AMA: Adaptive Memory Augmentation for Enhancing Image Captioning

Shuang Cheng[1,2]
chengshuang22s@ict.ac.cn
Jian Ye[1,2,†]
jye@ict.ac.cn

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences

Abstract

Memory-Augmented Image Captioning (MA-IC) has demonstrated significant performance improvements over standard neural image captioning systems. It effectively combines a well-trained captioning model with additional explicit knowledge from a memory bank to enhance captioning accuracy. However, the k-nearest neighbor algorithm used in MA-IC retrieves the same number of nearest neighbors for each target token, which may lead to prediction errors when the retrieved neighbors contain noise. In this paper, we propose an adaptive memory feedback mechanism to determine the number of k for each target token. We achieve this by introducing a lightweight network that can be efficiently trained using only a small number of training samples. By incorporating this adaptive memory-augmented method into various captioning baselines, the performance of the resulting captioners consistently improves on the evaluation benchmark. Notably, extensive experiments show that our approach is capable of efficiently adapting to larger training datasets by simply transferring the memory bank with a straightforward network.

1 Introduction

The image captioning task aims to describe the visual content of a given image. Inspired by neural machine translation, most existing models adopt encoder-decoder frameworks in the early study [2, 7, 23]. Recent advances in image captioning can be largely attributed to vision-language pre-training (VLP) , the current prevailing training paradigm for vision-language (VL) research [10, 14, 25]. In addition to these structures that only learn relational knowledge through parameter optimization from training data, an increasing number of hybrid captioning models combining retrieval-based memory mechanisms have gained attention[5, 24]. Most of these methods rely on effective sentence-level retrieval. They first employ an image-text matching model to search for the top-k similar caption candidates. Then, a specially designed network generates a sentence guided by the input image and these relevant sentence candidates. Different from sentence retrieval, memory-augmented image captioning introduces token-level retrieval to improve captioning. It equips a pre-trained

[1]† Corresponding author

captioning model with a kNN classifier over a memory bank of cached context representations and corresponding target tokens, demonstrating promising results for utilizing cached contextual information[8].

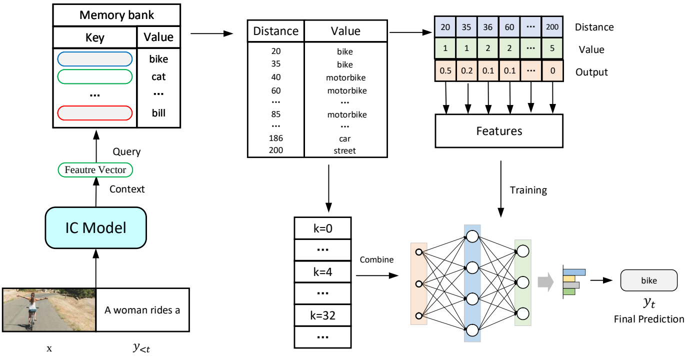

Figure 1: Adaptive Memory-Augmented Image Captioning

Existing retrieval-based methods still have limitations. Sentence-level retrieval methods struggle to find relevant examples for a given instance, and irrelevant retrieved results may mislead the final caption generation[8]. Moreover, these models can only exploit individual sentence-level retrieved results, resulting in high-performance variance[31]. Token-level retrieval methods apply a fixed hyper-parameter k for all cases, which may introduce noise when the target token is challenging to determine. Empirically, we find that caption quality is noise-sensitive, resulting in poor robustness.

Motivated by recent progress in retrieval-based methods[8, 32], we propose the Adaptive Memory-Augmented (AMA) framework for the image captioning task, effectively learning and adapting image captioning in token-level retrieval. As shown in Figure 1, the lightweight network adaptively selects k based on different target tokens. More specifically, we consider retrieval results with multiple kNN classifiers. First, instead of setting a fixed k, we employ a set of k values smaller than an upper bound K. Then, we introduce a lightweight network to adaptively measure the importance of all retrieved k-Nearest Neighbor results based on the current context, combining them to obtain the model's final prediction. In this manner, our framework adaptively determines the utilization of retrieved neighbor information for each target token, effectively filtering the noise in retrieved neighbors. To better measure its effects, we conduct extensive experiments to verify our method's effectiveness on the MS COCO benchmark[17]. Built upon recent strong captioners, our adaptive memory-incorporation mechanism demonstrates significant improvement over the base model when using the same training set to model history memory representations.

The lightweight network requires only thousands of parameters and can be easily trained using the validation dataset. We also demonstrate that our method is more robust than MA[8] when the database quality is suboptimal. Our contributions are summarized as follows:

- We propose an adaptive memory-augmented (AMA) approach that adaptively determines and utilizes neighbor information for each target token, filtering noise in retrieved neighbors. We achieve this by introducing a lightweight network that does not require numerous training samples.

- We apply AMA to strong baselines, achieving state-of-the-art performance on the

COCO benchmark dataset. Extensive experiments show that models equipped with AMA significantly outperform those without MA. We also analyze the effect of the memory bank scale. Encouragingly, the proposed memory mechanism can be easily incorporated into existing captioning models to improve their performance with minimal additional training.

2 Related Works

2.1 Image Captioning

Classical image captioning employs the encoder-decoder architecture to encode images into features and decode these features into sentences [11, 23]. With the emergence of the VilBERT model[19], the field of visual language pre-training models has rapidly developed, establishing the pre-training-fine-tuning paradigm for image captioning [10, 14, 15, 25]. In terms of visual representation, early studies utilized grid features extracted by Convolutional Neural Networks (CNN) for image embedding [9, 27]. Subsequently, region features, also known as object features, extracted by object detectors gained popularity for enhancing the granularity of visual embedding [10, 15, 18, 21]. However, due to the complexity and resource-intensive nature of the initial approaches and the advancements in Vision-Language Pre-training, the patch projection embedding schema [14, 25], first introduced by ViT, has become the prevalent solution for visual embedding in the multimodal domain. Despite these advancements, the structures mentioned above exhibit some disadvantages: they lack the ability to expand or update their prior memory freely and cannot directly provide insight into their current predictions.

2.2 Memory Augmentation

Inspired by advances in memory networks[28], models with memory mechanisms incorporate an external memory module accessed and manipulated by several trainable operations. Some methods store historical visual and semantic knowledge in memory, generating a global feature to enhance the attention model[4]. [5] further introduce a selective reading mechanism to retrieve past knowledge information. Unlike providing temporary variables to assist in caption decoding, [24] introduces a recall mechanism to use recalled words. The text-retrieval module is sentence-level and identical to solving the image-text matching task. [8] first introduce a word-level retrieval mechanism into the image captioning task, equipping a well-trained captioning model with a kNN classifier over a datastore of cached context representations and corresponding target tokens. Instead of utilizing only one kNN classifier, our memory mechanism considers information retrieved by multiple kNN classifiers, enabling the model to dynamically evaluate and utilize neighbor information conditioned on different target tokens.

3 Methodology

3.1 Background: Memory-Augmented Image Captioning

MA-IC[8] represents a captioning approach that approximates token distributions via interpolating a well-trained autoregressive captioning model's distribution and another distribution calculated using an external memory bank. Typically, a memory-augmented image captioning system consists of two phases: constructing a memory bank and making predictions dependent on it.

Memory Bank Construction. Given an image-sentence pair in the training set $(x,y) \in (\mathcal{X},\mathcal{Y})$, where x denotes input image features and y represents a ground-truth sentence. A well-trained image captioning decoder generates the t-th target word based on the context $(x,y_{<t})$. The memory bank comprises a set of key-value pairs obtained offline. Technically, each key is a semantic embedding of the image-text sample computed by a mapping function $f(\cdot)$, and the value is the corresponding ground truth word y_t. The memory bank $\mathcal{D} = (\mathcal{K},\mathcal{V})$ encompasses all key-value pairs constructed from the entire training examples.

$$\mathcal{D} = (\mathcal{K},\mathcal{V}) = \{(f(x,y_{<t}),y_t)|\forall y_t \in \mathcal{Y},(x,y) \in (\mathcal{X},\mathcal{Y})\} \tag{1}$$

Combined Inference. At the inference stage, the MA-IC (1) calculates the context embedding $f(x,y_{<t})$ utilizing the well-trained captioning model, (2) employs $f(x,y_{<t})$ as the query to retrieve k nearest neighbors $\mathcal{N} = (k_i,v_i)|i = 1,\cdots,k$ from the memory bank $\mathcal{D}$, and (3) aggregates the retrieved tokens to form the distribution $P_{MA}(y_t \mid x,y_{<t})$ as:

$$P_{MA}(y_t \mid x,y_{<t}) \propto \sum_{(k_i,v_i)\in\mathcal{D}} \mathbb{I}_{y_t=vi} \exp\left(\frac{-dis(k_i,f(x,y_{<t}))}{T}\right) \tag{2}$$

where T is the temperature to flatten the distribution, and $dis(\cdot,\cdot)$ denotes the l_2 distance.

The final probability is derived as the interpolation of the IC model's distribution $P_{IC}(y_t \mid x,y_{<t})$ and $P_{MA}(y_t \mid x,y_{<t})$:

$$P(y_t \mid x,y_{<t}) = \lambda P_{MA}(y_t \mid x,y_{<t}) + (1-\lambda)P_{IC}(y_t \mid x,y_{<t}) \tag{3}$$

where the fixed weight λ balances the two distributions.

3.2 Adaptive Memory-Augmented Image Captioning

The MA-IC method presents two main limitations. First, each query relies on the context information of k nearest neighbors, making it vulnerable to noise when there is an inadequate amount of relevant contexts in the memory bank, especially for large k values. In contrast, a small k could lead to overfitting problems in certain situations. Second, Equation 3 employs a fixed weight parameter λ to control the interpolation of the two distributions, which might not be optimal for all target tokens. To mitigate these concerns, we propose 1) adaptively leveraging information from varying numbers of neighbors, and 2) incorporating a learnable network to determine the weights for different target tokens adaptively.

Multiple kNN Classifiers. As shown in Figure 1, we employ information from multiple *kNN* classifiers based on the current retrieval results. More specifically, we consider a set of k-values whose upper bound is more minor than K, and $k = 0$ represents the distribution of the IC model. For simplicity, we choose the power of 2 as the choice of k in this paper.

$$\mathcal{S} = \{0\} \cup \{k \in \mathbb{N} \mid \log_2 k \in \mathbb{N}, k \leq K\} \tag{4}$$

where $\mathcal{S}$ is the space and k takes its value. Subsequently, we input the features, constructed using the results of each kNN classifier, into a lightweight network to determine the corresponding interpolation weight.

Importance Measurement. As depicted in Figure 1, we employ a lightweight network to estimate the importance of different distributions rather than relying on a fixed parameter. We design three features that consider both retrieved information and IC output.

Distance. We posit that the distance between q_t and each neighbor is the most direct evidence for evaluating importance. Neighbors closer to q are assigned greater weights, while those farther away receive smaller weights. We use the square of the l_2 distance $d_i = ||q_t, h_i||^2$to represent this metric. The feature has a dimension of k:

$$d = (d_1, \cdots, d_k) \tag{5}$$

Count. The distribution of target tokens in the retrieval results also impacts the prediction. If most retrieved results share the same token, the prediction is more reliable, and the model should rely more on the kNN probability. We count the number of unique values among the top i neighbors c_i:

$$c = (c_1, \cdots, c_k) \tag{6}$$

Output. Given that the final probability is an interpolation of two distributions, the IC output also plays a crucial role in the decision-making process. When the original captioning model exhibits low confidence in a target token, we should place more emphasis on the kNN decision to improve the prediction. Denote $w = w_1, \cdots, w_k$ as the target token of the retrieval result, and we use its probability in the IC output as the feature:

$$o = (o_1, \cdots, o_k) \tag{7}$$

We normalize the designed features to ensure they are on a similar scale and then concatenate them as the input features $[d : c : o]$ to a lightweight network, f_β, an FFN network. The normalized weights for each available k are computed as:

$$P_\beta(k) = \text{softmax}(f_\beta([d : c : o])) \tag{8}$$

Where $k \in \mathcal{S}$, and $k = 0$ corresponds to the distribution generated directly by the IC model.

3.3 Prediction

We eliminate the fixed hyper-parameter λ present in Equation 3, and the final prediction probability becomes a weighted ensemble of different kNN predictions combined with the output of the IC model:

$$P(y_t \mid x, y_{<t}) = \sum_{k \in \mathcal{S}} P_\beta(k) \cdot P_{k\text{NN}}(y_t \mid x, y_{<t}) \tag{9}$$

Where $P_{k\text{NN}}$ denotes the k nearest neighbor probability, as calculated in Equation 2. We then derive the final predicted word as follows:

$$w_t = \text{argmax}(P(y_t \mid x, y_{<t})) \tag{10}$$

3.4 Training Objectives

During training, only the parameters of the lightweight network need updating. We also follow a standard two-stage training strategy in image captioning: pre-training the model with cross-entropy loss (XE) and fine-tuning the model with reinforcement learning. First, we optimize the adaptive network by minimizing the cross-entropy loss between the mixed distribution and ground truth target tokens:

$$L_{XE}(\theta) = -\sum_{t=1}^{n} \log P(y_t \mid x, y_{<t})) \tag{11}$$

Where n is the length of the sentence. Subsequently, we perform self-critical reinforced training, optimizing the CIDEr score:

$$\nabla_\theta L(\theta) = -\frac{1}{n}\sum_{i=1}^{n} \left(\left(r\left(y^i\right) - b\right) \nabla_\theta \log P\left(y^i\right)\right) \tag{12}$$

Where y_i is the ith word in the sentence, $r(\cdot)$ is the reward function, and b is the average reward of the words to be selected in the captions.

4 Experiments

We conducted all experiments on the most popular image captioning dataset MSCOCO[17]. Following the standard evaluation protocol, we employ five standard automatic evaluation metrics: BLEU[20], METEOR [3], ROUGE[16], CIDEr[22], and SPICE[1].

4.1 Implementation Details

We equip Adaptive Memory Augmentation (AMA) method with various state-of-the-art models, including those utilizing the encoder-decoder paradigm (M2[6], RSTNet[30]) and the Vision-and-Language Pretraining (VLP) paradigm (LEMON[10], OFA[25]). The exact architecture and optimizations described in the corresponding papers are adopted. AMA is compared with two other methods: vanilla Memory Augment (MA)[8] and Uniform Memory Augment (UMA), where equal confidence is set for each k-NN prediction.

For implementation, the 512-dimensional representation input to the final layer feedforward network is used as the key. The training set is forward inferred with two trained models to create keys and values, and FAISS[12] is employed to represent the memory bank and search for nearest neighbors. For MA, the balancing parameter λ, temperature T, and nearest neighbor parameter k are carefully tuned, and the best scores for each model are reported. Specifically, $\lambda = 0.25$, $T = 100$, and $k = 64$ are set for one model, and $\lambda = 0.3$, $T = 100$, and $k = 64$ for the other. For UMA, λ and T are kept the same as in MA, and the values of k are varied using the set $2, 4, 8, 16, 32, 64$. The augmented memory is obtained as a uniform distribution of MA predictions for different values of k. In AMA, the hidden size of the FFN in a lightweight network is set to 64. The network is optimized using the validation set (5k examples), and the Adam optimizer[13] is employed for training. The learning rate is set to 5e-4, and the batch size is set to 32.

4.2 Quantitative Analysis

4.2.1 Offline Evaluation.

Table 1 summarizes the performance of state-of-the-art models, memory-utilized models, and baselines incorporated with the adaptive memory augmented (AMA) method on the offline COCO Karpathy test split. As illustrated in Table 1, four baselines employing the adaptive memory augmented approach achieve significant performance gains, demonstrating the advantages of adaptively determining and utilizing neighbor information for each target token. Remarkably, the proposed method surpasses previous memory augmentation methods for different models in both the encoder-decoder paradigm and the Vision-and-Language Pretraining (VLP) paradigm. A comparison of AMA with UMA, which assigns equal confidence to each kNN prediction, reveals that a simple aggregation does not yield better performance.

	Bleu-1	Bleu-4	Meteor	Rouge-L	CIDEr-D	Spice
State-of-the-art models						
AoANet[11]	80.2	38.9	29.2	58.8	129.8	22.4
HIP[29]	-	39.1	28.9	59.2	130.6	22.3
M2[6]	80.8	39.1	29.2	58.6	131.2	22.6
RSTNet[30]	81.1	39.3	29.4	58.8	133.3	23.0
LEMON$_{base}$[10]	82.1	40.3	30.2	59.8	133.3	23.3
OFA$_{base}$[25]	**82.5**	**41.0**	**30.9**	**60.2**	**138.2**	**24.2**
memory-utilized models						
ICMK[5]	81.9	38.4	28.7	58.7	125.5	-
Up-Down+SRT[24]	80.3	38.5	28.7	58.4	129.1	22.4
M2+MA[8]	80.9	39.3	29.3	58.7	132.0	22.7
RSTNet+MA[8]	81.2	39.7	29.5	59.0	134.0	23.1
LEMON$_{base}$+MA[8]	82.5	40.5	30.3	60.1	134.7	23.4
OFA$_{base}$+ MA[8]	**82.9**	**41.2**	**31.0**	**60.8**	**138.5**	**24.4**
Our adaptive memory-augmented models						
M2+UMA	80.7	39.2	29.2	58.5	131.7	22.5
RSTNet+UMA	81.0	39.6	29.4	58.8	133.4	23.0
LEMON$_{base}$ +UMA	82.4	40.4	30.2	59.9	133.5	23.3
OFA$_{base}$+ UMA	82.7	41.1	30.9	60.4	138.3	24.2
M2+AMA	81.1	39.8	29.5	58.8	133.4	22.9
RSTNet+AMA	81.6	40.3	29.6	59.3	135.2	23.3
LEMON$_{base}$ +AMA	82.9	40.6	30.4	60.3	136.3	23.5
OFA$_{base}$+ AMA	**83.1**	**41.3**	**31.1**	**61.1**	**138.8**	**24.5**

Table 1: Performance comparison with baseline methods.

4.2.2 Online Evaluation.

Adhering to the standard practice[11], we submit the generated captions for the official testing set to the online testing server and present the results in Table 2. Additionally, we report the method's performance on official test images with 5 reference titles (c5) and 40 reference titles (c40), as well as the top-ranking published works and other memory-utilizing methods on the leaderboard.

Methods	Bleu-1		Bleu-4		Meteor		Rouge-L		CIDEr-D	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
AoANet[11]	81.0	95.0	39.4	71.2	29.1	38.5	58.9	74.5	126.9	129.6
HIP[29]	81.6	95.9	39.3	71.0	28.8	38.1	59.0	74.1	127.9	130.2
M2[6]	81.6	96.0	39.7	72.8	29.4	39.0	59.2	74.8	129.3	132.1
RSTNet[30]	81.7	96.2	39.7	72.5	29.3	38.7	59.2	74.2	130.1	132.4
ICMK[5]	80.8	95.3	37.5	69.7	28.0	36.9	57.9	73.0	118.9	121.5
RSTNet+MA[8]	82.1	96.0	40.1	72.7	29.3	38.8	59.2	74.3	130.5	132.8
RSTNet+AMA	**82.3**	**96.2**	**40.5**	**73.2**	**29.4**	**38.9**	**59.4**	**74.5**	**131.5**	**134.0**

Table 2: Leaderboard of different methods on the online MS COCO test server.

4.2.3 Robustness Measurement.

The size and quality of the memory influence the performance of a model that incorporates a memory mechanism. We analyze the performance fluctuations of MA and AMA when utilizing a low-quality memory bank. Precisely, we assess the robustness of these models in the presence of noisy memory. We introduce token-level noise to the captions in the training data employing EDA[26]. Each word in a caption is altered with a 10% probability, with the same probability applying to the modification type. Subsequently, we use the noisy training data to create a noisy memory bank. We explore the impact of noise on captioning performance in Table 3. The BLEU and CIDEr scores of AMA show a less significant decrease, indicating that our approach exhibits increased robustness under low-quality memory conditions.

Method	Bleu-4	CIDEr-D
RSTNet+MA	39.7	134.0
RSTNet+MA(+10% noise)	39.5(-0.2%)	133.5(-0.5%)
RSTNet+AMA	40.3	135.2
RSTNet+AMA(+10% noise)	40.1(-0.1%)	134.9(-0.3%)

Table 3: Effect of noisy memory on different methods. AMA is more robust when the quality of database is not good.

4.2.4 Computational Cost

The experimental setup comprises an Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz and an NVIDIA GeForce RTX 3090 (24GB) GPU. We assess the generation time for the MS COCO test set across RSTNet, RSTNet+MA, and RSTNet+AMA under various K conditions, with GPU acceleration for FAISS disabled. The results are presented in Table 4. In our method, the results retrieved by the max K classifier can be reused in kNN classifiers with a smaller K. As a result, our design slightly increases the decoding time while achieving superior performance.

ms/token	K=8	K=16	K=32	K=64
RSTNet	0.446			
RSTNet+MA	3.23(×7.24)	4.43(×9.93)	6.00(×13.45)	10.01(×22.44)
RSTNet+AMA	3.26(×7.32)	4.63(×10.38)	6.31(×14.15)	10.57(×23.71)

Table 4: Generation time of different models.

4.3 Ablation Study

4.3.1 Effect of Designed Features.

We further estimate the effect of designed features. We conduct the ablation study with $K = 64$. It's evident that three features contribute significantly to the excellent performance of our approach, in which the distance feature is more important.

	Bleu-1	Bleu-4	Meteor	Rouge-L	CIDEr-D	Spice
RSTNet	81.1	39.3	29.4	58.8	133.3	23.0
+ AMA$_{Distance}$	81.5	40.1	29.6	59.2	134.8	23.2
+ AMA$_{Count}$	81.5	40.0	29.6	59.3	134.6	23.2
+ AMA$_{Output}$	81.4	39.9	29.5	59.2	134.5	23.1
+ AMA	81.6	40.3	29.6	59.3	135.2	23.3

Table 5: Effect of Three Designed Features.

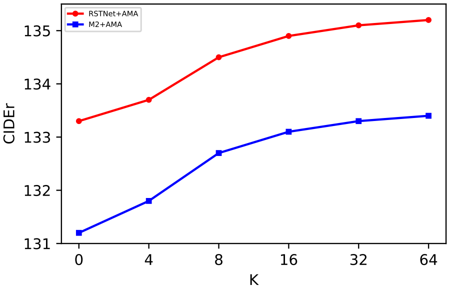

Figure 2: Effect of the number of *K*.

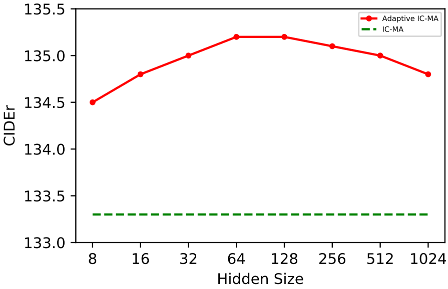

Figure 3: Effect of the hidden size.

4.3.2 Performance on low resource scenario.

We conducted experiments by varying the percentage of the training set while holding the entire training set as the memory for the kNN search. Table 6 shows that AMA can generate results comparable to MA with a memory bank constructed with only 40% of the training data.

4.3.3 Effectiveness of *K* and hidden size

We investigate the effect of the value *K* and hidden size. Figure 2 demonstrates the effectiveness of the adaptive memory mechanism. Besides, to make a trade-off between accuracy and speed, K = 16 is enough to achieve competitive performance. As shown in Figure 3. We can observe that the model reaches the highest metric value when the hidden size is 64. When the hiding decreases from 64 to 8 or increases from 128 to 1024, the CIDEr score decreases moderately due to the fitting problem of the network. Therefore, we set the hidden size 64 by default in this paper.

Rate	Memory Size	Bleu-1	Bleu-4	Meteor	Rouge-L	CIDEr-D	Spice
100%	6.5M	81.6	40.3	29.6	59.3	135.2	23.3
80%	5.2M	81.6	40.1	29.6	59.3	134.9	23.2
60%	3.9M	81.4	39.9	29.5	59.1	134.5	23.2
40%	2.6M	81.2	39.8	29.5	59.0	134.2	23.2
20%	1.3M	81.2	39.6	29.4	58.9	133.8	23.1
0%	0	81.1	39.3	29.4	58.8	133.3	23.0

Table 6: Effect of memory size.

5 Conclusion

In this paper, we propose an adaptive memory of feedback, which adaptively exploits the information from different numbers of neighbors by introducing a lightweight network. The network only requires thousands of parameters and can be easily trained with a validation dataset. Experiments conducted on the MS COCO benchmark prove that our adaptive memory incorporation mechanism can effectively filter noises and significantly outperforms the MA-IC approach. Additionally, studies show that our method is more robust with low-quality database.

6 Acknowledgement

The research work is supported by National Key R&D Program of China (No.2022YFB3904700), Industrial Internet Innovation and Development Project in 2021 (TC210A02M, TC210804D), Opening Project of Beijing Key Laboratory of Mobile Computing and Pervasive Device.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, pages 382–398. Springer, 2016.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[3] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[4] Hui Chen, Guiguang Ding, Zijia Lin, Yuchen Guo, and Jungong Han. Attend to knowledge: Memory-enhanced attention network for image captioning. In *International Conference on Brain Inspired Cognitive Systems*, pages 161–171. Springer, 2018.

[5] Hui Chen, Guiguang Ding, Zijia Lin, Yuchen Guo, Caifeng Shan, and Jungong Han. Image captioning with memorized knowledge. *Cognitive Computation*, 13(4):807–820, 2021.

[6] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587, 2020.

[7] Zhengcong Fei. Iterative back modification for faster image captioning. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3182–3190, 2020.

[8] Zhengcong Fei. Memory-augmented image captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence, Online*, pages 2–9, 2021.

[9] Jiuxiang Gu, Gang Wang, Jianfei Cai, and Tsuhan Chen. An empirical study of language cnn for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1222–1231, 2017.

[10] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pre-training for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17980–17989, 2022.

[11] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4634–4643, 2019.

[12] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[14] Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, Hehong Chen, Guohai Xu, Zheng Cao, et al. mplug: Effective and efficient vision-language learning by cross-modal skip-connections. *arXiv preprint arXiv:2205.12005*, 2022.

[15] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 121–137. Springer, 2020.

[16] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[18] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 375–383, 2017.

[19] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

[20] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[21] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024, 2017.

[22] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[23] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[24] Li Wang, Zechen Bai, Yonghua Zhang, and Hongtao Lu. Show, recall, and tell: image captioning with recall mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12176–12183, 2020.

[25] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022.

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

[27] Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton Van Den Hengel. What value do explicit high level concepts have in vision to language problems? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 203–212, 2016.

[28] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*, pages 2397–2406. PMLR, 2016.

[29] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Hierarchy parsing for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2621–2629, 2019.

[30] Xuying Zhang, Xiaoshuai Sun, Yunpeng Luo, Jiayi Ji, Yiyi Zhou, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Rstnet: Captioning with adaptive attention on visual and non-visual words. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15465–15474, 2021.

[31] Ying Zhang and Huchuan Lu. Deep cross-modal projection learning for image-text matching. In *Proceedings of the European conference on computer vision (ECCV)*, pages 686–701, 2018.

[32] Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. Adaptive nearest neighbor machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Online, August 2021. Association for Computational Linguistics.