Automatic image captioning using multi-task learning

Anna Fariha
afariha@cs.umass.edu

Abstract

Automatic image captioning refers to the problem of constructing natural language description of an image. This is an important problem with practical significance that involves two major artificial intelligence domains — computer vision and natural language processing. In this project, we used *multi-task learning* to solve the automatic image captioning problem. In the proposed multi-task learning setting, the primary task is to construct caption of an image and the *auxiliary task* is to recognize the *activities* in the image. The two tasks share a latent representation of images and it is empirically shown that introducing the auxiliary task helps improving the shared layer representation and thus improves the performance of the original task. The novelty in this project relies in incorporating the auxiliary task of activity recognition in a multi-task learning framework for solving the original task of generating captions of images. We evaluate the proposed multi-task learning model on publicly available benchmark Microsoft COCO dataset and the experiments show the effectiveness of the model.

1 Introduction

Automatic image captioning refers to the problem of constructing a natural language description of an image. This task is challenging than the image classification and object recognition task, because it not only requires detection of objects within the image, but also requires detection of their relationship, expression, and activity presented in the image. Furthermore, the perceived information must be translated to some human understandable natural language. The main obstacle is the task of detecting the salient visual information that comes naturally to human. An important application for automatic image captioning system is in aiding visually impaired persons by providing them information about the content of the image in natural language. Another application is in search engines where images can be searched by sentence fragments. Apart from the practical applications, image captioning requires the machine learning model to learn image understanding which is a significant computer vision challenge. The image captioning model can be further extended to video captioning which also has many practical applications including alert systems for enhancing security.

Key steps of image captioning task include extracting salient high level features from an image, detecting objects from those features, detecting salient visual information (relationship, interaction, expression, activity) involving those objects, and finally generating a natural language description as a sequence of words to express the content of an image. Some existing works [5, 10] address the image captioning problem by concatenating modules that solve these steps. More recent line of works [16, 15] aims to build an end to end system that uses Convolutional Neural Network (CNN) for salient feature detection and on top of that a Recurrent Neural Network (RNN) that generates sequential words to construct image captions. Several recent methods [18, 19] also proposed semantic attention based neural models for image captioning.

Our approach is different from the existing approaches since we model the image captioning problem as a multi-task learning problem with image activity detection as the auxiliary task. Unlike previous work [19], our approach does not explicitly fuse the semantic output to the RNN hidden layers, instead, through the auxiliary task of activity detection within an image, a bias is induced to the RNN.

In other words, the auxiliary task forces the shared layer to represent certain features significant for image captioning that might have been ignored if the auxiliary task was not included. To the best of our knowledge, there exists no such multi-task learning framework for image captioning in the existing literature.

Attention based approaches [18, 19] are designed to put attention on objects, rather than the activities. Our approach is also novel in a sense that it tries to capture the features of an image that can represent certain activities. This is certainly more challenging than simple object detection task. Since the caption of an image usually describes some event happening in that image, the features that represent activities are of primary importance. In this project, we aim to develop a system that will solve two tasks in parallel in a multi-task learning framework. The shared layer is responsible for extracting intermediate salient features from an image and forward those features as input to both tasks — primary task of caption generation and auxiliary task of activity detection. Formally, the primary task is: given an image I and a predefined set of words V, generate an ordered subset of words $C =< c_1, c_2, ..., c_n >$, where $\forall_{1 \leq i \leq n} \quad c_i \in V$ and the sentence constructed from the sequence of words describes the content of the image. The auxiliary task is a simple multi-label multi-class classification problem for activity detection. Our proposed model uses a CNN to generate high level features from an image that are used as input to the shared feature extractor. The output of this shared layer is used as input features to both learning tasks.

In our experiment, we measure performance under different settings of the multi-task learning framework. BLEU score [13] is used as similarity metric to measure similarity between ground truth captions and generated captions. The experiments show that incorporating the auxiliary task improves caption prediction. The rest of the report is organized as follows: Section 2 describes the related works and contrasts our proposed approach with them. In Section 3, we describe our proposed framework. We present the dataset, experiment design, and results in Sections 4, 5, and 6, respectively and conclude the report in Section 7.

2 Related Work

Recent methods for object detection and recognition have significantly motivated the image captioning problem. One of the early non-neural approaches on describing images was done by Farhadi et al. [5] who proposed a method based on multi-label Markov random field involving an intermediate meaning space to generate short descriptive sentences from images. The proposed approach works in two phases — mapping the image to a meaning space in the format of <object, action, scene>, and mapping the meaning space to a sentence using some predefined templates. Kulkarni et al. [10] also proposed a template based text generation method for describing images. The limitation in this approach is that it only describes the relative position of various objects detected in an image. Hence, the method is only capable of capturing the spatial relationship among objects. Being a neural model, our approach has much more capacity than these non-neural models and is able to learn from the provided captions to produce versatile captions.

More recent line of works on image captioning involves a deep convolutional neural network layer for high level feature extraction from images. Vinyals et al. [15, 16] proposed Neural Image Caption (NIC) — a generative model based on deep recurrent architecture that maximizes the likelihood of generating the target caption given an input image. In NIC, CNN is used as an image encoder to produce a fixed length feature vector to represent high level features of the input images. The idea is to chop off the final output layer of a CNN based image classification task and use the output of the last hidden layer as input to the caption generator recurrent network for sequence modeling. NIC combines pre-trained sub-networks for vision and language models. Our approach shares some similarity in the lower level of NIC, however, NIC model is an end-to-end single objective task, where our approach involves multi-task learning. Moreover, NIC makes use of language model that is trained from external corpora where we do not assume such external knowledge.

We use a slightly modified version of NIC as our baseline approach. We describe NIC model in detail here. NIC uses a probabilistic model to maximize the probability of the correct caption. The formulation is provided in Equation 1. Here θ denotes the model parameters, I is the input image and S is the corresponding target caption. The tuple (I, S) denotes a training image and caption pair.

$$\theta^* = \arg\max_{\theta} \sum_{(I,S)} \log P(S|I;\theta) \tag{1}$$

Since S denotes a sequence of words $< S_0, S_1, ..., S_N >$ of length N, the joint probability $P(S|I;\theta)$ can be expressed using chain rule as shown in Equation 2.

$$\log P(S|I;\theta) = \sum_{t=1}^{t=N} \log P(S_t|I, S_0, S_1, ..., S_{t-1};\theta) \tag{2}$$

For image representation NIC uses CNN and for word representation word embedding model is used. For modeling the structure in Equation 2, NIC uses Long Short Term Memory (LSTM) [6], which is very suitable for sequence prediction. For training, the output of t-th LSTM is fed as the input to $(t+1)$-th LSTM. For prediction, the authors mention two approaches — (1) Sampling words from the probability distribution obtained in the LSTM output layer at each time step, and (2) BeamSearch that generates top k sentences at time $t-1$ to generate sentences at time t and keeps top-k among them. The modified approach we use as the baseline in this report uses the ground truth captions instead of previous time step's LSTM output as input to the LSTM at next time step.

Karpathy et al. [8, 9] proposed a model that is built with a combination of CNN, bi-directional RNN, and a structured objective. Unlike other approaches that limit the image description to a sentence, the proposed approach aims at generating dense description of images. The key idea in their approach is to align sentence fragments in image description to corresponding image regions to better learn visual information in images. For text generation from input images, they introduced a multimodal RNN architecture. This model is particularly useful to generate description of previously unseen combination of known image regions. Instead of image regions, our proposed approach focuses on activity of an image that is overlooked by many state-of-the art approaches.

Xu et al. [18] and You et al. [19] proposed attention based image captioning models for better describing content of images. The approach in [18] focuses on spatial attention and the authors have shown visually how the trained model can fix its gaze on salient objects while generating sentences as captions. The model consists of CNN for extracting salient image features and RNN for learning word sequence generation. The key idea in this approach is incorporating attention that mimics human visual system while constructing the caption. Unlike the static features that are generated at once from the CNN, the attention allows features to be produced dynamically. Since the RNN is capable of accepting sequential inputs, the dynamic salient features can improve sentence generation as the RNN can attend different objects during different temporal steps. The approach described in [19] can attend multiple salient objects with differently assigned weights and dynamically switch attention among objects. A shortcoming of such attention based models is that they mostly attend to objects, but not activities within an image. Since objects will dominate activities in an image, without explicit supervision, it is unlikely that the salient features will represent activities. However, activities play an important role when it comes to caption generation. Our proposed approach can be augmented to such attention based models to attend activities.

Chen et al. [3] proposed a method for bidirectional mapping between images and captions using RNN. The model aims at both generating description from input image and reconstructing visual features from given description. Johnson et al. [7] proposed a dense captioning model that both localizes and describes salient image regions via producing rich annotations of objects. Lu et al. [12] proposed a modified version of attention based captioning where the model is not forced to always attend the visual features. There exists several other notable works in image captioning [1, 14, 11, 17, 4] that include attention based mechanism, nearest neighbor based approach, bidirectional LSTM etc. Our proposed approach is novel in two ways — (1) it incorporates multi-task learning with a relevant and well suited auxiliary task of activity detection, and (2) it addresses the issue of extracting salient features representing activities within an image that was not addressed by any of the previous works.

3 Methodology

In this project, we propose a multi-task learning framework to solve the image caption construction problem. The model aims at solving two different objectives — activity detection and caption generation. Both of the activities share a common layer. Our primary objective is to learn caption generation and activity detection is an auxiliary task. When the model tries to solve the auxiliary task, which is related to the primary task, it learns to generalize better to perform the original task. This happens due to the fact that the auxiliary task requires domain specific features which is often helpful to represent high level related features for the original task. Since both of the tasks share the learned

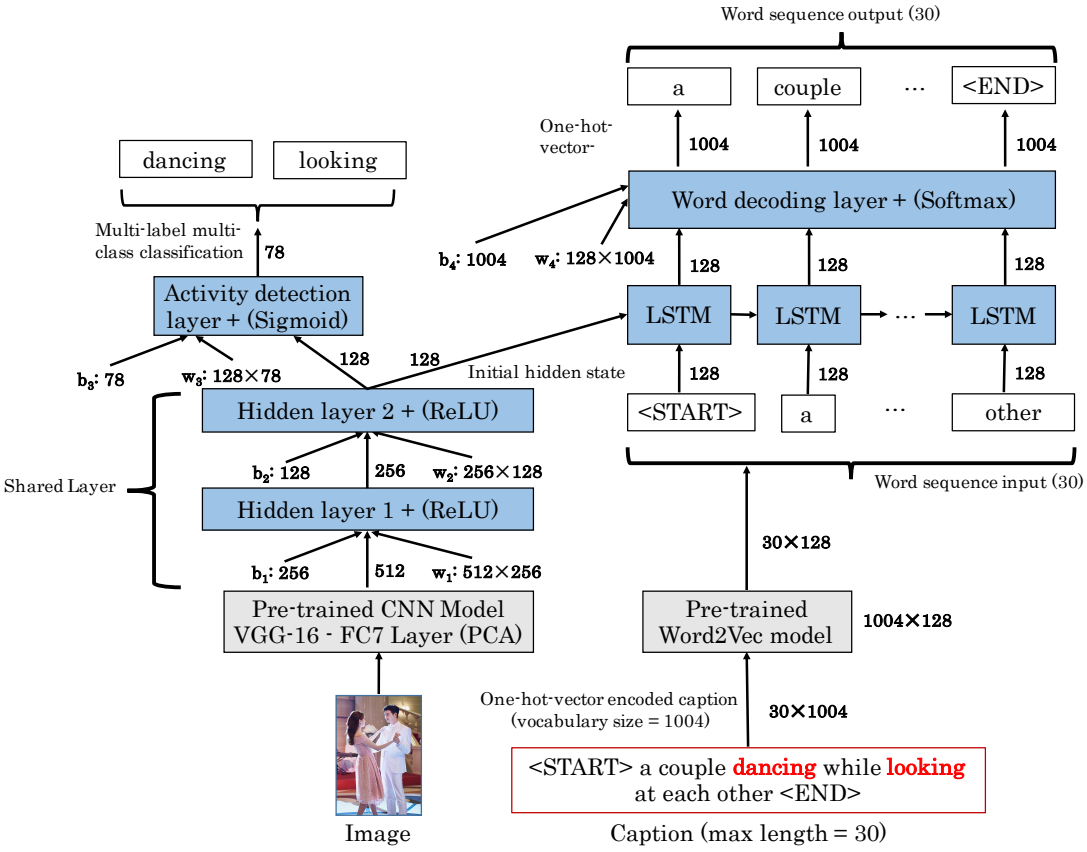

Figure 1: Multi-task learning architecture for image captioning

features, learning useful features for solving the auxiliary task can in turn help to solve the original task. Through training the model to solve the auxiliary task, we are "forcing" the model to solve the original task by making use of the features more suitable for solving the auxiliary task. This can be compared with the regularization technique. When we regularize a machine learning model, we force the model to learn weights under certain constraints (e.g., minimize the L2 norm of the weight vector). Similarly, in multi-task learning with auxiliary task, we force the model to learn to make use of the features better suitable for the auxiliary task.

The multi-task learning framework is shown in Figure 1. The framework can be divided into 5 components — (1) CNN layer for learning rich features from raw images, (2) Shared layer for learning features for optimizing two parallel tasks, (3) Word embedding layer for learning how to represent words in terms of feature vectors, (4) Multi-label multi-class classification module for solving the activity detection task, and (5) Long short term memory (LSTM) for caption generation. We discuss each of the components below.

3.1 CNN Layer

For generating high level features from the raw image, we have used VGG-16 — a deep pre-trained CNN model. We have extracted features from fc7 layer of that model. Te details is provided in Section 4. Using principal component analysis, a compressed representation of the high level features of dimension 512 is used. We did not take output of deeper layer since they tend to lose information that are not relevant for the task they were originally designed to solve. Most tasks tend to solve the image classification or object detection problem and do not focus on the activity expressed in the image. Hence, we use shared layers to learn features representing activities inside an image and we discuss that next.

3.2 Shared Layers

The high level features extracted from the VGG-16 fc7 layer is a vector of 512 elements and is fed as the input to the hidden layer 1. This hidden layer converts the input to a vector of 256 elements. ReLU non-linearity is applied to the output of this hidden layer and fed as input to the second hidden layer, hidden layer 2. This layer further compresses the 256 element vector to 128 element vector. Another ReLU non-linearity is applied to the hidden layer 2 output. Up to this point, the learning is shared among two parallel tasks. This 128 element feature vector is fed as input to both the activity detection module and the caption generation module. The reason for designing these two shared hidden layers is to make the model learn salient features for activity detection so that it can aid in caption generation. The shared layers aim at minimizing loss from both of the learning tasks in the proposed multi-task learning setting.

3.3 Word Embedding

For encoding the captions, we have produced a one-hot-vector representation of each word within the vocabulary of 1004 words. We allowed the caption length to be at most 30. If the caption length is less than 30, a special word <NULL> is assigned to fill up the empty slots. We also used two special words <START> and <END> for indicating the start and end of a caption, respectively. For each caption, we generated a 30×1004 matrix to represent the caption. Each row in that matrix represents one word and exactly one column in each row is set to 1. For vector representation of words, we have used Word2Vec[1]. We first trained the Word2Vec model with all captions available in the dataset. This results in a mapping between each word in the vocabulary to a feature vector of length 128. Using the weight vectors learned from the word embedding model, we can map the one-hot-vector of 1004 elements to a vector of 128 elements. Since our caption is encoded by a 30×1004 matrix, the word embedding layer converts it to a 30×128 matrix.

3.4 Multi-label Multi-class Classification for Activity Detection

The problem of activity detection from images is a multi-class problem since there are a number of different possible activities that an image can represent. We have found 78 activities from the captions of the dataset. This is also a multi-label problem since one image can represent multiple activities. For example, in Figure 1, the sample image represents two activities — dancing and looking. The activity detection layer takes a 128 element vector as its input and produces a 78 element vector as its output. Each element of this output vector can be interpreted as class score of the activities. Since it is a multi-label problem, we use sigmoid logistic function to map the scores to probability values for better interpretability. For activity prediction, we use a threshold of 0.5 to decide whether a certain activity is represented by an image or not. The pipeline for activity detection from input image I is expressed using Equations 3 – 7. We compute the loss using binary cross entropy loss as shown in Equation 8 where y^A is a 0-1 vector with 1 at correct class labels and $\hat{y}^A$ is a vector with probability scores for each activity label. We use C to represent the set of all possible activity labels.

$$X_1 = CNN(I) \tag{3}$$

$$H_1 = ReLU(w_1 X_1 + b_1) \tag{4}$$

$$H_2 = ReLU(w_2 H_1 + b_2) \tag{5}$$

$$H_3 = w_3 H_2 + b_3 \tag{6}$$

$$\hat{y}^A = \frac{1}{1 + e^{-H_3}} \tag{7}$$

$$loss(\hat{y}^A, y^A) = -\frac{1}{|C|} \sum_{c \in C} \Big(y_c^A \log \hat{y}_c^A + (1 - y_c^A) \log(1 - \hat{y}_c^A) \Big) \tag{8}$$

3.5 LSTM for Caption Generation

We have used LSTM for caption generation since it is a sequence prediction task and LSTM is very well suited for it. The input of each time step of LSTM is the 128 element feature vector for word at the previous time step. For training, we use the words from ground truth captions. However,

[1] code.google.com/archive/p/word2vec/

for the actual prediction task, we will not have any ground truth caption available. Hence, in that case, output word from the LSTM at previous time step is fed as input to the LSTM at next time step. The initial hidden layer of the LSTM at first time step consists of the 128 element feature vector representing the image. Note that, this feature vector is shared with the activity detection task. LSTM uses the hidden state and input to compute cell state and propagates both the hidden state and the cell state to the LSTM at next time step. The output of the LSTM at each time step is decoded using a word decoding layer that maps 128 element vector to a 1004 element vector. Since this is a single-label multi-class problem (i.e., at each time step, we expect exactly one word), we use softmax function to obtain normalized probability on the 1004 element vector. Then we pick the word with maximum probability as the predicted word. The computation inside LSTM at time step t is presented in Equations 9 – 16. S_c denotes the cell state and S_h denotes the hidden state of the LSTM. A_i, A_f, A_g, and A_o denote the input, forget, block, and output gates respectively.

$$X_2 = \text{WordEmbedding}(Caption) \tag{9}$$

$$S_h^{-1} = H_2 \tag{10}$$

$$A_i^t = \sigma(W_{ii}X_2^t + b_{ii} + W_{hi}S_h^{t-1} + b_{hi}) \tag{11}$$

$$A_f^t = \sigma(W_{if}X_2^t + b_{if} + W_{hf}S_h^{t-1} + b_{hf}) \tag{12}$$

$$A_g^t = \tanh(W_{ig}X_2^t + b_{ig} + W_{hg}S_h^{t-1} + b_{hg}) \tag{13}$$

$$A_o^t = \sigma(W_{io}X_2^t + b_{io} + W_{ho}S_h^{t-1} + b_{ho}) \tag{14}$$

$$S_c^t = A_f^t \times S_c^t + A_i^t \times A_g^t \tag{15}$$

$$S_h^t = A_o^t \times \tanh(S_c^t) \tag{16}$$

We perform word decoding using Equation 17. The loss function for caption generation is defined using Equation 18 using cross entropy loss. Here y^C is the one-hot-vector representation of words for ground truth caption and $\hat{y}^C$ denotes the normalized probability score for each word in the vocalbulary V.

$$\hat{y}^C = \text{softmax}(w_4 A_o + b_4) \tag{17}$$

$$loss(\hat{y}^C, y^C) = -\frac{1}{|V|}\sum_{v \in V} y_v^C \log(\hat{y}_v^C) \tag{18}$$

3.6 Multi-Task Learning

We use a trade-off parameter α to decide on relative weight on the losses obtained by two different tasks. The final loss is computed using Equation 19. We have used Adam as the optimizer. We have used gypsum cluster for the experiments and our implementation is written using PyTorch [2] in Python 3.6.

$$loss = \alpha \times loss(\hat{y}^C, y^C) + (1 - \alpha) \times loss(\hat{y}^A, y^A) \tag{19}$$

In the next sections, we discuss the dataset, experiment design and present the experimental results.

4 Dataset

The dataset we used is Microsoft COCO[3] dataset. It was collected using Amazon's Mechanical Turk[4]. The data collection procedure is described by Chen et al. [2]. The train set consists of 82K images where the validation set consists of 40K images. Each image is associated with around 5 captions. Each image caption is expected to have at least 8 words according to the data collection requirement. The vocabulary consists of 1004 English words. Instead of using the raw images as features, we have used VGG-16[5], a 16 layer deep pre-trained CNN model, that was trained on ImageNet[6] dataset. We have extracted features from fc7 layer of that model. Using principal component analysis, a

[2] http://pytorch.org/

[3] http://cocodataset.org/

[4] https://www.mturk.com/

[5] www.robots.ox.ac.uk/~vgg/research/very_deep/

[6] www.image-net.org

compressed representation of the high level features of dimension `512` is used. The feature is real valued. We have picked this dataset because it is a very well known benchmark for image captioning task. There exists a number of existing works that evaluate their approach on this dataset. We have identified words with the suffix "`ing`" as activities mentioned in captions. Excluding few non-activity words (e.g., something, during etc.), we have found `78` activity classes among the dataset.

5 Experiments

We evaluated the proposed model using different experiments on the COCO dataset. To measure the quality of the captions generated by the system, we have used BLEU [13]. We have chosen BLEU since it shows high correlation with human judgment and is one of the most popular and inexpensive sentence similarity metric. We experimented on different values of α within the range `[0,1]` to understand the effect of introducing auxiliary task. For $\alpha = 0.0$, the model ignores the caption generation objective and for $\alpha = 1.0$, the model ignores the activity detection objective. Therefore, we can interpret $\alpha = 1.0$ model as the baseline where the system is not learning from the auxiliary task of activity detection. We have investigated several different combinations of hyper-parameters for the optimizer. For learning rate we have tried the values — `[1e-5,1e-4,1e-3,1e-2,1e-1]` and for weight decay we have tried the values — `[1e-5,1e-4,1e-3,1e-2,1e-1]`. We found the best set of hyper parameters as follows: learning rate = `1e-4`, weight decay = `1e-4`. We expect the model to perform better with $0 < \alpha < 1$, i.e., when it attempts to learn from both tasks.

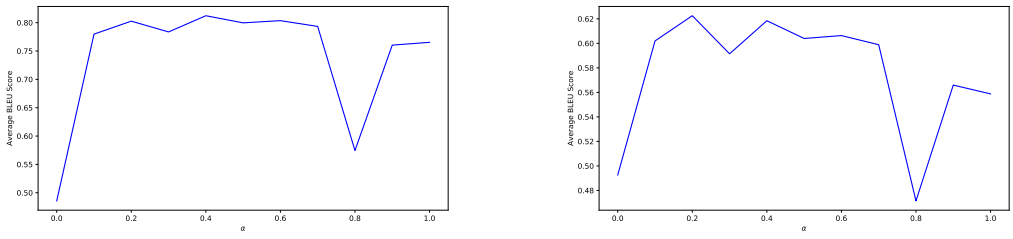

Figure 2: Left: average BLEU score for different values of α on train set with the ground truth captions as input to LSTM, Right: average BLEU score for different values of α on train set with the predicted captions as input to LSTM

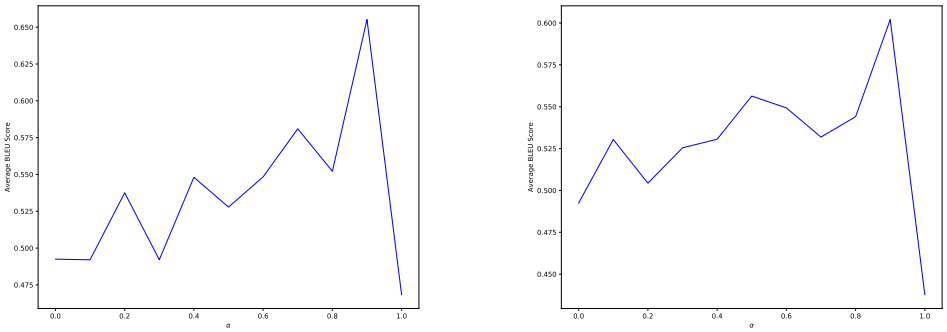

Figure 3: Left: average BLEU score for different values of α on validation set with the ground truth captions as input to LSTM, Right: average BLEU score for different values of α on validation set with the predicted captions as input to LSTM

6 Results

Figure 2 depicts the average BLEU score of the captions generated by the proposed multi-task learning model for different values of α on train set. We can see that for $\alpha \in [0.1, 0.7]$, the model performs better than the baseline model with $\alpha = 1$. Figure 3 shows similar results for validation

set. We have predicted the captions in two ways. In both Figures, the left one denotes the case where prediction was assisted by the ground truth caption, and the right one represents prediction where the ground truth caption was not provided. In the latter case, output generated from the previous LSTM was used as input to the next LSTM.

For understanding how performance is affected per activity, we have computed average BLEU score under few activity classes. Figure 4 shows that performance is better when the image has activities associated with it. Many images does not have any activity associated with it and that causes lower average BLUE score.

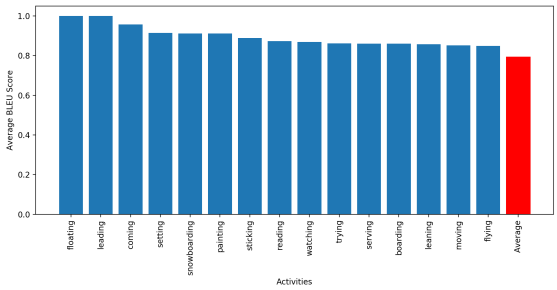

Figure 4: BLEU score grouped by activities vs. overall average BLEU score

Finally, we present few sample caption generation performance of our system in Figure 5. The ground truth and predicted captions are presented along with correctly predicted activity labels.

Figure 5: Ground truth and predicted captions with ground truth and predicted activity labels

7 Discussion and Conclusions

We did not implement several extensions to improve the model like batch normalization, drop-out, attention etc. However, our hypothesis was that under the same setting, multi-task learning with suitable auxiliary task should outperform the model with single objective of caption generation. The average BLEU score was pretty low compared to the state of the art approaches. Also, the performance on validation set was not good and it indicates that the system lacks generalization. However, techniques used to improve the baseline model is also applicable to the multi-task learning setting and in future we plan to investigate that. The takeaway from this project is — multi-task learning can improve generalized performance if the auxiliary task is related to the original task. The shared layer can learn to represent features more suitable for the original task while attempting to learn the auxiliary task.

Acknowledgments

In the project implementation, few utility functions provided by COMPSCI-682 course[7] staff were used.

[7] `https://compsci682.github.io/index.html`

References

[1] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and VQA. *CoRR*, abs/1707.07998, 2017.

[2] X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325, 2015.

[3] X. Chen and C. L. Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 2422–2431, 2015.

[4] J. Devlin, S. Gupta, R. Girshick, M. Mitchell, and C. L. Zitnick. Exploring nearest neighbor approaches for image captioning. *arXiv preprint arXiv:1505.04467*, 2015.

[5] A. Farhadi, S. M. M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. A. Forsyth. Every picture tells a story: Generating sentences from images. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, pages 15–29, 2010.

[6] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[7] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4565–4574, 2016.

[8] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):664–676, 2017.

[9] A. Karpathy and F. Li. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3128–3137, 2015.

[10] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12):2891–2903, 2013.

[11] C. Liu, J. Mao, F. Sha, and A. L. Yuille. Attention correctness in neural image captioning. In *AAAI*, pages 4176–4182, 2017.

[12] J. Lu, C. Xiong, D. Parikh, and R. Socher. Knowing when to look: Adaptive attention via A visual sentinel for image captioning. *CoRR*, abs/1612.01887, 2016.

[13] 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, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[14] K. Tran, X. He, L. Zhang, J. Sun, C. Carapcea, C. Thrasher, C. Buehler, and C. Sienkiewicz. Rich image captioning in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.

[15] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3156–3164, 2015.

[16] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):652–663, April 2017.

[17] C. Wang, H. Yang, C. Bartz, and C. Meinel. Image captioning with deep bidirectional lstms. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 988–997. ACM, 2016.

[18] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 2048–2057, 2015.

[19] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.