The Summary of ”Meshed-Memory Transformer for Image Captioning” [1] Paper

Furkan Gül

I. Motivation

Recently, self-attention based models have become widely used over recurrent neural networks (RNNs) and long-short-term-memory (LSTM) networks. Transformer [2] and BERT [3] models and many applications related to these two models such as video understanding and image retrieval can be given as examples for the dominance of self-attention-based models. Transformer models are the single-modal architectures. However, image caption is a type of multi-modal (processing different types of information together) problem. It means that the image captioning is dealing with not only forming a natural language sentence but also understanding the context and relations of objects in a given input image. That’s why it requires the multi-modal architecture which is different than the single-modal architectures implemented in standard transformer models.

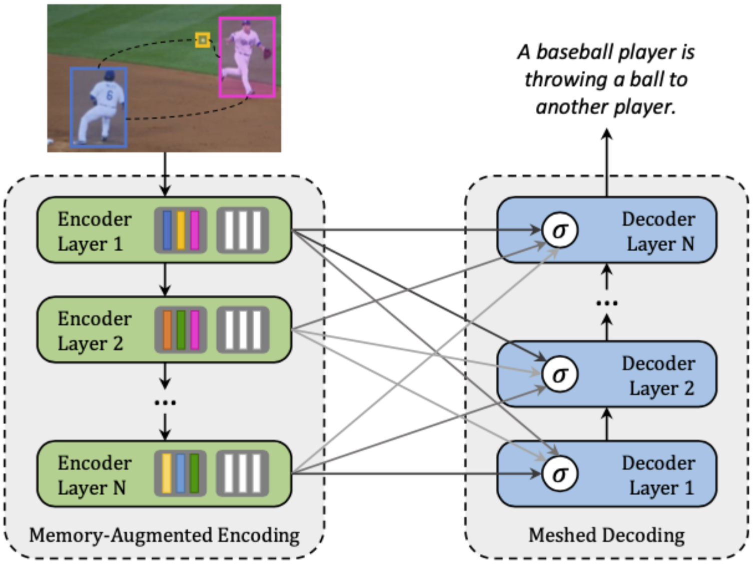

Fig. 1. Proposed image captioning model encodes relationships between image regions exploiting learned a prior knowledge. Multi-level encoding of image regions are linked to language decoder part through a meshed and learnable connectivity [1].

In this paper, to build the model architecture for image captioning, they add two key properties regarding all previous studies about image captioning algorithms to the Transformer model [2] for language (machine) translation. The first characteristic of image captioning problems is that image regions and their relations are encoded in a multi-level perspective in which both low-level and high-level interactions are considered. This is done by using memory vectors in the proposed model. The second one is that instead of getting just a single input from the image modality, low-level and high-level image relationships lie inside the captioning sentence generated by a multi-layer architecture. This is accomplished by a learned gating mechanism that weights multi-level contributions at every stage. Suggested architecture can be seen in Figure 1.

II. Approach

A. Meshed-Memory Transformer

M2 transformer has two parts: encoder and decoder. In each layer of encoder, image regions coming from an input image are processed and relationships among them are inferred. In the decoder, the output of each encoder layer is processed to create the output image caption word by word. The schematic for the M2 transformer architecture can be observed in Figure 2. The formula for the scaled dot-product attention can be found below:

$$Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^{\boldsymbol{T}}}{\sqrt{d}})\boldsymbol{V} \tag{1}$$

where $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ are a set of queries, keys, and values respectively and d is a scaling factor.

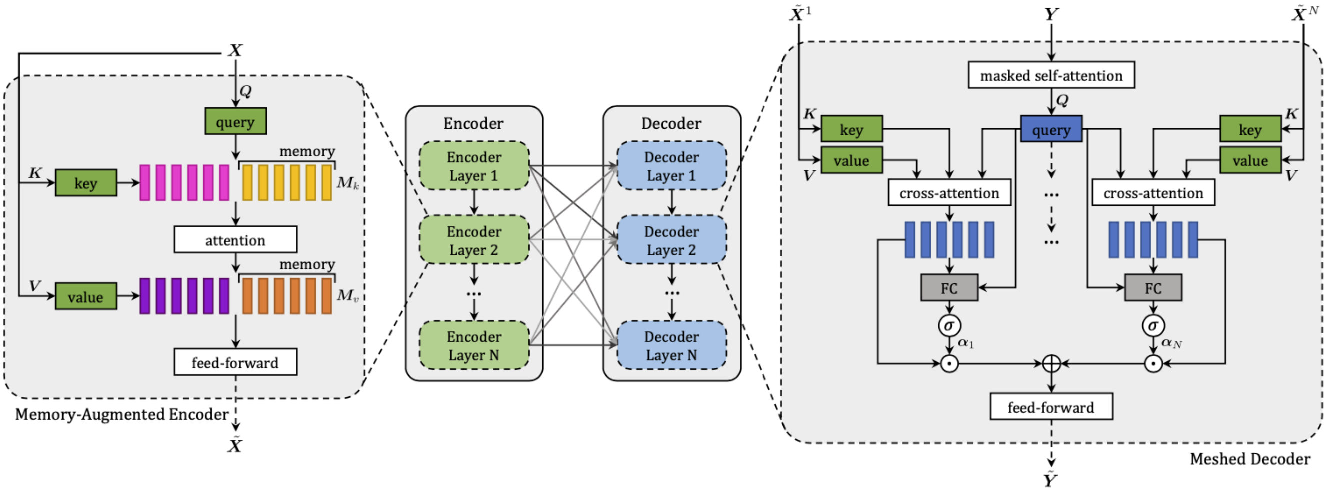

Fig. 2. M2 Transformer Architecture. It is composed of a stack of memory-augmented encoding layers, which encodes multi-level visual relationships with a prior knowledge, and a stack of decoder layers, responsible for generating textual tokens [1].

B. Memory Augmented Encoder

$\boldsymbol{X}$ is a set of image regions obtained from an input image. Self-attention operation can be defined as below where queries, keys, and values are extracted by linearly projecting the input features:

$$S(\boldsymbol{X}) = Attention(W_q\boldsymbol{X}, W_k\boldsymbol{X}, W_v\boldsymbol{X}) \tag{2}$$

where W_q, W_k, W_v are learnable weight matrices. The output of self-attention, $S(\boldsymbol{X})$, has the same cardinality as $\boldsymbol{X}$.

The self-attention operator encodes pairwise relationships among an input set of feature vectors or image regions that are required to understand the input image before captioning it. Since everything merely depends on pairwise similarities, self-attention can not represent whole prior knowledge on interactions/relationships between all image regions. To solve this limitation, a memory-augmented attention operator is proposed where the set of keys and values are extended with an extra term ($\boldsymbol{M_k}, \boldsymbol{M_v}$ respectively) which is intended to encode prior information. Newly updated attention operator can be formulated as:

$$M_{mem}(\boldsymbol{X}) = Attention(W_q\boldsymbol{X}, \boldsymbol{K}, \boldsymbol{V}) \tag{3}$$

$$\boldsymbol{K} = [W_k\boldsymbol{X}, \boldsymbol{M_k}] \tag{4}$$

$$\boldsymbol{V} = [W_v\boldsymbol{X}, \boldsymbol{M_v}] \tag{5}$$

where $\boldsymbol{M_k}$ and $\boldsymbol{M_v}$ are matrices of learnable weights, and $[.,.]$ denotes concatenation operation. As an intuition, memory-augmented attention can be able to retrieve learned information which does not exist in $\boldsymbol{X}$ by adding these learnable keys and values.

Feed-forward layer in each memory-augmented encoder is position-wisely implemented with two affine transformations with only one non-linearity. Details can be seen below:

$$F(\boldsymbol{X})_i = U\sigma(V\boldsymbol{X_i} + b) + c \tag{6}$$

where $\boldsymbol{X_i}$ represents the i-th vector of the input set, and $F(\boldsymbol{X})_i$ is the i-th output vector, $\sigma(.)$ denotes ReLU activation function, V and U are matrices of learnable weights, b and c are bias terms.

Two sub-parts of one encoder layer, memory-augmented attention and position-wise feed forward, are combined together with a residual connection and a layer normalization. Thus, the whole description for an encoding layer can be written as:

$$\boldsymbol{Z} = AddNorm(M_{mem}(\boldsymbol{X})) \tag{7}$$

$$\tilde{\boldsymbol{X}} = AddNorm(F(\boldsymbol{Z})) \tag{8}$$

where AddNorm represents the combination of a residual connection and a layer normalization.

At the end, N outputs of each encoding layer will be stacked together to form a multi-level output $\tilde{\boldsymbol{X}} = (\tilde{\boldsymbol{X}}^{\boldsymbol{1}}, ..., \tilde{\boldsymbol{X}}^{\boldsymbol{N}})$.

C. Meshed Decoder

The decoder part is responsible for generating the next tokens of the output image caption. The proposed decoder is conditioned on region encoding vectors and previously generated words. The decoder part will take the advantages of multi-level input image representations from the encoder part by using meshed-cross attention. The meshed attention operator connects $\boldsymbol{Y}$ to each element in $\tilde{\boldsymbol{X}}$ via gated cross- attentions, where $\boldsymbol{Y}$ is an input sequence of word embedding vectors and $\tilde{\boldsymbol{X}}$ is an another input coming from all encoding layers. In the general transformer models, the decoder only attends the last encoding layer. However, in the image captioning case where the nature of the multi-modal problem must be considered, it is tweaked a bit so that all encoding layers are attended to the decoder part by a cross-attention. As it is the case in encoder layers, the decoder layers also include a position-wise feed-forward layer. So, the formal definition of meshed attention operator and the structure of the encoder layer can be seen in Equations below:

$$M_{mesh}(\tilde{\boldsymbol{X}},\boldsymbol{Y}) = \sum_{i=1}^{N} \boldsymbol{\alpha_i} \odot C(\tilde{\boldsymbol{X}}^{\boldsymbol{i}},\boldsymbol{Y}) \tag{9}$$

$$C(\tilde{\boldsymbol{X}}^{\boldsymbol{i}},\boldsymbol{Y}) = Attention(W_q\boldsymbol{Y}, W_k\tilde{\boldsymbol{X}}^{\boldsymbol{i}}, W_v\tilde{\boldsymbol{X}}^{\boldsymbol{i}}) \tag{10}$$

$$\alpha_i = \sigma(W_i[\boldsymbol{Y}, C(\tilde{\boldsymbol{X}}^{\boldsymbol{i}},\boldsymbol{Y})] + b_i) \tag{11}$$

$$\boldsymbol{Z} = AddNorm(M_{mesh}(\tilde{\boldsymbol{X}}, AddNorm(S_{mask}(\boldsymbol{Y})))) \tag{12}$$

$$\tilde{\boldsymbol{Y}} = AddNorm(F(\boldsymbol{Z})) \tag{13}$$

where $C(.,.)$ means cross-attention for the encoder-decoder, $[.,.]$ is concatenation operations, σ is the sigmoid activation function, W_i is a learnable weight matrix, b_i is a bias term, $\boldsymbol{Y}$ denotes the input sequence of vectors, S_{mask} represents a masked self-attention over time.

D. Details of Training

Firstly, the cross-entropy loss is used for the training. Then, a type of reinforcement learning which is Self-Critical Sequence Sequence Training (SCST) [4] is used for finetuning the sequence generation process by using beam search [5].

III. Dataset & Results

MSCOCO [6] dataset which has more than 120000 iamges is used in their experiments to evaluate the suggested captioning model. There are at least 5 sentence descriptions for every image in this dataset. They use Karpathy split [7] which is extensively preferred for reporting results in previous works. For both the validation and test phases with MSCOCO dataset, 5000 images are used. For the evaluation of the caption model, following metrics are used: CIDEr [8], METEOR [9], SPICE [10], ROUGE [11], and BLEU [12].

A pre-trained Faster-RCNN [13] with ResNet-101 [14] model on Visual Genome [15] dataset is applied to extract a 2048-dimensional bottom-up feature vectors of input images. The original dimension of the feature vectors is 2048. But, they are linearly projected to a new space with the dimension of $d = 512$ in the paper. One hot vectors are used to represent words. Sinusoidal positional encodings [2] is implemented for the positions of the words. The number of heads is 8. The total number of memory vectors is 40. Drop-out of 0.9 is applied at the end of each encoding and decoding layer.

M2 transformer is trained with a mini-batch size of 50, ADAM [16] optimizer, and a beam size of 5. Fort the cross-entropy loss optimization, the learning rate schedule strategy in [2] is followed. For the optimization of the CIDEr-D score, SCST is applied at the fixed learning rate of 5e-6.

In Figure 3, they reports their results together with SCST [4], Up-Down [5], RFNet [17], Up-Down+HIP [18], GCN-LSTM [19], SGAE [20], ORT [21], and AoANet [22] on the Karpathy test split. As it can be seen in Figure 3, M2 Transformer model outperforms all SOTA models and becomes the new SOTA single-model for Karpathy split in terms of almost all evaluation metrics except ROUGE.

In Figure 4, they report the performance of their ensembled with 2 and 4 models trained on the official MSCOCO evaluation server. At the time of submission, around the earlier 2020, they take the lead in the leader board by outperforming all other submissions on the test server in terms of all metrics.

	B-1	B-4	M	R	C	S
SCST [33]	-	34.2	26.7	55.7	114.0	-
Up-Down [4]	79.8	36.3	27.7	56.9	120.1	21.4
RFNet [15]	79.1	36.5	27.7	57.3	121.9	21.2
Up-Down+HIP [49]	-	38.2	28.4	58.3	127.2	21.9
GCN-LSTM [48]	80.5	38.2	28.5	58.3	127.6	22.0
SGAE [46]	**80.8**	38.4	28.4	58.6	127.8	22.1
ORT [13]	80.5	38.6	28.7	58.4	128.3	**22.6**
AoANet [14]	80.2	38.9	**29.2**	**58.8**	129.8	22.4
$\mathcal{M}^2$ **Transformer**	**80.8**	**39.1**	**29.2**	58.6	**131.2**	**22.6**

Fig. 3. Single-model image captioning performance on the MSCOCO Karpathy test split. All values are reported as percentage (%).

	B-1	B-4	M	R	C	S
Ensemble/Fusion of 2 models						
GCN-LSTM [48]	80.9	38.3	28.6	58.5	128.7	22.1
SGAE [46]	81.0	39.0	28.4	58.9	129.1	22.2
ETA [24]	81.5	**39.9**	28.9	59.0	127.6	22.6
GCN-LSTM+HIP [49]	-	39.1	28.9	**59.2**	130.6	22.3
$\mathcal{M}^2$ **Transformer**	**81.6**	39.8	**29.5**	**59.2**	**133.2**	**23.1**
Ensemble/Fusion of 4 models						
SCST [33]	-	35.4	27.1	56.6	117.5	-
RFNet [15]	80.4	37.9	28.3	58.3	125.7	21.7
AoANet [14]	81.6	40.2	29.3	59.4	132.0	22.8
$\mathcal{M}^2$ **Transformer**	**82.0**	**40.5**	**29.7**	**59.5**	**134.5**	**23.5**

Fig. 4. Leaderboard of several models on the online MSCOCO test server.

References

[1] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara. Meshed-memory transformer for image captioning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10575–10584, 2020.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L. Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[3] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[4] Steven J. Rennie, E. Marcheret, Youssef Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195, 2017.

[5] Peter Anderson, X. He, C. Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018.

[6] Tsung-Yi Lin, M. Maire, Serge J. Belongie, J. Hays, P. Perona, D. Ramanan, Piotr Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. *ArXiv*, abs/1405.0312, 2014.

[7] A. Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:664–676, 2017.

[8] Ramakrishna Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, 2015.

[9] Michael J. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *WMT@ACL*, 2014.

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

[11] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *ACL 2004*, 2004.

[12] Kishore Papineni, S. Roukos, T. Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[13] Shaoqing Ren, Kaiming He, Ross B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[14] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[15] R. Krishna, Yuke Zhu, O. Groth, J. Johnson, Kenji Hata, J. Kravitz, Stephanie Chen, Yannis Kalantidis, L. Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2016.

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[17] W. Jiang, Lin Ma, Yu-Gang Jiang, W. Liu, and T. Zhang. Recurrent fusion network for image captioning. *ArXiv*, abs/1807.09986, 2018.

[18] Ting Yao, Yingwei Pan, Yehao Li, and T. Mei. Hierarchy parsing for image captioning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2621–2629, 2019.

[19] Ting Yao, Yingwei Pan, Yehao Li, and T. Mei. Exploring visual relationship for image captioning. In *ECCV*, 2018.

[20] X. Yang, Kaihua Tang, Hanwang Zhang, and J. Cai. Auto-encoding scene graphs for image captioning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10677–10686, 2019.

[21] Simao Herdade, Armin Kappeler, K. Boakye, and J. Soares. Image captioning: Transforming objects into words. In *NeurIPS*, 2019.

[22] Lun Huang, Wenmin Wang, J. Chen, and Xiao-Yong Wei. Attention on attention for image captioning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4633–4642, 2019.