Data Augmentation for Text-based Person Retrieval Using Large Language Models

Anonymous ACL submission

Abstract

Text-based Person Retrieval aims to retrieve person images that match the description given a text query. The performance of the TPR model relies on high-quality data. However, it is challenging to construct a large-scale, high-quality TPR dataset due to expensive annotation and privacy protection. Recently, Large Language Models (LLMs) have approached human performance on many NLP tasks, creating the possibility to expand high-quality TPR datasets. This paper proposes the first LLM-based Data Augmentation (LLM-DA) method for TPR. LLM-DA uses LLMs to rewrite the text in the TPR dataset, achieving high-quality expansion concisely and efficiently. These rewritten texts are able to increase text diversity while retaining the original key semantic concepts. To alleviate hallucinations of LLMs, LLM-DA introduces a Text Faithfulness Filter to filter out unfaithful rewritten text. To balance the contributions of original and augmented text, a Balanced Sampling Strategy is proposed to control the proportion of original and augmented text used for training. LLM-DA is a plug-and-play method that can be integrated into various TPR models. Comprehensive experiments show that LLM-DA can improve the retrieval performance of current TPR models.

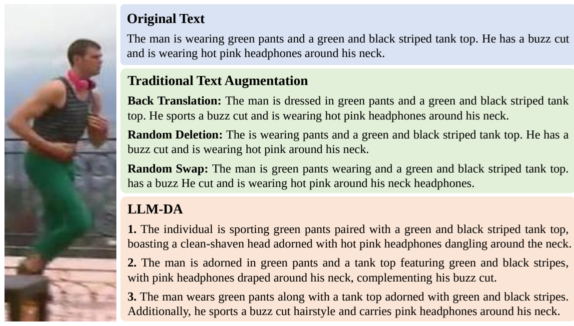


Figure 1: Original person image, original text, and augmented text.

1 Introduction

Text-based Person Retrieval (TPR) (Jiang and Ye, 2023) aims to retrieve person images that match the description given a text query, which is a sub-task of image-text retrieval (Chen et al., 2020a) and person re-identification (Re-ID) (Ye et al., 2021). TPR can assist in identifying individuals captured in surveillance footage based on textual descriptions. TPR has implications for surveillance and security applications, where identifying individuals based on textual descriptions can aid in law enforcement and public safety efforts.

Current studies (Jiang and Ye, 2023; Bai et al., 2023) on TPR mainly focus on extracting discriminative feature representations and fine-grained feature alignment to achieve competitive retrieval performance. As a multi-modal learning task, the performance improvement of the TPR model relies on high-quality data for supervised training. However, it is challenging to construct a large-scale, high-quality TPR dataset. Due to the following two reasons: 1) **Lack of data.** Due to privacy protection, it is challenging to obtain large-scale person images. 2) **Lack of high-quality annotation.** Text annotation is tedious and inevitably introduces annotator biases. Therefore, the texts in the current TPR datasets are usually short and cannot comprehensively describe the characteristics of the target person. In order to solve this problem, Yang *et al.* (Yang et al., 2023) construct a large-scale multi-attribute dataset, MALS, for the pre-training of the TPR task. It takes a lot of manpower and material resources to construct MALS, and we are grateful for their contribution to the TPR field.

In addition to constructing large-scale datasets, data augmentation is also an effective way to expand data scale and facilitate model training. Compared with dataset construction, data augmentation has lower labor and material costs. Cao *et al.* (Cao et al., 2024) conduct a comprehensive empirical study on data augmentation in the TPR task, including image and text augmentation. Image augmentation methods include traditional removal and alteration. Text augmentation methods include back translation, random deletion, *etc*. Most of these traditional image augmentation methods can improve the retrieval performance of TPR models. However, we find that these traditional text augmentation methods do not significantly improve retrieval performance, and some methods even reduce retrieval performance. These text augmentation methods have limited improvement in text diversity. More seriously, some crude text augmentation methods, such as random deletion and random swap, can destroy the correct sentence structure and even change the original semantic concept of the text, as shown in Figure 1. These low-quality augmented texts can have a negative impact on model training.

Recently, Large Language Models (LLMs) have approached or even surpassed human performance on many NLP tasks, creating the possibility to expand high-quality TPR datasets. LLM can be used to rewrite the original text to generate new text, thereby achieving text augmentation. Thanks to the powerful semantic understanding and generation capabilities of LLMs, these rewritten texts are able to increase the diversity of vocabulary and sentence structure while retaining the original key concepts and semantic information. We first explore using LLM for data augmentation in the TPR task. Figure 1 shows the augmented text we generated using the open-source LLM Vicuna (Chiang et al., 2023). The augmented text generated by LLM can enhance the diversity of the text while maintaining the correct sentence structure. Although LLM has powerful generation capabilities, hallucinations have always been a thorny problem that LLM cannot solve. It is possible for LLM to generate augmentation text that does not meet expectations, which is an issue that needs to be addressed. In addition, how to balance the original data and augmented data to give full play to the role of data augmentation is also a challenge that needs to be solved.

This paper proposes the first LLM-based Data Augmentation (LLM-DA) method for TPR. LLM-DA uses LLMs to rewrite the text in the current dataset, achieving high-quality expansion concisely and efficiently. These rewritten texts are able to increase the diversity of vocabulary and sentences while retaining the original key semantic concepts. To alleviate hallucinations of LLMs, LLM-DA introduces a Text Faithfulness Filter (TFF) to filter out unfaithful rewritten text. To balance the contributions of original and augmented text, a Balanced Sampling Strategy (BSS) is proposed to control the proportion of original text and augmented text used for training. LLM-DA neither changes the original model architecture nor affects the form of the original loss function. Therefore, LLM-DA is a plug-and-play method that can be easily integrated into various TPR models. The major contributions of this paper are summarized as follows:

- We propose an LLM-DA method for TPR, using LLMs to rewrite the text in the dataset, achieving high-quality expansion. This is the first exploration of using LLM for data augmentation in TPR.
- We propose a TFF to filter out unfaithful rewritten text to alleviate hallucinations in LLMs.
- We propose a BSS to control the proportion of original text and augmented text used for training.
- LLM-DA can be plug-and-play integrated into various TPR models. Comprehensive experiments on TPR benchmarks show that LLM-DA can improve the retrieval performance of TPR models.

2 Related work

2.1 Text-based Person Retrieval

TPR (Jiang and Ye, 2023) aims to retrieve person images that match the description given a text query. Feature extraction and alignment are the core steps to achieving TPR.

Feature Extraction refers to extracting discriminative features from input person images and text descriptions. Li *et al.* (Li et al., 2017a,b) use LSTM to extract text features and CNN to extract image features. Zhu *et al.* (Zhu et al., 2021) use ResNet-50 (He et al., 2016) to extract image features and Bi-GRU to extract text features. In recent years, with the emergence of Transformer (Vaswani et al., 2017) and BERT (Devlin et al., 2018), large-scale pre-trained models are used to extract features. Han *et al.* (Han et al., 2021) first introduce Contrastive Language-Image Pre-Training (CLIP) (Radford et al., 2021) for feature extraction. Yang *et al.* (Yang et al., 2023) apply Swin Transformer (Liu et al., 2021) to extract image features and BERT to extract text features. Bai *et al.* (Bai et al., 2023) use the large-scale vision-language pre-trained model ALBEF (Li et al., 2021) to extract image and text features.

Feature Alignment refers to the process of effectively matching image and text features. Li *et al.* (Li et al., 2017a) use cross-modal cross-entropy loss for feature alignment. Li *et al.* (Li et al., 2017b) propose a RNN with gated neural Attention mechanism to capture the relationship between images and text. In addition to loss functions and attention mechanisms, recent studies (Zhu et al., 2021; Niu et al., 2020; Wang et al., 2020; Jing et al., 2020) use more complex models for feature alignment. Zhu *et al.* (Zhu et al., 2021) use five different modules and loss functions for feature alignment. Jing *et al.* (Jing et al., 2020) propose a moment alignment network to solve the cross-domain and cross-modal alignment problems. Later studies *et al.* (Jiang and Ye, 2023) focus more on the fine-grained alignment of multimodalities. Yang *et al.* (Yang et al., 2023) incorporate the tasks of image-text contrastive Learning, image-text matching learning, and masked language modeling to impose the alignment constraints. Bai *et al.* (Bai et al., 2023) propose relationship-aware learning and sensitivity-aware learning.

Most TPR studies focus on improving retrieval performance through the feature level, but high-quality data is crucial to improving the performance of supervised learning models. Privacy protection and annotation make building large-scale, high-quality datasets challenging. In order to solve this problem, Yang *et al.* (Yang et al., 2023) construct a large-scale TPR dataset, MALS, for pre-training, which takes a lot of manpower and material resources. In order to obtain large-scale, high-quality data at a low cost, this paper first considers using LLMs for data augmentation in TPR.

2.2 Data Augmentation

Data augmentation increases the diversity of the data and improves the robustness of the model by changing and expanding the original data. TPR datasets are usually constructed in the form of image-text pairs. Therefore, the data augmentation of TPR datasets requires considering both image augmentation and text augmentation.

Image Augmentation. There are a lot of methods of image augmentation. Commonly used traditional methods include random cropping, flipping, scaling, *etc*. In addition, some novel image augmentation methods, such as Mixup (Zhang et al., 2017) and CutMix (Yun et al., 2019), are also widely used. Mixup randomly selects two images in each batch and mixes them in a certain ratio to generate a new image. Previous studies (Simonyan and Zisserman, 2014; Szegedy et al., 2016) have demonstrated that the data augmentation of images can effectively improve the generalization and robustness of the model. In particular, Cao *et al.*(Cao et al., 2024) point out that image augmentation can improve the retrieval performance of TPR.

Text Augmentation. Text augmentation faces more challenges because of the complexity, abstraction, flexibility, scarcity, and diversity of text. EDA (Wei and Zou, 2019) is a simple text augmentation method, including synonym replacement, random insertion, *etc*. Back translation (Fadaee et al., 2017) generates new sentences by translating text into another language and then back. Although back translation is widely used and has achieved certain success, due to cultural differences between different languages, it may lead to semantic inconsistency. CutMixOut (Fawakherji et al., 2024) combines Cutout (DeVries and Taylor, 2017) and CutMix (Yun et al., 2019) to randomly replace and remove text subsequences through a binary mask. However, these methods may destroy the structural and semantic information of sentences, and the augmented texts lack diversity. With the widespread application of LLMs, text augmentation can be performed using LLMs. While ensuring the semantic integrity of the sentence, LLMs can also increase the diversity of sentence structure. Fan *et al.* (Fan et al., 2024) improve CLIP performance by augmenting text with LLMs. Vertical applications such as TPR are short on high-quality data and need to be supplemented by high-quality data augmentation. However, there is currently no research on using LLM to perform data augmentation on TPR.

2.3 Large Language Models

The Transformer architecture provides the basis for the subsequent generation of LLMs. Radford *et al.* (Radford et al., 2018) introduce GPT, which is based on the Transformer architecture and serves as the foundation for the advancement of LLMs. Subsequently, the emergence of a series of GPT models (Radford et al., 2019; Brown et al., 2020; Achiam et al., 2023) further promotes the development of this field. Moreover, the release of open-sourced models like LLaMA (Touvron et al., 2023) and GLM (Du et al., 2022), fine-tuned for various tasks, has served as the backbone for numerous applications. Vicuna (Chiang et al., 2023) introduces a more economical option with its 7B and 13B versions while maintaining impressive performance. These models collectively achieve comparable performances across various benchmarks, creating the possibility to expand high-quality TPR datasets.

Although LLMs can perform well on many tasks,

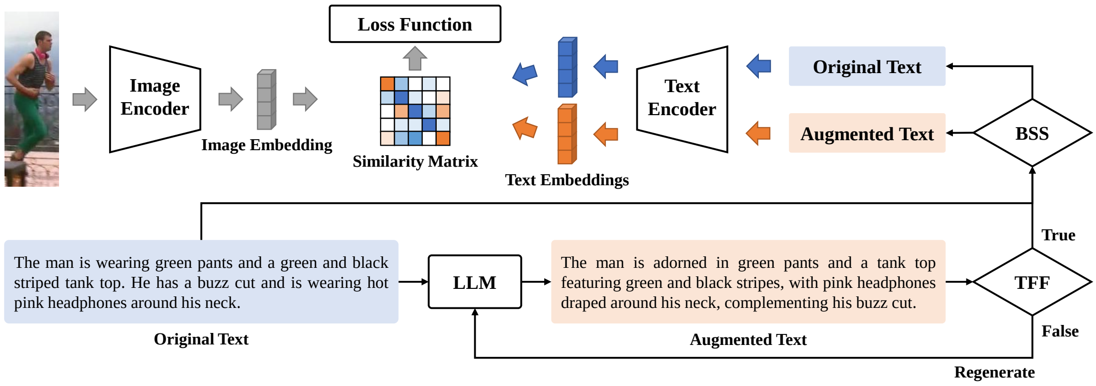


Figure 2: The framework of LLM-based Data Augmentation (LLM-DA) in TPR model training. LLM-DA introduces a Text Faithfulness Filter (TFF) to alleviate the hallucinations of LLMs and a Balanced Sampling Strategy (BSS) to balance the contributions of original text and augmented text.

there are still some problems that need to be solved when applying LLMs for text augmentation. One of the key issues is the hallucination, which refers to the situation where the grammatical correctness, fluency, and authenticity of the generated text are inconsistent with the original text or even inconsistent with the facts (Ye et al., 2023). Hallucination not only reduces the reliability of generated text but may also lead to an uneven quality of output text and sometimes even abnormal text. Therefore, it is necessary to slove the hallucination of LLMs.

3 Methodology

3.1 Preliminary

TPR is defined as retrieving person images relevant to the description of a given text query. We denote $\mathcal{V} = \{V_i\}_{i=1}^{I}$ as a collection of person images and $\mathcal{T} = \{T_i\}_{i=1}^{I}$ as a collection of text descriptions, where V_i is a person image and T_i is a text description. In TPR, given T_i, the goal is to find the most relevant V_i from $\mathcal{V}$. Current TPR models generally follow a common framework, which contains an image encoder $\boldsymbol{f}_{img}(\cdot)$ and a text encoder $\boldsymbol{f}_{text}(\cdot)$. The similarity $s(V_i, T_i)$ between V_i and T_i is computed based on the encoded image feature $\boldsymbol{f}_{img}(V_i)$ and text feature $\boldsymbol{f}_{text}(T_i)$. Finally, the retrieval results are obtained by ranking the similarities.

3.2 LLM-based Data Augmentation

Figure 2 shows the framework of LLM-DA in TPR model training. LLM-DA first utilizes an LLM to rewrite the original text to generate augmented text. Then, to alleviate the hallucinations of LLMs, LLM-DA introduces a TFF to filter out unfaithful rewritten text. On the one hand, the faithfully rewritten text is used as augmented text for model training. On the other hand, LLM-DA discards the unfaithful rewritten text and uses LLM again to rewrite the original text to generate augmented text. Finally, to balance the contributions of original text and augmented text, LLM-DA introduces a BSS to control the proportion of original text and augmented text used for training through sampling. Through the BSS, the caculated similarity matrix between person images and texts is a mixed similarity matrix, which contains both the similarity between the image and the original text and the similarity between the image and the augmented text. This mixed similarity matrix is used to calculate the loss function and implement model training.

Figure 3: Using LLM for text augmentation.

Figure 3 shows how to use LLMs to generate augmented text. This paper chooses the LLM Vicuna (Chiang et al., 2023) for text augmentation, which is an open-source chatbot trained by fine-tuning LLaMA on user-shared conversations collected from ShareGPT. Preliminary evaluation using GPT-4 as a judge shows Vicuna achieves more than 90% of the quality of OpenAI ChatGPT and Google Bard. We concatenate the original text T_i^{ori} and prompt "*Rewrite this image caption.*" and enter them into Vicuna together. Vicuna rewrites the original text T_i^{ori} and returns the augmented text:

$$T_i^{aug} = \text{LLM}(\text{Concat}(T_i^{ori}, \text{Prompt})). \quad (1)$$

Thanks to the powerful generalization of LLMs,

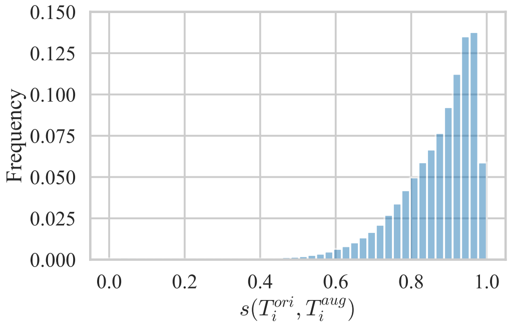


Figure 4: Distribution of $s(T_i^{ori}, T_i^{aug})$ on the CUHK-PEDES dataset.

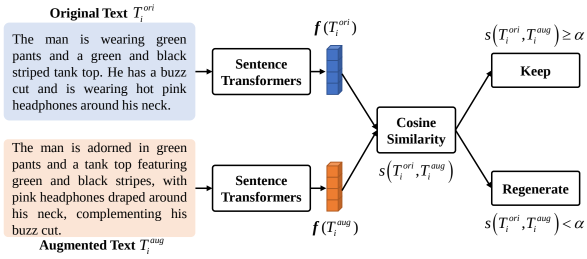


Figure 5: Text Faithfulness Filter (TFF).

most of the text rewritten using LLMs can maintain the same key concepts and semantic information as the original text. In addition, with the powerful generation capabilities of LLMs, using LLMs to rewrite text can enrich the diversity of text data.

3.3 Text Faithfulness Filter

Although LLMs have demonstrated powerful capabilities in various tasks, hallucination is still a prominent problem with LLMs. In the process of using LLMs for text augmentation, we find that the rewritten text output by LLMs may not be semantically consistent with the original text, and LLMs may even output text in other languages or garbled characters. We calculate the semantic similarity between the original text and the augmented text, as shown in Figure 4. More than 90% of the augmented text has a semantic similarity greater than 0.6 with the original text. But there are still a small number of augmented texts that are semantically inconsistent with the original texts. To alleviate the hallucinations of LLMs, LLM-DA introduces a TFF to filter out unfaithful rewritten text.

The architecture of TFF is shown in Figure 5. The purpose of TTF is to filter out augmented text that does not match the semantics of the original text. Therefore, there is a need to measure the semantic similarity between the original text and the augmented text. To this end, we introduce the Sentence Transformers framework to implement semantic similarity calculation. Sentence Transformers is a Python framework for state-of-the-art sentence, text and image embeddings. First, we use Sentence Transformers $\boldsymbol{f}_{st}(\cdot)$ to encode the original text T_i^{ori} and augmented text T_i^{aug} to obtain original text embedding $\boldsymbol{f}_{st}(T_i^{ori})$ and augmented text embedding $\boldsymbol{f}_{st}(T_i^{aug})$. Then, the semantic similarity between the original text and augmented text can be calculated using cosine similarity:

$$s(T_i^{ori}, T_i^{aug}) = \frac{\boldsymbol{f}_{st}(T_i^{ori})^\top \cdot \boldsymbol{f}_{st}(T_i^{aug})}{\|\boldsymbol{f}_{st}(T_i^{ori})\| \, \|\boldsymbol{f}_{st}(T_i^{aug})\|}. \quad (2)$$

We set a threshold α. When $s(T_i^{ori}, T_i^{aug}) < \alpha$, the augmented text is considered to be semantically inconsistent with the original text. LLM-DA discards the unfaithful rewritten text and uses LLM again to rewrite the original text to generate augmented text. When $s(T_i^{ori}, T_i^{aug}) \geq \alpha$, the augmented text is considered to be semantically consistent with the original text. The faithfully rewritten text is used as an augmented text for model training. Through TFF filtering, noise data in augmented text can be effectively removed, and the quality of training data can be improved.

3.4 Balanced Sampling Strategy

After obtaining the augmented text, the simplest way to use the augmented text for training is to directly add the augmented text to the original dataset. However, there may still be a small amount of noise data in the augmented text, which can have a negative impact on model training. In addition, the distribution of augmented text may be different from that of original text. Introducing too much augmented text for training may be detrimental to the generalization of the model. Therefore, in order to balance the contributions of original text and augmented text, LLM-DA introduces a BSS to control the proportion of original text and augmented text used for training through sampling.

We define T_i^* as the text ultimately used for training. The process of BSS can be expressed as:

$$T_i^* = \begin{cases} T_i^{ori}, & r_i > \beta, \\ T_i^{aug}, & r_i \leq \beta, \end{cases} \quad (3)$$

where r_i is a random number following a uniform distribution with a value range of $[0, 1]$. β is a predefined sampling threshold hyperparameter used to control the proportion of original text and augmented text for training. Balancing the contributions of original text and augmented text can reduce the interference of noisy data on model training while increasing the diversity of training data.

Through the BSS, the caculated similarity matrix between person images and texts is a mixed similarity matrix:

$$\boldsymbol{S} = \begin{bmatrix} s(V_1, T_1^*) & \dots & s(V_N, T_1^*) \\ \vdots & \ddots & \vdots \\ s(V_1, T_N^*) & \dots & s(V_N, T_N^*) \end{bmatrix}, \quad (4)$$

where N is the batch size. $\boldsymbol{S}$ contains both the similarity $s(V_i, T_i^{ori})$ between the image and the original text and the similarity $s(V_i, T_i^{aug})$ between the image and the augmented text. This mixed similarity matrix is used to calculate the loss function and implement model training. In this paper, we use CLIP as a baseline model to implement TPR. The contrastive learning loss used by CLIP after applying LLM-DA can be written as:

$$\mathcal{L}_{\text{Contrastive}}^{v \to t} = -\sum_{i=1}^{N} \log \frac{\exp(s(V_i, T_i^*)/\tau)}{\sum_{j=1}^{N} \exp(s(V_i, T_j^*)/\tau)}, \quad (5)$$

where τ is a temperature coefficient. $\mathcal{L}_{\text{Contrastive}}^{v \to t}$ is the loss of image-to-text retrieval, and the loss $\mathcal{L}_{\text{Contrastive}}^{t \to v}$ of text-to-image retrieval is symmetrical to $\mathcal{L}_{\text{Contrastive}}^{v \to t}$. LLM-DA neither changes the model architecture nor affects the form of the loss function. Therefore, LLM-DA is a plug-and-play method that can be easily integrated into various TPR models without increasing complexity.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct comprehensive experiments on three TPR datasets: CUHK-PEDES (Li et al., 2017b), ICFG-PEDES (Ding et al., 2021), and RSTPReid (Zhu et al., 2021).

• **CUHK-PEDES** (Li et al., 2017b) contains 40,206 images and 80,412 sentences for 13,003 identities. The training set consists of 11,003 identities, 34,054 images, and 68,108 sentences. The validation set and test set contain 3,078 and 3,074 images, 6158 and 6156 sentences, respectively, and both of them have 1,000 identities.

• **ICFG-PEDES** (Ding et al., 2021) contains a total of 54,522 images for 4,102 identities. The dataset is divided into a training set and a test set; the former comprises 34,674 image-text pairs of 3,102 identities, while the latter contains 19,848 image-text pairs for the remaining 1,000 identities.

• **RSTPReid** (Zhu et al., 2021) contains 20,505 images of 4,101 identities. Each identity has 5 corresponding images taken by different cameras, and each image is annotated with two textual descriptions. The training, validation, and test sets contain 3,701, 200, and 200 identities, respectively.

Method	Rank-1	Rank-5	Rank-10	mAP
CLIP (ViT-B/32)	60.82	81.47	88.50	54.51
+ LLM-DA	**61.45**	**82.41**	**88.68**	**54.77**
CLIP (ViT-B/16)	64.59	83.59	89.51	58.02
+ LLM-DA	**66.33**	**85.31**	**91.03**	**59.92**

Table 1: Experimental results on the CUHK-PEDES dataset.

Evaluation Metrics. We adopt the popular Rank-K metrics (K = 1, 5, and 10) as the primary evaluation metrics. Rank-K reports the probability of finding at least one matching image within the top-K candidate list when given a textual description as a query. In addition, for a comprehensive evaluation, we also adopt the mean Average Precision (mAP) as a retrieval criterion. The higher Rank-K and mAP indicate better performance.

Implementation Details. We use CLIP as a baseline model to implement TPR. Many TPR methods (Cao et al., 2024) use CLIP as the backbone of the model. Since this paper mainly focuses on data augmentation, in order to reflect the gains of data augmentation, we do not use the various tricks proposed for TPR and only use the original CLIP for experiments. CLIP-ViT-B/16 and CLIP-ViT-B/32 are used as the image encoders, and CLIP Text Transformer is used as the text encoder.

4.2 Improvements to TPR Models

In this section, we present the performance improvements of three TPR datasets on two baseline models. We use two CLIP models used in the latest TPR research (Cao et al., 2024) as baseline models.

Improvements on the CUHK-PEDES Dataset. Table 1 shows the experimental results on the CUHK-PEDES dataset. The performance after applying LLM-DA is better than the original baseline on both models. The performance improvement on the more powerful CLIP (ViT-B/16) model is more significant than that of the CLIP (ViT-B/32) model. Specifically, after applying LLM-DA, the retrieval performance metrics Rank-1 and mAP can be improved by 2.69% and 3.27%, respectively,

Method	Rank-1	Rank-5	Rank-10	mAP
CLIP (ViT-B/32)	51.40	77.05	84.95	41.21
+ LLM-DA	**52.15**	**77.65**	**85.00**	**41.57**
CLIP (ViT-B/16)	55.75	80.20	88.20	44.73
+ LLM-DA	**58.70**	**81.20**	**88.35**	**45.93**

Table 2: Experimental results on the RSTPReid dataset.

Method	Rank-1	Rank-5	Rank-10	mAP
CLIP (ViT-B/32)	52.75	72.27	79.52	31.29
+ LLM-DA	**53.04**	**72.58**	**79.84**	**32.00**
CLIP (ViT-B/16)	56.70	75.25	81.55	35.20
+ LLM-DA	**58.05**	**75.43**	**81.74**	**37.33**

Table 3: Experimental results on the ICFG-PEDES dataset.

Method	Rank-1	Rank-5	Rank-10	mAP
CLIP (ViT-B/16)	55.75	80.20	88.20	44.73
+ Random Deletion	56.50	80.05	88.00	44.13
+ Random Swap	56.95	80.05	88.25	45.13
+ Back Translation	55.95	80.85	**88.50**	45.17
+ LLM-DA	**58.70**	**81.20**	88.35	**45.93**

Table 4: Comparisons with traditional text augmentation methods on the RSTPReid dataset.

DA	TFF	BSS	Rank-1	Rank-5	Rank-10	mAP
-	-	-	64.59	83.59	89.51	58.02
✓	-	-	64.78	84.06	89.93	58.95
✓	✓	-	65.66	85.14	90.98	59.17
✓	-	✓	64.94	84.29	90.59	58.12
✓	✓	✓	**66.33**	**85.31**	**91.03**	**59.92**

Table 5: Ablation studies on the CUHK-PEDES dataset.

compared with the original CLIP (ViT-B/16).

Improvements on the RSTPReid Dataset. Table 2 shows the experimental results on the RSTPReid dataset. On both models, the performance after applying LLM-DA is superior to the initial baseline. The performance improvement on the more powerful CLIP (ViT-B/16) model is more significant than the CLIP (ViT-B/32) model. In particular, compared to the original CLIP (ViT-B/16), the retrieval performance metrics Rank-1 and mAP are improved by 5.29% and 2.68%, respectively, after applying LLM-DA.

Improvements on the ICFG-PEDES Dataset. Table 3 shows the experimental results on the CUHK-PEDES dataset. Applying LLM-DA improves performance on both models over the baseline. In particular, Rank-1 and mAP retrieval performance metrics are improved by 2.38% and 6.05%, respectively, following the application of LLM-DA in comparison to the initial CLIP (ViT-B/16). In summary, LLM-DA can improve the performance of all metrics on all three datasets. This demonstrates the generalization of LLM-DA.

4.3 Comparisons with Text Data Augmentation Methods

LLM-DA is a text augmentation method. There are many traditional text augmentation methods:

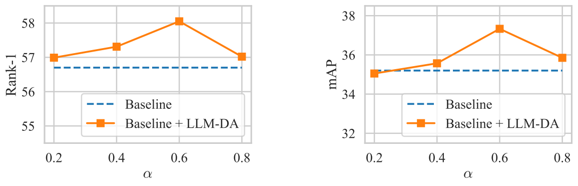
- **Random Deletion** randomly removes words from text.
- **Random Swap** randomly selects two words from the text and swaps their positions.
- **Back Translation** translates the original text into a specific language and back again.

We compare LLM-DA with the above traditional text augmented methods. For back translation, we use French as the intermediate language. It has a relatively closer form to English and introduces fewer changes to the translated back text in semantics than other languages.

Table 4 shows the performance comparisons with traditional text augmentation methods on the RSTPReid dataset. LLM-DA shows significant performance gains compared with other text augmentation methods. Several traditional text augmentation methods fall below the baseline on some evaluation metrics. Random deletion may remove keywords from the text. Random swap may change the original grammatical structure of the text. Both methods may destroy the correct sentence structure and even change the original semantic concept of the text, which may have a negative impact on model training. Back translation can maintain the semantic concepts and grammatical structure of the original text, but the text diversity it can increase is relatively limited. LLM-DA utilizes the powerful generalization and generation capabilities of LLMs, which can not only maintain the semantic concepts and grammatical structure of the original text but also significantly improve the text diversity, thus achieving the most significant performance gain.

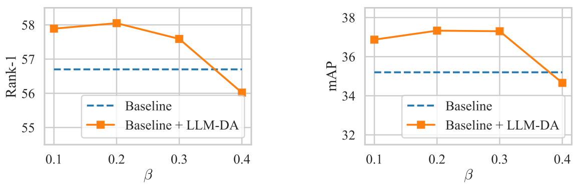
4.4 Ablation Study

Impact of Different Modules. LLM-DA mainly consists of three components: LLM-based Data Augmentation (DA), TFF and BSS. DA first utilizes an LLM to rewrite the original text to generate augmented text. Then, in order to alleviate the hallucinations of LLMs, TFF filters out unfaith-



(a) Rank-1 (b) mAP

Figure 6: The impact of hyperparameter α on retrieval performance on the ICFG-PEDES dataset.



(a) Rank-1 (b) mAP

Figure 7: The impact of hyperparameter β on retrieval performance on the ICFG-PEDES dataset.

ful rewritten text. Finally, in order to balance the contributions of original text and augmented text, BSS controls the proportion of original text and augmented text used for training through sampling.

Table 5 shows the impact of different modules in LLM-DA. The experiment is conducted on the CUHK-PEDES dataset. We adopt the CLIP (ViT-B/16) model as the baseline for the experiment. Compared with the baseline, only data augmentation of text can improve retrieval performance, but the performance improvement is not significant. After TFF filtering, the retrieval performance is significantly improved, since TFF filters out augmented text that is inconsistent with the semantic concepts of the original text, reduces the noise in the training data, and alleviates the negative impact of noisy data on model training. There is a little improvement in retrieval performance following BSS sampling, since balancing the proportion of original and augmented text can also alleviate the negative impact of noisy data to a certain extent and improve generalization. Combining the three modules can achieve optimal performance. This shows that the three modules introduced by LLM-DA can not only improve performance individually but also complement each other.

Hyperparameter Analysis. There are two hyperparameters (α and β) in LLM-DA that can be tuned. α is a predefined similarity threshold in TFF, which is used to decide whether the augmented text should be retained for training. β is a predefined sampling threshold in BSS, which is used to control the proportion of original text and augmented text for training. We experiment with several hyperparameter settings on the ICFG-PEDES dataset using the CLIP (ViT-B/16) model.

As shown in Figure 6, as α increases, the retrieval performance first increases and then decreases. At $\alpha < 0.4$, LLM-DA does not significantly improve performance since more noisy data is used for training, which has a negative impact for training. When $\alpha = 0.6$, the performance reaches the optimal level. However, a larger α is not always better. When $\alpha > 0.8$, since the augmented text is similar to the original text, the diversity of the text data is insufficient and the retrieval performance is reduced, which is not conducive to the generalization of the model. Therefore, the choice of α requires a trade-off between reducing noise data and increasing the diversity of text data.

As shown in Figure 7, as β increases, the retrieval performance first increases and then decreases. When the value of β is small, only less augmented text participates in training, and the contribution to model performance improvement is not significant. When $\beta = 0.2$, the retrieval performance reaches the optimal level. When $\beta > 0.3$, the retrieval performance drops significantly. There are two reasons why the performance decreases when the value of β is large. On the one hand, there may still be a small amount of noise data in the augmented text, which has a negative impact on model training. On the other hand, the distribution of augmented text may be different from the distribution of the original text. To sum up, the value of β needs to balance the proportion of original text and augmented text participating in training.

5 Conclusion

This paper proposes an LLM-DA method for TPR. Specifically, we use LLMs to rewrite the text in the TPR dataset, achieving high-quality expansion of the dataset concisely and efficiently. To alleviate the hallucinations of LLMs, we introduce a TFF to filter out unfaithful rewritten text. To balance the contributions of original and augmented text, a BSS is proposed to control the proportion of original and augmented text used for training. LLM-DA is a plug-and-play method that can be integrated into various TPR models and improve their retrieval performance. In future work, we plan to expand LLM-DA to more cross-modal retrieval tasks.

Limitations

We believe that our LLM-DA can be applied to various text-based cross-modal models as a plug-and-play method.

(1) **Applicable to other domains tasks:** Our method is designed for TPR models, and experimental results show that it significantly improves TPR models. However, we have not yet conducted comprehensive experiments for performance in other domains, so performance in some domains remains unknown.

(2) **Uncertainty in time spent:** During the experiments, the optimal choice of hyperparameters depends on the specific TPR model and dataset. Finding the optimal combination of hyperparameters can be a time-consuming process. The time required for the data augmentation part using the LLM-DA method depends on the number of texts to be augmented and the performance of the LLM used. Therefore, there is uncertainty in the time consumption of the LLM-DA.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yang Bai, Min Cao, Daming Gao, Ziqiang Cao, Chen Chen, Zhenfeng Fan, Liqiang Nie, and Min Zhang. 2023. Rasa: relation and sensitivity aware representation learning for text-based person search. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 555–563.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Min Cao, Yang Bai, Ziyin Zeng, Mang Ye, and Min Zhang. 2024. An empirical study of clip for text-based person search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 465–473.

Hui Chen, Guiguang Ding, Xudong Liu, Zijia Lin, Ji Liu, and Jungong Han. 2020a. Imram: Iterative matching with recurrent attention memory for cross-modal image-text retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12655–12663.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and GeoffreyE. Hinton. 2020b. A simple framework for contrastive learning of visual representations. *Cornell University - arXiv,Cornell University - arXiv*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Terrance DeVries and Graham W Taylor. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Zefeng Ding, Changxing Ding, Zhiyin Shao, and Dacheng Tao. 2021. Semantically self-aligned network for text-to-image part-aware person re-identification. *arXiv preprint arXiv:2107.12666*.

Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. 2020. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 736–740. IEEE.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. 2022. Dg-stgcn: Dynamic spatial-temporal modeling for skeleton-based action recognition.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. *arXiv preprint arXiv:1705.00440*.

Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. 2024. Improving clip training with language rewrites. *Advances in Neural Information Processing Systems*, 36.

Mulham Fawakherji, Eduard Vazquez, Pasquale Giampa, and Binod Bhattarai. 2024. Textaug: Test time text augmentation for multimodal person re-identification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 320–329.

Xiao Han, Sen He, Li Zhang, and Tao Xiang. 2021. Text-based person search with limited data. *arXiv preprint arXiv:2110.10807*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Ding Jiang and Mang Ye. 2023. Cross-modal implicit relation reasoning and aligning for text-to-image person retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2787–2797.

Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. 2020. Cross-modal cross-domain moment alignment network for person search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10678–10686.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705.

Shuang Li, Tong Xiao, Hongsheng Li, Wei Yang, and Xiaogang Wang. 2017a. Identity-aware textual-visual matching with latent co-attention. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1890–1899.

Shuang Li, Tong Xiao, Hongsheng Li, Bolei Zhou, Dayu Yue, and Xiaogang Wang. 2017b. Person search with natural language description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1970–1979.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022.

Xinhao Mei, Xubo Liu, Jianyuan Sun, Mark D Plumbley, and Wenwu Wang. 2022. On metric learning for audio-text cross-modal retrieval. *arXiv preprint arXiv:2203.15537*.

Kai Niu, Yan Huang, Wanli Ouyang, and Liang Wang. 2020. Improving description-based person re-identification by multi-granularity image-text alignments. *IEEE Transactions on Image Processing*, 29:5542–5556.

Mathis Petrovich, Michael J Black, and Gül Varol. 2023. Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9488–9497.

Matthias Plappert, Christian Mandery, and Tamim Asfour. 2016. The kit motion-language dataset. *Big data*, 4(4):236–252.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Omer Terlemez, Stefan Ulbrich, Christian Mandery, Martin Do, Nikolaus Vahrenkamp, and Tamim Asfour. 2014. Master motor map (mmm) — framework and toolkit for capturing, representing, and reproducing human motion on humanoid robots. In *2014 IEEE-RAS International Conference on Humanoid Robots*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Zhe Wang, Zhiyuan Fang, Jun Wang, and Yezhou Yang. 2020. Vitaa: Visual-textual attributes alignment in person search by natural language. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 402–420. Springer.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Shuyu Yang, Yinan Zhou, Zhedong Zheng, Yaxiong Wang, Li Zhu, and Yujiao Wu. 2023. Towards unified text-based person retrieval: A large-scale multi-attribute and language search benchmark. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 4492–4501.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. 2023. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*.

Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven CH Hoi. 2021. Deep learning for person re-identification: A survey and outlook. *IEEE* *transactions on pattern analysis and machine intelligence*, 44(6):2872–2893.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Yuhao Zhang, Hang Jiang, Yasuhide Miura, ChristopherD. Manning, and CurtisP. Langlotz. 2020. Contrastive learning of medical visual representations from paired images and text. *Cornell University - arXiv,Cornell University - arXiv*.

Aichun Zhu, Zijie Wang, Yifeng Li, Xili Wan, Jing Jin, Tian Wang, Fangqiang Hu, and Gang Hua. 2021. Dssl: Deep surroundings-person separation learning for text-based person retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 209–217.

A Appendix

We include here extra information that supports the results presented in the main body of the paper.

A.1 TPR Experimental Setup

Datasets. We conduct comprehensive experiments on three TPR datasets: CUHK-PEDES (Li et al., 2017b), ICFG-PEDES (Ding et al., 2021), and RSTPReid (Zhu et al., 2021).

- **CUHK-PEDES** (Li et al., 2017b) is the first dataset dedicated to TPR, which contains 40,206 images and 80,412 textual descriptions for 13,003 identities. Following the official data split, the training set consists of 11,003 identities, 34,054 images, and 68,108 textual descriptions. The validation set and test set contain 3,078 and 3,074 images, 6158 and 6156 textual descriptions, respectively, and both of them have 1,000 identities.

- **ICFG-PEDES** (Ding et al., 2021) contains a total of 54,522 images for 4,102 identities. Each image has only one corresponding textual description. The dataset is divided into a training set and a test set; the former comprises 34,674 image-text pairs of 3,102 identities, while the latter contains 19,848 image-text pairs for the remaining 1,000 identities.

- **RSTPReid** (Zhu et al., 2021) contains 20,505 images of 4,101 identities from 15 cameras. Each identity has five corresponding images taken by different cameras, and each image is annotated with two textual descriptions. Following the official data split, the training, validation, and test sets contain 3,701, 200, and 200 identities, respectively.

Evaluation Metrics. We adopt the popular Rank-K metrics (K = 1, 5, and 10) as the primary evaluation metrics. Rank-K reports the probability of finding at least one matching person image within the top-K candidate list when given a textual description as a query. In addition, for a comprehensive evaluation, we also adopt the mean Average Precision (mAP) as another retrieval criterion. The higher Rank-K and mAP indicate better performance.

Implementation Details. Our all experiments are conducted on an NVIDIA GeForce RTX 3090 GPU using PyTorch. We use CLIP as a baseline model to implement TPR. CLIP is a neural network trained on a variety of image-text pairs. Many TPR methods use CLIP as the backbone of the model. Since this paper mainly focuses on data augmentation, in order to reflect the gains of data augmentation, we do not use the various tricks proposed for TPR and only use the original CLIP for experiments. CLIP-ViT-B/16 and CLIP-ViT-B/32 are used as the image encoders, and CLIP Text Transformer is used as the text encoder. All person images are resized to 224×224. The maximum length of the textual token sequence is set to 77. The model is trained with the AdamW optimizer with a learning rate initialized to 1×10^{-5}. The training batch size is 80. We use an early stopping strategy to select the optimal model. When the mAP of five consecutive epochs after an epoch no longer grows, the model saved in this epoch is selected as the final model for subsequent testing.

A.2 Qualitative Results of LLM-DA

Figure 8 presents the qualitative results of different text data augmentation methods on the CUHK-PEDES dataset. We compare the proposed LLM-DA method with three traditional text augmention methods. Text augmented using traditional methods may destroy the semantic concepts of the original text. In addition, these texts are similar to the sentence structure of the original text and lack diversity. On the other hand, the text augmented by LLM-DA has more complete semantics and richer sentence structure than the traditional method. This shows that the LLM-DA method has significant advantages in text augmentation, can better retain the semantic information of the original text, and can generate more natural and fluent sentences.

A.3 Other Text-based Cross-modal Retrieval Experiment

We also make an effort to apply the LLM-DA to other text-based cross-modal retrieval models, text-based audio retrieval (TAR) and text-based motion retrieval (TMR). The details of the experimental setup and results are given below.

A.3.1 Experimental Setup

Datasets.

- **TMR Dataset** KIT Motion-Language Dataset (Plappert et al., 2016) contains 3,911 recordings of fullbody motion in the Master Motor Map form (Terlemez et al., 2014), along with textual descriptions for each motion.

Original Text

A woman wearing a purple shirt, a pair of purple shorts and a gray head scarf.

Traditional Text Augmentation

Back Translation: A woman dressed in a purple shirt, purple shorts, and a gray headscarf.

Random Deletion: A ~~woman~~ wearing a purple shirt, a pair of ~~purple~~ shorts and a gray head ~~scarf~~.

Random Swap: A shirt wearing a purple woman, a head of purple shorts and a gray pair scarf.

LLM-DA

A lady dressed in a purple shirt, matching shorts, and adorned with a gray headscarf.

(a)

Original Text

The man is wearing a light t-shirt with dark pants and light sneakers, accessorized with a large black backpack and glasses.

Traditional Text Augmentation

Back Translation: A man wears a light T-shirt with dark pants and light sneakers, accessorized with a large black backpack and glasses.

Random Deletion: The man is ~~wearing~~ a light t-shirt with dark pants and light ~~sneakers~~, accessorized with a ~~large~~ backpack and glasses.

Random Swap: The glasses is wearing a light backpack with dark pants and light sneakers, accessorized with a large black t-shirt and man.

LLM-DA

The gentleman is clad in a light-colored t-shirt, paired with dark trousers and light sneakers, complemented by a sizable black backpack and glasses.

(b)

Figure 8: Qualitative results of different text data augmentation methods on the CUHK-PEDES dataset.

Method	KIT Motion Language Dataset				
	Rank-1 ↑	Rank-5 ↑	Rank-10 ↑	mean ↓	med ↓
Baseline	8.3	30.0	44.2	43.0	13
+ LLM-DA	**9.4**	**31.4**	**47.0**	**39.1**	**11**

Table 6: Experimental results on the KIT Motion Language Dataset.

It has a total of 6,278 annotations in English, where each motion recording has one or more annotations that explain the action. The data is split into 4888, 300, 830 motions for training, validation, and test sets, respectively. In this dataset, each motion is annotated 2.1 times on average.

- **TAR Dataset** Clotho v2 (Drossos et al., 2020) has 3839 audio clips in the training set and 1045 audio clips in the validation and test sets respectively. The length of the audio clips ranges uniformly from 15 to 30 seconds. All the audio clips have five diverse human-annotated captions of eight to 20 words in length.

Evaluation Metrics. Similarly,We adopt the popular Rank-K metrics (K = 1, 5, and 10) as the primary evaluation metrics for TAR and TMR models. We also adopt the median and mean ranks for TAR model, which represent the median and mean rank of the exact result computed among all the queries. The higher Rank-K and mAP indicate better performance. The lower mean and median indicate better performance.

Implementation Details. Our all experiments are conducted on an NVIDIA GeForce RTX 3090 GPU using PyTorch.

- **TAR Experiment** The Bert-base-uncased model is used as the text encoder, and ResNet38 is used as the audio encoder. These pre-trained models are both frozen. We train the model with a batch size of 24 for 50 epochs. The learning rate is 1×10^{-4} and decayed to 1/10 of itself every 20 epochs when training the model. We choose the nex-ent (Chen et al., 2020b) as the loss function.

- **TMR Experiment** We use CLIP Text Transformer to encode text and DG-STGCN (Duan et al., 2022) to encode motion.Info-nce (Zhang et al., 2020) as the loss function for the training model. The model is trained with the AdamW optimizer with a learning rate initialized to 5×10^{-5}. The training batch size is 64, and the epoch is set at 120. The latent dimensionality of the embeddings is $d = 256$. We set the temperature τ to 0.1, and the weight of the contrastive loss term λ_{NCE} to 0.1. The threshold to filter negatives is set to 0.8.

A.3.2 Improvements on the TMR Dataset.

Table 6 presents the performance improvements of the KIT Motion-Language Dataset on the model used in (Petrovich et al., 2023). After applying the LLM-DA, the performance shows significant improvement compared to baseline, indicating that LLM-DA has a significant effect on the performance improvement of the TMR model. In particular, Rank-1 is improved by 13.3% and mean is improved by 9.1% compared to baseline.

A.3.3 Improvements on the TAR Dataset.

Table 7 shows the performance improvements of Clotho v2 on the model used in (Mei et al., 2022).

Method	Text-to-Audio			Audio-to-Text		
	Rank-1	Rank-5	Rank-10	Rank-1	Rank-5	Rank-10
Baseline	7.73	22.99	34.53	8.52	24.98	37.89
+ LLM-DA	**8.36**	**24.13**	**35.37**	**8.61**	**28.13**	**38.37**

Table 7: Experimental results on the Clotho Dataset.

Observing Table 7, we can find that the application of LLM-DA not only improves the performance of Text-to-Audio significantly, but also improves the performance of Audio-to-Text. For the Text-to-Audio task, Rank-1 is improved by 8.2% compared to baseline. For the Audio-to-Text task, Rank-1 is improved by 1.0% compared to baseline.

TLLM-DA is not only suitable for TPR, but also excels in other text-based cross-modal retrieval model. Performance improvements on the TAR and TMR datasets further demonstrate the effectiveness and generalizability of LLM-DA.