Negative Sample is Negative in Its Own Way: Tailoring Negative Sentences for Image-Text Retrieval

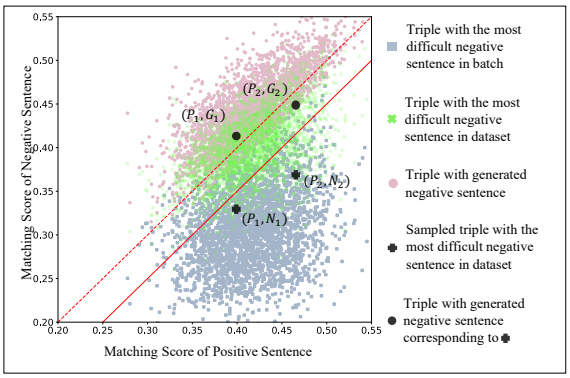
Zhihao Fan[1], **Zhongyu Wei**[1], **Zejun Li**[1], **Siyuan Wang**[1], **Xuanjing Huang**[1], **Jianqing Fan**[2]
[1]Fudan University, [2]Princeton University
{fanzh18,zywei,zejunli20,wangsy18,xjhuang}@fudan.edu.cn, jqfan@princeton.edu

Abstract

Matching model is essential for Image-Text Retrieval framework. Existing research usually train the model with a triplet loss and explore various strategy to retrieve hard negative sentences in the dataset. We argue that current retrieval-based negative sample construction approach is limited in the scale of the dataset thus fail to identify negative sample of high difficulty for every image. We propose our TAiloring neGative Sentences with Discrimination and Correction (TAGS-DC) to generate synthetic sentences automatically as negative samples. TAGS-DC is composed of masking and refilling to generate synthetic negative sentences with higher difficulty. To keep the difficulty during training, we mutually improve the retrieval and generation through parameter sharing. To further utilize fine-grained semantic of mismatch in the negative sentence, we propose two auxiliary tasks, namely word discrimination and word correction to improve the training. In experiments, we verify the effectiveness of our model on MS-COCO and Flickr30K compared with current state-of-the-art models and demonstrates its robustness and faithfulness in the further analysis.

1 Introduction

The task of image-text retrieval takes a query image (sentence) as input and finds out matched sentences (images) from a candidate pool. The key component of the retrieval framework is the similarity computation of an image-sentence pair and it aims to assign higher scores to positive pairs than negative ones. Triplet loss is widely applied for training. Take image-to-text as example[1], it constructs two image-sentence pairs using an image and two sentences (one is relevant and the other is not), and the optimization process increases the similarity of the positive pair while decreasing that of the negative one. Previous research (Xuan et al., 2020) reveals that models trained with harder negative samples, i.e., sentences that are more difficult to be distinguished, can generally achieve better performance. In this line of work, researchers explore various strategies to search mismatched sentences for a query image, from randomly choosing mismatched sentences to using the most similar one. The search



(a) The diagram plots a triplet (image, positive sentence, negative sentence) as a dot is defined by matching score of the positive pair on the X-axis and that of the negative pair on the Y-axis. The matching scores are also computed by CLIP(ViT-B/32) (Radford et al., 2021).

Image	Sentence	Score
	P_1: A man with a gray beard rides his bike on the beach of the ocean.	0.40
	N_1: Man on bike, with bike clothing and helmet on, having trouble maneuvering through sand from beach.	0.34
	G_1: A woman with a gray beard rides his bike on the beach of the ocean.	0.41
	P_2: A little girl is posing on some pumpkins within an area surrounded by flowers.	0.47
	N_2: A girl wearing a red and black striped shirt is sitting on a brick wall near a flower garden .	0.36
	G_2: A little girl is posing on some pumpkins within a beach surrounded by flowers.	0.45

(b) Two images with the positive sentence (P), the most difficult negative one (N) retrieved from dataset by CLIP and the generated negative one (G). The score is the cosine similarity computed by CLIP and larger is better. The underlined red words are non-correspondence ones to the image.

Figure 1: Diagram of matching scores (a) and two examples (b) in Flickr30K (Plummer et al., 2015).

[1]To keep the presentation simple and clear, we use image-to-text as example to represent tasks in both ways throughout the paper.

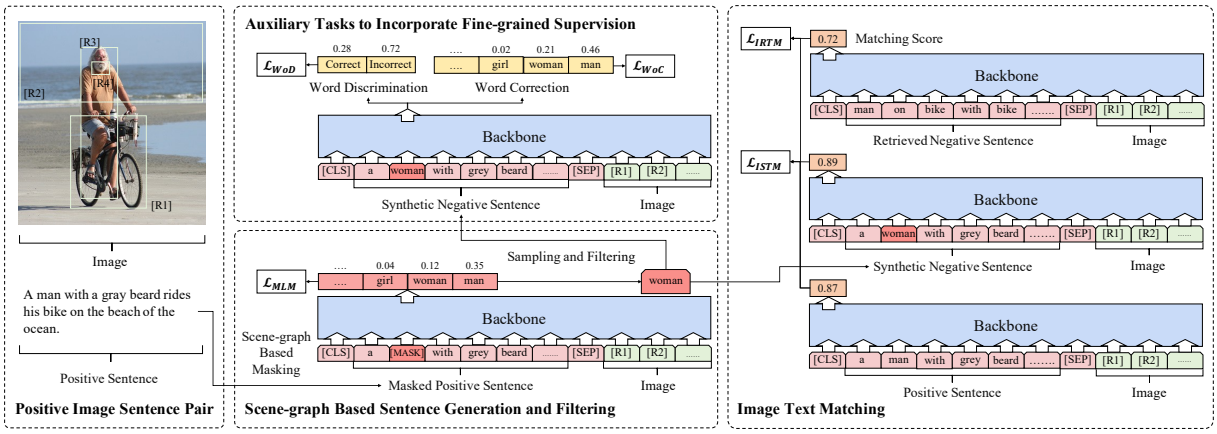


Figure 2: Framework of TAiloring neGative Sentences with word Discrimination and Correction (TAGS-DC).

scope moves from a single training batch (Karpathy and Fei-Fei, 2015; Faghri et al., 2018; Kiros et al., 2014; Socher et al., 2014; Lee et al., 2018; Li et al., 2019) to the whole dataset (Chen et al., 2020a; Zhang et al., 2020). Although promising results have been reported by searching for harder negative samples in a larger scope, the effectiveness is limited by the scale of the dataset.

To compare the effectiveness of these strategies, we randomly sample $3,000$ images in Flickr30K (Plummer et al., 2015) and plot training triples constructed in Figure 1. Each dot stands for a triple (image, positive sentence, negative sentence), and X-axis is the matching score of the positive image-sentence pair while Y-axis is that of the negative one. In general, triples located on the left of the dotted line are more difficult to be distinguished because matching score of the negative pair is higher than the positive one or comparable. As we can see, triples obtained by searching the most difficult mismatched sample in the batch are largely located on the right of the dotted line, and the matching scores of negative pairs are much smaller with a gap larger than 0.05 on average (in the right of the solid line). When enlarging the searching scope to the whole dataset, triples move up in positions, and around 40% of negative pairs obtain higher matching scores than positive ones. However, there are still 18% of images that can only recruit negative samples with a matching score 0.05 lower than its positive counterpart. This confirms the limitation of retrieve-based negative sample construction strategy.

To have a better understanding, we present two triples in Figure 1 i.e., (P_1, N_1) and (P_2, N_2) (denoted as black cross). It shows that negative sentences N_1 and N_2 describe scenes with significant differences compared with the query images, therefore, they are easy to be distinguished. Given that a high percentage of images obtain these low-quality negative sentences in the dataset, we believe it is necessary to collect negative samples beyond retrieval. Instead of searching for original sentences in the dataset, we explore constructing artificial negative samples by editing positive sentences. We demonstrate two generated sentences in Figure 1, G_1 replaces “man" with “woman" on P_1 and G_2 replaces “area" with “beach" on P_2. The generated sentences obtain comparable or even higher matching scores than positive ones. We further generate artificial sentences for all images to form a new set of triples. These triples are plotted in Figure 1 as pink dots. We can see all of them located on the left side of the dotted line, which means they are more difficult to be distinguished.

In this paper, we propose TAiloring neGative Sentences (TAGS) by rewriting keywords in positive sentences of a query image to construct negative samples automatically. In specific, we employ the strategy of *masking* and *refilling*. In masking, we construct scene graph for the positive sentence and mask elements in the graph (objects, attributes, and relations). Refilling replaces the masked original words with mismatched ones to construct the negative sample. In the training process, we further propose two word-level tasks, *word discrimination* and *word correction*, to incorporate fine-grained supervision into consideration. Word discrimination requires the model to distinguish which words lead to the mismatch, and word correction demands the regeneration of the original words. Both tasks evaluate the capability of the model to identify minor differences between synthetic sentences and positive ones. During inference, the output of two tasks

can provide fine-grained information through highlighting and revising mismatched words, and these can be regarded as the explanation for the decision made by the model to improve the interpretability. We evaluate our model on MS-COCO (Lin et al., 2014) and Flickr30K (Plummer et al., 2015). Experiment results show the effectiveness of our model.

Our contributions are three-fold: (1) We propose a generation-based method to construct negative samples to improve the training efficiency of image-text retrieval model. (2) To fully exploit the synthetic negative sentences, we propose two training tasks, word discrimination and word correction, to incorporate the fine-grained supervision to enhance the multi-modal local correspondence modeling. (3) Our model generates state-of-the-art performance on two public datasets MS-COCO and Flickr30K.

2 Framework

The overall framework of TAiloring neGative Sentences with word Discrimination and Correction (TAGS-DC) is shown in Figure 2. For each positive image-text pair (I_i, T_i), we first generate negative sentences $\mathbb{T}_i^-$ through scene-graph based masking and refilling T_i on the basis of masked language model (MLM) in §2.1. Second, we utilizes both retrieved and synthetic negative sentences for the training of image-text matching (IRTM and ISTM) in §2.2, where synthetic negative sentences are exploited in sentence-level. Third, we propose to train the synthetic sentence generator in a dynamic way to keep pace with the upgrading of matching model. Fourth, in §2.4, we apply word-level tasks of word discrimination (WoD) and word correction (WoC) on $\mathbb{T}_i^-$ to discover their differences with T_i for further training. MLM, IRTM, ISTM, WoC and WoD share the same backbone M_θ and have their own heads, namely, H_{MLM}, H_{ITM}, H_{WoC} and H_{WoD}. The detailed training step is illustrated in Algorithm 1 in appendix.

2.1 Scene-graph based Sentence Generation and Filtering

In general, negative sentences with more overlapped words with positive sentences tend to obtain higher matching scores with the query image, thus are more difficult to be distinguished. Therefore, we propose to edit relevant sentences to construct negative samples for a query image. After the sentence generation, we control the quality by filtering the false negative sentences. To ensure the editing operates on key semantic units of the sentence, we use a strategy based on scene-graph.

2.1.1 Scene-graph based Sentence Editing

The module of sentence editing takes a relevant sentence of the query image as input and outputs a synthetic sentence. It first identifies some key semantic units in the sentence and replaces them with other words. We employ a masked language model for this process following two steps namely, masking and refilling.

To identify the key semantic of a sentence, we construct the scene graph for a relevant sentence through scene graph parser of SPICE (Anderson et al., 2016) following SGAE[2] (Yang et al., 2019). We then collect objects, relations, and attributes as candidates for masking. To control the semantic offset of the synthetic sentence $T_i^{(k)}$, we randomly mask 15% tokens of sentence.

In the step of refilling, we use the output head H_{MLM}, which is a two-layer feed-forward network (FFN), on top of the backbone M_θ for masked language modeling. Thus, image I_i also gets involved in MLM to guide the refilling later. The detailed computation of $\mathcal{L}_{MLM}$ is shown in Eq. (1), where $\circ$ is the function composition and *NLL* is the loss of negative log-likelihood.

$$\begin{aligned} MLM &: H_{MLM} \circ M_\theta\big(I_i,\ T_i^{(k)}\big) \rightarrow T_i/T_i^{(k)} \\ \mathcal{L}_{MLM} &= NLL\big(MLM\big(I_i, T_i^{(k)}\big), T_i/T_i^{(k)}\big) \end{aligned} \tag{1}$$

Then during refilling process, we put $T_i^{(k)}$ into MLM to produce the logit scores, then sample the synthetic sentence $T_i^{(k,l)}$ following the distribution which originates from the logit with temperature τ as Eq. (2).

$$T_i^{(k,l)} \sim Softmax\left(MLM\big(I_i, T_i^{(k)}\big)/\tau\right) \tag{2}$$

We conduct the masking and refilling steps for K and L times to generate candidate synthetic sentences.

2.1.2 False Negative Sample Filtering

It hurts the training of using sentences that are relevant to the query image as negative samples (Chuang et al., 2020; Huynh et al., 2020). Therefore we propose a filtering process to remove

[2] https://github.com/yangxuntu/SGAE

false negative ones of synthetic sentences. In vision and language datasets, each image is annotated with multiple descriptive sentences. For example, there are five in MSCOCO and Flickr30K. For a synthetic sentence, if its replaced tokens are completely included in these annotated sentences, we will treat it as relevant. Based on this, we filter synthetic sentences which are relevant.

2.2 Image Text Matching

Given an image I_i and a sentence T_j, the retrieval model assigns a matching score $s \in [0,1]$ of (I_i, T_j) with an output head H_{ITM}, which is a one-layer FFN, as Eq. (3).

$$ITM : H_{ITM} \circ M_\theta(I_i, T_j) \rightarrow s \tag{3}$$

Triplet loss (TripL) is widely applied in image text matching. With a hyper-parameter α, it takes a query image (text) U as an anchor for the matched (positive) image-text pair (U, V) against the mismatched (negative) pair (U, W) as the following equation.

$$\begin{aligned} &TripL_\alpha(U, V, W) \\ &= max\left(\alpha - ITM(U, V) + ITM(U, W),\ 0\right) \end{aligned} \tag{4}$$

Matching on Retrieved Cases During training, for each positive image-text pair (I_i, T_i), we retrieve a negative image I_i^- and a sentence T_i^-, then employ the loss of ITM in Eq. (5) for training,

$$\mathcal{L}_{IRTM} = TripL_\alpha\left(I_i, T_i, T_i^-\right) + TripL_\alpha\left(T_i, I_i, I_i^-\right) \tag{5}$$

Matching on Synthetic Sentences First, we pick up these relatively better generated negative sentences. In practice, we compute the matching score between each synthetic negative sentence and I_i as Eq. (6), and keep a synthetic negative sentence pool $\mathbb{T}_i^-$ to make each of them as difficult as possible.

$$\mathbb{T}_i^- = \underset{T_t^- \in \{T_i^{(k,l)} | T_i^{(k,l)} \neq T_i\}}{argmax\text{-}m} ITM(I_i, T_t^-) \tag{6}$$

where *argmax-m* is to pick out m sentences that earn the top-m matching scores.

Second, with synthetic sentences $\mathbb{T}_i^-$ in Eq. (6), we utilize them and the positive one T_i to compute the triplet loss, and get $\mathcal{L}_{ISTM}$ in Eq. (7).

$$\mathcal{L}_{ISTM} = \frac{1}{|\mathbb{T}_i^-|} \sum_{T_t^- \in \mathbb{T}_i^-} TripL_\alpha\left(I_i, T_i, T_t^-\right) \tag{7}$$

2.3 Dynamic Training Strategy of Negative Sample Generation for Image-Text Matching

The naive choice of MLM is to keep a pre-trained static one: pre-training a MLM in advance and fixing its parameters during the training of ITM. Recall that $\mathcal{L}_{ISTM}$ encourages the ITM model to learn the pattern of synthetic sentences and keep them away from the image, we consider that negative sentences generated by the static MLM would be no longer difficult for the ITM model as the training goes on. We propose to use the dynamic MLM that shares the M_θ with ITM for mutual improvement. Through the sharing, MLM continuously learns what is more relevant to the positive sentences and produces challenging negative ones for the improvement of ITM. The stronger ITM helps MLM to better identify the semantic alignment of image and keywords. MLM achieves the improvement synchronously with ITM through interaction.

2.4 Auxiliary Tasks to Incorporate Fine-grained Supervision

$\mathcal{L}_{ISTM}$ only provides sentence-level supervision and we argue it does not fully exploit the synthetic negative sentence. We introduce two auxiliary tasks to utilize the word-level difference and further enhance the model capability in multi-modal local correspondence modeling.

Word Discrimination The task is to determine whether each word of the synthetic sentence $T_t^- \in \mathbb{T}_i^-$ is matched with I_i, and we regard the replaced words of T_t^- as mismatched ones and others as matched ones. The target label G_t of $T_t^- \in \mathbb{T}_i^-$ is determined following $G_{t,j} = 1$ if $s_{i,j} = s_{t,j}$ else 0, where $s_{i,j}$ and $s_{t,j}$ are the j-th token of T_i and T_t^-. We set up a new output head H_{WoD}, and the objective of word discrimination is in Eq. (8).

$$\begin{aligned} &WoD : H_{WoD} \circ M_\theta\left(I_i, T_t^-\right) \rightarrow G_t \\ &\mathcal{L}_{WoD} = NLL\left(WoD\left(I_i, T_t^-\right), G_t\right) \end{aligned} \tag{8}$$

Word Correction This task is to correct these mismatched words in T_t^- as Eq. (9). The task not only requires the model to comprehensively understand the gap between the synthetic negative sentences and the original positive ones, but also word-dependency knowledge and local cross-modal alignment to fill the gap. H_{WoC} is the output head for word correction, and the objective is

Model	MS-COCO							Flickr30K						
	Image-to-Text			Text-to-Image				Image-to-Text			Text-to-Image			
	R@1	R@5	R@10	R@1	R@5	R@10	RSum	R@1	R@5	R@10	R@1	R@5	R@10	RSum
SCAN	50.4	82.2	90.0	38.6	69.3	80.4	410.9	67.4	90.3	95.8	48.6	77.7	85.2	465.0
MMCA	54.0	82.5	90.7	38.7	69.7	80.8	416.4	74.2	92.8	96.4	54.8	81.4	87.8	487.4
AOQ	55.1	83.3	90.8	41.1	71.5	82.0	423.8	72.8	91.8	95.8	55.3	82.2	88.4	486.3
UNITER+DG	51.4	78.7	87.0	39.1	68.0	78.3	402.5	78.2	93.0	95.9	66.4	88.2	92.2	513.9
Unicoder-VL	62.3	87.1	92.8	46.7	76.0	85.3	450.2	86.2	96.3	99.0	71.5	90.9	94.9	538.8
LightningDOT(B)	64.6	87.6	93.5	50.3	78.7	87.5	462.2	86.5	97.5	98.9	72.6	93.1	96.1	544.7
ERNIE-ViL(B)	-	-	-	-	-	-	-	86.7	97.8	99.1	**75.1**	**93.4**	96.3	548.4
UNITER(B)	64.4	87.4	93.1	50.3	78.5	87.2	460.9	85.9	97.1	98.8	72.5	92.3	96.1	542.7
TAGS-DC(B)	**66.6**	**88.6**	**94.0**	**51.6**	**79.1**	**87.5**	**467.4**	**87.9**	**98.1**	**99.3**	74.5	93.3	**96.3**	**549.4**
CLIP	58.4	81.5	88.1	37.8	62.4	72.2	400.4	88.0	98.7	**99.4**	68.7	90.6	95.2	540.6
LightningDOT(L)	65.7	89.0	93.7	53.0	**80.1**	88.0	469.5	87.2	98.3	99.0	75.6	94.0	96.5	550.6
ERNIE-ViL(L)	-	-	-	-	-	-	-	89.2	98.5	99.2	76.7	94.1	96.7	554.4
UNITER(L)	65.7	88.6	93.8	52.9	79.9	88.0	468.9	87.3	98.0	99.2	75.6	94.1	96.8	551.0
TAGS-DC(L)	**67.8**	**89.6**	**94.2**	**53.3**	80.0	**88.0**	**472.9**	**90.6**	**98.8**	99.1	**77.3**	**94.3**	**97.3**	**557.4**

Table 1: Overall performance of the image-text retrieval. *B* and *L* are the base and large settings.

Model	MS-COCO							Flickr30K						
	Image-to-Text			Text-to-Image				Image-to-Text			Text-to-Image			
	R@1	R@5	R@10	R@1	R@5	R@10	RSum	R@1	R@5	R@10	R@1	R@5	R@10	RSum
TAGS w/ WM	64.9	87.8	93.3	51.1	78.9	87.4	463.4	85.9	97.6	99.1	74.2	93.0	96.1	545.9
TAGS w/ SG	64.1	87.6	93.4	50.9	78.8	87.3	462.1	85.5	97.4	98.9	73.3	92.6	96.0	543.7
TAGS	65.4	88.4	93.6	51.3	79.0	87.5	465.2	87.2	97.8	99.2	74.4	93.1	96.1	547.8

Table 2: Effectiveness of Different Modules. *TAGS w/ WM* means replace the scene-graph based masking with word masking in TAGS. *TAGS w/ SG* means replace dynamic generator with static generator in TAGS.

shown in Eq. (9).

$$\begin{aligned} &WoC : H_{WoC} \circ M_\theta\left(I_i, T_t^-\right) \rightarrow T_i \\ &\mathcal{L}_{WoC} = NLL\left(WoC\left(I_i, T_t^-\right), T_i\right) \end{aligned} \tag{9}$$

2.5 Overall Training

Details of our training step are shown in Algorithm 1 in appendix. The overall training loss of our model has five components as Eq. (10) with hyperparameters λ_{IRTM}, λ_{MLM}, λ_{ISTM}, λ_{WoD} and λ_{WoC}.

$$\begin{aligned} \mathcal{L} = \lambda_{IRTM}\mathcal{L}_{IRTM} + \lambda_{MLM}\mathcal{L}_{MLM} \\ +\lambda_{ISTM}\mathcal{L}_{ISTM} + \lambda_{WoD}\mathcal{L}_{WoD} + \lambda_{WoC}\mathcal{L}_{WoC} \end{aligned} \tag{10}$$

During inference, we employ the ITM to determine the matching score of the query image (text) and the candidate text (image) as Eq. (3).

3 Experiment

Dataset We evaluate our model on MS-COCO (Lin et al., 2014) and Flickr30K (Plummer et al., 2015). In MS-COCO, each image is accompanied with 5 human annotated captions. We split the dataset following (Karpathy and Fei-Fei, 2015) with 113,287 images in the training set and 5,000 images in the validation and test sets, respectively. Flickr30K (Plummer et al., 2015) consists of 31000 images collected from the Flickr website, and every image contains 5 text descriptions. We take the same splits as in (Karpathy and Fei-Fei, 2015), with 1000 images for validation and 1000 images for testing, and the rest for training. 500

Models for Comparison We compare our model with some competitive approaches, including MMCA (Wei et al., 2020), and AOQ (Chen et al., 2020a). We also compare with methods based on vision language pre-trained models: UNITER+DG (Zhang et al., 2020), Unicoder-VL (Li et al., 2020), LightningDOT (Sun et al., 2021), UNITER (Chen et al., 2020b), CLIP (Radford et al., 2021) and ERNIE-ViL (Yu et al., 2020).

Implementation We employ the pre-trained UNITER (Chen et al., 2020b) with base (B) and large (L) settings as our backbone.

Evaluation Metrics We report recall at K (R@K) and Rsum. R@K is the fraction of queries for which the correct item is retrieved among the closest K points to the query. RSum is the sum of R@1+R@5+R@10 in both image-to-text and text-to-image.

3.1 Overall Performance

The overall result is shown in Table 1. TAGS is the model trained with generated negative samples, using the dynamic training strategy. TAGS-DC is our model built on top of TAGS, further trained using two auxiliary tasks. In the base setting, our model achieves the best performance in terms of all metrics except R@1 and R@5 of in text-to-image on Flickr30K. In the large setting, our model also outperforms other models across all metrics except R@5 MS-COCO text-to-image and Flickr30K image-to-image R@10. Compared with UNITER(L), our model achieves an improvement of 4.0 and 6.4 RSum points in MS-COCO and Flickr30K.

3.2 Ablation Study

We further demonstrate the effectiveness of different modules, namely, scene-graph based masking (denoted as PM), dynamic sentence generation (denoted as DG), and fine-grained training tasks (denoted as WoD and WoC) in Flickr30K. Original TAGS is trained with PM and DG. TAGS-DC is further trained with WoD and WoC.

Scene-graph VS Word based Masking We replace the scene-graph based masking with word-based masking (denoted as WM) to form TAGS w/ WM. Detailed results are shown in Table 2. WM follows the original sampling method of UNITER (Chen et al., 2020b) that randomly sample 15% tokens to mask, and PM is introduced in §2.1. TAGS outperforms TAGS w/ WM in terms of all metrics, and this verifies the effectiveness of PM.

Dynamic VS Static Generator We replace DG with a static sentence generator (denoted as SG) to form TAGS w/ SG. The difference between TAGS and TAGS w/ SG lies in that the former shares the parameters of ITM and MLM while the latter does not. Both of them are initialized with the pre-trained UNITER-base and share the same hyperparameters. In detail, we set $\lambda_{MLM} = 0.1$ and $\lambda_{ISTM} = 0.001$. The static generator is fixed as a fine-tuned UNITER+MLM model. The performance of TAGS w/ SG is not so good as TAGS. This demonstrates the effectiveness of DG.

WoD and WoC In Table 2, TAGS-DC outperforms TAGS in both MS-COCO and Flickr30K. This reveals that word discrimination and correction contribute to the performance of ITM.

4 Further Analysis

4.1 Difficulty Distribution of Samples from Dynamic and Static Generator

To see the difficulty of negative samples constructed by various generation strategies, we plot the value distribution of samples. To evaluate the difficulty, we compute the similarity gap between the positive pair ITM(I_i, T_i) and the negative one ITM(I_i, T_t^-). We plot the value of negative pair minus positive one with respect to training steps (X-axis). In general, higher value means higher difficulty. The result is shown in Figure 3 where the darker color means more samples. The overall values of TAGS w/ SG (Figure 3 (a)) are higher than TAGS w/ DG (Figure 3 (b)). This implies that the static generator fails to provide negative sentences close to the image for ITM during training while our generator with dynamic generating strategy is effective.

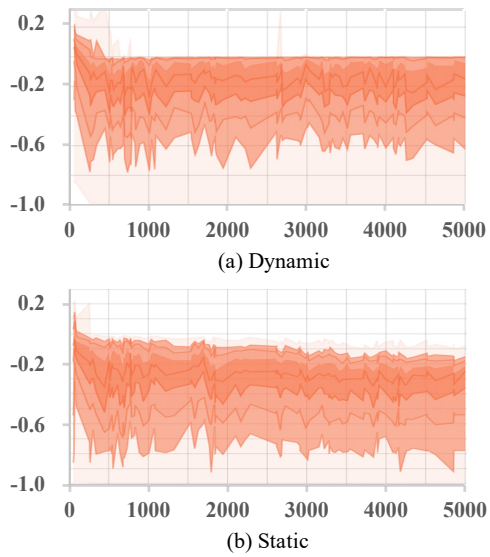


Figure 3: Value $\{\text{ITM}(I_i, T_t^-) - \text{ITM}(I_i, T_i)\}$ distribution of triples generated by dynamic and static generators respectively during the training. X-axis is training steps.

4.2 Quality Evaluation of Synthetic Sentences

We evaluate the quality of generated synthetic sentences in terms of automatic metrics and human evaluation.

Fluency We utilize the pre-trained language model GPT-2 (Radford et al., 2019) to compute the perplexity of synthetic negative sentences for the measurement of their fluency. We use positive sentences in the test set of Flickr30K as original ones and generate negative samples by TAGS and VSE-C. Furthermore, we look into sentences after correction. The overall results are shown in Table 3. Compared with sentences produced by VSE-C, our synthetic sentences have much smaller perplexity. After correction, the fluency of synthetic sentences can be improved.

Human Evaluation We perform a human evaluation to see whether all negative sentences generated are true negative. We randomly sample 200 sentences generated by TAGS and ask two annotators to determine whether the synthetic sentences are mismatched to the corresponding images. The result shows that 96.5% of synthetic sentences generated are true negative.

	Positive	Synthetic	Corrected	VSE-C
Perplexity	51.13	87.63	70.87	292.76

Table 3: Perplexity of synthetic negative sentences.

4.3 Negative Sentences Discrimination

In this section, we explore to see if the generator can discriminate positive sentences from synthetic ones. We compare UNITER and TAGS. For a pair of sentences (one is positive and the other is a synthetic negative one), the generator should assign a higher score to the positive one. We report the accuracy of discrimination. We utilize two negative sentence generators TAGS and VSE-C (Shi et al., 2018). Two versions of TAGS with different seeds are used for cross-validation. Results are shown in Table 4. We have several findings as follows. (1) TAGS2 is trained with a different seed with TAGS1, but the performance of TAGS1 almost makes no difference in discriminating their generated sentences. (2) Although the synthetic sentences of VSE-C are constructed with human efforts, TAGS also outperforms UNITER by about 9%. (3) Three generators produce negative sentences with different distributions, but TAGS performs better than UNITER consistently. These facts validate the robustness of TAGS.

Generator	Discriminator	Accuracy
TAGS1	*TAGS1*	98.7%
	UNITER	2.3%
TAGS2	*TAGS1*	99.7%
	UNITER	2.8%
VSE-C	*TAGS1*	96.3%
	UNITER	87.5%

Table 4: Accuracy of TAGS1 and UNITER in discriminating the negative sentences constructed by TAGS1, TAGS2 and VSE-C (Shi et al., 2018).

4.4 Effectiveness of Two Auxiliary Tasks

Image	Type	Sentence	U	T
	Positive	A man wearing a helmet, floating in the water	92.35	99.90
	Synthetic	A man carrying white helmet , swimming in the water	93.17	98.92
	Corrected	A man wearing a helmet, swimming in the water	-	-
	Positive	A young man about to throw a football	89.54	99.90
	Synthetic	A man playing playing to catch a ball	90.61	75.39
	Corrected	A man player about to throw a ball	-	-

Figure 4: Examples of TAGS-DC. The second column is the sentence type including positive one, synthetic one and corrected one. The third column is the corresponding sentence of the second column. The fourth and fifth columns are the UNITER(U) and TAGS-DC(T) scores for the sentence in the third column, respectively. The word color in synthetic sentences from green to yellow means the increase of the word mismatching scores. Words with underline mean the regenerated words are different from the original ones.

We show the performance of our model in two auxiliary tasks, namely, word discrimination and correction in the testing set of Flickr30K. In word discrimination, we use a threshold of 0.5 to split the positive and negative ones in terms of probability. The accuracy of word discrimination is 66.5%. In word correction, the accuracy is 87.3%. With the probability, we can provide additional support information accompanied to the final decision of our model.

Two examples are presented in Figure 4. (1) TAGS-DC assigns lower scores for synthetic negative sentences than positive ones, but UNITER

fails. (2) Color of "carrying" and "playing playing" are yellow which means that our word discrimination successfully detects these mismatched words. Our model finds the local alignment in word-level and grammatical errors, then generates "wearing" and "man player" for correction. In the examples, word discrimination marks the mismatched components and word correction provides reasons for mismatching. (3) Our model fails to identify two mismatched words, "swimming", and "ball". Considering they are partially related to the image, our model is less effective in determining the relevance of these fuzzy words.

5 Related Work

Image-Text Retrieval Most works in image-text retrieval focus on better feature extraction and cross-modal interaction. Nam et al. (2017) and Ji et al. (2019) represent the image by semantics gathered from block-based attention. A line of research (Lee et al., 2018; Li et al., 2019; Wang et al., 2020; Wei et al., 2020; Li et al., 2021) detects features by pre-trained Faster R-CNN (Ren et al., 2015). Some other methods also focus on enhancing cross-modality relationship modeling, such as the dual attention network (Nam et al., 2017), the stacked cross attention (Lee et al., 2018; Liu et al., 2019; Hu et al., 2019), the graph structure attention (Liu et al., 2020), and the multi-modal transformer modeling (Wei et al., 2020). UNITER (Chen et al., 2020b), Unicoder (Li et al., 2020) and ERNIE-ViL (Yu et al., 2020) follow BERT (Devlin et al., 2019) to pre-train the vision-language transformer model on the large-scale image-text datasets, and finetune in image-text retrieval.

Negative Samples in Contrastive Learning Selection strategies for negative samples have been widely studied in metric learning (Schroff et al., 2015; Oh Song et al., 2016; Harwood et al., 2017; Suh et al., 2019; Zhang et al., 2020; Chen et al., 2020a). Wu et al. (2017) employ distance weighted sampling to select more informative and stable examples. Ge (2018) present a novel hierarchical triplet loss capable of automatically collecting informative training samples. In image-text retrieval, early works (Kiros et al., 2014; Karpathy and Fei-Fei, 2015; Socher et al., 2014) utilize random negative samples for training. VSE++ (Faghri et al., 2018) incorporates difficult negative ones in the multi-modal embedding learning. The method is widely applied in the following works (Lee et al., 2018; Wei et al., 2020), and achieves significant performance improvement. UNITER (Chen et al., 2020b) randomly samples a portion of texts ($\sim$512) from the dataset and picks up the hardest ones. AOQ (Chen et al., 2020a) selects these hard-to-distinguish cases from the whole dataset through a pre-trained ITM model and assigns hierarchical and adaptive penalties for samples with different difficulties. UNITER+DG (Zhang et al., 2020) samples hard negative sentences according to the structure relevance based on denotation graph (Plummer et al., 2015). These methods are retrieval-based and inspire us to find more difficult negative sentences through generation. Chuang et al. (2020) propose a method for debiasing, i.e., correcting for the fact that some negative pairs may be false negatives. In our work, we mask keywords (objects, attributes, and relationships) in the positive sentence then refilling, and exclude these sentences of which each token is included in image annotated sentences. This method introduces new keywords and alleviates the generation of false negative samples. Kalantidis et al. (2020) consider applying mixup to produce hard negatives in latent space. In our work, we directly rewrite the positive sentences that is missing in the latent space based method, and this improves the robustness and faithfulness. The most similar work is VSE-C (Shi et al., 2018) that attacks the VSE++ (Faghri et al., 2018) through replacing the nouns, numerals, and relations according to language priors of human and the WordNet knowledge base. Compare with VSE-C (Shi et al., 2018), our method has three advantages. (1) Our model does not depend on rules. (2) Our model is more flexible and can generate negative sentences with any number, but this is intractable for VSE-C. (3) The generated sentences of our model are more fluent than these of VSE-C as the results in Table 4.

6 Conclusion

In this paper, we focus on the image-text retrieval task and find that retrieve-based negative sentence construction methods are limited by the dataset scale. To further improve the performance, we propose TAiloring neGative Sentences (TAGS). It utilizes masking and refilling to produce synthetic negative sentences as negative samples. We also set up the word discrimination and word correction to introduce word-level supervision to better ex-

ploit the synthetic negative sentences. Our model shows competitive performance in MS-COCO and Flickr30k compared with current state-of-the-art models. We also demonstrate the behavior of our model is robust and faithful.

7 Acknowledgements

This work is partially supported by Natural Science Foundation of China (No.6217020551), Science and Technology Commission of Shanghai Municipality Grant (No.20dz1200600, 21QA1400600, GWV-1.1, 21511101000) and Zhejiang Lab (No.2019KD0AD01).

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *ECCV*.

Tianlang Chen, Jiajun Deng, and Jiebo Luo. 2020a. Adaptive offline quintuplet loss for image-text matching. In *European Conference on Computer Vision*, pages 549–565. Springer.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020b. Uniter: Universal image-text representation learning. In *ECCV*.

Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. 2020. Debiased contrastive learning. *arXiv preprint arXiv:2007.00224*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. Vse++: Improving visual-semantic embeddings with hard negatives. In *Proceedings of the British Machine Vision Conference (BMVC)*.

Weifeng Ge. 2018. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–285.

Ben Harwood, Vijay Kumar B G, Gustavo Carneiro, Ian Reid, and Tom Drummond. 2017. Smart mining for deep metric learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Zhibin Hu, Yongsheng Luo, Jiong Lin, Yan Yan, and Jian Chen. 2019. Multi-level visual-semantic alignments with relation-wise dual attention network for image and text matching. In *IJCAI*, pages 789–795.

Tri Huynh, Simon Kornblith, Matthew R Walter, Michael Maire, and Maryam Khademi. 2020. Boosting contrastive self-supervised learning with false negative cancellation. *arXiv preprint arXiv:2011.11765*.

Zhong Ji, Haoran Wang, Jungong Han, and Yanwei Pang. 2019. Saliency-guided attention network for image-sentence matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5754–5763.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. *arXiv preprint arXiv:2010.01028*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. 2014. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–216.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. 2019. Visual semantic reasoning for image-text matching. In *ICCV*.

Zejun Li, Zhongyu Wei, Zhihao Fan, Haijun Shan, and Xuanjing Huang. 2021. An unsupervised sampling approach for image-sentence matching using document-level structural information. *arXiv preprint arXiv:2104.02605*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *ECCV*.

Chunxiao Liu, Zhendong Mao, An-An Liu, Tianzhu Zhang, Bin Wang, and Yongdong Zhang. 2019. Focus your attention: A bidirectional focal attention network for image-text matching. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 3–11.

Chunxiao Liu, Zhendong Mao, Tianzhu Zhang, Hongtao Xie, Bin Wang, and Yongdong Zhang. 2020. Graph structured network for image-text matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10921–10930.

Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 299–307.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. 2016. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4004–4012.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.

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. *arXiv preprint arXiv:2103.00020*.

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

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Haoyue Shi, Jiayuan Mao, Tete Xiao, Yuning Jiang, and Jian Sun. 2018. Learning visually-grounded semantics from contrastive adversarial samples. *arXiv preprint arXiv:1806.10348*.

Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218.

Yumin Suh, Bohyung Han, Wonsik Kim, and Kyoung Mu Lee. 2019. Stochastic class-based hard example mining for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7251–7259.

Siqi Sun, Yen-Chun Chen, Linjie Li, Shuohang Wang, Yuwei Fang, and Jingjing Liu. 2021. Lightningdot: Pre-training visual-semantic embeddings for real-time image-text retrieval. *arXiv preprint arXiv:2103.08784*.

Sijin Wang, Ruiping Wang, Ziwei Yao, Shiguang Shan, and Xilin Chen. 2020. Cross-modal scene graph matching for relationship-aware image-text retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1508–1517.

Xi Wei, Tianzhu Zhang, Yan Li, Yongdong Zhang, and Feng Wu. 2020. Multi-modality cross attention network for image and sentence matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10941–10950.

Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. 2017. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2848.

Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. 2020. Hard negative examples are hard, but useful. In *European Conference on Computer Vision*, pages 126–142. Springer.

Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. 2019. Auto-encoding scene graphs for image captioning. In *CVPR*.

Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie-vil: Knowledge enhanced vision-language representations through scene graph. *arXiv preprint arXiv:2006.16934*.

Bowen Zhang, Hexiang Hu, Vihan Jain, Eugene Ie, and Fei Sha. 2020. Learning to represent image and text with denotation graph. *arXiv preprint arXiv:2010.02949*.

A Appendix

A.1 Implementation Details

We have two settings, base and large. The base setting of model has 12-layers, 768 hidden size and 12 attention heads and the large one has 24-layers, 1024 hidden size and 16 attention heads.

We utilize grid search to determine the hyperparameters. In retrieval-based matching, we randomly samples 399 negative sentence (image) from the whole dataset for the query image (sentence), and pick out the top 31 ones from them according to the matching scores. In the masked language modeling, we utilize the scene graph parser in SPICE to extract the phrases of objects, relationships and attributes from the positive sentence, and take these phrases as a whole to sample and mask. The mask

probability is 0.15. In the generation enhanced matching, the temperature $\tau \in \{1.0, 1.5\}$, and we set $K = L = 20$ and $|\mathbb{T}_i^-| = 31/23$ for the base and large settings. λ_{ITM}, λ_{MLM}, λ_{ISTM}, λ_{WoD} and λ_{WoC} is sampled from $\{1.0\}$, $\{$5e-2, 1e-1$\}$, $\{$1e-4, 5e-4, 1e-3$\}$, $\{$5e-4, 1e-3$\}$ and $\{$5e-4, 1e-3$\}$, where we set $\lambda_{WoD} = \lambda_{WoC}$.

Our training is composed of two steps, (1) we train with *ITM*, *MLM* and *ISTM* with 5,000 steps as *NSG*; (2) we further train the model with the whole loss function as *NSGDC* with 1,500 steps. The learning rate lr is sampled from $\{$5e-5, 4e-5, 1e-5$\}$. We use a linear learning rate scheduler with 10% warmup proportion. The Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.98$ is taken as the optimizer. The dropout is 0.1.

Our code is implemented with pytorch. For base setting in Flickr30K, we utilize 8 V100 for training and the computation time is about 8 hours.

A.2 Algorithm of TAGS-DC

Algorithm 1 Training step of TAGS-DC

Input: A positive image-text pair (I_i, T_i).
Parameter: Backbone M_θ, the head of masked language model H_{MLM}, image-text matching H_{ITM}, word discrimination H_{WoD} and word correction H_{WoC}.

 # negative sentence generation.
Initializing $\widehat{\mathbb{T}}_i^- := \{\}$.
for k in $1, \ldots, K$ **do**
  Randomly masking T_i to get the masked one $T_i^{(k)}$.
  Computing $\mathcal{L}_{MLM}$ in Eq. (1) with M_θ and H_{MLM}.
  for l in $1, \ldots, L$ **do**
    Refilling $T_i^{(k)}$ to generate a synthetic sentence $T_i^{(k,l)}$ following Eq. (2).
    if $T_i^{(k,l)}$ satisfies criteria C1 **then**
      Adding $T_i^{(k,l)}$ to $\widehat{\mathbb{T}}_i^-$ and computing its matching score with I_i.
    end if
  end for
end for
 # image text matching.
Sampling negative image I_i^- and negative sentence T_i^- to compute $\mathcal{L}_{IRTM}$ in Eq. (5) with M_θ and H_{ITM}.
Picking out top-m synthetic sentences from $\widehat{\mathbb{T}}_i^-$ by the matching scores to constitute $\mathbb{T}_i^-$.
Utilizing $\mathbb{T}_i^-$ and I_i to compute $\mathcal{L}_{ISTM}$ in Eq. (7) with M_θ and H_{ITM}.
 # word discrimination and word correction.
for T_t^- in $\mathbb{T}_i^-$ **do**
  Utilizing T_t^- and I_i to compute $\mathcal{L}_{WoD}$ in Eq. (8) with M_θ and H_{WoD}.
  Utilizing T_t^- and I_i to compute $\mathcal{L}_{WoC}$ in Eq. (9) with M_θ and H_{WoC}.
end for

Dataset	Model	lr	α	τ	$\lvert\mathbb{T}_i^-\rvert$	λ_{ITM}	λ_{MLM}	λ_{ISTM}	λ_{WoD}	λ_{WoC}
Flickr30k	*NSG(B)*	5e-5	0.2	1.5	31	1.0	1e-1	1e-3	-	-
	NSGDC(B)	1e-5	0.2	1.5	31	1.0	1e-1	1e-3	1e-3	1e-3
	NSG(L)	4e-5	0.2	1.5	23	1.0	1e-1	5e-4	-	-
	NSGDC(L)	1e-5	0.2	1.5	23	1.0	1e-1	5e-4	5e-4	5e-4
MS-COCO	*NSG(B)*	5e-5	0.2	1.5	31	1.0	5e-2	1e-4	-	-
	NSGDC(B)	1e-5	0.2	1.5	31	1.0	5e-2	1e-4	5e-4	5e-4
	NSG(L)	4e-5	0.2	1.5	23	1.0	5e-2	5e-4	-	-
	NSGDC(L)	1e-5	0.2	1.5	23	1.0	5e-2	5e-4	5e-4	5e-4

Table 5: Hyper-parameters