 Retrieval-Augmented Transformer for Image Captioning

Sara Sarto
University of Modena and Reggio Emilia
Modena, Italy
237128@studenti.unimore.it

Lorenzo Baraldi
University of Modena and Reggio Emilia
Modena, Italy
lorenzo.baraldi@unimore.it

Marcella Cornia
University of Modena and Reggio Emilia
Modena, Italy
marcella.cornia@unimore.it

Rita Cucchiara
University of Modena and Reggio Emilia
Modena, Italy
rita.cucchiara@unimore.it

ABSTRACT
Image captioning models aim at connecting Vision and Language by providing natural language descriptions of input images. In the past few years, the task has been tackled by learning parametric models and proposing visual feature extraction advancements or by modeling better multi-modal connections. In this paper, we investigate the development of an image captioning approach with a kNN memory, with which knowledge can be retrieved from an external corpus to aid the generation process. Our architecture combines a knowledge retriever based on visual similarities, a differentiable encoder, and a kNN-augmented attention layer to predict tokens based on the past context and on text retrieved from the external memory. Experimental results, conducted on the COCO dataset, demonstrate that employing an explicit external memory can aid the generation process and increase caption quality. Our work opens up new avenues for improving image captioning models at larger scale.

CCS CONCEPTS
• Computing methodologies → Natural language generation;
• Visual content-based indexing and retrieval: Matching; Computer vision tasks.

KEYWORDS
image captioning, image retrieval, vision-and-language.

ACM Reference Format:

1 INTRODUCTION
Image captioning has recently emerged as an important task at the intersection of Computer Vision, Natural Language Processing, and Multimedia, thanks to the key role it can have to connect Vision and Language in multimedia systems [3, 8, 31]. Recent advances on image captioning, indeed, have demonstrated that fully-attentive architectures can provide high-quality image descriptions, even in few or zero-shot settings [1, 15, 36, 43]. Regardless of this progress, accurately describing the concepts contained in input image is still a considerable challenge. While some recent works have addressed this by increasing the size of the model [1], thus improving its memorization capabilities, this comes at the cost of increasing the number of learnable parameters and, ultimately, the training cost.

The same issue is currently being faced in large-scale language models, where the adoption of retrieval components is being explored as a viable solution [9, 58]. The basic idea behind these models is that of letting the language model attend textual chunks or hidden states retrieved from an external memory, instead of relying solely on its own activations. In this manner, the memorization requirements of the model are relieved in favor of the external memory, which can handle larger-scale data and can easily be accessed via approximate nearest neighbor searches.

In this paper, we investigate the development of a retrieval component for image captioning. We propose a fully-attentive architecture with a knowledge retriever based on approximate kNN searches, which can provide the language model with suitable hints from an external memory. Our language model then employs a kNN-augmented attention layer to predict tokens based on the past context and on text retrieved from the external memory. To the best of our knowledge, our proposal is the first model to integrate a retrieval-based memory into an image captioning pipeline.

We conduct experiments on the COCO dataset for image captioning, in comparison with a fully-attentive baseline that does not employ an external memory and with an adaptation of the RETRO architecture [9], which has been devised for large-scale language models. Our experiments show that using an external memory can significantly improve the generation quality and that adding a retrieval component to multi-modal models can be a viable solution. Further, we show that our proposed architecture overcomes the RETRO design [9] by a significant margin.

2 RELATED WORK
Image Captioning. Image captioning is a broad topic that has witnessed research on visual information extraction, text generation, and semantics incorporation. Over the years, different approaches have been proposed. Early works relied on CNN-based encoders and RNN-based language models [18, 34, 48, 57]. Nowadays, attentive and Transformer-based architectures [55] are often employed both in the visual encoding stage [14, 40], either applied to image patches directly [19, 54] or to refine features from a visual backbone, and as language models [11, 13, 25]. Regarding language models, in the last few years, large performance improvements have come from
increasing the amount of training data, model size, and performing large-scale training [10, 46].

The introduction of Transformer-based models in image captioning has also brought to the development of effective variants of the self-attention operator [16, 22, 25, 27, 39, 43] and to that of vision-and-language early-fusion approaches [26, 36, 65] based on BERT-like architectures [17]. On the image encoding side, a recent paradigm is that of employing visual features extracted from large-scale multi-modal architectures [6, 7, 15, 50] like CLIP [45].

Convolutional [4] and fully-attentive language models [41, 60, 64] based on the Transformer paradigm have been used due to the limited representation power and sequential nature of RNN-based language models and thanks to their success in NLP tasks such as machine translation and language understandings [17, 52, 55]. In this paper, we follow the dominant track of employing a Transformer-based language model and propose a fully-attentive architecture augmented with retrieval abilities.

**Retrieval-Augmented Approaches.** Image retrieval has evolved over time from methods based on local descriptors to convolutional encoders until the use of visual Transformers [20, 21]. Large-scale language models can perfectly memorize parts of their training data [12] and increasing model size predictably improves performance on a wide range of downstream tasks [10, 30, 46]. This suggests that enhancing models with retrieval may lead to further improvements and savings in terms of model size. In this work, we take inspiration from this line of research and investigate the incorporation of retrieval in image captioning. We conduct kNN searches [29] on the extracted visual features to integrate the corresponding k-most similar retrieved captions with the rest of our architecture. In this way, the model can access the entire training dataset through the retrieval mechanism and is also, in principle, not limited to the data seen during training [9]. This idea allows us to go conceptually beyond the traditional Transformer language model in which the benefits of model size and data size are linked.

Retrieval-augmented language models have been recently gaining a lot of attention [23, 28, 32, 35]. Most works tackling this direction have devised novel forms of attention to model the connections with the retrieved chunks of text [58, 62]. Borgeaud et al. [9] split input sequences into chunks, which are augmented with the k-nearest neighbors using a chunked cross-attention module to incorporate the retrieved text. Wu et al. [58] proposed a gated attention module to attend the internal states of a Transformer, seen during past training iterations. The usage of a learned attention gate is closely related to our formulation, even though in our case retrieval is applied towards an external memory rather than on internal activations, and we employ a single scalar gate instead of a learned per-head parameter.

### 3 PROPOSED METHOD

The goal of an image captioner is that of modeling a distribution \( p(y|I) \) over possible captions \( y \) given an input image \( I \). During the pre-training stage, the captioner is trained with a time-wise distribution over the vocabulary while being conditioned on both the input image and the set of retrieved captions.

Having defined a knowledge retrieval strategy that provides captions from similar images of an external memory, we devise a retrieval-augmented language model which can predict a time-wise distribution over the vocabulary while being conditioned on both the input image and the set of retrieved captions.

The bone structure of the model is a vanilla encoder-decoder Transformer [55] in which the encoder is employed to process the entire caption \( y \) without relying on previous ground-truth words. The generated caption is then usually matched with ground-truth captions to obtain a reward signal [48].

Our approach decomposes the probability distribution \( p(y|I) \) into two steps: retrieval and prediction. Firstly, given an image \( I \) we retrieve possibly related descriptions \( \{z_i\}_i \) from an external memory, thanks to a visual similarity space in which k-NN searches can be carried out. Then, we condition our language model on both the input image \( I \) and the set of retrieved descriptions \( \{z_i\}_i \), thus effectively modeling \( p(y|\{z_i\}_i, I) \) and marginalizing over the set of retrieved captions.

#### 3.1 Knowledge Retriever

Given a corpus of image-text pairs and an input query image \( I \), we model \( p(z|I) \) by building a knowledge retrieval component. This performs an approximate k-nearest-neighbor search into the external memory, defined through an inner product similarity between image embeddings, i.e.:

\[
f(I_1, I_2) = \text{Embed}(I_1)^T \text{Embed}(I_2),
\]

where \( \text{Embed}(\cdot) \) is a function that maps an image to a vector. The relevance \( f(\cdot, \cdot) \) between the query image and images in the corpus is employed to sort images by decreasing similarity. Then, the knowledge retriever returns all captions associated with the selected images, as a source of conditioning for the language model.

To model the visual embedding function, we employ the visual encoder of one of the CLIP models [45], which have been trained contrastively to match image-text pairs. Empirically, we found this relevance function to be more robust in our scenario when compared to vision-only descriptors, as also reported in recent literature [6, 50]. While the maximum inner product search is carried out employing visual queries and textual keys, however, our strategy is computationally lighter as it does not require to forward through a textual encoder.

Specifically, we select a CLIP ResNet-based [24] visual encoder. In this kind of encoder, the image is processed through a sequence of residual layers, then the grid of activations from the last convolutional layer is fed to an attention pool layer. Here, a single query is built from the global average-pooled feature vector, and all elements of the grid act as keys and values. To get a more fine-grained representation of the image and have a higher control on the pooling strategy, we directly take the grid of features from the last convolutional layer and define the Embed function as an aggregation (e.g., average, max) of the features contained in the grid (see Fig.1).

#### 3.2 Retrieval-Augmented Language Model

Having defined a knowledge retrieval strategy that provides captions from similar images of an external memory, we devise a retrieval-augmented language model which can predict a time-wise distribution over the vocabulary while being conditioned on both the input image and the set of retrieved captions.

The bone structure of the model is a vanilla encoder-decoder Transformer [55] in which the encoder is employed to process the
input image and the decoder acts as a language model. The input of the encoder consists of a flattened sequence of grid feature vectors (as described in Sec. 3.1) which are linearly projected into a vector space. The resulting vectors are then passed through a sequence of encoder layers, each of which does dense self-attention, followed by a feed-forward network (FFN). In the encoder, attention operations are not masked, thus allowing a bidirectional encoding of input image features with complete connectivity. The input text is instead tokenized and embedded into a vector space. The embedding vectors are then passed through a sequence of decoder layers, each of which performs a kNN-augmented attention, a cross-attention with the last layer of the encoder, and a feed-forward network. In the decoder, we use a causal attention mask and the token embeddings of the last layer are used to predict the next token.

The kNN-augmented attention layer combines two types of attention: like a normal self-attention layer, it attends the input subsequence encoding the past context. Plus, it performs attention over the set of retrieved captions, thus connecting to the external memory. This is modeled by first encoding all retrieved captions independently through a Transformer encoder, and then performing cross-attention over its outputs. Crucially, the same queries are employed for the self-attention over the input subsequence and for the cross-attention over the encoded retrieved captions.

Given the input sequence of tokens \( \{ w_i \} \), \( i = \{ w_0, ..., w_t, ..., w_T \} \) and the set of retrieved captions \( \{ z_i \} \), \( i = \{ z_0, ..., z_t, ..., z_N \} \), the kNN-augmented attention can be written as follows:

\[
\tilde{z}_k = \text{MSA}(z_k, z_k)
\]

\[
\tilde{w}_t = \text{MSA}(w_t, \{ w_i \}_{i \leq t})
\]

\[
\tilde{w}_t^m = \text{MCA}(w_t, \{ \tilde{z}_k \}, k)
\]

where \( k \) indicates a generic item from the set of retrieved captions, \( t \) the \( t \)-th element of the sequence of tokens, \( \{ w_i \}_{i \leq t} \) is the sequence of tokens up to the \( t \)-th element, \( \text{MSA}(x, y) \) indicates a multi-head self-attention with \( x \) mapped to query and \( y \) mapped to key-values, and \( \text{MCA}(x, y) \) a multi-head cross-attention with \( x \) as query and \( y \) as key-values. The first equation refers to a self-attention between tokens of each retrieved caption, and we drop the dependency to single tokens for readability. The last equation, instead, refers to the cross-attention operation with retrieved captions. Here, given \( w_t \) as query, all tokens from all retrieved captions are attended.

The outputs of the self-attention over the input subsequence and that of the cross-attention over the external memory are combined using a learned gate, which allows the model to choose between local context and retrieved captions. Formally,

\[
\tilde{w} = \alpha \cdot \tilde{w}^t + (1 - \alpha)\tilde{w}^m,
\]

where gate \( \alpha \) is learned as the sigmoid of a single scalar parameter.

As it might be noticed, gradients are not backpropagated into the external memory, which is critical to the scalability of our technique. Following a standard practice in image captioning, we first pre-train our language model with a time-wise cross-entropy loss. Then, we fine-tune using the self-critical sequence learning paradigm (SCST), which employs reinforcement learning over sampled sequences. Specifically, we employ the variant proposed in [16] that employs beam search for sampling, and sets the baseline reward equal to the mean of rewards of generated captions inside a beam. We refer the reader to [48] for a comprehensive treatment of the RL-based fine-tuning stage.

4 EXPERIMENTAL EVALUATION

4.1 Setup

Dataset. Following standard image captioning approaches [3, 16, 27], we train and evaluate our model on the COCO dataset [38], thus not relying on large-scale image-text datasets [15]. COCO is composed of more than 120,000 images, each of them associated with 5 human-collected captions. We follow the splits defined by Karpathy et al. [31], using 5,000 images for both validation and testing and the rest for training.

Metrics. According to the standard evaluation protocol, we employ the complete set of captioning metrics: BLEU [44], METEOR [5], ROUGE [37], CIDEr [56], and SPICE [2].

Retrieval Index. We build our retrieval index on the COCO training set. During training, to reduce overfitting risks, we avoid retrieving captions that belong to the current training image. We employ
approximate kNN search rather than exact kNN search because it significantly improves the computational speed of our model. To this aim, we employ the Faiiss library [29] and a graph-based HNSW index with 32 links per vertex, which has a size of 6.7 GB. For simplicity, we do not employ any vector transform (e.g. PCA) or vector quantization, although they might be employed to reduce the index size and scale to larger datasets.

### Implementation Details

To represent images, we employ CLIP-RN50x16 [45] intermediate features. To represent words, of both the input subsequence and retrieved sentences, we use Byte Pair Encoding (BPE) [49] with a vocabulary size of 49, 408. We use standard sinusoidal positional encodings [55] to represent word positions. For efficiency, the length of the output token sequence is limited to 40 tokens. Visual features and word tokens are projected into \( d \times 5 \) intermediate features. To represent words, of both intermediate features and retrieved sentences, we use Byte Pair Encoding (BPE) [49] with a vocabulary size of 49, 408. We use standard sinusoidal positional encodings [55] to represent word positions. For efficiency, the length of the output token sequence is limited to 40 tokens. Visual features and word tokens are projected into \( d \)-dimensional vectors with \( d = 384 \) and fed to our Transformer-based model, which has \( L = 3 \) layers in both encoder and decoder with six attention heads. The external memory encoder has the same number of heads and dimensionality as the rest of the model. The gate \( \alpha \) is initialized to zero at the beginning of the training.

Pre-training with cross-entropy loss is performed using the LAMB optimizer [63] and following the learning rate scheduling strategy of [55] with a warmup equal to 6,000 iterations and a batch size of 1,080. For the CIDEr-based fine-tuning, we adopt the SCST strategy of [55] with a warmup equal to 6,000 iterations and a batch size of 1,080. For the CIDEr-based fine-tuning, we adopt the SCST strategy of [55] with a warmup equal to 6,000 iterations and a batch size of 1,080.

All experiments are performed by paralyzing training on two Quadro RTX-5000 GPUs, using five gradient accumulation steps during both cross-entropy pre-training and CIDER optimization. ZeRo memory offloading [47] and mixed-precision [42] are used to accelerate training and save memory.

### 4.2 Quality of Nearest Neighbor Captions

To prove the appropriateness of using nearest neighbor captions, we first investigate their quality with respect to ground-truth captions of a given test image. Specifically, given a sample image from the test set, we retrieve the \( k \) nearest captions from the training set according to our relevance function. We then compare the retrieved set with ground-truth captions by computing their mean scores as well as the scores obtained by the retrieved caption with maximum similarity with the ground-truth.

Results are reported in Table 1. As it can be seen, retrieving a relatively limited number of captions (e.g. \( k = 5 \)) produces a set of captions that have a significant, although low, overlap with the ground-truth. Increasing the number of retrieved captions degrades mean scores. The maximum (oracle) score, instead, reaches significantly high levels, up to 130.4 CIDEr points when retrieving \( k = 40 \) captions. The quality reached by the oracle caption for

### Table 1: Performance of the \( k \) nearest-neighbor captions.

<table>
<thead>
<tr>
<th>( k )</th>
<th>( k = 5 )</th>
<th>( k = 10 )</th>
<th>( k = 20 )</th>
<th>( k = 40 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B-1</td>
<td>B-4</td>
<td>M</td>
<td>R</td>
</tr>
<tr>
<td>mean</td>
<td>49.4</td>
<td>10.6</td>
<td>1.70</td>
<td>36.1</td>
</tr>
<tr>
<td>max</td>
<td>65.5</td>
<td>14.4</td>
<td>24.8</td>
<td>49.4</td>
</tr>
</tbody>
</table>

### Table 2: Performance of a base Transformer captioner and our retrieval-augmented Transformer, by varying the aggregation function, the number \( k \) of retrieved sentences, and the number of layers in the external memory encoder. Results are reported after cross-entropy pre-training.

<table>
<thead>
<tr>
<th>Aggregation Function</th>
<th>Layers</th>
<th>( k )</th>
<th>( k = 5 )</th>
<th>( k = 10 )</th>
<th>( k = 20 )</th>
<th>( k = 40 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \ell_1 )-norm sum</td>
<td>-</td>
<td>-</td>
<td>78.1</td>
<td>38.1</td>
<td>28.5</td>
<td>58.0</td>
</tr>
<tr>
<td>( \ell_1 )-norm sum</td>
<td>1</td>
<td>5</td>
<td>78.3</td>
<td>38.6</td>
<td>28.9</td>
<td>58.3</td>
</tr>
<tr>
<td>( \ell_1 )-norm sum</td>
<td>1</td>
<td>10</td>
<td>78.5</td>
<td>38.6</td>
<td>28.8</td>
<td>58.3</td>
</tr>
<tr>
<td>( \ell_1 )-norm sum</td>
<td>1</td>
<td>20</td>
<td>78.3</td>
<td>38.6</td>
<td>28.9</td>
<td>58.3</td>
</tr>
<tr>
<td>( \ell_1 )-norm sum</td>
<td>1</td>
<td>40</td>
<td>78.2</td>
<td>39.1</td>
<td>28.7</td>
<td>57.9</td>
</tr>
<tr>
<td>max</td>
<td>-</td>
<td>-</td>
<td>78.6</td>
<td>38.7</td>
<td>29.1</td>
<td>58.5</td>
</tr>
<tr>
<td>max</td>
<td>1</td>
<td>5</td>
<td>78.4</td>
<td>38.8</td>
<td>29.2</td>
<td>58.3</td>
</tr>
<tr>
<td>max</td>
<td>1</td>
<td>10</td>
<td>78.3</td>
<td>38.5</td>
<td>28.9</td>
<td>58.2</td>
</tr>
<tr>
<td>max</td>
<td>1</td>
<td>20</td>
<td>78.3</td>
<td>38.6</td>
<td>29.0</td>
<td>58.3</td>
</tr>
<tr>
<td>max</td>
<td>1</td>
<td>40</td>
<td>78.3</td>
<td>38.3</td>
<td>28.9</td>
<td>58.3</td>
</tr>
<tr>
<td>mean</td>
<td>-</td>
<td>-</td>
<td>78.8</td>
<td>38.9</td>
<td>29.1</td>
<td>58.5</td>
</tr>
<tr>
<td>mean</td>
<td>1</td>
<td>5</td>
<td>78.9</td>
<td>38.9</td>
<td>28.9</td>
<td>58.5</td>
</tr>
<tr>
<td>mean</td>
<td>1</td>
<td>10</td>
<td>78.3</td>
<td>38.5</td>
<td>28.9</td>
<td>58.3</td>
</tr>
<tr>
<td>mean</td>
<td>1</td>
<td>20</td>
<td>78.3</td>
<td>38.6</td>
<td>28.9</td>
<td>58.3</td>
</tr>
<tr>
<td>mean</td>
<td>1</td>
<td>40</td>
<td>78.3</td>
<td>38.3</td>
<td>28.9</td>
<td>58.3</td>
</tr>
</tbody>
</table>

### 4.3 Model Ablation and Analysis

#### Role of Different Aggregation Functions

We then move to our full model, and first analyze the results of different aggregation functions to embed visual features and retrieve the most similar images. Specifically, we consider a standard average pooling over grid features, a max pooling, and a sum of \( \ell_2 \)-normalized features followed by an \( \ell_2 \)-norm of the result, which has demonstrated to be effective in previous image and video retrieval works [53]. Results are reported in Table 2, after cross-entropy pre-training, in comparison with a standard Transformer-based encoder-decoder model without retrieval. We can first notice that all configurations with the external memory encoder achieve better performance than the baseline which obtains 121.6 CIDEr points, thus demonstrating the effectiveness of our retrieval-augmented architecture. When
Table 3: Ablation study results, in comparison with RETRO.

<table>
<thead>
<tr>
<th></th>
<th>B-1</th>
<th>B-4</th>
<th>M</th>
<th>R</th>
<th>C</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours w/ RETRO block (C = 2) [9]</td>
<td>75.2</td>
<td>34.5</td>
<td>26.2</td>
<td>55.7</td>
<td>108.6</td>
<td>20.2</td>
</tr>
<tr>
<td>Ours w/ RETRO block (C = 6) [9]</td>
<td>74.8</td>
<td>33.9</td>
<td>26.3</td>
<td>55.8</td>
<td>106.2</td>
<td>19.9</td>
</tr>
<tr>
<td>Transformer (w/o external memory)</td>
<td>78.1</td>
<td>38.1</td>
<td>28.5</td>
<td>58.0</td>
<td>121.6</td>
<td>21.8</td>
</tr>
<tr>
<td>RA-Transformer (w/o gate)</td>
<td>78.3</td>
<td>38.3</td>
<td>28.9</td>
<td>58.1</td>
<td>122.5</td>
<td>21.9</td>
</tr>
<tr>
<td>RA-Transformer</td>
<td>78.9</td>
<td>38.9</td>
<td>28.9</td>
<td>58.5</td>
<td>124.5</td>
<td>22.1</td>
</tr>
</tbody>
</table>

Comparing the different aggregation functions, the results show that a standard mean of grid features performs generally better than the other considered aggregation functions, also according to a different number \( k \) of retrieved captions.

**Number of Layers and Retrieved Captions.** We also evaluate the effect of changing the number \( k \) of retrieved sentences and the number of layers in the external memory encoder. In particular, we evaluate the captioning results employing \( k = 5, 10, 20, 40 \) retrieved elements. From Table 2, it can be seen that \( k = 10 \) and \( k = 20 \) generally lead to the best results according to almost all evaluation metrics. Additionally, we compare the results using a different number of Transformer layers (i.e., 1, 2, and 3) in the external memory encoder. The architecture with a single encoding layer obtains better performance than those with an increased number of parameters. Overall, the best performance is obtained by the model with the mean as aggregation function, \( k = 10 \) retrieved captions, and a single Transformer layer in the external memory encoder, with a CIDEr score equal to 124.5 points. This configuration is used in all experiments reported in the rest of the paper.

**Retrieval-Enhanced Transformer (RETRO) Baseline.** To evaluate the effectiveness of our retrieval strategy, we devise a variant of our model in which we replace the cross-attention between input tokens and retrieved captions with the chunked cross-attention mechanism proposed in [9]. We report the results in Table 3 by varying the chunk size \( C \) (i.e., \( C = 2 \) and \( C = 6 \)). Also in this case, the results are obtained after pre-training with cross-entropy loss. When comparing the results by using different chunk sizes, it can be noticed that increasing the chunk size leads to decreasing the final results. Overall, the performance of our retrieval-enhanced Transformer with multi-head cross-attention mechanism (i.e., RA-Transformer in the Table) is consistently better than that obtained by the version with chunked cross-attention.

**Role of External Memory and Learned Gate.** We finally evaluate the effectiveness of both the external memory encoder and learned gate to modulate the contribution of retrieved captions. To do this, as shown in Table 3, we design two different baselines. In the first model, we remove the external memory encoder from our architecture thus obtaining a vanilla Transformer-based encoder-decoder model. In the second baseline, instead, we only remove the learned gating mechanism. In this case, masked self-attention and cross-attention between input tokens and retrieved captions are performed in sequence, and the result is fed to the cross-attention with visual features.

As it can be seen, even just by introducing the external memory encoder in the architecture we can obtain enhanced results, with an improvement of 0.9 CIDEr points (i.e., 121.6 vs 122.5). The final results are further improved by the introduction of the learned gate \( \alpha \), with an improvement of 2 CIDEr points with respect to the architecture without gate and 2.9 CIDEr points compared to the vanilla Transformer model without external memory.

**Role of Approximated kNN Search.** We also test when employing exact kNN search, in place of the HNSW index. Removing the approximation in the retrieval phase brings to an increase of 0.3 CIDEr points on the COCO test set, and no significant improvement on all other metrics, thus confirming that approximate searches provide a convenient efficacy-efficiency balance when employing external memories.

**Training and Inference Time Analysis.** Training with cross-entropy takes around 24 hours for the model without external memory and around 30 hours for our complete model, while fine-tuning with reinforcement learning employs four and five days for a standard encoder-decoder model and for our retrieval-augmented Transformer, respectively. In fact, with respect to a basic encoder-decoder captioner, our model requires running approximate kNN searches and introduces additional attention layers and operations. At decoding time, this entails a 26% increase in execution times (from 0.170 seconds to 0.215 seconds to decode a single caption)\(^1\).

### 4.4 Comparison to the State of the Art

We compare the results of our model with those of several recent image captioning models trained exclusively on the COCO dataset. In our analysis, we include methods with LSTM-based language models and attention over image regions such as Up-Down [3], GCN-LSTM [61], SGAE [59], and MT [51], eventually enhanced with self-attention mechanisms such as AoANet [27], X-LAN [43], and DPA [39], and captioning architectures entirely based on fully-attentive mechanisms such as ORT [25], \( M^2 \) Transformer [16], X-Transformer [43], DLCT [41], and RSTNet [64].

Table 4 shows the results on the Karpathy-test split after fine-tuning with CIDEr optimization. We report the results of our complete retrieval-augmented Transformer and of the model trained without retrieval. As it can be seen, the effectiveness of the kNN-augmented

---

\(^1\) Execution times have been measured on a machine with Intel(R) Xeon(R) W-2235 CPU and Quadro RTX 5000 GPU, with a mini-batch composed of a single image and running kNN searches on CPU.
attention layer is confirmed also when training with reinforcement learning, with an improvement of 1.2 CIDEr points compared to the standard Transformer-based version. Additionally, we can notice that our model obtains competitive performance compared to other state-of-the-art approaches, surpassing them according to all evaluation metrics.

4.5 Qualitative Results

Finally, in Fig. 2 we show some qualitative results generated by our model and those generated by a Transformer-based model without external memory. For each image, we also report sample captions retrieved from the external memory. As it can be seen, the majority of retrieved captions adequately match the visual content of input images and can provide a helpful external source during the generation process to improve the final results. For example, in the top-right example, we can notice that the predicted caption has very similar content to that of the retrieved sentences (“i.e. “a group of elderly people”), while the model without external memory fails to generate a detailed description. Similarly, in the bottom-left example, the generated caption contains several words that also appear in the retrieved textual items (e.g. “a man jumping” and “a tennis ball”). This further demonstrates the effectiveness of our approach from a qualitative point of view.

In the bottom-right, we instead show an example in which the knowledge retriever partially fails to return textual sentences that effectively describe the visual content of the input image. In fact, while the image contains a painting with a vase, oranges, and other objects, some of the retrieved sentences refer to a painting with a locomotive. This confirms that additional efforts can be done to improve the quality of the retrieval embedding space and that, at the same time, the model maintains the ability to rely on input visual features when the quality of the retrieved captions is poor.

5 CONCLUSION

In this paper, we have presented a retrieval-augmented Transformer for image captioning that integrates kNN-augmented attention layers to generate word tokens based on textual sentences retrieved from an external memory. This enables the model to access an external corpus of textual items during the generation process thus improving the quality of generated captions. Experimental results conducted on the COCO dataset demonstrate the effectiveness of equipping a captioning architecture with retrieval abilities, opening up further research in this direction.

ACKNOWLEDGMENTS

We thank CINECA, the Italian Supercomputing Center, for providing computational resources. This work has been supported by “Fondazione di Modena” and by the PRIN project “CREATIVE: CRoss-modal understanding and gEnerATIon of Visual and tExtual content” (CUP B87G2200460001), co-funded by the Italian Ministry of University and Research.

REFERENCES


Karima Badreddine, C Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based Image Description Evaluation. In CVPR.

Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In NeurIPS.

Karima Badreddine, C Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based Image Description Evaluation. In CVPR.


