Violet: A Vision-Language Model for Arabic Image Captioning with Gemini Decoder

Abdelrahman Mohamed[ξ] **Fakhraddin Alwajih**[λ] **El Moatez Billah Nagoudi**[λ]
Alcides Alcoba Inciarte[λ] **Muhammad Abdul-Mageed**[λ,ξ]
[λ] Deep Learning & Natural Language Processing Group, The University of British Columbia
[ξ]Department of Natural Language Processing & Department of Machine Learning, MBZUAI
{fakhr.alwajih,moatez.nagoudi,muhammad.mageed}@ubc.ca

Abstract

Although image captioning has a vast array of applications, it has not reached its full potential in languages other than English. Arabic, for instance, although the native language of more than 400 million people, remains largely underrepresented in this area. This is due to the lack of labeled data and powerful Arabic generative models. We alleviate this issue by presenting a novel vision-language model dedicated to Arabic, dubbed *Violet*. Our model is based on a vision encoder and a Gemini text decoder that maintains generation fluency while allowing fusion between the vision and language components. To train our model, we introduce a new method for automatically acquiring data from available English datasets. We also manually prepare a new dataset for evaluation. *Violet* performs sizeably better than our baselines on all of our evaluation datasets. For example, it reaches a CIDEr score of 61.2 on our manually annotated dataset and achieves an improvement of 13 points on Flickr8k.

1 Introduction

Captioning images involves describing the visual elements of a picture using natural language. This requires a system that combines the strengths of two models: one that can represent the visual elements of an image, and another that can translate this representation into natural language. The latter employs a language model to produce *fluent* (i.e., grammatically accurate) and *adequate* (i.e., capturing sufficient semantic information) descriptions. In recent years, research on vision language models (VLMs) and their applications has boomed (Alayrac et al., 2022; Wang et al., 2022; Huang et al., 2023). Owing to the rapid advancements in large language models (LLMs), the performance of VLMs has improved dramatically. More concretely, VLMs have progressed from merely providing descriptions that vaguely resemble a given image (Vinyals et al., 2015) to accurately describ-

Figure 1. Examples of captions generated by our model.

ing complex visual cues within the image. The *pretraining-then-finetuning* paradigm also plays a significant role in achieving such impressive results, as it allows models to first grasp general language structures and then specialize in the specific task of image captioning (Gan et al., 2022).

Progress in VLMs, however, has been witnessed thus far primarily on English Awais et al. (2023). This leaves behind a large number of other languages for which no sufficient image captioning data or language models exist. Arabic is a case in point where image captioning lags far behind (Elbedwehy and Medhat, 2023). Similar to other low-resource languages, progress in Arabic image captioning has been hampered by the lack of publicly available datasets and limited efforts in creating any such data. Manual creation of image datasets, after all, requires a huge amount of time and labor. Again, the unavailability of powerful Arabic language models that understands the struc-

ture of the language and can capture its rich morphology has also caused a delay in the development of VLMs. Given the rapid progress in vision language technologies and their wide applications in society, limited progress in this area can have negative consequences for the Arabic-speaking world.
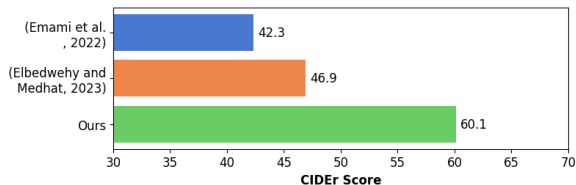

Figure 2. Performance of our model compared to previous works on Flickr8k using CIDEr metric.

To address this important issue, we introduce a novel Arabic image captioning model dubbed *Violet*. Our new model is comprised of two main components: a vision encoder and a text decoder. For the vision encoder, we employ an object detector network based on FasterRCNN (Ren et al., 2015) to extract visual features that are then passed to a compact transformer encoder. At the decoder side, we leverage the recently developed generative pretrained model JASMINE (Nagoudi et al., 2022). Taking inspiration from (Yu et al., 2022), we split our text decoder into two halves: the first half functions as a text decoder, whereas the second incorporates cross-attention layers, effectively serving as a fusion decoder. Given the dual nature of our decoder, we refer to it as *Gemini*. Drawing parallels with VisualGPT (Chen et al., 2022) and the meshed transformer (Cornia et al., 2020), we also adopt a meshed connection between the transformer vision encoder and the text decoder to foster enhanced communication between the encoder and decoder layers.

The other major challenge we face in our work is the unavailability of native Arabic captioning data. We alleviate this challenge by introducing a method for automatically acquiring captions that is based on first employing a powerful machine translation model followed by a quality assurance mechanism for removing poor captions. For evaluation, in addition to reporting on Arabic translated dataset, we task five human annotators to manually caption an image dataset. Compared to previous works and baselines, our novel model excels in captioning images in fluent Arabic. Figure 1 offers four examples of fluent Arabic captions generated by our novel model. Figure 2 shows a comparison of our model performance with prior research on

Flickr8k in CIDEr score.

In summary, our contributions are as follows:

- We present a novel image captioning model that employs an effective pretrained Arabic decoder capable of outputting rich captions.

- Our model achieves competitive performance for Arabic image captioning on both the MSCOCO (Lin et al., 2014) and Flicker8k (Jia et al., 2014) datasets, establishing a new state-of-the-art in this area.

- In the process of developing our new model, we release a translated version of MSCOCO dataset that has gone through our quality assurance pipeline. Our released dataset can help further advance research in Arabic VLMs.

- We also release our manually captioned dataset, a subset of MSCOCO test set, that we dub *AraCOCO*.

2 Related Work

Image captioning. Early methods for image captioning involve either retrieving descriptions (Karpathy et al., 2014) or using template filling combined with manually designed natural language generation techniques (Yang et al., 2011; Li et al., 2011). However, modern image captioning primarily relies on deep learning models. In early work, image captioning is framed as an image-to-sequence task using encoder-decoder models, with Convolutional Neural Networks (CNNs) as encoders and Recurrent Neural Networks (RNNs) as decoders while incorporating attention mechanisms (Xu et al., 2015; You et al., 2016; Huang et al., 2019). Soon after, using a transformer architecture of a vision encoder with a text decoder became the defacto direction towards solving the problem of image captioning (Stefanini et al., 2022). Some approaches use a detection model to extract visual features and then pass it to a transformer text decoder as in Oscar (Li et al., 2020; Chen et al., 2022), while others like CoCa (Yu et al., 2022) train a transformer vision encoder with a text decoder from scratch on a large-scale dataset.

More recently, there has been a shift towards using pre-trained LLMs and vision models. Generative Image-to-text Transformer (GIT) (Wang et al., 2022) is a decoder-only transformer that utilizes a CLIP (Radford et al., 2021) visual encoder to incorporate both visual and textual inputs. Another method to consider is VisualGPT (Chen et al.,
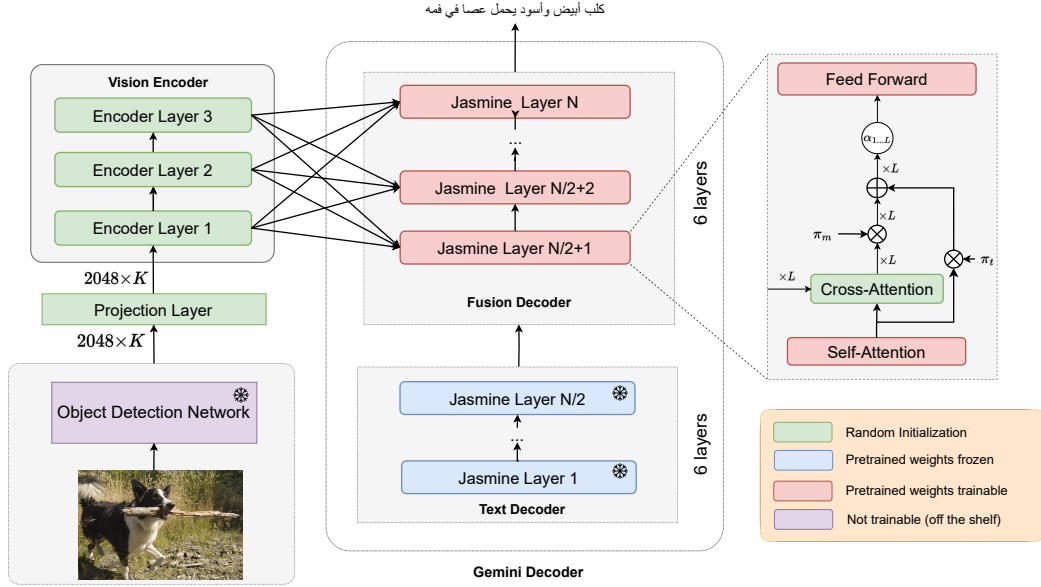
كلب أبيض وأسود يحمل عصا في فمه

Figure 3. The architecture and output generated by our model. We use an object detection network to extract K object features (K equal 50 in our case) from an image. After projecting to a lower dimension, the features are fed to an L-layer (three-layer in our architecture) transformer encoder. Meshed connection is employed between the encoder and decoder layers, where each encoder layer contributes to the cross-attention output. Our text decoder is split into two halves, the first half is the standard frozen pretrained text decoder layers, while the second half has cross-attention layers inserted after each self-attention layer. We call this design a *Gemini decoder*. We employ a gating mechanism through π_t and π_m that controls the flow of information from the vision and language sides. The final input to the feed forward network in each cross-attention layer is the weighted sum of each encoder-decoder attention controlled by the α parameters.

2022) which uses a pretrained FasterCNN to extract visual features that it passes to a small vision encoder. For the decoder side, it uses the text-pretrained model GPT2 (Radford et al., 2019).

Arabic image captioning. Arabic poses significant challenges to image captioning. This is due to the lack of native Arabic captioning datasets in the public domain, the morphological complexity of Arabic, and the large number of diverse dialects (Attai and Elnagar, 2020). However, a number of Arabic image captioning works exist. For instance, approaches such as root-word based RNNs and deep neural networks are used for direct Arabic caption generation (Jindal, 2017). Al-Muzaini et al. (2018) employ a generative merge model with three components: an LSTM-based language model, a CNN-based image feature extraction model, and a decoder that processes outputs from the first two models. ElJundi et al. (2020) introduce an Arabic captioning model trained on a translated Flickr8K dataset, discussing issues related to translation. Afyouni et al. (2021) present AraCap, a hybrid design that combines a CNN with object detection using attention mechanisms and produces captions through an LSTM. They train their model on MSCOCO and Flickr30k (Plummer et al., 2015)

datasets and test on an Arabic translated subset of MSCOCO. Lasheen and Barakat (2022) propose an encoder-decoder structure, incorporating attention mechanisms with CNN encoding and LSTM decoding. In another study (Emami et al., 2022), various Arabic image captioning models are formulated and assessed using standard metrics. The authors use transformers pretrained on diverse Arabic datasets following the architecture and training method introduced in OSCAR (Li et al., 2020). Elbedwehy and Medhat (2023) present a model employing transformers for both encoding and decoding. It uses feature extraction from images in the encoding stage and a pretrained word embedding model in the decoding stage, all tested on the Arabic-translated Flickr8k dataset in ElJundi et al. (2020). This work is closest to ours in that we also utilize transformer encoders and decoders. However, we use a GPT-styled decoder that endows our approach with high Arabic fluency.

3 Approach

3.1 Model Architecture

Our model is a vision-encoder-decoder architecture. For the vision encoder part, we employ an object

detection network (Anderson et al., 2018) and a three-layer transformer. For the text decoder, we use the pretrained transformer decoder JASMINE (Nagoudi et al., 2022). To align visual and textual features, we utilize cross-attention. In standard attention, also known as self-attention, the attention output is computed using three matrices derived from the same input: the query matrix Q, the key matrix K, and the value matrix V. More concretely, given an input sequence represented as a matrix S_t, where each row corresponds to a vector in the sequence, the attention is calculated as:

$$Attn(S_t) = \text{softmax}\left(\frac{S_t W_q (S_t W_k)^T}{\sqrt{d_k}}\right) S_t W_v \quad (1)$$

Where W_q, W_k, and W_v are the learnable weight matrices for the query Q, key K, and value V respectively. d_k is the dimensionality of the query/key vectors. The division by $\sqrt{d_k}$ is a scaling factor to ensure the dot products don't grow too large as the dimensionality increases.

In the case of cross-attention, the query is derived from the output of the text decoder's self-attention, while the key and value are sourced from the vision encoder. Mathematically, given image visual features output S_m, and the textual features S_t, the formula becomes:

$$XAttn(S_t, S_m) = \text{softmax}\left(\frac{(S_t W_q)(S_m W_k)^T}{\sqrt{d_k}}\right) \times S_m W_v \quad (2)$$

Now that the attention mechanism foundations are laid out, we describe our vision encoder and text decoder in detail.

3.1.1 Vision Encoder

Our vision encoder consists of two components: a pretrained object detection network, and a three-layer transformer encoder. For the object detection network, we employ bottom-up attention network (Anderson et al., 2018). In our initial experiments, it results in superior visual features compared to using the vanilla FasterRCNN model (Ren et al., 2015). Previous works (Li et al., 2020; Cornia et al., 2020; Chen et al., 2022) also show the effectiveness of this network in feature extraction. The transformer encoder, on the other hand, is a three-layer standard transformer architecture that takes the output of the detection network to further refine the visual features. For each image, the detection network detects the potential objects and

extracts the visual features from their bounding boxes.[1] These visual features are passed through a projection layer and then fed to the three-layer transformer encoder as input. We adapt meshed connection (Cornia et al., 2020) in our architecture between the encoder layers and the text decoder. This allows all the encoder layers to contribute to the input of the cross-attention rather than using only the output of the last encoder layer. The contribution of each encoder layer is determined by the learnable parameters matrix α. For each layer i, α_i is calculated as:

$$\alpha_i = \sigma(W_i[S_t \parallel XAttn(S_{m_i}, S_t) + b_i]) \quad (3)$$

Where S_t is the input sequence of each decoder layer, σ is the sigmoid activation function, W_i is a learnable weight matrix, b_i is a bias term and $\parallel$ indicates concatenation. This measures the relevance between the input for each decoder layer S_t, and the output of each encoder layer.

3.1.2 Gemini Decoder

We employ the pretrained Arabic decoder JASMINE (Nagoudi et al., 2022) as our text decoder. JASMINE is a decoder-based transformer that follows GPTNeo architecture (Black et al., 2021). JASMINE models range in complexity from 300 million to 13 billion parameters and are trained on a text dataset of approximately 400GB, covering diverse Arabic varieties from multiple domains. We utilize the JASMINE base variant in our architecture, which is a 12-layer transformer decoder with a 768-dimensional embedding.

Although the meshed connection introduced in Cornia et al. (2020) proved to have positive improvements on performance due to the richer visual features, calculating the cross-attention of each encoder layer with each decoder layer is computationally expensive. Inspired by Yu et al. (2022), we split our pretrained text decoder into two parts. The first part acts as a vanilla text decoder, while the second part acts as a fusion decoder that aligns visual and textual features. This design choice serves two purposes. First, it reduces the computations and the number of parameters by removing cross-attention layers and the mesh connections in the first half of the decoder. Second, having its first half intact acting as a vanilla text decoder, allows our decoder to keep its innate generative capabilities, while also enabling smoother convergence.

---

[1] A bounding box is a region in the image that contains the object.

As shown in Figure 3, the first half has only the pretrained self-attention layers of JASMINE. While the second half got cross-attention blocks inserted in-between each layer, acting as a fusion decoder. To ensure maintaining the functionality of our pretrained decoder, we freeze the first part that acts as the text decoder. This modification not only decreases computational cost but also positively impacts overall performance. In order to further enhance the quality of the features generated by both the vision encoders and the text decoder, we employ self-resurrecting activation unit (SRAU) introduced in Chen et al. (2022). The process of generating a caption relies on visual cues to convey the image's content and textual cues to provide relationships between words for a coherent and fluent output. To allow the important information to flow without distortion, SRAU selectively permits the activation above a certain threshold through a gating mechanism. This effectively filters out any weak signal produced by either the vision or language part.

Concretely, as shown in Figure 3, for each encoder-decoder connection, the output Z_i to the feedforward layer is calculated as:

$$Z_i = \pi_m \otimes XAttn(S_t, S_{m_i}) + \pi_t \otimes Attn(S_t), \quad (4)$$

in which π_m is the gating parameter for the vision part and π_t for the text part, calculated as:

$$\pi_m = \sigma(A_n)\mathbb{1}(\sigma(A_n) > \tau), \qquad \forall n \in Attn(S_t)$$
$$\pi_t = (1 - \sigma(A_n))\mathbb{1}(1 - \sigma(A_n) > \tau) \quad \forall n \in Attn(S_t)$$

where σ is the sigmoid function, A_n is an element in the attention matrix, $\mathbb{1}$ is an indicator function that equals one if the condition is true and zero otherwise, and τ is a hyperparameter. This negates any disturbance caused by weak activations below the threshold τ by zeroing them out. The final output Z to the feedforward layer will be the sum of each encoder-decoder connection weighted by the learned parameter α introduced earlier, mathematically:

$$Z = \frac{1}{\sqrt{L}} \sum_{i=1}^{L} \alpha_i Z_i \qquad (5)$$

Where L is the number of encoder layers, set to three in our architecture.

Figure 4. A comparison between the translations produced by Google translate API and NLLB for MSCOCO dataset. Unlike NLLB, Google API tends to give literal translations without incorporating the context.

3.2 Data Collection

Owing to the unavailability of high-quality Arabic captioning training data, we first start by creating a training dataset for our model. Manually labeling and creating a new dataset would be both time-consuming and expensive; therefore, we opt for translating the commonly used captioning dataset Microsoft Common Objects in Context (MSCOCO) (Lin et al., 2014). There are two famous training/validation splits for this dataset, the 2014 Karpathy's split, and the 2017 split. Both splits contain the same images and only differ in the split ratio. The dataset covers around 80 different objects in a total of 123k images with 5 captions per image. The dataset is annotated manually, which makes it suitable for evaluation. We create our dataset in two steps, (i) translating the English MSCOCO, followed by (ii) a quality assurance step to filter poor translations.

3.2.1 Machine Translation

In all of the previous attempts at Arabic image captioning pretraining (ElJundi et al., 2020; Sabri, 2021; Emami et al., 2022), Google translate API (Google, 2023) was used for translating the datasets. However, the quality of the translations produced by it is not satisfactory. In Sabri (2021) it is reported that from a random sample of 150 examples, a whooping 46% of the translations obtained by Google API are unintelligible. Motivated by that, we investigate Meta's *No Language Left Behind model (NLLB)* model (Costa-jussà et al., 2022)

for translation. Figure 4 illustrates a comparison between the translations produced by the Google Translate API and NLLB for four sentences sampled from MSCOCO dataset.

We conduct our comparison between the two translation models, Google Translate API[2] and NLLB, on two aspects. First, we manually check the quality of 200 sentences translated by both models. Second, we calculate the perplexity of the translations of both models using our JASMINE decoder. Perplexity calculates the probability of a given sequence, providing insight into the fluency of the output translations. Lower perplexity scores indicate better fluency, while higher scores indicate poor fluency. This metric helps us to quantitatively gauge how good the translations are, supplementing our manual evaluation to offer a comprehensive understanding of the models' performance. Subsequently, our observations reveal that the Google API tends to provide a more literal translation in comparison to NLLB. Empirically speaking, we find that 42% of Google's translations are unintelligible, a stark contrast to the mere 15% from NLLB. Interestingly, this observation is consistent with findings presented in Sabri (2021). Furthermore, when pitted against ChatGPT (Ouyang et al., 2022), the latter displays an impressive error rate of only 7% in its translations. However, we opted for NLLB due to its open-source nature.

Sim	Original Caption	Translated Caption
0.03	AN older man smiles while holding his luggage	أَنْتَ أَبُو بَكْرٍ؟
0.08	m m m m m m m m mmm m m m m	لَا ، لَا ، لَا ، لَا ، لَا
0.19	Red and white shower curtain in household bathroom.	ستارة حمام حمام حمام حمام حمام حمام حمام حمام حمام حم ـــــــ
0.19	A teddy bear with a pacifier and a baby bottle.	دب بـ (ما) و (ما) و (ما)
0.26	AN IAMGE OF A BATHROOM WITH A TOILET AND A SHOWER	حلم حمام مع مرحاض ومستحمام
0.31	Two doge have their paws out in an overexposed picture.	دوجيان يرفعان كفيهما في صورة مفرطة
0.51	white and gold plates with various arranged fruits	صالون " صحون " بيضاء وذهبية فيها " أي " الفاكهة " مرتبة .
0.57	This is a thing that is straightforward and plain.	هذا شيء واضح وواضح

Figure 5. Examples of the rejected translations from the dataset and their semantic similarity to the English caption. Where orange highlighting refers to poor translation, and red highlighting refers to poor original caption.

3.2.2 Data Quality Assurance

Although NLLB in general provides better translations compared to that of Google API, it can still

---

[2]The Google Translate API, integrated into Google Sheets, was used to translate the subset of data utilized in the comparison.

output 'hallucinations' and ultimately poor translations. This can be seen in the orange highlighted instances in Figure 5. Moreover, our manual inspection reveals that some English captions in the original dataset are indeed incorrect. The MSCOCO training set can have incomprehensible samples, typos, and even unrelated captions. Examples highlighted in red in Figure 5 illustrate these poor cases. To mitigate this issue, we employ a simple method based on semantic similarity that allows us to identify and reject any such examples.

The *semantic similarity* of two sentences, as the term suggests, is an indicator of the extent to which these two sentences align. A simple comparison between the embeddings of the two sentences can be obtained by passing each of them through a model and a metric such as cosine similarity can be calculated to determine how alike the two embeddings are. The smaller the angle between the two vectors, the higher the similarity score, indicating that the sentences are closer in meaning. When the sentences are in different languages, it is crucial to employ a multilingual model to generate accurate embeddings, ensuring the semantic comparison remains valid across languages. In our experiments, we employ sentence-BERT (Reimers and Gurevych, 2019) to calculate the semantic similarity between each original caption and its translation. We empirically chose a similarity score threshold of 0.6, rejecting all captions below that threshold. This results in removing a total of 60K samples from the whole dataset, which amounts to approximately 10% of the data.

3.2.3 AraCOCO Evaluation Dataset

Evaluating the performance of an Arabic captioning model presents a significant challenge due to the limited availability of human captioned data. To tackle this issue, we manually annotate a subset of 500 images from the MSCOCO test set, dubbing our resulting dataset *AraCOCO*. For each of the 500 images, we acquire five distinct captions. To ensure diversity of image descriptions, we acquire captions from five native Arabic-speaking annotators. The human labeling process is carried out using *Label Studio*, a platform designed for such tasks. Each annotator is presented with the same set of images and is asked to write an Arabic caption describing the image given a unique English caption as a reference. We encourage annotators to provide an Arabic caption that is more descriptive whenever possible. That is, in cases where the

English Caption	NLLB Translation	AraCOCO
An airport with large jetliners and a bus traveling on a tarmac.	مطار مع طائرات كبيرة وحافلة تسافر على المدرج	أشجار النخيل أمام مطار به طائرتان كبيرتان للركاب وحافلات مكوكية.
a group of buses driving around at the airport	مجموعة من الحافلات تسير في المطار	مجموعة من الحافلات تتجول في المطار
Airplanes sit at the gate as transportation vehicles move about.	الجيران تجلس عند البوابة بينما تتحرك مركبات النقل.	طائرات متوقفة عند بوابة المطار وهناك مركبات نقل
A busy runway with buses and luggage carts driving around	مدرج مزدحم مع الحافلات وعربات الأمتعة التي تقود حولها	مدرج مزدحم مع حافلات وعربات أمتعة تتجول
An airplane and busses are lined up at the airport.	طائرة وحافلات منتظمة في المطار	طائرة وحافلات منتظمة في المطار

Table 1: A comparison between original MSCOCO captions (first column), their NLLB translations (second column), and AraCOCO captions (third column) for the image in Figure 6.

Figure 6. A sample from MSCOCO included in our Ara-COCO.

English caption is not capturing all details in the image, annotators are encouraged to capture these lacking details in their Arabic captions. Each annotator gets to provide only one caption per image, This approach ensures having multiple perspectives to the captions on the same image. We provide an example image from AraCOCO in Figure 6, along with five different captions each acquired from one annotator in Table 1.

4 Experiments

We analyze the performance of three variations of our architecture: (i) using the normal decoder with cross-attention in each layer, (ii) using Gemini decoder without freezing the text part, and (iii) using Gemini decoder while freezing the text part. As a baseline, we train a VisualGPT model (Chen et al., 2022) on the English MSCOCO training set then translate output into Arabic using NLLB. Our trained VisualGPT achieves a 117.8 CIDEr score on the English MSCOCO validation set. We conduct our experiments on three datasets, as follows:

(i) Our translated MSCOCO: Following the Karpathy split, our translated and filtered MSCOCO contains $543, 817$ samples for training (Train), $22, 845$ samples for validation (Dev), and $22, 912$ samples for testing (Test). We refer to this dataset as MSCOCO.

(ii) Translated Flickr8K: Similar to the original Flickr8k, the translated dataset introduced in ElJundi et al. (2020) consists of $6, 000$ images for Train, $1, 000$ images for Dev, and $1, 000$ for Test. Each Image has three captions, all translated using Google translate API. We refer to this dataset simply as Flickr8K.

(iii) AraCOCO: As described in Section 3.2.3, AraCOCO consists of 500 images from Karpathy test split. Each image has five captions, all obtained from human annotators.

4.1 Implementation Details

We use JASMINE base (300m) as our text decoder. While for the detection network, following previous works (Li et al., 2020; Cornia et al., 2020; Chen et al., 2022), we employ bottom-up attention network (Anderson et al., 2018) based on Resnet-101 backbone (He et al., 2016) with $2, 048$ output features. We also limit the maximum number of detections per image to 50 bounding boxes. The three-layer transformer encoder contains 12 attention heads per layer with 768 embeddings dimension.

As we utilize the JASMINE decoder (Nagoudi et al., 2022), we adopt its byte-pair encoding (BPE) vocabulary where frequent character pairs are merged to form subwords. This vocabulary encompasses $63, 999$ tokens. For data preprocessing, we employ a custom normalizer that removes punctuation and repeated characters.

For the optimization part, in all experiments, we use AdamW Loshchilov and Hutter (2019) with a

Model	BLEU-1 ↑	BLEU-4 ↑	Rouge ↑	CIDEr ↑
VisualGPT	56.2	21.4	44.1	82.1
Violet (w/o Gemini)	45.1	11.3	34.1	41.2
Violet (w/ Gemini)	59.2	21.5	46.3	83.2
Violet (w/ Gemini) *	**60.3**	**24.8**	**47.2**	**84.9**

Table 2: Results on the translated MSCoco test set. VisualGPT is trained by us on the MSCOCO dataset, and the outputs were translated using NLLB (Costa-jussà et al., 2022). (w/o Gemini) means using a normal text decoder with meshed cross-attention in each layer. * indicates freezing the first part of the text decoder.

learning rate of $1e^{-4}$, and empirically set τ to 0.3. The model is trained using a batch size of 60 for 20 epochs while employing early stopping with a patience of 5 on the validation loss. For Flickr8k, we use our MSCOCO-pretrained model and only finetune it for one epoch on Flickr8k's training data. We employ a cross-entropy loss and train the model in an auto-regressive manner, where the decoder predicts the next token given the visual features and the previously generated textual tokens.

4.2 Results and Discussion

We evaluate the performance of our models against previous methods on the popular evaluation metrics BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and CIDEr (Vedantam et al., 2015). The results of our models on our MSCOCO dataset are displayed in Table 2. Our Gemini decoder with six frozen layers (last row in Table 2) achieves better performance while having fewer computations than the unfrozen counterpart. Furthermore, it achieves around three points higher CIDEr score compared to the translated VisualGPT outputs (first row in Table 2). The poor performance observed using the full decoder with cross-attention layers (second row in Table 2), compared to other variants may be due to sensitivity of the decoder parameters which end up being changed significantly with full cross-attention across all its layers.

Model	BLEU-1	BLEU-4	Rouge	CIDEr
Elbedwehy and Medhat (2023)	**58.7**	**16.5**	<u>38.0</u>	<u>46.9</u>
Emami et al. (2022)	39.0	09.0	33.4	42.3
Violet	<u>44.2</u>	<u>13.0</u>	**38.4**	**60.1**

Table 3: Results on Flickr8k test set from (ElJundi et al., 2020). The results are taken from the respective papers.

To compare our Arabic captioning model with previously published Arabic models, we evaluate our model on the Flickr8k test set from (ElJundi et al., 2020). As shown in Table 3, our model achieves 2 points better score on the ROUGE metric, while having a substantial improvement over previous published results in the CIDEr metric. Our model scores 13 points higher than the best model of the two previous models. On the other hand, our model falls behind in BLEU score against Elbedwehy and Medhat (2023). It is worth noting, however, that we are only comparing to published results of Elbedwehy and Medhat (2023) since their model is not available (i.e., not released). They have also used the validation set of Flickr8k in their training, and applied self-critical (Rennie et al., 2017) with no mention of the target data, thus giving their model an advantage over our own model. Regardless, for image captioning, it is known that the CIDEr (where our model excels) is a more relevant evaluation metric than BLEU. Finally, we

Model	BLEU-1	BLEU-4	Rouge	CIDEr
VisualGPT	52.7	17.6	40.2	58.5
Violet	**54.5**	**19.0**	**41.8**	**61.2**

Table 4: Results of our model against translated outputs of VisaulGPT on AraCOCO.

score our model on our manually annotated dataset, AraCOCO. As shown in Table 4, our model again exhibits sizeable gains compared to our baseline model (i.e., the translated output of VisualGPT). This means that we cannot expect a satisfactory performance by simply taking output from a VLM trained on English data and translating it into Arabic, further corroborating our previous findings and motivating future work on developing VLM models that natively tailored to Arabic language.

5 Conclusion

In this paper, we introduced *Violet*, an Arabic image captioning model leveraging the pretrained text decoder JASMINE. Our results demonstrated the efficacy of our Gemini decoder in enhancing performance while simultaneously reducing the number of model parameters and computations. We also presented a new method that is effective for acquiring Arabic captioning data from available English data. In addition, we manually annotated a new dataset for evaluating Arabic image captioning models. Our model outperforms all of our baselines and promises to enable benchmarking in this area. We will release our model and datasets to advance Arabic vison-language research.

6 Limitations

Similar to other image detection-based captioning models, the dependence on an external network to provide the visual features introduces an additional layer of complexity to the model. Since the model is not trained end to end, during inference, the visual features must first be obtained from the detection network before passing it to the vision encoder. Another limitation arises from the constraints of the training data. Since MSCOCO focuses solely on 80 class objects, the model's applicability in real-world scenarios is restricted. In our future work, we aim to address both of these limitations to enhance Arabic models' efficiency and broaden their practical usage.

7 Ethics Statement and Broad Impact

Bridging the Gap in Multilingual Image Captioning. Image captioning serves as a crucial bridge between vision and language, with its applications touching numerous domains such as accessibility, education, and search engines. For a long time, the privilege of these advancements has been constrained to a handful of languages, primarily due to the lack of necessary datasets and dedicated research in other languages. Arabic, with its vast speakers and rich history, has unfortunately been left behind in this domain. Our work with *Violet* seeks to rectify this disparity, providing a robust foundation for Arabic image captioning. By releasing *Violet* and the datasets, we aim to invigorate research in this direction, promoting inclusivity and equal opportunity in NLP and computer vision advancements across languages.

Automated Data Acquisition and Transparency. To overcome the challenge of limited labeled data for Arabic image captioning, we employed a novel method for data acquisition using available English datasets. While this approach provides a solution, it also warrants a discussion on the accuracy, bias, and quality of the automatically acquired data. We emphasize that while our method provides a foundational dataset, manual annotations and human evaluations remain paramount for ensuring data quality and avoiding propagation of errors.

Acknowledgment of Data Sources and Fair Credit. Similar to ensuring proper credit assignment for benchmarking tasks, we emphasize the importance of acknowledging the original data sources we leveraged, especially in the context of automated data acquisition. Users and researchers utilizing our datasets and model are encouraged to cite and acknowledge the original datasets and sources. This practice ensures that original creators receive the recognition they deserve and promotes a culture of transparency and fairness in the research community.

Acknowledgments

References

Imad Afyouni, Imtinan Azhar, and Ashraf Elnagar. 2021. Aracap: A hybrid deep learning architecture for arabic image captioning. *Procedia Computer Science*, 189:382–389.

Huda A Al-Muzaini, Tasniem N Al-Yahya, and Hafida Benhidour. 2018. Automatic arabic image captioning using rnn-lstm-based language model and cnn. *International Journal of Advanced Computer Science and Applications*, 9(6).

---

[3]https://alliancecan.ca
[4]https://arc.ubc.ca/ubc-arc-sockeye

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. 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.

Anfal Attai and Ashraf Elnagar. 2020. A survey on arabic image captioning systems using deep learning models. In *2020 14th International Conference on Innovations in Information Technology (IIT)*, pages 114–119. IEEE.

Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. 2023. Foundational models defining a new era in vision: A survey and outlook. *arXiv preprint arXiv:2307.13721*.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large scale autoregressive language modeling with meshtensorflow.

Jun Chen, Han Guo, Kai Yi, Boyang Li, and Mohamed Elhoseiny. 2022. Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18030–18040.

Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10578–10587.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Samar Elbedwehy and T Medhat. 2023. Improved arabic image captioning model using feature concatenation with pre-trained word embedding. *Neural Computing and Applications*, pages 1–17.

Obeida ElJundi, Mohamad Dhaybi, Kotaiba Mokadam, Hazem M Hajj, and Daniel C Asmar. 2020. Resources and end-to-end neural network models for arabic image captioning. In *VISIGRAPP (5: VISAPP)*, pages 233–241.

Jonathan Emami, Pierre Nugues, Ashraf Elnagar, and Imad Afyouni. 2022. Arabic image captioning using pre-training of deep bidirectional transformers. In *Proceedings of the 15th International Conference on Natural Language Generation*, pages 40–51.

Zhe Gan, Linjie Li, Chunyuan Li, Lijuan Wang, Zicheng Liu, Jianfeng Gao, et al. 2022. Vision-language pre-training: Basics, recent advances, and future trends. *Foundations and Trends® in Computer Graphics and Vision*, 14(3–4):163–352.

Google. 2023. Google translate api. Accessed: 15/07/2023.

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

Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. 2019. Attention on attention for image captioning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4634–4643.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, et al. 2023. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*.

Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. 2014. Caffe: Convolutional architecture for fast feature embeding. *arXiv preprint*, arXiv:1408.5093.

Vasu Jindal. 2017. A deep learning approach for arabic caption generation using roots-words. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Andrej Karpathy, Armand Joulin, and Li F Fei-Fei. 2014. Deep fragment embeddings for bidirectional image sentence mapping. *Advances in neural information processing systems*, 27.

Moaz T Lasheen and Nahla H Barakat. 2022. Arabic image captioning: the effect of text pre-processing on the attention weights and the bleu-n scores. *Int J Adv Comput Sci Appl*, 13(7):11.

Siming Li, Girish Kulkarni, Tamara Berg, Alexander Berg, and Yejin Choi. 2011. Composing simple image descriptions using web-scale n-grams. In *Proceedings of the fifteenth conference on computational natural language learning*, pages 220–228.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. 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.

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

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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *In ICLR*.

El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, AbdelRahim Elmadany, Alcides Alcoba Inciarte, and Md Tawkat Islam Khondaker. 2022. Jasmine: Arabic gpt models for few-shot learning. *arXiv preprint arXiv:2212.10755*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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.

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. In *International conference on machine learning*, pages 8748–8763. PMLR.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

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

Sabri Monaf Sabri. 2021. *Arabic image captioning using deep learning with attention*. Ph.D. thesis, University of Georgia.

Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. 2022. From show to tell: A survey on deep learning-based image captioning. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):539–559.

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

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

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR.

Yezhou Yang, Ching Teo, Hal Daumé III, and Yiannis Aloimonos. 2011. Corpus-guided sentence generation of natural images. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 444–454.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.