码

本文的数据处理与文本编码阶段主要针对数据集中存在的三种任务进行优化设计。数据集包含以下任务：1) 段落级写作风格改变检测，给定两位作者的文本，定位风格改变的位置；2) 文本归属分类，根据写作风格将段落分配给特定作者；3) 句子级写作风格改变标注，在段落内标注句子间的风格改变位置。为了提高模型的泛化能力，我们对任务 2 进行了重构：将原本的作者归属分类问题转换为段落间写作风格相似性判断问题，即通过两两配对的方式，将段落输入模型进行风格相似性判断，再通过聚类算法实现作者划分。这种转换不仅统一了任务形式，还增强了模型对风格特征的捕捉能力。预处理后的文本段落对将作为 BERT 编码器的输入。

文本编码模块采用预训练的 BERT 模型作为核心编码器。BERT 基于 Transformer 架构，通过双向上下文信息捕捉文本的深层语义特征。对于输入的文本段落对，BERT 首先将其转换为模型可理解的 token 序列，然后通过多层 Transformer 编码器生成高维语义向量表示，这些表示不仅包含文本的语义和语法信息，还隐含了作者的写作风格特征，这些特征将为后续的图注意力网络处理提供高质量的输入。这种多层次的特征提取机制使模型能够更好地理解文本的多义性和复杂性，为写作风格改变检测奠定基础。

在将任务 2 (文本归属分类)转换为与任务 1 (段落级风格改变检测)和任务 3 (句子级风格改变标注)相似的形式后，我们得到了一个通用的任务表示形式。设输入文本由 $n$ 个段落或句子组成，记为 $X=\{x_1,x_2,\ldots,x_n\}$，其中 $x_i$ 表示一个段落或句子。通过两两配对，生成 $n(n-1)/2$ 个文本对，记为 $P=\{(x_i,x_j)\mid 1\le i<j\le n\}$。对于每个文本对 $(x_i,x_j)$，我们将其输入预训练的 BERT 模型中，生成对应的特征表示。BERT 模型的输出包括每个 token 的向量表示以及特殊[CLS]标记的向量表示。我们采用[CLS]标记的向量作为初始的文本风格特征表示，使用风格增强机制，对[CLS]向量进行风格特征表示增强，再输入图注意力网络(GAT)进行后续处理。

### 3.2. 图神经网络处理

本模块旨在通过融合 BERT 的特征表示能力和图注意力网络(GAT)的结构化建模能力，捕捉文本间的风格差异，从而提升写作风格改变检测的准确性和鲁棒性。具体来说，BERT 用于生成文本的高维特征表示，而图注意力网络则用于建模文本之间的复杂关系，进一步增强模型的分类性能。

基于 BERT 的特征向量表示，对图的构建进行了设计，具体方法如下：采用 BERT 的增强 CLS 向量作为图神经网络的节点，使用节点与节点之间的余弦相似度来构建图神经网络边的权重，为了进一步捕捉节点之间的复杂关系，我们引入了图注意力卷积层(GATConv) [21]。在模型设计中，我们采用余弦相似度作为初始边权重，这一策略为 GATConv 提供了良好的初始语义关系表示。基于此，GAT 通过其注意力机制自适应地学习节点间的重要性权重，特别聚焦于写作风格特征的差异建模。同时，边标签作为监督信号，进一步引导模型准确捕捉写作风格变化的关键特征。这种双重优化机制显著提升了模型对风格差异的识别能力，从而获得更优的分类性能。对于节点 $v_i$ 和 $v_j$，GATConv 的注意力系数 $a_{ij}$ 通过以下方式计算：

$$a_{ij}=\frac{\exp\left(\mathrm{LeakyReLU}\left(\mathrm{a}^T\left[\mathbf{W}\mathbf{h}_i\,\|\,\mathbf{W}\mathbf{h}_j\right]\right)\right)}{\sum_{k\in\mathcal{N}(i)}\exp\left(\mathrm{LeakyReLU}\left(\mathrm{a}^T\left[\mathbf{W}\mathbf{h}_i\,\|\,\mathbf{W}\mathbf{h}_k\right]\right)\right)} \tag{1}$$

其中，$\mathbf{W}$ 是可学习的权重矩阵，a 是注意力机制的参数向量，$\|$表示向量拼接，$\mathcal{N}(i)$表示节点 $v_i$ 的邻居节点集合。

节点 $v_i$ 的表示通过加权聚合其邻居节点的表示来更新，公式如下：

$$\mathrm{h}_{\mathrm{i}}^{'}=\sigma\left(\sum_{\mathrm{j}\in\mathcal{N}(\mathrm{i})}\mathrm{a}_{\mathrm{ij}}\mathrm{Wh}_{\mathrm{j}}\right) \tag{2}$$

其中 σ 是激活函数 ReLU。

基于任务 2 的重构结果，我们采用全连接图构建图结构。这种设计不仅与文本预处理阶段的两两配对策略高度契合，还能够充分建模所有节点之间的潜在关系。全连接图意味着每个节点都与其他节点相连，边的权重则由余弦相似度决定，从而确保模型能够捕捉到局部与全局之间的文本关系，从而更全面地建模文本间的风格差异。

GATConv 输出的节点表示高维向量，而任务的目标是对节点进行分类，判断写作风格是否改变。对于每对节点 $\left(v_i,v_j\right)$，使用更新后的节点表示 $\mathrm{h}_i^{'}$ 和 $\mathrm{h}_j^{'}$ 进行边分类，在此次任务中使用多层感知机(MLP)作为前馈神经网络，一共设置了两层，分别是隐藏层和输出层。先将两个节点的隐藏向量 $\mathrm{h}_i^{'}$ 和 $\mathrm{h}_j^{'}$ 拼接为一个向量：

$$\mathbf{x}=\left[\mathrm{h}_i^{'}\left\|\mathrm{h}_j^{'}\right.\right] \tag{3}$$

其中||表示向量拼接。然后使用 MLP 对拼接向量 $\mathbf{x}$ 进行相关预测：

$$\hat{W}_{ij}=\mathrm{MLP}\left(\mathbf{x}\right) \tag{4}$$

其中 $\hat{W}_{ij}$ 是预测的边标签。MLP 的作用就是将这些高维向量映射到分类任务的输出空间，如二分类的相关概率。隐藏层对输入拼接向量 $\mathbf{x}$ 进行线性变换，并通过激活函数 ReLU 引入非线性关系：

$$\mathrm{z}_1=\mathrm{ReLU}\left(\mathbf{W}_1\mathrm{x}+\mathrm{b}_1\right) \tag{5}$$

其中 $\mathbf{z}_1$ 是第一层的激活输出，$\mathbf{W}_1$ 是第一层的权重矩阵，$\mathrm{b}_1$ 是第一层的偏置向量，ReLU 是激活函数。输出层则是对隐藏层的输出进行线性变换，最终通过 sigmoid 函数生成最终输出：

$$\hat{W}_{ij}=\sigma\left(\mathbf{W}_2\mathrm{z}_1+\mathrm{b}_2\right) \tag{6}$$

$$\sigma\left(x\right)=\frac{1}{1+e^{-x}} \tag{7}$$

$\mathbf{W}_2$ 是第二层的权重矩阵，$\mathrm{b}_2$ 是第二层的偏置向量，$\sigma\left(x\right)$ 为 sigmoid 函数，输出范围为[0, 1]。

### 3.3. 对比损失

在本模型中，我们采用两种损失函数共同优化模型：对比损失函数用于增强 BERT 提取的文本特征表示，交叉熵损失函数用于优化图注意力网络的分类性能。以下对两种损失函数进行详细说明。

对比损失函数的主要目标是优化 BERT 对输入文本的特征表示，特别是增强[CLS]向量中的风格信息。BERT 作为一种编码器，将输入文本映射至高维空间，以获得文本的特征表示。BERT 输出的三种常见文本表示方式有 CLS 表示、最大池化表示以及平均池化表示。CLS 表示是对整个输入序列的总结，包含了输入序列的总体信息；最大池化表示是对 BERT 输出的所有 token 表示在每个维度上取最大值所得到的向量，能够捕捉文本中最显著的特征，但是会造成一些细节信息的丢失；平均池化表示则是对 BERT 输出的所有 token 表示在每个维度上取平均值所得到的向量，能够表示文本的整体语义信息，但会丢失一定的关键信息。鉴于所针对任务的特殊性，我们取 CLS 向量来开展整个工作。

为了增强 CLS 中所包含的风格信息占比，我们通过对比学习的方式优化文本的 CLS 表示，使得同一作者的文本特征表示具有更高的余弦相似度，相对而言，不同作者文本特征表示的余弦相似度更低，对

比损失函数如下所示：

$$\mathcal{L}_e = \frac{1}{N}\sum_{(i,j)\in\mathcal{P}}\left(1-sim\left(x_i,x_j\right)\right)+\frac{1}{M}\sum_{(i,j)\in\mathcal{N}}\max\left(0,sim\left(x_i,x_j\right)-m\right) \tag{8}$$

其中 $\mathrm{sim}\left(\mathrm{x_i},\mathrm{x_j}\right)$ 为 $\mathrm{x_i}$ 与 $\mathrm{x_j}$ 之间的余弦相似度，$\mathcal{P}$ 是正样本对集合，$\mathcal{N}$ 是负样本对集合，N 为正样本对的数量，M 是负样本对的数量，m 是一个超参数表示负样本对的余弦相似度的间隔。通过对比损失函数，模型能够学习到更具判别性的风格特征表示，从而提升后续任务的性能。

最终对图注意力网络进行分类预测时，使用交叉熵损失函数对模型训练效果进行考量，反映预测结果 $\hat{y}_{ij}$ 和真实标签 $y_{ij}$ 之间的差异，从而指导模型参数更新。交叉熵损失公式如下：

$$\mathcal{L}_c = -\sum_{(i,j)\in\varepsilon}\left(y_{ij}\log\left(\hat{y}_{ij}\right)+\left(1-y_{ij}\right)\log\left(1-\hat{y}_{ij}\right)\right) \tag{9}$$

其中 $y_{ij}$ 是真实标签(0 或 1)，$\hat{y}_{ij}$ 是模型预测的概率，$\varepsilon$ 是边的集合。通过最小化交叉熵损失函数，模型能够更准确地预测文本对的风格一致性，从而提升写作风格改变检测的性能。

## 4. 实验结果及分析

### 4.1. 实验设置

实验环境和参数配置：本文所进行的所有实验均在搭载有 NVIDIA RTX 3090 Ti GPU 的工作站上完成，使用 Python 3.9 和 PyTorch 2.1.0 框架完成模型训练与测试。在模型配置方面，文本表示部分采用预训练模型 BERT-Base 模型，含有 12 层 Transformer，向量维度为 768，风格检测部分引入图注意力网络(GAT)对文本之间的风格关系进行建模，GAT 包含两层，每层使用 8 个注意力头。具体的硬件环境配置和模型训练超参数如表 1 所示。

**Table 1.** Experimental environment and parameter configuration
**表 1.** 实验环境和参数配置

| 环境 | 配置 | 参数 | 配置 |
|---|---|---|---|
| 操作系统 | Windows 10 (64 位) | Dropout | 0.1 |
| 处理器 | Intel Core i9-10900X | 学习率 | 1e−5 |
| GPU | 3090 Ti | Batch_size | 32 |
| 内存 | 128 G | 词向量维度 | 768 |
| 编程语言 | Python 3.9 | 优化器 | Adam |

对比模型：为了验证所提出模型的有效性，实验选取了四个比较有代表性的预训练模型进行了相关实验，其中包括 GPT-2 [18]、RoBERTa [16]、SBERT [17]和 BERT [15]。

数据集：我们选用 PAN 2022 [20]作为此次模型训练与测试的数据集。PAN 2022 是一个针对风格变化检测任务所提出的数据集，该数据集用来解决以下三类任务：1) 整个文本由两个作者撰写，在所给文本中，仅存在一个写作风格发生改变的地方，需要找到正确的位置；2) 整个文本由两个或多个作者撰写，需要将这些段落进行一个作者归属分类，即将每一段分配给其真正的作者；3) 给定一个段落，该段落由两个或多个作者撰写，逐句读取文本内容，找出句与句之间由不同作者完成的地方。该数据集以英文提供，并且可能包含任意数量的风格变化，但是作者数量最多不超过五个。由于该数据集更贴近日常生活

中存在的抄袭情景，所以采用该数据集作为训练和测试数据集。我们将验证集进行了拆分，将其中一部分用于训练过程中的验证，另一部分则作为测试数据集，以便测试模型的训练效果，处理后的数据集如表 2 所示。

**Table 2.** Composition of the dataset
**表 2.** 数据集组成情况

| 数据集 | 数据集 1 | | 数据集 2 | | 数据集 3 | |
|---|---|---|---|---|---|---|
| | 作者数 | 段落数 | 作者数 | 段落数 | 作者数 | 句子数 |
| 训练集 | 2800 | 10,989 | 21,000 | 52,723 | 21,000 | 111,992 |
| 验证集 | 300 | 1159 | 1515 | 3648 | 1512 | 7904 |
| 测试集 | 300 | 1282 | 2985 | 7389 | 2988 | 15,701 |

## 4.2. 评价指标

### 4.2.1. 余弦相似度

由于我们使用 CLS 特征对文本的整体内容进行表示，其中包含了一定的隐式风格信息，为了增强风格信息的表示，我们采用余弦相似度来衡量 CLS 表示中的风格信息相似程度。余弦相似度是一种衡量两个向量之间相似性的方法，通过计算两个向量的夹角余弦值来评估它们的相似程度，取值在−1 到 1 之间，余弦相似度的计算公式如下：

$$\cos\theta = \frac{\mathbf{X}\cdot\mathbf{Y}}{\|\mathbf{X}\|\|\mathbf{Y}\|} \tag{10}$$

其中，$\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 分别表示两个待检测文本的向量表示。

### 4.2.2. Spearman 相关系数

Spearman [22]相关系数是一种非参数统计方法，用于衡量两个变量之间的单调关系，即一个变量随另一个变量的增加而增加或减少的趋势。在此次任务中，Spearman 相关系数用于衡量文本的风格特征相似度与真实风格标签之间的单调关系，从而判断模型是否能够正确捕捉风格特征信息。Spearman 相关系数具有对异常值不敏感的特点，且其取值范围在−1 到 1 之间，能够帮助我们在此次任务中，直观地观察模型提取风格特征的性能。Spearman 相关系数具体公式如下所示：

$$\rho_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n\left(n^2-1\right)} \tag{11}$$

式中 $d_i$ 对应变量的秩次差，即两个变量分别排序后成对的变量位置(等级)差，$n$ 是样本数量。

### 4.2.3. 准确率(Accuracy)

准确率是分类任务中最常用的评估指标之一，表示模型正确预测的样本占总样本数的比例。在进行风格变化检测任务中，准确率可以反映模型在整体数据集上的表现。具体公式如下所示：

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{12}$$

其中，TP 和 TN 表示正确预测的样本，FP 和 FN 表示错误预测的样本。

#### 4.2.4. macro F1-score

macro F1-score 是 F1-score 的宏平均版本，用于多分类任务中评估模型的整体性能。F1-score 是精确率(Precision)和召回率(Recall)的调和平均值。在风格变化检测任务中，macro F1-score 能够平等对待每个类别，并综合反映模型的分类性能。macro F1-score 的计算公式如下：

$$\text{macro F1} = \frac{1}{C}\sum_{i=1}^{C} F1_i \tag{13}$$

其中 $C$ 是类别数量，$F1_i$ 是第 $i$ 个类别的 F1-score。

### 4.3. 风格特征强化

为了更直观地展示风格特征强化机制的实际效果，本文采用 t-SNE 降维方法对原始 BERT 向量与经过风格特征强化后的向量进行了可视化分析，并比较二者在风格空间中的分布差异。具体而言，我们分别提取了文本在原始 BERT 模型下和引入风格强化机制后所得的[CLS]向量表示，并使用 t-SNE 将其从高维空间降至二维空间进行可视化。不同写作风格的文本在图 3 中以不同颜色进行标注，从而便于观察各类风格在向量空间中的聚类情况。

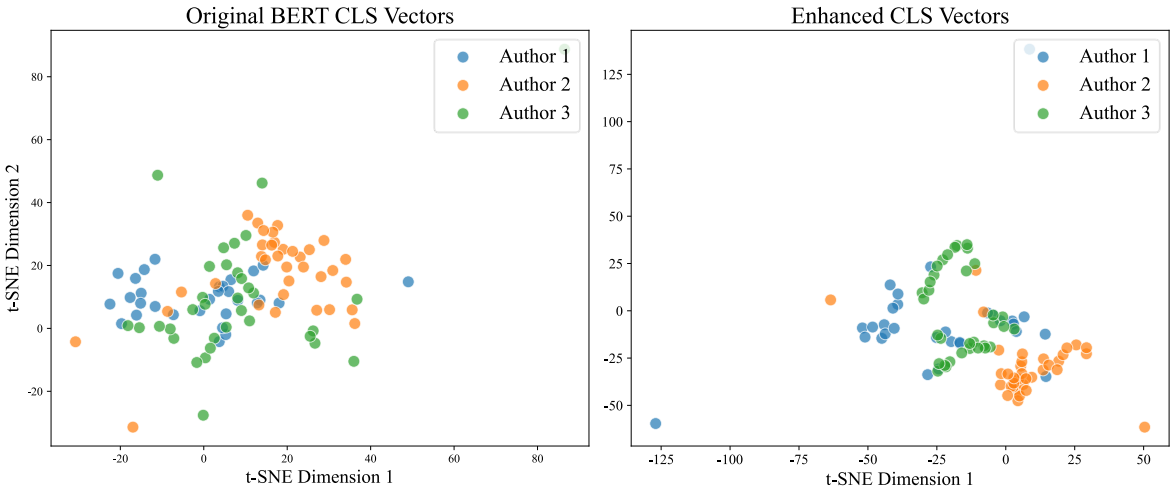



**Figure 3.** Comparison of style features before and after enhancement
**图 3.** 风格特征强化的前后对比图

从图 3 中可以看出，未经过风格强化机制的向量在空间中的分布较为混杂，风格类别之间的边界模糊，甚至存在较严重的重叠现象；而经过风格特征强化机制处理后的向量分布则更为清晰，各类写作风格呈现出明显的聚类趋势，类间边界显著增强，整体分布具有更强的可区分性。这一结果表明，所提出的风格特征强化机制在提升写作风格表征能力方面具有显著作用，为后续基于图注意力网络的文本分类任务提供了更加可靠的特征基础。

### 4.4. 模型性能对比

为了能够更直观地凸显所提出模型在风格改变检测任务上的性能，我们将所提出模型与较有代表性的语言模型进行了比较，在整个比较过程中，条件变量均保持一致，使用相同的数据集对不同的模型进行训练、验证以及测试，对 AUC_ROC、macro F1_score、Accuracy、Spearman 等指标进行了评估，评估结果如表 3~5 所示。

**Table 3.** Performance comparison of different models on task 1
**表 3.** 不同模型在任务 1 上的表现

| | AUC_ROC | macro F1-score | Accuracy | Spearman |
|---|---|---|---|---|
| BERT | 0.7493 | 0.7096 | 0.7214 | 0.7043 |
| RoBERTa | 0.8015 | 0.7519 | 0.7793 | 0.7417 |
| SBERT | 0.7751 | 0.7318 | 0.7602 | 0.7382 |
| GPT-2 | 0.7493 | 0.7024 | 0.7221 | 0.7021 |
| Ours | 0.8297 | 0.7931 | 0.8156 | 0.7792 |

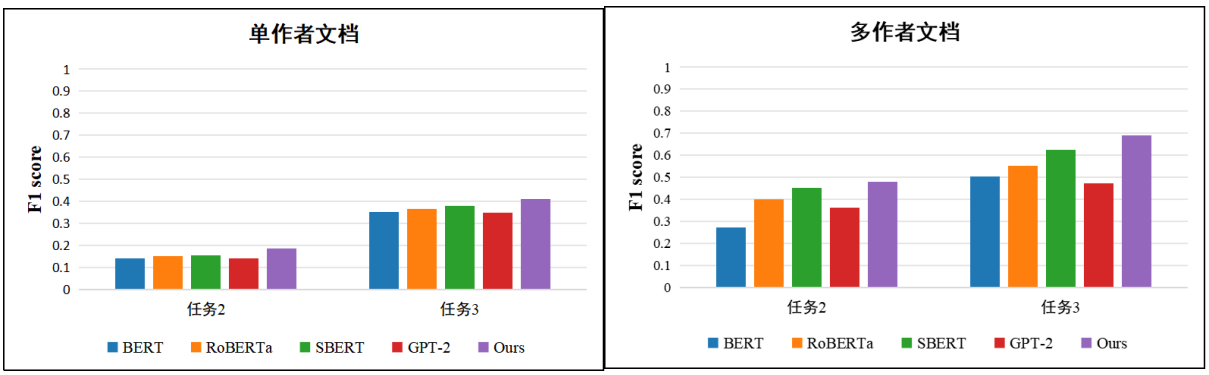
**Table 4.** Performance comparison of different models on task 2
**表 4.** 不同模型在任务 2 上的表现

| | AUC_ROC | macro F1-score | Accuracy | Spearman |
|---|---|---|---|---|
| BERT | 0.5589 | 0.5109 | 0.5132 | 0.5119 |
| RoBERTa | 0.5701 | 0.5122 | 0.5508 | 0.5219 |
| SBERT | 0.5991 | 0.5397 | 0.5743 | 0.5439 |
| GPT-2 | 0.5274 | 0.4793 | 0.5099 | 0.4713 |
| Ours | 0.6617 | 0.6124 | 0.6441 | 0.6013 |

**Table 5.** Performance comparison of different models on task 3
**表 5.** 不同模型在任务 3 上的表现

| | AUC_ROC | macro F1-score | Accuracy | Spearman |
|---|---|---|---|---|
| BERT | 0.7541 | 0.7102 | 0.7391 | 0.7199 |
| RoBERTa | 0.7713 | 0.7228 | 0.7569 | 0.7235 |
| SBERT | 0.7752 | 0.7237 | 0.7513 | 0.7219 |
| GPT-2 | 0.7481 | 0.6914 | 0.7202 | 0.6905 |
| Ours | 0.8198 | 0.7862 | 0.7843 | 0.7451 |

## 4.5. 作者数量对模型性能的影响分析

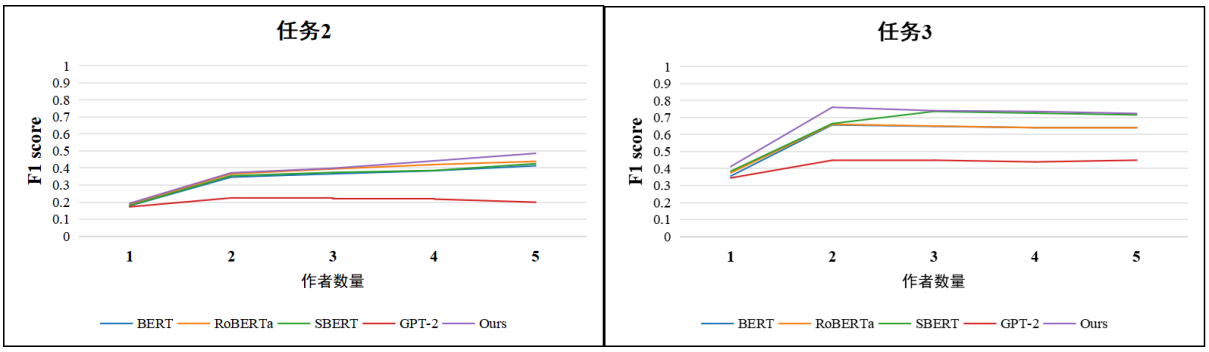



(a) 单作者文档　　(b) 多作者文档

**Figure 4.** Comparative analysis of model performance on single-author and multi-author documents
**图 4.** 模型在单作者文档与多作者文档上的表现

为了探究文本作者数量是否对模型的性能有所影响，我们使用 F1-score 测试文本作者数量对模型性能的影响。由于任务 1 的文本仅由两个作者完成，所以只对任务 2 和任务 3 进行了相关实验，探讨作者数量对模型效果的影响，具体结果如图 4 所示。

从图 4 中我们可以看出待检测文本的作者数确实对模型的判别效果有一定影响，当作者人数大于 1 时，所有模型的判别效果均优于对单一作者文本的判别效果。为了进一步探究文本作者数对模型判别效果的影响，我们基于 F1-score 对模型进行了相关实验：将测试集文本按作者数(1~5)划分，分别计算了在各个作者数量情况下，模型的 F1-score。检测结果如图 5 所示，图中横坐标为文本的作者数量。




(a) 任务 2 上的表现　　(b) 任务 3 上的表现

**Figure 5.** Effect of varying author numbers on detection model performance
**图 5.** 作者数量对模型检测效果的影响

从图 5 中我们可以得出结论：所有模型在作者数为 2 的时候，模型检测效果最好；当文本的作者数量超过 2 时，随着作者数量的增加，模型检测性能下降。

## 4.6. 消融实验

**Table 6.** Performance analysis of GAT and SFEM on task 1
**表 6.** GAT 和 SFEM 对任务 1 的性能影响分析

| | w/GAT | | w/o GAT | |
|---|---|---|---|---|
| | w/SFEM | w/o SFEM | w/SFEM | w/o SFEM |
| AUC_ROC | 0.8297 | 0.7901 | 0.7683 | 0.7612 |
| macro F1-score | 0.7931 | 0.7541 | 0.7302 | 0.7247 |
| Accuracy | 0.8156 | 0.7701 | 0.7431 | 0.7401 |
| Spearman | 0.7792 | 0.7471 | 0.7186 | 0.7147 |

**Table 7.** Performance analysis of GAT and SFEM on task 2
**表 7.** GAT 和 SFEM 对任务 2 的性能影响分析

| | w/GAT | | w/o GAT | |
|---|---|---|---|---|
| | w/SFEM | w/o SFEM | w/SFEM | w/o SFEM |
| AUC_ROC | 0.6617 | 0.6411 | 0.6093 | 0.5633 |
| macro F1-score | 0.6124 | 0.5926 | 0.5517 | 0.5133 |
| Accuracy | 0.6441 | 0.6195 | 0.5667 | 0.5393 |
| Spearman | 0.6013 | 0.5902 | 0.5501 | 0.5129 |

**Table 8.** Performance analysis of GAT and SFEM on task 3
**表 8.** GAT 和 SFEM 对任务 3 的性能影响分析

| | w/GAT | | w/o GAT | |
|---|---|---|---|---|
| | w/SFEM | w/o SFEM | w/SFEM | w/o SFEM |
| AUC_ROC | 0.8198 | 0.8113 | 0.7621 | 0.7314 |
| macro F1-score | 0.7862 | 0.7422 | 0.7203 | 0.7014 |
| Accuracy | 0.7843 | 0.7692 | 0.7476 | 0.7136 |
| Spearman | 0.7451 | 0.7126 | 0.6971 | 0.6791 |

为了证明所添加的风格特征增强模块以及图注意力网络的有效性，对所提出模型进行了消融实验，通过改变模型的构成来获得不同的模型效果。通过是否添加风格特征增强模块(Style Feature Enhancement module, SFEM)以及是否使用图注意网络(GAT)，获得了模型的四种变体，并对四种变体进行性能检测，具体实验结果如表 6~8 所示。

由上表可知，本文所使用的风格特征增强模块和图注意力网络能够在一定程度上增强模型对风格改变检测的效果，证明了所提出方法的有效性。

## 5. 总结

本文提出了一种面向风格转变的抄袭检测算法，通过分析文本内部的写作风格一致性来识别潜在的抄袭行为。针对大语言模型改写文本的特点，我们设计了统一的检测框架：首先，通过对比学习机制增强 BERT 模型对文本进行特征表示；然后再使用风格增强机制对风格特征进行增强，以便于后续任务的进行；其次，在获得文本的增强特征表示之后，利用图注意力网络(GAT)对文本特征进行结构化建模，将风格分析任务转化为基于节点关系的二分类任务，有效降低了模型复杂度。实验结果表明，本文提出的模型在 PAN 2022 数据集上整体性能优于其他抄袭检测模型。该模型能够更准确地识别文本段落中写作风格改变的位置，为后续的抄袭检测任务提供了可靠的前置条件。此外，通过消融实验验证了风格特征增强模块和图注意力网络在风格改变检测任务中的有效性，进一步证明了模型设计的合理性。

## 基金项目

上海市教委人工智能促进科研范式改革赋能学科跃升计划项目。

## 参考文献

[1] Sindhu, B., Prathamesh, R.P., Sameera, M.B. and KumaraSwamy, S. (2024) The Evolution of Large Language Model: Models, Applications and Challenges. 2024 *International Conference on Current Trends in Advanced Computing* (*ICCTAC*), Bengaluru, 8-9 May 2024, 1-8. https://doi.org/10.1109/icctac61556.2024.10581180

[2] Pudasaini, S., Miralles-Pechuán, L., Lillis, D. and Llorens Salvador, M. (2024) Survey on AI-Generated Plagiarism Detection: The Impact of Large Language Models on Academic Integrity. *Journal of Academic Ethics*. https://doi.org/10.1007/s10805-024-09576-x

[3] Lee, J., Le, T., Chen, J. and Lee, D. (2023) Do Language Models Plagiarize? *Proceedings of the ACM Web Conference* 2023, Austin, 30 April-4 May 2023, 3637-3647. https://doi.org/10.1145/3543507.3583199

[4] Foltýnek, T., Meuschke, N. and Gipp, B. (2019) Academic Plagiarism Detection: A Systematic Literature Review. *ACM Computing Surveys*, **52**, 1-42. https://doi.org/10.1145/3345317

[5] 郭凯威, 杨奎武, 张万里, 胡学先, 刘文钊. 面向文本识别的对抗样本攻击综述[J]. 中国图象图形学报, 2024, 29(9): 2672-2691.

[6] Franke, J. and Oberlander, M. (1993) Writing Style Detection by Statistical Combination of Classifiers in Form Reader Applications. *Proceedings of* 2*nd International Conference on Document Analysis and Recognition* (*ICDAR*'93), Tsukuba,

20-22 October 1993, 581-584. https://doi.org/10.1109/icdar.1993.395668

[7] Chong, M. and Specia, L. (2011) Lexical Generalisation for Word-Level Matching in Plagiarism Detection. *Proceedings of the International Conference Recent Advances in Natural Language Processing*, Hissar, 12-14 September 2011, 704-709.

[8] Alzahrani, S. and Salim, N. (2010) Fuzzy Semantic-Based String Similarity for Extrinsic Plagiarism Detection. CLEF 2010 LABs and Workshops, Notebook Papers, 1-8.

[9] Bergroth, L., Hakonen, H. and Raita, T. (2000) A Survey of Longest Common Subsequence Algorithms. Proceedings Seventh International Symposium on String Processing and Information Retrieval. SPIRE 2000, A Curuna, 27-29 September 2000, 39-48. https://doi.org/10.1109/spire.2000.878178

[10] Christian, H., Agus, M.P. and Suhartono, D. (2016) Single Document Automatic Text Summarization Using Term Frequency-Inverse Document Frequency (TF-IDF). *ComTech*: *Computer*, *Mathematics and Engineering Applications*, **7**, 285. https://doi.org/10.21512/comtech.v7i4.3746

[11] Jakkula, V. (2006) Tutorial on Support Vector Machine (SVM). School of EECS, Washington State University.

[12] 周大为, 徐一搏, 王楠楠, 刘德成, 彭春蕾, 高新波. 针对未知攻击的泛化性对抗防御技术综述[J]. 中国图象图形学报, 2024, 29(7): 1787-1813.

[13] Chen, Q. and Wu, R. (2017) CNN Is All You Need. arXiv: 1712.09662. https://doi.org/10.48550/arXiv.1712.09662

[14] Huang, Z., Ye, Z., Li, S. and Pan, R. (2017) Length Adaptive Recurrent Model for Text Classification. *Proceedings of the* 2017 *ACM on Conference on Information and Knowledge Management*, Singapore, 6-10 November 2017, 1019-1027. https://doi.org/10.1145/3132847.3132947

[15] Devlin, J., Chang, M.W., Lee, K., *et al*. (2019) BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv: 1810.04805. https://doi.org/10.48550/arXiv.1810.04805

[16] Liu, Y., Ott, M., Goyal, N., *et al*. (2019) Roberta: A Robustly Optimized Bert Pretraining Approach. arXiv: 1907.11692. https://doi.org/10.48550/arXiv.1907.11692

[17] Reimers, N. and Gurevych, I. (2019) Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. arXiv: 1908.10084. https://doi.org/10.48550/arXiv.1908.10084

[18] Radford, A., Wu, J., Child, R., *et al*. (2019) Language Models Are Unsupervised Multitask Learners. *OpenAI Blog*, **1**, 1-24.

[19] Xu, K., Hu, W., Leskovec, J., *et al*. (2018) How Powerful Are Graph Neural Networks? arXiv: 1810.00826. https://doi.org/10.48550/arXiv.1810.00826

[20] Bevendorff, J., Chulvi, B., Fersini, E., Heini, A., Kestemont, M., Kredens, K., *et al*. (2022) Overview of PAN 2022: Authorship Verification, Profiling Irony and Stereotype Spreaders, and Style Change Detection. *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, Bologna, 5-8 September 2022, 382-394. https://doi.org/10.1007/978-3-031-13643-6_24

[21] Veličković, P., Cucurull, G., Casanova, A., *et al*. (2017) Graph Attention Networks. arXiv:1710.10903. https://doi.org/10.48550/arXiv.1710.10903

[22] Popescu, M.C., Balas, V.E., Perescu-Popescu, L., *et al*. (2009) Multilayer Perceptron and Neural Networks. *WSEAS Transactions on Circuits and Systems*, **8**, 579-588.