Image Captioning with Vision/Text Transformers

Yinghuan Zhang
Carnegie Mellon University
yinghuan@andrew.cmu.edu

Qi Xin
Carnegie Mellon University
qix@andrew.cmu.edu

Bowen Tan
Carnegie Mellon University
btan2@andrew.cmu.edu

Abstract

Following the recent success of Transformer, we implement a Transformer-Transformer architecture image captioning model, with Vision Transformer (ViT) as the encoder and a standard Transformer decoder. After training, we use the pretrained GPT-2 to generate captioning. We further utilize data augmentation, including image augmentation and text augmentation. Each realization itself can improve the performance. However, combining image augmentation and text augmentation fails to further enhance the result. We experimented with the common evaluation metrics for image captioning, including BLEU-1, BLEU-4, METEOR, ROUGE, and CIDEr, where our model achieves 74.9, 34.3, 27.2, 55.5, and 104.2 respectively. Our results outperform the CNN-LSTM baselines. We also apply transfer learning to our model, which does not render better results.

1 Introduction

Image captioning is the task of producing a natural-language sentence describing visual content of an image. On the one hand, there are many practical applications of image caption generation, such as helping visually impaired people understanding pictures, Content-Based Image Retrieval, and auto-description of photos on social platforms. On the other hand, large research efforts have been devoted to image captioning because it is a multi-modal problem connecting computer vision and natural language processing, which plays an essential role in generative intelligence. Therefore, image caption generation has become a popular research area over the past few years.

Encoder-decoder architecture based models were widely used a couple years ago and allegedly have beat human baselines. The choice for encoder and decoder varies and has been updated all the time to take advantages of the latest breakthrough in computer vision and natural language processing. Common choice for encoders are AlexNet [11], VGGNet [7], ResNet [14], Attention-Based CNN [18], etc. Common choice for decoders are RNN [11] and LSTM [7] [14] [18]. After training, CNN or other type of encoder would be able to detect the objects along with their relationships, and output the global features as an encoded image vector. Then RNN or LSTM, as the decoder, takes the global image features as its initial state and generates a sequence of word tokens. Though this Encoder-decoder architecture outperforms those previous bottom-up approaches, CNN and LSTM face bottleneck if we want to further improve accuracy. Global CNN tends to over compress the information and thus it may not capture detailed information well enough. LSTM has challenges in generating long length sentences because a word generated at the beginning would have weaker effect towards the words at the end of the sentence.

Final Project Report, Team 24.
10-701 Introduction to Machine Learning. Carnegie Mellon University.

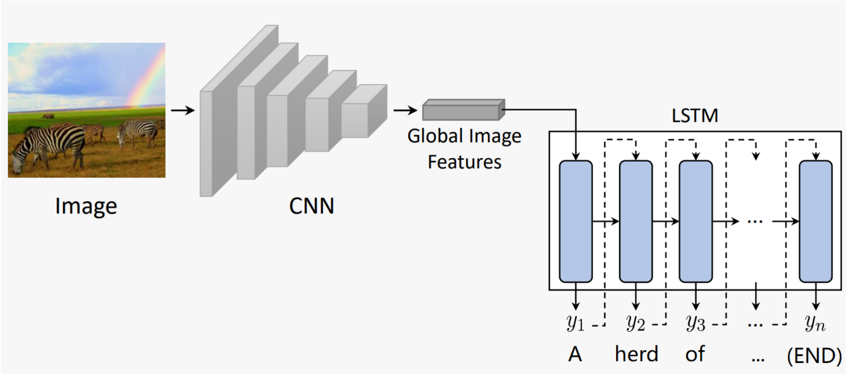

Figure 1: An illustration of CNN-LSTM image captioning model.

The recent trend of transformer-based machine learning technique [16] and BERT-like pre-training approaches [3] stands out, and have greatly improved the benchmark of almost all kinds of computer vision and natural language processing tasks, including image captioning. Carion et al. [2] first use a CNN-transformer architecture, which outperforms well-established and highly-optimized CNN-LSTM baseline. After Dosovitskiy et al. [4] invent Vision Transformer and show it has better performance and greater efficiency than CNN models on image classification task, a recent paper by Liu et al. [10] come up with full transformer architecture for image captioning, which push the benchmark to a new level. Zhang et al. [19] further implemented the idea of using a pre-trained cross-modal fusion model, Oscar [9], as the encoder and BERT as the decoder. They also used fine-tuning strategy to adapt to image captioning task.

In this work, we implement our architecture with pre-trained Vision Transformer (ViT) as the encoder and GPT-2 as the decoder. After fine-tuning, our full-transformer model indeed outperforms CNN-LSTM baseline. More importantly, we further improve our model performance with data augmentation, including both image augmentation and text augmentation.

2 Related Work

2.1 CNN-LSTM image captioning model

Most image captioning models have an encoder-decoder paradigm as is shown in Figure 1. [6] [15] Similar to neural machine translation, where a source sentence is converted to a single vector by a neural network and then another neural network generates sentence of the target language based on the vector, the encoder of image captioning model use computer vision models like CNN to extract scene type or location, object properties and their interactions. The extracted vector should contains all the necessary information about the image, just like the encoded vector in machine translation model represents the semantics of the source sentence in an abstract way. The decoding part of image captioning model is nearly identical to the decoding part of machine translation model since they all take in an encoded vector and generate the caption. For example, LSTM takes the vector as its initial state, and then generates word one by one until and an END sign shows up. Each next word is only based on the current time step and the previous hidden state.

CNN-LSTM, as one of the most popular encoder-decoder architecture in the past, demonstrate the advantages of top-down image captioning approaches over bottom-up approaches. Here, top-down means we have a representation of global image features that can be used to translate into a sentence, like the encoded vector generated by CNN as we described in the last paragraph. On the contrary, bottom-up approaches explicitly detect and extract the objects in the figure, then use language model to link those objects and output a sentence. Though CNN-LSTM model is no longer the state-of-the-art method anymore, the encoder-decoder architecture is still being extensively used in today's best performing models.

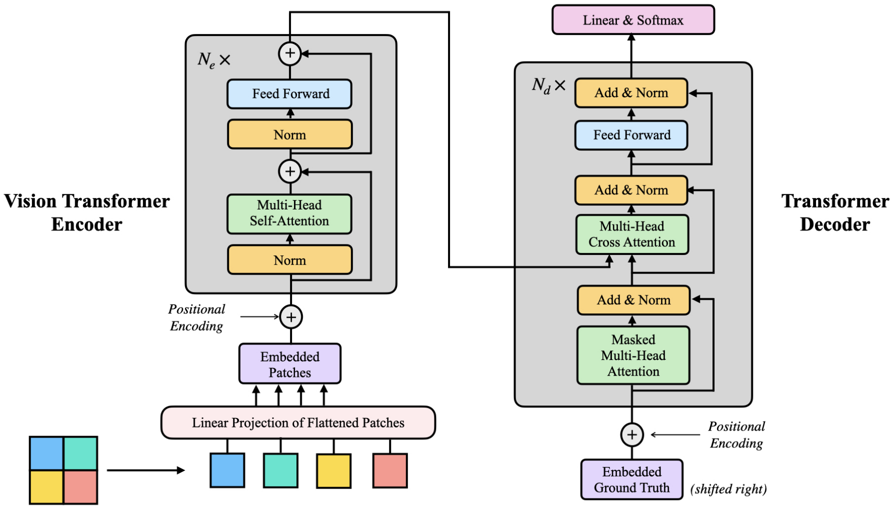

Figure 2: Transformer-Transformer Model Architecture Overview

2.2 Transformer and large transformer-based pre-trained models

The Transformer models an encoder-decoder structure [16]. The encoder maps an input sequence of symbol representations to a sequence of continuous representations. Given the output representations of the encoder, the decoder then generates an output sequence of symbols. The Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder. An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and outputs are all vectors. Self-Attention is an attention mechanism relating different positions of a single sequence to compute a representation of the same sequence. In addition to attention sub-layers, each transformer block contains a fully connected feed-forward network, which is applied to each position separately and identically. Radford et al. [13] later trained a large Transformer with 1.5B parameters, named GPT-2, by language modeling (the task of predicting the next word based on previous n words).

Inspired by the success of self-attention-based architectures, in particular Transformers [16], in NLP tasks, multiple works try combining CNN-like architectures with self-attention, some even replacing the convolutions entirely. Dosovitskiy et al. [4] proposed Vision Transformer(ViT) with applying a standard Transformer directly to images with the fewest possible modifications. Basically, ViT separates the image into hundreds of patches, each patches can be viewed as a word in original Transformer setting. Similar to the masked language modeling (MLM) in BERT [3], pre-training of ViT can just mask a patch for self-supervision.

3 Method

3.1 Transformer-Transformer Model

An overview of the model is depicted in Figure 2. Our Transformer-Transformer captioning model has an encoder-decoder structure like the standard Transformer. Instead of convolutional architectures, the transformer-based encoder maps an input sequence of symbol representations to a sequence of continuous representations. Given these representations, the decoder then generates an output sequence of symbols. Unlike the standard Transformer, which takes 1D token embeddings as the encoder's input, our model takes 2D images embeddings as encoder's input, followed by the standard Transformer decoder, which generated 1D sequences.

3.1.1 Encoder

To handle 2D images in the encoder side, we follow the standard image embedding process for Vision Transformer[4]. We firstly resize the input image into a fixed resolution $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$, then divides the image into sequence of flattened 2D patches and reshape them into 1D patch sequence $\mathbf{x}_p \in \mathbb{R}^{N \times \left(P^2 \cdot C\right)}$, where (H, W) denotes the resolution of the original images, C denotes the number of channels, P denotes the patch size, (P, P) is the resolution of each image patch, and $N = \frac{H}{P} \times \frac{W}{P}$ is the resulting number of patches. After that, we flatten the patches and map to D dimensions, which is used as a constant latent vector size through all layers in the Transformer, with a trainable linear projection (Eq.1. We refer to the output of this projection as the patch embeddings. Positional encodings are added to the patch embeddings to retain positional information. Following ViT standard modification, we use standard learnable 1D position embeddings. The resulting sequence serves as input to the Vision Transformer Encoder.

Transformer encoder consists of N_e stacked Transformer blocks which are composed of multi-head self-attention (MSA)(Eq.2) and fully connected feed-forward network (FFN)3. Layernorm (LN)[1] and residual connections[5] are applied at each part in one Transformer block. The FFN contains two layers with GELU non-linearity.

The encoder performs as follows:

$$\mathbf{z}_0 = \left[\mathbf{x}_{class}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^N\mathbf{E}\right] + \mathbf{E}_{pos}, \quad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}, \mathbf{E}_{pos} \in \mathbb{E}^{(N+1) \times D} \tag{1}$$

$$\mathbf{z}' = \mathrm{MSA}(\mathrm{LN}(\mathbf{z}_{\ell-1}) + \mathbf{z}_{\ell-1}, \quad \ell = 1...L \tag{2}$$

$$\mathbf{z}_\ell = \mathrm{FFN}(\mathrm{LN}(\mathbf{z'}_\ell)) + \mathbf{z'}_\ell, \quad \ell = 1...L \tag{3}$$

Multi-head self-attention (MSA) sub-layer contains H parallel heads, each corresponding to an independent scaled dot-product attention function h_i. The scaled dot-product attention is particular attention proposed in the Transformer model. The multi-head mechanism allows the model to attend to different sub-spaces jointly. The attention results of different heads use a linear transformation W^O to aggregate. MSA is computed as:

$$\mathrm{MSA}(Q, K, V) = \mathrm{Concat}(h_1, \cdots, h_H)W^O \tag{4}$$

where $Q \in \mathbb{R}^{N_q \times d_k}$,$K \in \mathbb{R}^{N_k \times d_k}$,$V \in \mathbb{R}^{N_k \times d_v}$ denotes the query, key and value matrix respectively. $h_i(Q, K, V) = \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right), \forall i = 1...H$ is scaled dot-product attention function.

The followed positional FFN layer is implemented as two linear layers with GELU activation function, formulated as:

$$\mathrm{FFN}(x) = \mathrm{FC}_2(\mathrm{Dropout}(\mathrm{GELU}(\mathrm{FC}_1(x)))) \tag{5}$$

3.1.2 Decoder

The transformer-based decoder consists of N_d stacked identical transformer block similar to the encoder. Each transformer decoder block is composed of a masked multi-head self-attention sublayer followed by a multi-head cross attention sublayer and a positional feed-forward sublayer sequentially. The decoder in our model takes in the encoded image embeddings and the embedded ground truth caption sequences. In addition, we add positional embedding to it for word embedding features, which is added to make use of the order of the ground truth caption sequences. The positional encodings have the same dimension as the sequence embeddings to be summed. We follow the standard Transformer and use sine and cosine functions of different frequencies:

$$\mathrm{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\mathrm{model}}}}\right)$$

$$\mathrm{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\mathrm{model}}}}\right)$$

where pos denotes the position and i denotes the dimension.

The decoder block utilizes the last decoder block's output feature to predict the next word via a linear layer whose output dimension equals the vocabulary size. For example, given a ground truth sentence, $y^\star_{1:T}$ and the prediction $y^\star_t$ of the generated caption by the captioning model with parameters θ, the cross-entropy loss can be computed as:

$$L_{XE}(\theta) = -\sum_{t=1}^{T} \log\left(p_\theta\left(y^\star_t | y^\star_{1:t-1}\right)\right). \tag{6}$$

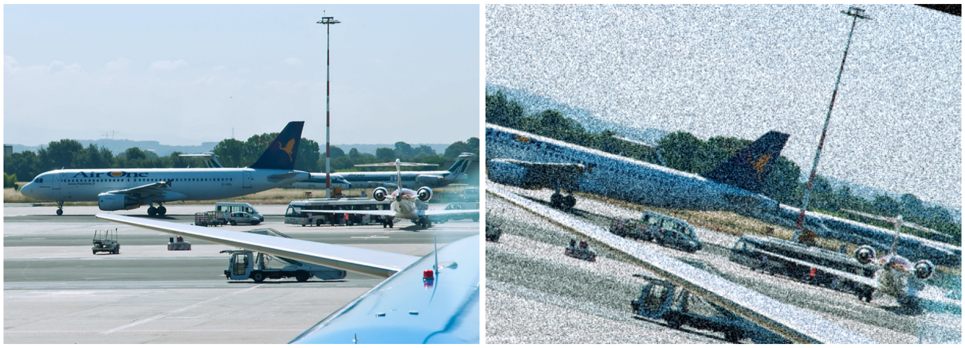

Figure 3: Image augmentation instance. **Left:** the original image; **Right:** the augmented image.

3.2 Data augmentation

The performance of machine learning models always heavily depend on the quality and the amount of training data. *Data augmentation* is the most commonly used strategy to manipulate data to improve model learning. Rich types of augmentation strategies have been developed for both computer vision and natural language processing. In this project, we tried simple and efficient approches of them.

Augmentation for images We augment images by following transformation sequences:

- rotate the image, by a randomly sampled angle between -30 to 30 degree.
- add Gaussian noise, whose variance value is sampled from 10 to 60.
- randomly crop the image, the ratio for each side is sampled between 0 and 0.2.

Augmentation for captions We apply synonym augmentation on the captions of images. Specifically, for each randomly sampled word in a sentence, we replace it with its synonym which is obtained in the WordNet.

3.3 Transfer learning

Transfer learning has been ubiquitous in the machine learning community, such as BERT, GPT3, ResNet, etc. Transfer learning techniques make the knowledge learned from one task and benefit the learning of other tasks. In this project, we tried one of the simplest ways, which is to initialize our model with another task. Specifically, we use the TrOCR model from the charactor recognition task, which is also a image-to-text model.

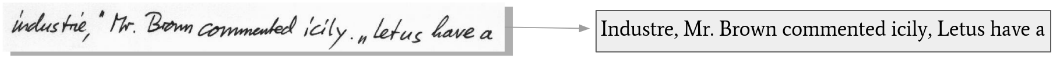
industrie, " Mr. Brown commented icily. " Let us have a
Industre, Mr. Brown commented icily, Letus have a

Figure 4: Charactor recognition task.

4 Experiment

4.1 Setup

We use MS Coco Dataset, which contains 120,000 images. Each image has five ground truth captions. We initialize our encoder and decoder with ViT and GPT2 models, and apply Adam optimization with 4×10^{-5} learning rate and 0.01 weight decay, and our training batch size is 16. Our experiments run on 1080Ti GPUs.

Table 1: Results. Best numbers are in bold.

	BLEU-1	BLEU-4	METEOR	ROUGE	CIDEr
Baseline (Attention CNN+LSTM) [18]	72.4	31.4	25.0	53.1	97.2
Ours	73.9	33.4	26.4	54.2	100.2
Ours + Aug (caption)	**74.9**	34.2	26.9	**55.5**	103.3
Ours + Aug (image)	74.3	**34.3**	**27.2**	55.0	**104.2**
Ours + Aug (caption + image)	74.6	34.1	26.8	54.8	103.4
Ours + Transfer	72.7	32.4	25.9	53.8	96.8

4.2 Evaluation Metrics

BLEU BiLingual Evaluation Understudy (BLEU)[12] is firstly proposed as a method for automatic evaluation of machine translation task. It is based on n-gram based precision. Four sub metrics are denotes as BLEU_n, for $n = 1, 2, 3, 4$. For a candidate sentence a and a set of reference sentences b, the BLEU score is calculated as:

$$\mathrm{BLEU}_n(a, b) = \frac{\sum_{w_n \in a} \min\left(count_a(w_n), \max_{j=1,\cdots,|b|} count_{b_j}(w_n)\right)}{\sum_{w_n \in a} count_a(w_n)},$$

where w_n denotes n-gram, $count_x(y_n)$ denotes count of n-gram y_n in sentence x.

BLEU or $\mathrm{BLEU}_{\mathrm{overall}}$ is a geometric mean of n-gram scores from 1 to 4. In our work, we take the BLEU_4 score as one of our main evaluation metric.

METEOR Metric for Evaluation of Transflation with Explicit ORdering (METEOR) [8] is an automatic metric for machine translation evaluation with improved correlation with human judgment. For a candidate sentence a and a set of reference sentences b, an alignment between a and b is first computed. Then the METEOR score is calculated as:

$$\mathrm{METEOR} = \max_{j=1,\cdots,|b|} \left(\frac{10PR}{R - 9P}\right)\left(1 - \frac{1}{2}\left(\frac{\#\mathrm{chunks}}{\#\mathrm{matched} \quad \mathrm{unigrams}}\right)^3\right),$$

where P is the unigram precision, R is the unigram recall, $\#\mathrm{chunks}$ is the number of set of unigrams adjacent in a and b_j.

ROUGE ROUGE is the abbreviation of Recall-Oriented Understudy for Gisting Evaluation. It is also n-gram based and considers precesion and recall given the hypothesis and reference. It is a standard metric to evaluate summarization and image captioning.

$$ROUGE_n(a, b) = \frac{\sum_{j=1}^{b} \sum_{w_n \in b_j} \min\left(c_a(w_n), c_{b_j}(w_n)\right)}{\sum_{j=1}^{b} \sum_{w_n \in b_j} c_{b_j}(w_n)}$$

where a is the candidate sentence, b is the set of reference sentence, w_n is n-gram, and $c_x(y_n)$ stands for the count of n-gram y_n in sentence x.

CIDEr CIDEr is a consensus-based image description evaluation [17]. For a candidate sentence a and a set of reference sentences b, the CIDEr score is computed as:

$$\mathrm{CIDEr}_n(a, b) = \frac{1}{|b|} \sum_{j=1}^{|b|} \frac{\mathbf{g^n}(a) \cdot \mathbf{g^n}(b_j)}{\|\mathbf{g^n}(a)\| \|\mathbf{g^n}(b_j)\|},$$

$$\mathrm{CIDEr}(a, b) = \sum_{n=1}^{N} w_n \mathrm{CIDEr}_n(a, b),$$

where $\mathbf{g^n}(x)$ is the vector formed by TF-IDF scores of all n-grams in x.

4.3 Results

All of our results and comparisons are shown in Table 1.

CNN-RNN vs full transformer Our Transformer model outperforms the CNN+LSTM baseline by a significant margin across all the metrics.

Data augmentation The data augmentation methods, both on captions and images, get further improvements. Data and model are two orthogonal aspects in learning, so it is intuitive for them to work together. However, augmenting images and captions together doesn't get even better results. It should be because the augmented examples are too far away to the data distribution if we change both the image and caption.

Tranfer Learning It is a bit surprising that the transfer learning doesn't work well and even shows a drop of performance. It is possible that the pretrain task, character recognition, is too different with our generic image captioning task, so it doesn't share enough knowledge to our model.

5 Conclusion

In this project, for a sequence-to-sequence task, image captioning, we apply the vision and text transformer both on the encoder and decoder, and get significant better results than the basic CNN+LSTM. We also investigated the data augmentation and transfer learning techniques, and some of them provide further performance improvements.

It should be interesting to also try to apply reinforcement learning on our model, which is able to leverage external reward signals (e.g., CIDER, METEOR scores). Moreover, making use of other supervision signals, such as the image segmentation and object detection results, should also benefit our image captioning task. We'll keep those interesting and promising directions in our future study and research.

References

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization, 2016.

[2] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers, 2020.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[6] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga. A comprehensive survey of deep learning for image captioning, 2018.

[7] J. Jin, K. Fu, R. Cui, F. Sha, and C. Zhang. Aligning where to see and what to tell: image caption with region-based attention and scene factorization, 2015.

[8] A. Lavie and A. Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, page 228–231, USA, 2007. Association for Computational Linguistics.

[9] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, Y. Choi, and J. Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks, 2020.

[10] W. Liu, S. Chen, L. Guo, X. Zhu, and J. Liu. Cptr: Full transformer network for image captioning, 2021.

[11] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks, 2014.

[12] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL `https://doi.org/10.3115/1073083.1073135`.

[13] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[14] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning, 2017.

[15] M. Stefanini, M. Cornia, L. Baraldi, S. Cascianelli, G. Fiameni, and R. Cucchiara. From show to tell: A survey on deep learning-based image captioning, 2021.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.

[17] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation, 2015.

[18] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator, 2015.

[19] P. Zhang, X. Li, X. Hu, J. Yang, L. Zhang, L. Wang, Y. Choi, and J. Gao. Vinvl: Revisiting visual representations in vision-language models, 2021.