Modeling and Simulation 建模与仿真, 2024, 13(4), 4279-4288
Published Online July 2024 in Hans. https://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2024.134388

Hans汉斯

基于深度学习的翻译洗稿抄袭检测算法

贺小玲，周元鼎

上海理工大学，光电信息与计算机工程学院，上海

收稿日期：2024年6月12日；录用日期：2024年7月5日；发布日期：2024年7月12日

摘　要

为应对多媒体技术和互联网快速发展带来的多样化和新型化洗稿抄袭问题，本文提出了一种基于深度学习的翻译洗稿的抄袭检测算法，该算法通过融合多轮翻译后的特征来增强翻译文本的特征，从而得到高质量的文本表示，并利用对比学习架构拉近原文本在语义向量空间中与翻译文本的距离，同时保持其与负样本的距离。此外，本文通过改进的对比损失函数增强模型检测洗稿文本的能力。最后利用所构建的多元组翻译洗稿数据集来进行训练和验证，使之达到检测翻译洗稿抄袭的能力。实验结果表明，本文所提出的算法产生了质量更高的文本表示，从而在翻译洗稿抄袭检测任务上优于先前的方法，Spearman相关系数的结果也证明了所构建模型的优越性。

关键词

翻译洗稿，语义向量空间，对比损失函数，Spearman相关系数

Deep Learning-Based Translation Laundering Plagiarism Detection Algorithm

Xiaoling He, Yuanding Zhou

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

Received: Jun. 12th, 2024; accepted: Jul. 5th, 2024; published: Jul. 12th, 2024

Abstract

In response to the increasingly diverse and novel plagiarism issues arising from the rapid development of multimedia technologies and the Internet, this paper introduces a deep learning-based

文章引用: 贺小玲, 周元鼎. 基于深度学习的翻译洗稿抄袭检测算法[J]. 建模与仿真, 2024, 13(4): 4279-4288.
DOI: 10.12677/mos.2024.134388

plagiarism detection algorithm for translation laundering. This algorithm enhances the characteristics of translated texts by fusing features from multiple translation rounds, thereby achieving high-quality text representations. It utilizes a contrastive learning framework to narrow the distance between the original text and the spun text within the semantic vector space, while maintaining separation from negative samples. Additionally, the model's ability to detect spun texts is bolstered by an improved contrastive loss function. The algorithm was trained and validated on a specially constructed multiset translation laundering dataset, to effectively detect plagiarism via translation laundering. Experimental results show that the proposed algorithm produces higher quality text representations and surpasses previous methods in detecting translation laundering plagiarism. The effectiveness of the constructed model is further affirmed by the Spearman correlation coefficient results.

Keywords

Translation Laundering, Semantic Vector Space, Contrastive Loss Function, Spearman Correlation Coefficient

1. 引言

随着互联网技术的高速发展，资源获取和信息共享的途径越来越多，带来便利的同时降低了抄袭成本，使得近些年来抄袭乱象频现[1]，尤其是洗稿现象日益普遍，其影响从新闻报道扩展到文学创作。回译洗稿涉及将文本从一种语言翻译到另一种语言，然后再翻译回原始语言的改写行为[2]。这种不道德的行为侵犯了原作者的知识产权。为解决这一问题，自然语言处理领域出现了各种抄袭检测方法[3]，包括成本高昂的人工判断方法，由于每日生成的数字内容量巨大，使得这种方法效率低下。此外，人工检测还引入了主观性，因为不同个人的判断可能会有所不同。Alzahrani 等人[4]将抄袭检测任务分为两个正式类别：外部抄袭[5]和内部抄袭[6]。外部抄袭检测根据一个或多个源文档评估抄袭行为，另一方面，内在抄袭检测通过单独检查可疑文档来评估抄袭文本。现代抄袭检测方法可以分为三种类型：传统的人工方法、基于机器学习的方法和基于深度的方法。传统的人工方法需要繁琐的特征向量构建、分析反向翻译前后的特征变化，并设计二元分类器来区分原始文本和回译文本[7]。由于特征提取能力有限和依赖离散的手工特征，这些方法实际上并不可行。此外，机器翻译[8]的进步，特别是神经网络，可以产生与原始文本几乎无法区分的回译文本，为传统方法带来挑战。与传统的人工抄袭检测不同，基于机器学习的方法可以从大量训练数据中自动学习特征，无须提前进行特征设计。然而，一部分基于机器学习的方法忽略了文本中的上下文依赖性，仅依赖于卷积神经网络(CNN)进行局部语义特征提取[9]，而其他方法则使用循环神经网络(RNN)来捕捉上下文信息。尽管如此，RNN 结构面临诸如有限的长范围上下文利用和与梯度相关的问题[10]。Pennington 等人[11]通过将局部信息与全局词频统计相结合，提取全面的语义特征。

2018 年末，BERT 的出现[12]标志着自然语言处理进入预训练模型时代。BERT 的自注意力机制能够捕捉局部特征和全局语义关系。然而，直接使用 BERT 作为嵌入模型的性能并不如预期[13]。因此，Reimers 等人[14]提出了 SBERT，采用双生网络结构生成具有语义信息的固定长度句向量，便于相似性比较。随后的发展，包括 BERT-Flow [15]和 BERT-Whitening [16]，解决了各向异性问题，而 Yan 等人[17]在 2021 年引入了 NLP 语义计算中的对比学习通用框架 ConSERT。Su 等人[18]在 2022 年引入了 CoSENT，一种

e 新的监督句向量方案，与 SBERT 相比，显示出更快的收敛速度和更好的结果。然而，这些模型将所有特征平等对待，未能考虑高维特征或特征质量对模型性能的影响。对于现有的抄袭检测方法，评估回译文本与原始文本之间的相似性是一个挑战。提高特征质量对于改善检测模型的性能至关重要。目前，针对翻译洗稿的机器检测方法还很少。因此，本文提出了一种基于深度学习的翻译洗稿抄袭检测算法。本文方法从回译文本中提取语言风格、习语表达和全局长期依赖性的潜在表征，并采用所提出的翻译洗稿特征融合机制生成分组特征的重要性系数。此机制可以加强重要特征，削弱不重要的特征，最终提高文本表示质量[19]。我们的模型还采用对比损失函数通过多轮翻译的文本表示融合来优化检测能力。本文的主要贡献如下：

1) 基于深度学习提出了一个针对翻译洗稿的抄袭检测模型，并根据模型结构改进了对比损失函数。

2) 提出了一个创新性的特征融合机制，通过对回译文本特征进行分组、增强和融合来构建高质量的文本表示。

3) 构建了中英文多元组机器翻译洗稿数据集，并观察到在对我们所制作的数据集进行训练后，本文方法的 Spearman 相关系数可以达到 0.86，均高于目前所提出的一些模型。

2. 数据集的构建

为了对翻译洗稿抄袭进行检测，首先从哈尔滨工业大学整理的基于新闻媒体在微博上发布的 LCSTS 新闻摘要数据集中抽取了摘要文本。其次，通过从百度翻译、Google 翻译、有道翻译和 Bing 翻译这四个机器翻译器中随机选择一个作为翻译器。紧接着从汉语、英语、西班牙语、德语、日语和藏语等语言中随机选择翻译的过渡语言。最后分别随机翻译 2 到 4 轮得到回译文本 1 和回译文本 2 以及不相似的矛盾文本并创建了四元组翻译洗稿抄袭检测数据集，从数据集中随机选取其他回译后的文本作为矛盾文本，从而构建出了翻译洗稿数据集。测试集和验证集分别有 1000 条数据，每个语料库都通过多人交叉人工标注的方式用数字 0 和 1 标注了每一对句子的相似性，其中 0 表示是原文本和矛盾文本，1 表示原文本和回译文本，数据集标签说明表见表 1。

Table 1. Dataset label description sheet
表 1. 数据集标签说明表

训练集标签	说明	测试集与原文组成文本对的标签
Origin	原文	-
Translation 1	多轮回译文本 1	1
Translation 2	多轮回译文本 2	1
Contradiction	矛盾文本	0

3. 所提方法

如图 1 所示，本文所提出的基于深度学习的翻译洗稿抄袭检测算法主要由文本编码、特征融合和损失函数这三部分组成。首先将待检测的文本输入到 BERT 中进行文本编码，随后输出更高维度的抽象表示形式，该形式能够捕捉和表达输入数据的关键信息和内在结构，为了提升特征质量和模型识别能力，本文在训练的过程中设计了一个特征融合机制，通过对特征表示进行分组、增强和融合操作，形成一个综合的且高质量的特征表示。最后通过优化的对比损失函数将模型训练的焦点放在使相似文本表示更加接近，而使不相似文本表示更加疏远上，从而提高模型的区分能力。

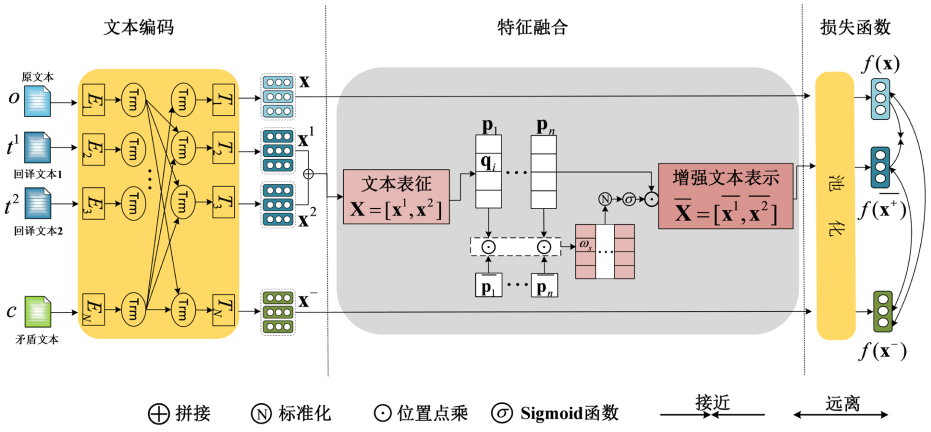


Figure 1. Overall framework of our method
图 1. 本文方法模型框架

3.1. 文本编码

编码层通过学习输入数据的特征，将原始数据转换为模型可以理解和处理的格式。对于文本数据，这意味着将单词、短语或句子映射到一个抽象的特征空间，这些特征能够表示词义、语法结构、上下文关系等语言属性。编码层能够从输入数据中提取最重要的信息，并以更紧凑和抽象的形式表示。在基于 Transformer 的模型 BERT 中，编码层通过自注意力机制能够捕捉单词间的长距离依赖关系，从而实现对整个输入序列的全局理解。这种机制允许模型根据上下文动态调整对每个单词的关注，从而更好地理解语言的多义性和复杂性。

首先将文本输入一个共享的 BERT 编码器，对于输入的原始文本 o，多轮翻译文本 t^1, t^2 以及矛盾文本 c，通过 BERT 中堆叠的多层 Transformer 块进行编码转换分别得到文本表示 $\mathbf{x},\mathbf{x}^1,\mathbf{x}^2,\mathbf{x}^-$，将转换后的多轮翻译洗稿的文本表示进行拼接得到联合表征图 $\mathbf{X}=\left[\mathbf{x}^1,\mathbf{x}^2\right]$。

3.2. 特征融合

本模块旨在通过融合多轮翻译所得的回译文本的表示，以强化关键特征并弱化不重要的特征，以便获得更高质量的文本表示。首先，根据特征的维度，将回译后的联合表征图 $\mathbf{X}$ 分为 n 个组记为：

$$\mathbf{G}=\left[\mathbf{g}_1,\mathbf{g}_2,\cdots,\mathbf{g}_n\right]\in\mathbb{R}^{l\times k} \tag{1}$$

由于 n 组的处理方法相同，因此选择一个组记为：

$$\mathbf{g}_j=\left[\mathbf{f}_1,\mathbf{f}_2,\cdots,\mathbf{f}_l\right]\in\mathbb{R}^{l\times m},\ j=1,2,\cdots,n \tag{2}$$

其中 $m=\dfrac{k}{n}$，其子特征为：

$$\mathbf{f}_i=\left[a_1,a_2,\cdots,a_m\right]^{\mathrm{T}}\in\mathbb{R}^m,i=1,2,\cdots,l \tag{3}$$

在网络学习过程中，假设每个组都包含特定的语义信息。因此，可以通过计算每组的平均子特征来近似该组的语义向量，并对每一组进行相应的处理：

$$\overline{\mathbf{g}_j}=\frac{1}{l}\sum_{i=1}^{l}\mathbf{f}_i \tag{4}$$

其次，使用该平均化后的回译文本语义向量，通过一个简单的位置点积来获得每个子特征相应的重

要性系数，该点积在一定程度上衡量了平均语义向量与子特之间的相似性：

$$\omega_i = \overline{\mathbf{g}}_j \cdot \mathbf{f}_i \tag{5}$$

为消除不同样本间的度量影响，进行数据的归一化处理，归一化不仅加速了模型收敛，也保持了特征之间的相关性。

$$\widehat{\omega}_i = \frac{\omega_i - E(\omega)}{\sqrt{D(\omega)+\varepsilon}} \tag{6}$$

$$E(\omega) = \frac{1}{l}\sum_{i=1}^{l}\omega_i \tag{7}$$

$$D(\omega) = \frac{1}{l}\sum_{i=1}^{l}\left(\omega_i - E(\omega)\right)^2 \tag{8}$$

归一化过程中引入的常数 ε，以及每组的均值 $E(\omega)$ 和方差 $D(\omega)$，都是为了数值稳定性而设。归一化后的输出通过引入的缩放参数 α 和平移参数 β 进行调整，确保归一化操作后能有效表示文本的变化，然后通过 sigmoid 函数 $\sigma(\cdot)$ 生成一个新的归一化重要系数：

$$\overline{\omega}_i = \sigma\left(\alpha\widehat{\omega}_i + \beta\right) \tag{9}$$

最后，用归一化后的重要性系数对原始特征进行缩放，强化重要特征，弱化不重要特征，得到增强的子向量：

$$\overline{\mathbf{f}}_i = \overline{\omega}_i \cdot \mathbf{f}_i \tag{10}$$

所有增强子向量形成了增强向量组 $\overline{\overline{\mathbf{g}}}_j = \left[\overline{\mathbf{f}}_1, \overline{\mathbf{f}}_2, \cdots, \overline{\mathbf{f}}_l\right] \in \mathbb{R}^{l\times m}$。在所有组增强后，就能够得到一个增强的文本表示，记为 $\overline{\mathbf{X}} = \left[\overline{\mathbf{x}^1}, \overline{\mathbf{x}^2}\right] \in \mathbb{R}^{l\times k}$。

3.3. 损失函数

对比损失函数对输出的文本表示 $f(\mathbf{x}), f\left(\overline{\mathbf{x}^+}\right), f\left(\mathbf{x}^-\right)$ 进行处理。在此之前通过池化层，在保留主要特征的情况下进行特征降维，最终得到文本语义表示，常用的池化策略有[CLS]向量、最大池化、平均池化三种。[CLS]向量的池化策略直接将模型最后一层输出的[CLS]向量作为输入文本的语义表示。最大池化策略取所有词向量每一个维度的最大值作为输入文本的语义表示，突出了关键信息但是由于丢弃了太多值存在信息丢失问题。平均池化策略对所有向量求平均值，用所有词向量的平均值作为输入文本的语义表示，虽然所有词向量每个维度的特征值都参与了运算，但是求和平均的操作使得在计算文本的语义表示时，每一个词向量所占的权重相等，导致模型无法聚焦于关键信息。本文选择 CLS 池化来对文本表示进行降维。

随后通过优化的对比损失函数来计算和最小化正样本间的距离，同时最大化负样本间的距离。在每个训练步骤中，模型从训练集中随机抽取 N 个文本表示构建一个 mini-batch，并将通过不同轮数翻译的回译文本表示进行融合，作为对比损失函数的优化学习的目标。在最大限度地提高原文本表示与多轮回译文本表示之间相似性的同时保持其与同一 batch 中的其他矛盾文本表示之间的距离。优化的对比损失函数如公式 11 所示：

$$\ell_i = -\log\frac{e^{\mathrm{sim}\left(f(\mathbf{x}_i), f\left(\overline{\mathbf{x}_i^+}\right)\right)/\tau}}{\sum_{j=1}^{N}\left(e^{\mathrm{sim}\left(f(\mathbf{x}_i), f\left(\overline{\mathbf{x}_j^+}\right)\right)/\tau} + e^{\mathrm{sim}\left(f(\mathbf{x}_i), f\left(\mathbf{x}_j^-\right)\right)/\tau}\right)} \tag{11}$$

式中，sim( ) 函数表达的是样本之间的余弦相似度，N 为 mini-batch 的数量，$\mathbf{x}_i$ 代表的是原文本表示，$\overline{\mathbf{x}_i^+}$ 代表的是将多轮翻译洗稿后得到的回译文本向量进行融合后的表示，$\mathbf{x}_j^-$ 表示 mini-batch 内的矛盾文本表示。

4. 实验结果及分析

4.1. 实验设置

本文中所有用来进行对比的基准模型的实验配置都保持了一致，模型的具体参数设置如下：网络层数为 12，隐藏网络层数为 768，遮蔽的自注意力机制的级数为 12，全局的自注意力机制的级数为 12；训练时每次输入的 Batch 为 32；为防止模型出现过拟合的问题，Dropout 参数设置为 0.2。整个模型搭建使用的是 Pytorch 框架，训练过程中使用 NVIDIA GeForce RTX 2080 Ti GPU 进行计算。本文实验环境配置和实验参数具体设置如表 2 所示。

Table 2. Experimental environment and parameter configuration
表 2. 实验环境和参数配置

环境	配置	参数	配置
操作系统	Windows (64 位)	Dropout	0.2
CPU	i7	内存	64g
内存	64g	词向量维度	768
编程语言	Python3.9	Batch_size	32
计算框架	Pytorch	Epoch	1

4.2. 评价指标

4.2.1. 余弦相似度

在抄袭检测任务中，通常需要计算不同文本之间的相似性。余弦相似度是量化语义空间中两个嵌入之间的相似程度的度量。余弦相似度通过计算两个嵌入之间的夹角的余弦值来测量它们之间的差异。余弦值越接近 1，这两个向量就越相似。本文中使用余弦相似度来计算样本之间的距离，余弦相似度的计算公式如下：

$$\cos\theta = \frac{\boldsymbol{X}\cdot\boldsymbol{Y}}{\|\boldsymbol{X}\|\|\boldsymbol{Y}\|} \tag{12}$$

式中，$\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 分别表示两个待检测的文本表示。

4.2.2. Spearman 相关系数

在多轮翻译洗稿抄袭检测任务中，通过将两个文本序列转换为向量表示来计算它们的相似性。正如 Reimers 等人[20]所证明的，Spearman 相关系数是一种非参数统计方法，因为它不依赖于数据的具体分布，而是通过对数据进行排序来评估两个变量之间的关系。具体计算流程如下：首先，将文本对分别输入到模型中，得到文本表示对，然后计算文本表示对的余弦相似度。在计算完所有文本表示对的余弦相似度后，最后使用 Spearman 相关系数比较模型生成的余弦相似度与手工标记相似度的相关性。从−1 到 1，相关系数越接近 1 或−1，相关性越强，相关系数越接近 0，相关性越弱。Spearman 相关系数的计算公式为：

$$\rho_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n\left(n^2-1\right)} \tag{13}$$

式中，d_i 对应变量的秩之差，即两个变量分别排序后成对的变量位置(等级)差，n 表示观测对象的数量。

4.3. 抄袭检测结果

使用本文方法分别对通过设定翻译的过渡语言及翻译轮次的中英文数据集进行实验。首先对 LCSTS 新闻摘要数据集[21]中的文本使用百度翻译器，随机经过 2 至 4 轮翻译，并且随机选择过渡语言来分别得到原文本为中文和英文的数据集。过渡语言用语种的英文首字母来表示，不同过渡语言的字母表示如表 3 所示：

Table 3. Pivot language abbreviations
表 3. 过渡语言首字母缩写表

Language	English	Chinese	Spanish	German	Japanese	Tibetan
Capital Initial	E	C	S	G	J	T

我们通过翻译器对数据集中的文本进行了 r 轮翻译，其中 r = 2，3，4，在实验测试过程中，随机选择过渡语言和翻译轮数，多轮翻译洗稿抄袭检测的 Spearman 相关系数的结果如表 4 所示：

Table 4. Spearman correlation results of our method under different translation rounds and translation processes
表 4. 本文方法在不同翻译轮数和翻译过程下的 Spearman 相关系数结果

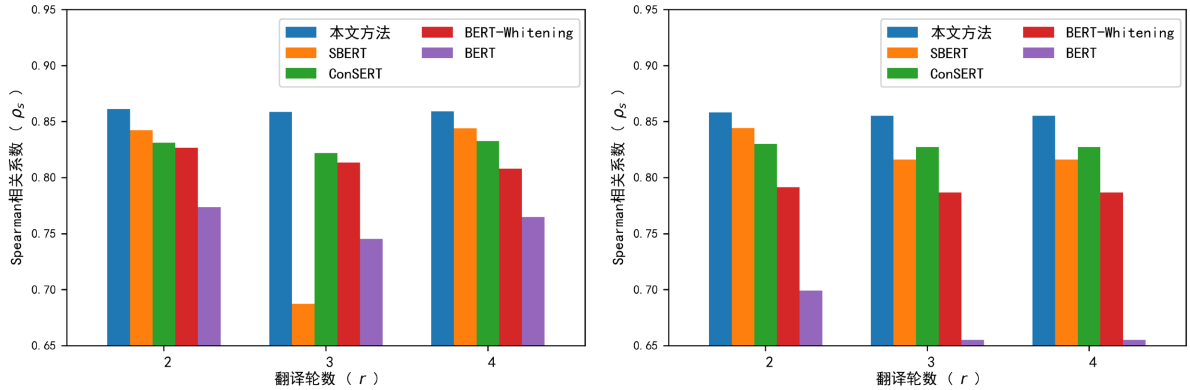
Round	English as the Source Language		Chinese as the Source Language	
	Translation Process	ρ_s	Translation Process	ρ_s
r = 2	E → C → E	0.8574	C → E → C	0.8564
	E → G → E	0.8583	C → G → C	0.8544
	E → T → E	0.8563	C → T → C	0.8611
r = 3	E → C → G → E	0.8561	C → E → G → C	0.8413
	E → G → T → E	0.8555	C → G → T → C	0.8547
	E → T → J → E	0.8549	C → T → J → C	0.8551
r = 4	E → T → S → C → E	0.8543	C → T → S → E → C	0.8547
	E → G → C → S → E	0.8442	C → G → E → S → C	0.8473
	E → C → J → T → E	0.8511	C → E → J → T → C	0.8445

由实验结果可得，以英语为原文本的实验通过使用“E → G → E”方法的两轮翻译过程获得了最佳的性能。另一方面，在以中文为原文本的实验中，使用“C → T → C”过程进行两轮翻译，获得了最好的结果。分析原因可得，英语和德语都属于日耳曼语系，而汉语和藏语则属于汉藏语系。同一语系中的语言在词源和语法方面有很大程度的相似性。因此，来自同一语系的语言之间的相互翻译的容易程度明显高于来自不同语系的语言。

4.4. 抄袭检测性能对比

此外，本文还分别在以英文和中文为原文本的翻译洗稿数据集进行了实验。不同模型的 Spearman 相关系数的结果如下图所示。其中如图 2(a)所示，在以中文作为源语言的实验中，选用“C → E → C”，

“C → E → G → C”和“C → T → S → E → C”作为 2 至 4 轮翻译洗稿抄袭检测来进行实验。如图 2(b)所示，在以英文作为原文本实验中，本文选择了“E → C → E”，“E → C → G → E”和“E → T → S → C → E”作为 2 至 4 轮翻译洗稿抄袭检测来进行实验。实验结果显示本文方法在多轮翻译洗稿抄袭检测任务上性能的显著提升。



(a) 中文为源语言　　(b) 英文为源语言

Figure 2. Performance comparison of different models on translation laundering dataset
图 2. 不同模型在翻译洗稿数据集上的性能比较

4.5. 文本相似度性能对比

为了验证本文方法能否完成传统的文本相似度检测任务，将本文方法与之前 STS 任务上最先进的相似度计算方法进行了比较并使用受监督的数据集 NLI 来训练本模型。实验在中英文 STS 数据集上通过预测两个句子之间的关系，判断其是否为蕴涵、中性或矛盾。然后对预测结果使用 Spearman 相关系数来进行评估。在 NLI 数据集中，给定一个前提，标注者需要手动写出一个绝对正确的句子(蕴含)、一个可能正确的句子(中性)和一个绝对错误的句子(矛盾)。因此，对于每个前提及其蕴涵假设，都有一个伴随的矛盾假设。形式上为三元组$\left(x_i, x_i^+, x_i^-\right)$，其中 x_i 是前提，x_i^+ 和 x_i^- 是蕴含假设和矛盾假设[22]。不同模型在 STS 数据集上的结果如表 5 所示，实验显示本文算法将最佳 Spearman 相关系数从 0.7329 提高到 0.7850。

Table 5. Comparison of results of different models on STS data set
表 5. 不同模型在 STS 数据集上的结果比较

	本文方法	SBERT	ConSERT	BERT-Whitening	BERT
中文	**0.7850**	0.7329	0.7175	0.6722	0.5600
英文	**0.7641**	0.6881	0.6748	0.6755	0.5818

5. 总结

本文提出了一种基于深度学习的翻译洗稿抄袭检测算法。该算法采用对比学习架构，在训练的过程中设计了一个特征融合机制，通过对特征表示进行分组、增强和融合，构建了一个综合且高质量的特征表示。此外，通过改进的对比损失函数将模型训练的焦点放在提高模型的区分能力上。相比于此前的方法，本文方法重视高维特征重要性的差异，为不同的特征分配权重系数，从而提高了算法的检测能力。最后利用所构建的多元组翻译洗稿数据集来对模型进行训练和验证。实验结果表明，本文方法在翻译洗

稿数据集上的 Spearman 相关系数结果达到了 86.11%，比其他算法在多轮翻译洗稿任务上有更好的表现。

参考文献

[1] 刘宏更. 基于小样本学习的文档查重系统的设计与实现[D]: [硕士学位论文]. 北京: 北京邮电大学, 2023.

[2] Jones, M. (2009) Back-Translation: The Latest form of Plagiarism. *The* 4*th Asia Pacific Conference on Educational Integrity*, Wollongong, 28-30 September 2009, 1-7.

[3] Anchal, P. and Urvashi, G. (2023) A Review on Diverse Algorithms Used in the Context of Plagiarism Detection. 2023 *International Conference on Advancement in Computation & Computer Technologies* (*InCACCT*), Gharuan, 5-6 May 2023, 1-6.

[4] Alzahrani, S.M., Salim, N. and Abraham, A. (2012) Understanding Plagiarism Linguistic Patterns, Textual Features, and Detection Methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* (*Applications and Reviews*), **42**, 133-149. https://doi.org/10.1109/tsmcc.2011.2134847 .

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

[6] Alzahrani, S. and Salim, N. (2010) Fuzzy Semantic-Based String Similarity for Extrinsic Plagiarism Detection Lab Report for PAN at CLEF 2010. *CLEF* 2010 *LABs and Workshops, Notebook Papers*, Padua, 22-23 September 2010, 1-8.

[7] El-Rashidy, M.A., Mohamed, R.G., El-Fishawy, N.A. and Shouman, M.A. (2023) An Effective Text Plagiarism Detection System Based on Feature Selection and SVM Techniques. *Multimedia Tools and Applications*, **83**, 2609-2646. https://doi.org/10.1007/s11042-023-15703-4

[8] Poibeau, T. (2017) Machine Translation. MIT Press. https://doi.org/10.7551/mitpress/11043.001.0001

[9] Yoon, K. (2014) Convolutional Neural Networks for Sentence Classification. In *Proceedings of the* 2014 *Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Doha, 25-29 October 2014, 1746-1751.

[10] 厍向阳, 刘哲, 董立红. 基于多尺度注意力特征融合的场景文本检测[J]. 计算机工程与应用, 2024, 60(1): 198-206.

[11] Jeffrey, P., Richard, S. and Christopher, D.M. (2014) Glove: Global Vectors for Word Representation. *Proceedings of the* 2014 *Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Doha, 25-29 October 2014, 1532-1543.

[12] Jacob, D., Ming-Wei, C., Kenton, L. and Kristina, T. (2019) BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT* 2019, Minneapolis, 2-7 June 2019, 4171-4186.

[13] Jun, G., Di, H. and Xu, T. (2018) Representation Degeneration Problem in Training Natural Language Generation Models. *International Conference on Learning Representations*, New Orleans, 6-9 May 2018.

[14] Nils, R. and Iryna, G. (2019) Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the* 2019 *Conference on Empirical Methods in Natural Language Processing and the* 9*th International Joint Conference on Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong, 3-7 November 2019, 3982-3992.

[15] Li, B., Zhou, H. and He, J.X. (2020) On the Sentence Embeddings from Pre-Trained Language Models. *Proceedings of the* 2020 *Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), 16-18 November 2020, 9119-9130. https://doi.org/10.18653/v1/2020.emnlp-main.733

[16] Su, J.L., Cao, J.R., Liu, W.J. and Ouyang, Y.W. (2021) Whitening Sentence Representations for Better Semantics and Faster Retrieval. arXiv: 2103.15316.

[17] Yan, Y.M., Li, R.M. and Wang, S.R. (2021) ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. *Proceedings of the* 59*th Annual Meeting of the Association for Computational Linguistics and the* 11*th International Joint Conference on Natural Language Processing* (*Volume* 1: *Long Papers*), August 2021, 5065-5075. https://doi.org/10.18653/v1/2021.acl-long.393

[18] Wikipedia (2022) Spaces.Ac.cn. https://spaces.ac.cn/archives/8860

[19] Li, X., Hu, X.L. and Yang, J. (2019) Spatial Group-Wise Enhance: Improving Semantic Feature Learning in Convolutional Networks. arXiv: 1905.09646.

[20] Nils, R., Philip, B. and Iryna, G. (2016) Task-Oriented Intrinsic Evaluation of Semantic Textual Similarity. *Proceedings of COLING* 2016, *the* 26*th International Conference on Computational Linguistics*: *Technical Papers*, Osaka, 11-16 December 2016, 87-96.

[21] Hu, B.T., Chen, Q.C. and Zhu, F.Z. (2015) LCSTS: A Large Scale Chinese Short Text Summarization Dataset. *Proceedings of the* 2015 *Conference on Empirical Methods in Natural Language Processing*, Lisbon, 17-21 September

2015, 1967-1972. https://doi.org/10.18653/v1/D15-1229

[22] Gao, T.Y., Yao, X.C. and Chen, D.Q. (2021) Simcse: Simple Contrastive Learning of Sentence Embeddings. 2021 *Conference on Empirical Methods in Natural Language Processing*, 7-11 November 2021, 6894-6910. https://doi.org/10.18653/v1/2021.emnlp-main.552