Image Captioning Models using Bayesian Meshed Memory Transformers

Jeffrey Barahona, Ratna Darbhamalla, and Joshua Kalyanapu

***Abstract*—In this project, we improve on prior results by incorporating Self Critical Reinforcement Learning into our training schema of the Meshed Memory Transformer presented in the prior project. We also implement a smooth probabilistic form of attention by imposing a Weibull distribution on the weights generated by the attention operation. Additionally, we combine global a priori information gathered by the memory vectors described in the previous project with the contextual priors generated for Bayesian Attention. We find that using a probabilistic form of attention outperforms its deterministic counterparts, and we find that the aggregation of global a priori information into the prior of the Bayesian Attention can improve overall performance on image captioning tasks, especially on longer caption n-grams.**

[1]

I. Introduction

Image captioning bridges the gap between object detection and scene understanding. By forcing a model to not only find and identify salient regions and objects in an image but to also express the relationship between these entities in natural language, we improve the interpretability of these models and better approximate human level scene understanding in addition to developing a useful and interesting application. In the previous project, we approached this by developing a Meshed Memory transformer[2] which works by encoding a priori information into the attention operation and provides residual connections to the decoder from all layers of the encoder.

In that project, we had found that the transformer variation improves performance dramatically over the CNN-LSTM approach. In this project, we expand on that by converting the standard deterministic attention operation into a probabilistic interpretation and investigating the effects of combining attention with the a priori information collected by the memory vectors.

II. Data

We use the Flickr30K dataset to train and test the data. The data is split into training, validation, and test sets in a 75:15:10 proportion. Since [5] mentions that using or not using the Karpathy splits should not affect performance, we used a random split on the data for simplicity. Furthermore, we additionally preprocess the data by using Faster RCNN to extract salient regions and detection features[1]. These features effectively tokenize the image for use with the the model and simultaneously preprocesses the important regions of the image.

III. Methodology

A. Self Critical Sequence Training

Self-critical sequence training was developed as a technique to address one of the fundamental issues with teacher-forced training; that is, under teacher-forcing, a model trains only on the ground-truth whereas under evaluation, the model generates its next predicted word in the sequence contingent on the previous model outputs. Self-critical sequence training addresses this issue by reinterpreting the problem of sequence generation into a reinforcement learning framework[4]. In this case, the ”policies” of the model are the next possible words to be output, and the ”reward” to the network can be one of the metrics’ scores. For our reward function, we chose the equation 1

$$r = meteor(Y, \hat{Y}) - \frac{1}{n}\sum_{i=1}^{n} meteor(y_i, \hat{y}_i) \tag{1}$$

Then, the loss that the policy model is optimized against is

$$L_y(\hat{y}) = -\log(p_\theta(x))r$$

Using this, we were able to successfully have the model be exposed to its previous predictions during training. We largely based our implementation off of [4], with a few notable differences. First, the beam search we implemented for evaluation in Project 2 was incapable of being used in SCST as in the paper due to the fact that it was not a batch-wise implementation and did not work well with PyTorch. Instead, we returned to the greedy approach of the training in Projects 1 and 2 where instead of maintaining the 5 best sequences, we simply choose the highest-scoring word at each sequence step. This difference contributed to the difference in results between our project and the reference paper. Secondly, we used a different reward than in [4] due to the differences in how our models were implemented.

B. Bayesian Attention

We define the weights of the attention operation by

$$W = softmax\left(\frac{QK^T}{\sqrt{n_k}}\right)$$

[1] Jeffrey Barahona wrote and tested the Bayesian forms of attention with and without memory. Jeffrey Barahona also provided assistance for other tasks Joshua Kalyanapu wrote and tested the reinforcement learning algorithm. Ratna Darbhamalla wrote the visualization code and performed additional postprocessing of the dataset. All members contributed to the report

. Then, we define the attention operation

$$Attention(Q, K, V) = WV.$$

where Q, K, and V are the queries, keys, and values generated for the attention operation.

The attention expression is scaled by the dimensionality of the keys n_k. This form of attention is deterministic and can be described as the expectation of the values. Conceptually, we are computing which value vectors contribute most to the generation of a token in the predicted sequence. The benefit of this form of attention is that it lends itself to usage with gradient descent techniques for training, allowing for scaleable model architectures using attention, e.g. transformer models. There are probabilistic interpretations of attention that already exist, but these are non-differentiable and must be learned using reinforcement learning techniques like the REINFORCE algorithm[3]. Fan et. al. circumvents this by devising a variational approach to attention.

1) Constraints: There are some constraints that must be satisfied: the simplex constraint, a reparametrizable distribution to assume[3]. The simplex constraint means that the generated weights are non-negative and the sum of all the weights are unity. Our distribution must also be reparameterizable to remove the stochastic components from the path down the network backpropagation takes, allowing differentiability. For this implementation, we use the Weibull distribution to satisfy those constraints as described by Fan et al.[3]. We only approach a single variational distribution in this project for simplicity. The Weibull distribution pdf is defined by

$$p(S|k\lambda) = \frac{k}{\lambda^k} S^{k-1} e^{-(S/\lambda)^k}$$

where $S \sim Weibull(k, \lambda)$, $S \in_+$ with expectation $E[S] = \lambda\Gamma(1 + 1/k)$ and variance $Var(S) = \lambda^2(\Gamma(1 + 2/k) - (\Gamma(1 + 1/k))^2$. We may reparameterize this by setting $S = \lambda(-\log(1 - \epsilon))^{1/k}$, $\epsilon \sim Uniform(0, 1)$. Let γ be the Euler-Mascheroni constant and the KL divergence is defined by

$$\begin{aligned} KL(Weibull(k, \lambda)||Gamma(\alpha, \beta)) = \frac{\gamma\alpha}{k} - \alpha \log \lambda + \log k \\ + \beta\lambda\Gamma(1 + \frac{1}{k} - \gamma - 1 - \alpha \log \beta + \log \Gamma(\alpha) \end{aligned} \tag{2}$$

2) Contextual Prior: This form of Bayesian Attention makes use of the data in the form of the keys to construct the prior. This provides a more natural interpretation of the prior. Rather than using a fixed parameter to represent the knowledge of the image, Fan et. al. uses the data itself to encode global information about the image. In our implementation, we extend it further to include global information about the entire dataset through the use of memory vectors. We define the computation for computing the contextual priors as

$$\Psi = softmax(F_2(F_{NL}(F_1(K)))) \in^{n \times 1}$$

where F_{NL} is a nonlinear activation like ReLU and F_1 and F_2 are linear layers. When using the global prior information, the computation above is done normally, and the global information data is masked during the computation.

3) Amortized Variational Inference: We start by imposing a variational distribution, q_ϕ, over the attention weights, W. Then, we can consider a Bayesian model with prior $p_\eta(W)$ and likelihood $p_\theta(y|x, W)$. Now, we can learn the variational distribution by minmizing the KL divergence ,$KL(q_\theta(x)||p(W|x, y))$, between the posterior distribution given the data and the variational distribution of the attention weights.

In the case of the Weibull distribution, optimizing the parameters of the network is equivalent to optimizing the ELBO defined here as

$$\begin{aligned} L_\lambda(x, y, \epsilon) = \log p_\theta(y|x, S(\epsilon)) \\ - \lambda \sum_{l=1}^{L} KL(Weibull(k, \lambda)||Gamma(\alpha, \beta)). \end{aligned} \tag{3}$$

C. Hyperband Hyperparameter Optimization

To simplify hyperparameter search, we used the Hyperband algorithm to quickly iterate over the training data and Ray Tuning library is used for this purpose. The Adam optimizer was tuned with a dropout rate by sampling uniformly in the range of 0.1 to 0.2 and learning rate was selected from a loguniform distribution defined from 10^{-4} to 10^{-2}. The batch size was selected in the range of 8 to 64 and the number of encoder, decoder layers was sampled uniformly between 2 and 5. Hyperparameters for Bayesian Attention Model include the shape parameter k for Weibull distribution. However, k was left static as the model was tuned for the base version of the meshed memory transformer with standard attention without memory. This was done for consistency across trials. The best performing parameters were used for training subsequent models. The Async Successive Halving Algorithm (ASHA) scheduler was used to improve the overall efficiency of the hyperparameter optimization by terminating unpromising trials early. It provided better parallelism and avoided straggler issues during eliminations.

IV. Results and Discussion

From Table I, we see that the Bayesian formulations of attention perform better than their deterministic counterparts on all the language metrics. During the validation and testing stages, the posterior mean of the model predictions was calculated for inference. Interestingly enough, we also see that adding memory vectors improves performance over attention variations without memory vectors. On a superficial examination, it appears that the memory vectors encode global information about the dataset, allowing the attention mechanism to focus more on salient features rather than recapturing redundant information

Model/Metric	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	METEOR
Attention	49.61	29.44	17.45	10.72	37.11	46.62
Bayesian Attention	**55.68**	**33.05**	19.50	12.11	38.85	47.08
Attention with Memory	52.99	31.61	19.15	11.9	38.47	47.38
Bayesian Attention with Memory	53.97	32.35	**19.80**	**12.61**	**39.16**	**47.95**

TABLE I
Model Scores on Common Language Metrics

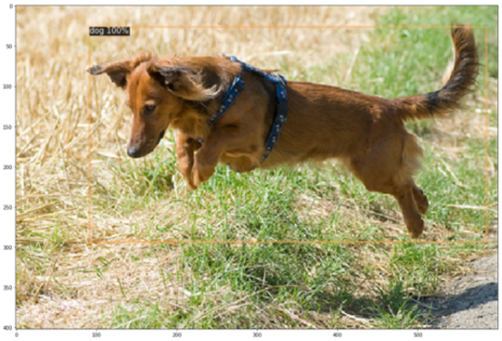

Fig. 1. The predicted caption for this image is ”a black and white dog runs through the grass”. One of the ground truth captions is ”a dog is jumping in the grass”

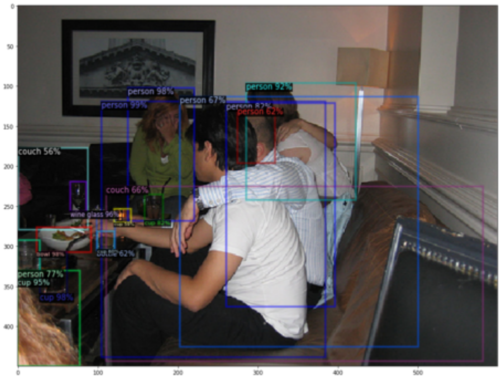

Fig. 2. The predicted caption for this image is ”a group of people are sitting at a table in a restaurant”. One of the ground truth captions is ”a group of people is gathered around a table of food with their arms around each other”

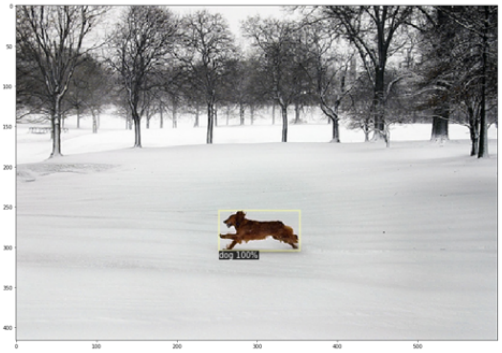

Fig. 3. The predicted caption for this image is ”two dogs are running through the snow”. One of the ground truth captions is ”a brown dog runs across a snowy field”

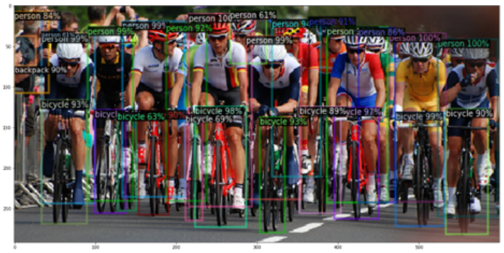

Fig. 4. The predicted caption for this image is ”a group of people ride bicycles down a street in front of a building”. One of the ground truth captions is ”a group of people riding a bicycle down a street”

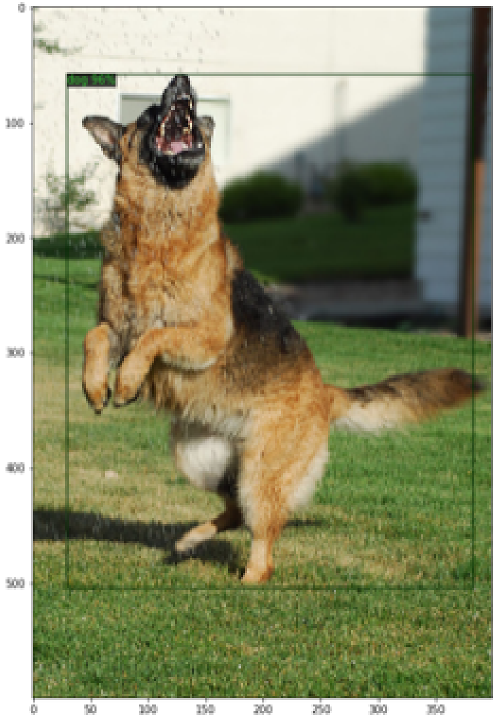

Fig. 5. The predicted caption for this image is ”a black and white dog runs through the grass”. One of the ground truth captions is ”a dog plays on the grass”

across samples. In Bayesian attention, the memory vectors improve performance in the metrics measuring overall sentence performance; we have better performance on higher n-gram BLEU, ROUGE, and METEOR scores. In the images provided, we visualize the captured salient features from bottom up attention to demonstrate what the model sees as its inputs.

V. Conclusion

In this project, we implemented a full pipeline for Bayesian Meshed Memory Transformers by deploying a bottom up attention pipeline to extract salient features from the images before further processing. Then, we performed hyperparameter optimization using the Hyperband algorithm to find a reasonable set of hyperparameters for the deterministic Meshed Memory Transformer without memory and reused those hyperparameters across the variants of the models tested. We attempted to configure a reinforcement learning routine to perform self critical sequence training, but we did not find a significant improvement to performance. We tested the effects of memory vectors on the testing performance of the transformer and found that adding memory vectors to the transformer significantly improves performance. Next, we reformulated the attention mechanism of the transformer from a Bayesian perspective, imposing a Weibull distribution on the attention weights by using the keys and queries as the distribution parameters. We found that the Bayesian variants of attention led to better performance. Finally, we incorporated memory vectors into the prior assumed by Bayesian attention, further improving performance, especially on longer matching n-grams.

References

[1] Peter Anderson et al. “Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering”. In: *arXiv:1707.07998 [cs]* (Mar. 14, 2018). arXiv: 1707.07998. URL: http://arxiv.org/abs/1707.07998 (visited on 03/12/2022).

[2] Marcella Cornia et al. “Meshed-Memory Transformer for Image Captioning”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA: IEEE, June 2020, pp. 10575–10584. ISBN: 978-1-72817-168-5. DOI: 10.1109/CVPR42600.2020.01059. URL: https://ieeexplore.ieee.org/document/9157222/ (visited on 02/08/2022).

[3] Xinjie Fan et al. “Bayesian Attention Modules”. In: (), p. 15.

[4] Steven J Rennie et al. “Self-critical sequence training for image captioning”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7008–7024.

[5] Kelvin Xu et al. “Show, Attend and Tell: Neural Image CaptionGeneration with Visual Attention”. In: (), p. 10.