Detection-Aided Adaptive Attention Model for Image Captioning

Xudong Sun
UC San Diego
A53247312
xus022@eng.ucsd.edu

Haocheng Li
UC San Diego
A53240824
hal461@eng.ucsd.edu

Yi Luo
UC San Diego
A53239037
yil901@eng.ucsd.edu

Feihua Fang
UC San Diego
A53238573
f5fang@eng.ucsd.edu

Abstract

Image captioning is a popular research topic in computer vision and natural language processing and has wide applications. The usual architecture is encoder-decoder framework. However, to attain better image captioning model, motivated by two state-of-art works, we introduce detection-aided features into adaptive attention model and propose to employ two attention units to aggregate visual information from CNN and detection-aided features. Besides, we changed the backbone of encoder model into DenseNet and fine tune the model. In experiments, we added our improving strategies step by step on the basic model and the experiments results demonstrates the effectiveness of our approaches. Detailed discussions are also conducted to illustrate our results.

1 Introduction

Image captioning has become an interesting research topic in recent years, thanks to the fast development of deep learning. The task of image captioning is to generate a human-readable sentence given an input image. Convolutional Neural Networks (CNNs) has been widely used in various image-related tasks, such as classification [1], detection [2][3] and segmentation [4], and have demonstrated the capability of encoding the information of the image. Recurrent Neural Networks (RNNs), on the other hand, are also frequently applied to sequence-based tasks, such as language modeling [5] and machine translation [6]. Given this big picture, it becomes straight-forward to apply the encoder-decoder framework [7] for image captioning tasks, where CNNs are used for encoding image information, and RNNs for generating descriptions based on the encoding vectors.

Recently many researchers introduce attention mechanism to highlight some image regions when generating a word and achieve some amazing results. Further, sometimes decoder does not need to attend to the image when predicting some words such as functional words such as 'the', 'of' and some fixed phrases such as 'cell phone'. To handle this problem, [8] proposes an adaptive attention model which could automatically decide

when and where to attend to the image or information during sequential words generation. This adaptive attention mechanism is novel and reasonable but as for spatial image features, the model just adopts the spatial feature outputs of the last convolutional layer in ResNet[9] and does not exploit more effective feature signals from images. Thus, we explore to enrich useful information extracted from images and during the paper investigation, we get inspired by [10] 's work. They proposed a top-down attention LSTM utilizing Faster RCNN [2] detection result as input image features and language LSTM architecture.

Motivated by the two state-of-art models, we incorporate detection-aided features into adaptive attention model in image captioning and also creatively introduce location information of detected objects into image features. To sufficiently incorporate both abstract CNN features and detection-aided features with location, we adjusted attention mechanism accordingly. Finally, we also changed the backbone CNN models into DenseNet-201 to improve the encoder model and visualize attention areas during caption generation.

In experiments, we compared our model with simplified adaptive attention model[8] without beam search strategy. It turns out that by introducing detection-aided features with localization, our model enhances four criterion greatly. Also, with changing the backbone and fine tuning the model, our model finally achieves 0.6343 in BLEU1, 0.4575 in ROUGE and 0.4614 in CIDEr.

Overall, we propose a detection-aided adaptive attention encoder-decoder framework for image captioning. The main contributions of this project are that

- Incorporate detected-aided features into adaptive attention models.
- Propose to employ location information in detected-aided features
- Design two attention units for abstract image features from CNN and detection-aided features with localization.
- Change the backbone of image model into DenseNet with fine tuning
- Use Adamax to optimize the model

2 Related Work

Image captioning refers to automatically generate caption describing the content of an image. Most of the recent works applied the encoder-decoder framework. The network is separated into two sub-nets, one for encoding the information from the image and another utilizing the recurrent network to decode the current information to predict a meaningful sentence. Some well known trials e.g. show-and-tell [7] have explored this architecture sufficiently and gained relative good results compared with traditional NLP methods.

2.1 Image features extraction

In order to look into more details, people developed the attention mechanism which was treated as weighted spatial features on an image. The attention makes use of the last feature map of the backbone network and partially summarize the useful information inside different location.

One of the largest drawback is that it strongly restricted the information exchange between encoder and the decoder network, hence if an error-prone perturbation occurs in one of them the network performance will fall a lot. Some of the previous work are dedicated to solve the pitfall. One reorganized the image information and made it more like the visual signal accepted by human. Anderson et al[10] used bottom up and top down methods to reinforce the encoding subnetwork where traditional detection network like Faster-RCNN contributes to bottom up visual features. In such a way, the framework is capable to predict from the region of interest(ROI) instead of the fixed spacial locations. Another trail developed by Lu et al [8] is by using the visual sentinel which adds a gate onto the visual output. Such mechanism allows the network to

learn the mixture ratio between the information gained from the previous words and the image. On the other hand, we can not only encode the spatial-wise attention but also add attention mechanism into the channel selection. [11] chooses to implement two-branch attention network with one focusing on spatial information and another operating on the channel weighting. Their result largely outperforms the show-and-tell network.

2.2 Caption generation

As our candidate sentence follows a generative model, it makes more sense to predict according word joint distribution. One trick to achieve that is called beam search. It is a simple heuristic algorithm by looking ahead for more steps and keeps the best choices so far in the candidate pool. However, some recent paper [12] tried to use sampling techniques to obtain more precise distribution. The job aims to deploying a decision-making framework like the deep reinforcement learning. Given the natural language process metric as the final reward the decoder learns to modify its policy. The learning strategy is called actor-critic that allows the network to largely learn from each mistakes it makes on generating the intermediate words.

Skeleton is a fancy idea in image captioning these years. Other than more delicate analysis in image feature extraction it's more interested in making semantic analysis on the sentence. Wang et al [13] decomposite the sentence with skeleton and attributes and trained them separately. It's quite evident to distinguish the trunk with adjective/adverb while still there are many tricky task like decomposite adverb with adjective are mentioned by their paper. The central idea is to establish a hierarchical semantic tree and corporate trunk network and the attributes network.

Instead of exploiting different architecture of the sentence generating model, we can always try to optimize the different part inside our current network. Along with the outstanding performance the bidirectional LSTM achieved in image generating task, i.e. mostly in PixelCNN as well as VAE, BLSTM also plays a promising role in caption generating. The main idea of [14] is to generate sentence in forward and backward directions and retrieve their weighted combination. The generation process resembles human recognision system which can be considered as a top-down bottom-up method on the sentence.

Some other applications incorporates with GAN [15] which gives more randomness when producing a new sentence. This time we made an attempt by combining the adaptive image caption model along with the detection features.

3 Model Architecture

3.1 Basic Structure

The basic model comes from [8] and its framework is illustrated in Figure 1.

3.1.1 Spatial attention model

In their work, they first used a spatial attention model to ask the LSTM to attend to specific regions of the input image at different time steps.

Image Features The image features come from the last convolutional layer of ResNet, indicated by $A = [a_1, ..., a_k], a_i \in R^{2048}$. A global image feature vector is computed by

$$a^g = \frac{1}{k}\Sigma_{i=1}^{k} a_i$$

As for the final representation of images, the visual feature vectors V and v_g are obtained by

$$v_i = RELU(W_a a_i)$$

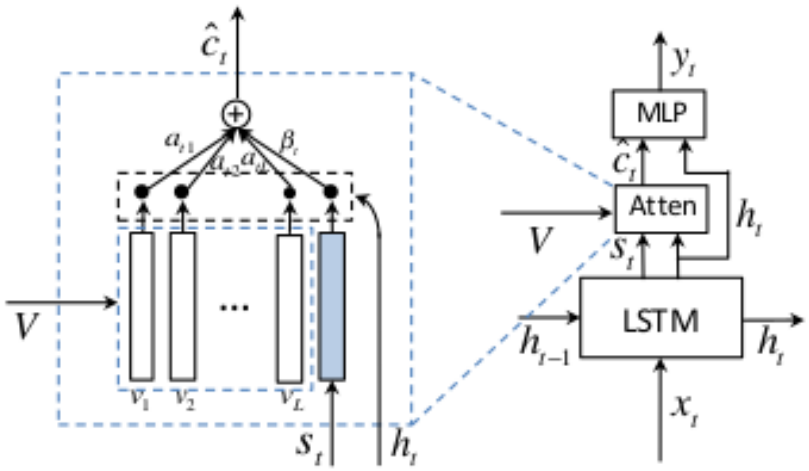


Figure 1: Architecture of the adaptive attention model for image classification [8]

$$v_g = RELU(W_g a^g)$$

where W_a and W_g are learned weights.

Attention Given image features $V = [v_1, ..., v_k], v_i \in R^d$ from CNN model, the attention distribution over k different regions comes from the following equation.

$$z_t = w_h^T tanh(W_v V + (W_g h_t) I^T)$$

$$\alpha_t = softmax(z_t)$$

where W_v and w_h are learned weights and $I \in R^k$ is an all-one vector.

Therefore, the context vector with attention mechanism is obtained by

$$c_t = \Sigma_{i=1}^k \alpha_{ti} v_{ti}$$

3.1.2 Adaptive attention model

Then they extend the spatial attention model to an adaptive attention model by introducing the *visual sentinel* concept.

The input to the language LSTM model is $x_t = [w_t; v^g]$ where w_t is word embedding vector for previous word and v_g is the global image vector. In this way, during caption generation, the decoder exploits the information from previous generated word and also image features.

Visual sentinel concept s_t stores the information that the decoder already knew. In other words, it decides how much information the decoder needs to obtain from visual features as opposed to the linguistic part when generating the next word.

$$g_t = \sigma(W_x x_t + W_h h_{t-1})$$

$$s_t = g_t \odot tanh(m_t)$$

where W_x and W_h are weights to learn and g_t is the gate on memory cell m_t. $\odot$ is element-wise operator and σ is sigmoid activation function.

Based on spatial image features c_t and visual sentinel s_t, the mixed adaptive context vector $\hat{\mathbf{c_t}}$ is computed by

$$\hat{\mathbf{c_t}} = \beta_t s_t + (1 - \beta_t) c_t$$

where $\beta_t = \hat{\alpha_\mathbf{t}}[k+1]$. This new context vector combines visual image features and also information from visual sentinel. And thus this implements what the model claims in terms of when to attend to images.

The new sentinel gate β_t comes from modified version of attention component. The modified attention vector $\hat{\alpha_\mathbf{t}}$ blends attention weights z_t of visual features from images and also visual sentinel information s_t.

$$\hat{\alpha_\mathbf{t}} = softmax([z_t; w_h^T tanh(W_s s_t + W_g h_t)])$$

where W_s and W_g are weight parameters.

Finally, during word generation, a single layer neural network aggregates the context vector $\hat{\mathbf{c_t}}$ and h_t to produce a probability for each word in vocabulary.

$$p_t = softmax(W_p(\hat{\mathbf{c_t}} + h_t))$$

3.2 DenseNet CNN backbone

While searching different CNN models, DenseNet arose our interests. With connections between each layers and every other layer in a feedfoward fashion , it achieves higher performance with even less parameters. Figure 2 shows a 5-layer dense block. From it we can roughly understand the idea. To be more specific, we first consider the traditional structure of ResNet. [16] Consider a single image x_0 . Suppose $H_l(.)$ is the non-linear transformation and x_l is the output at l th layer. In ResNet we have:

$$x_l = H_l(x_{l-1}) + x_{l-1}$$

In a DenseNet we introduce connections from any layer to all subsequent layers as following equation shows:

$$x_l = H_l([x_0, x_1, ..., x_{l-1}])$$

Where $[x_0, x_1, ..., x_{l-1}]$ refers to the concatenation of the feature maps from layers 0, 1, ...l-1. That's the main difference between DenseNet and other classical CNNs. Actually, DenseNet outperforms many typical CNNs including ResNet in basic model.
Here we replaced ResNet152 in the origin code with DenseNet201 in pytorch. To adapt to our model, we eliminate the last fully connected layer of DenseNet201. Not surprisingly, we achieved a higher result than that of basic model. Experiments show that DenseNet201 works better than ResNet152. Thus, in all the following models, we keep using DenseNet201.

3.3 Detection-Aided Features

3.3.1 Extraction of detection features

To obtain the detection features needed for image captioning, we basically follow the procedure introduced in [10]. Specifically, we used the pre-trained detection model provided by the authors. The model was trained on the Visual Genome dataset [17], using ResNet-101 as the backbone architecture, and Faster R-CNN [2] as the detection pipeline. In addition to the traditional Faster R-CNN framework, the authors added

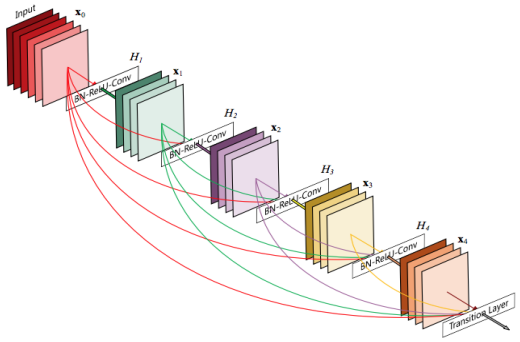


Figure 2: The structure of DenseNet

a new branch for attribute analysis in the second stage of Faster R-CNN, along with class prediction and bounding box regression tasks. The attribute analysis task was able to describe specific attributes, such as color, open/closed, long/short, etc.

Using the aforementioned pre-trained model, we generated intermediate features (final features in the second stage of Faster R-CNN) of the flickr30k dataset. In particular, for each image, we keep 36 region proposals with highest confidence after non-maximum suppression (NMS). A feature of 2048-dimension is attached to each region proposal.

We also plotted some detection results, as shown in Figure 3.

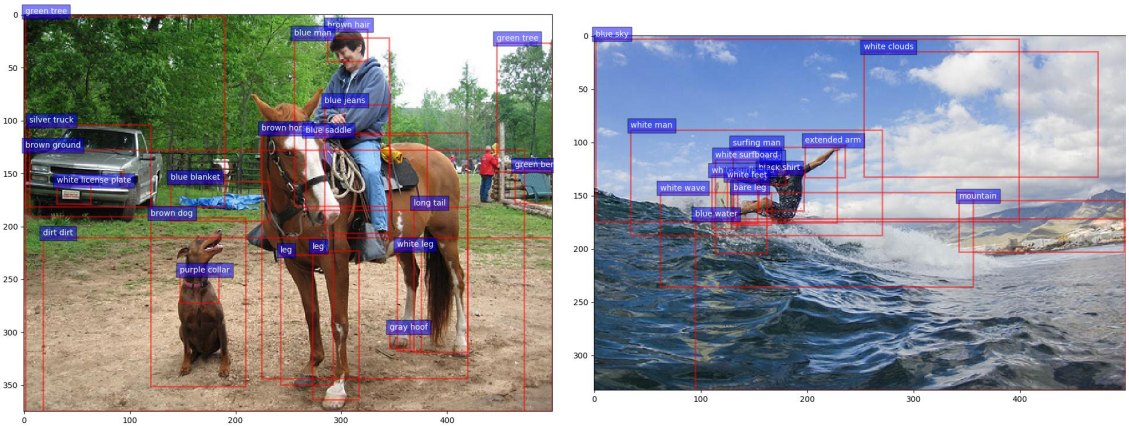


Figure 3: An illustration of the detection result with attribute analysis. In addition to the traditional classification and localization tasks, the model is also able to describe detailed attribues, such as color, open/closed, long/short, etc.

From the results, we can see that the detection model is not only able to detect objects in the picture, but also describe the attributes, such as ”brown shirt”, ”extended arm”, ”surfing man”, which we believe are crucial for correctly describing the details of the images.

3.3.2 Incorporation into the model

Right now, we obtained two types of visual features, namely spatial features $V \in R^d$ from encoder CNN and detection-aided features $V_d \in R^l$ from the procedure in [10]. Besides, we consider the significance of **localization** of detected objects in image captioning task and therefore originally incorporate 6 localization features into V_d. Given a $w \times l$ image and a $w_b \times l_b$ bounding box $(x_b, y_b), (x_b \leq w, y_b \leq l)$ of detected region, the localization features can be expressed as $[\frac{x_b}{w}, \frac{y_b}{l}, \frac{x_b}{w} + \frac{w_b}{w}, \frac{y_b}{l} + \frac{l_b}{l}, w_b, l_b]$.

The two types of image features have internal heterogeneity. It is because the spatial features $V \in R^d$ from encoder CNN are high-level abstract features whereas detection-aided features V_d are features from detected regions which functions like hard-attention to some extent. Therefore, how to synthesize the visual features is a key problem here.

Concatenating V and V_d is an easy way to handle the problem but may not perform well due to its internal heterogeneity. Here, we introduce two attention units $\hat{\mathbf{c_t}}$ and $\hat{\mathbf{c_{td}}}$for V and V_d, as shown in Figure 4. And the final probability for each word is computed as follows utilizing detection attention and visual attention information.

$$p_t = softmax(W_p(\hat{\mathbf{c_t}} + h_t)) + softmax(W_p(\hat{\mathbf{c_{td}}} + h_t))$$

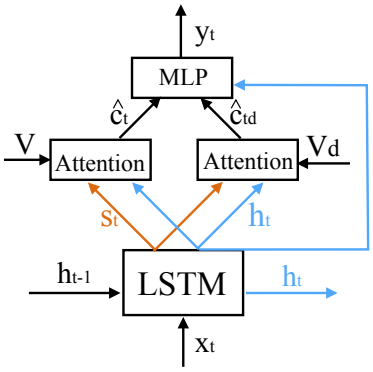


Figure 4: Proposed model

3.4 Model Fine-Tuning

Many of deep learning algorithms require massive datasets like ImageNet. However, for particular problems, we only have access to relatively small datasets such as Flickr30k in our case. Considering the fact that the previous part of the network is just a feature extractor, we can just "freeze" them with the pretrained parameters. Then we create our new last output layer and train it merely. This is how we finetune our network. Here we use ImageNet for pretrained model.

4 Experiments

4.1 Experimental Settings

Dataset The dataset we use is a standard benchmark dataset, Flickr30k dataset [18] from flickr, for training and evaluation. It contains 31,783 images, most of which describe human-involved actions and each image has five captions. We adopt the same split approach to obtain training, validation and test dataset.

Preprocessing Following the procedures in [8], we set the maximum length of caption to be 20 and build a vocabulary of 7649 words.

Training details In our experiments, we use a single layer LSTM with hidden size of 512. For the best model, we use the Adamax optimizer with parameters beta1 = 0.8 and beta2 = 0.999. The base learning rate is 0.0002 for LSTM part and 0.0001 for CNN. The LSTM learning rate starts to decay exponentially after 20 epoch, and reaches 0.000152 in epoch 40. We set the batch size to be 40, and train for 40 epochs.

4.2 Results

We show the captions generated by our best model on 3 images in Figure 5. We can see that the model is not only able to describe what the subject is doing (for example "a man is riding the wave"), but it can also describe some details, such as "a man in a blue shirt".

(a) a woman in a black dress is walking down the street

(b) a man in a blue shirt is riding a wave in the water

(c) a man in a white shirt is standing in front of a large crowd of people

Figure 5: Caption examples

For quantitative analyses, we evaluate the result on the Flickr30k validation set, using the BLEU [19], ROUGE-L [20] and CIDEr [21] metrics. The results are shown in Table 1. These are a little bit lower than those in the original paper, but since we do not implement the beam search algorithm in our work, this is understandable.

Further, since the main point of this project is to prove the effectiveness of the ideas that we came up with, we will show in the next section that, all of the ideas mentioned above have a positive effect on the final captioning results.

Table 1: Result of our best experiment settings

BLEU1	BLEU2	BLEU3	BLEU 4	ROUGE	CIDEr
0.6359	0.4570	0.3206	0.2252	0.4575	0.4614

4.3 Ablation Analysis

To gain further insights into the effects of our ideas, we conducted some ablation experiments. Due to limited GPU resources, we terminate those experiments after 15 epochs, but we can already see the trend through those results, which are shown in table 2. Here M stands for the model from [8] with Res-Net-152 but without beam search and M-det stands for detection-aided M with DenseNet-201.

Table 2: Results of our ablation experiments

Model	BLEU1	BLEU2	BLEU3	BLEU 4	ROUGE	Cider
Baseline Model (M)	0.5492	0.3602	0.2292	0.1490	0.3967	0.2354
M with DenseNet-201	0.5641	0.3713	0.2394	0.1578	0.4019	0.2483
M-det concatenated V and V_d	0.6059	0.4191	0.2843	0.1952	0.4321	0.3713
M-det with two attentions	0.6188	0.4326	0.2977	0.2085	0.4358	0.3780
M-det with two attentions + Adamax	0.6246	0.4391	0.3026	0.2109	0.4379	0.3805

4.3.1 ResNet-152 VS DenseNet

We can observe that DenseNet backbone improves all four criterion by around 0.01 than ResNet-152. Both DenseNet and ResNet could alleviate the vanishing gradient problem and makes training deep networks easy. For DenseNet, the design of Dense connection makes each layer accessible to the inputs and loss regardless of the depth whereas ResNet adopts residual block to allow the gradient pass directly through layers.

The better performances of DenseNet over ResNet on our model may come from the better feature aggregation. For DenseNet, the connections between layers enable the model to better exploit the features from all the previous layers. In this way, when extracting higher level features such as peoples' faces, the ability of checking all preceding different low-level features such as edges, nose parts, eyelids may facilitate the feature extraction abilities. The better feature propagation and reuse makes it outperform ResNet in our problem.

4.3.2 Effect of detection features

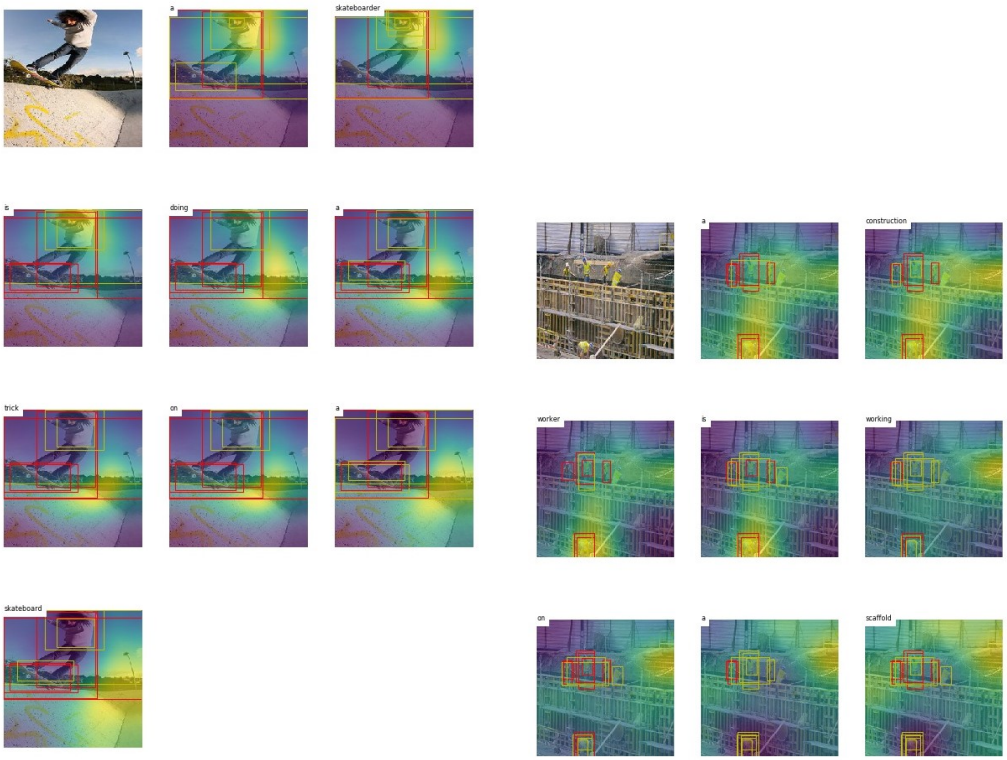
With detection-aided features without localization, the model improves BLEU-series by around 0.02 and enhances CIDEr by around 0.08. This validates our assumption that detection-aided features are useful in our caption generation task. The evaluation criteria CIDEr evaluates the generated sentence through TF-IDF and is proven to better capture human judgement of consensus. The higher improvement of CIDEr validates the effectiveness of detection-aided features.

By incorporating localization features, the model improves in four criterion, which demonstrates that the spatial position information of detected objects in images contributed to the caption generation process. For instance, it is obvious that the size of an object tends to provide critical information such as the main object or background objects.

4.3.3 Effect of two attention units

As for how to aggregate visual features V and detection-aided features V_d, the two approaches we try are concatenation and attention-based synthesis. And it turns out that attention-based method outperforms the counterpart. This indicates that the two various image features are heterogeneous and needs to contribute separately into generating image descriptions.

4.4 Visualization



(a) A skateboarder is doing a trick on a skateboard (b) A construction work is working on a scaffold

Figure 6: spatial & detection attention visualization

In addition to ablation analysis visualization for detection and spatial attention is also drawn for each generated word. Figure 6a shows how detection attention provides extra information regarding the original spatial attention for the network and aids the network in locating the most interesting portion of image each step. In the experiment, we draw red bounding box for detection with high weight and the yellow one for relatively lower weight. Besides, the effect of spatial attention is depicted as a heat map where highlight part stands for higher weight while darker region stands for lower weight. We can tell the network realize a "skateboarder" largely relying on the detection features and it also catches a sight of "skateboard" implied by another red box. Notice in this example the spatial attention only contributes to "skateboarder" recognition but fails on activating the "skateboard" segmentation.

In contrast, in figure 6b the network gives more credit to spatial attention and we can tell from figure that, it changes along with the object in each step. However, the detection only stresses on the "workers". One possible reason is it keeps on embedding the "workers" information into the network such that the word "scaffold" is more likely to stand out in the end. Both visualization show how successfully the joint attention mechanism works within the current network architecture.

5 Conclusions

Most image captioning works apply encoder-decoder framework to deal with. Built on two state-of-art works, we explore how to apply detection-aided features into adaptive attention model to facilitate caption generation task. The adaptive attention model with visual sentinel concept is capable of deciding when to attend to the image or the previous generated captions. And the detection features provide useful image features besides the visual feature vector from CNN model. To aggregate the two visual features, we adopt two attention units and try DenseNet model in encoder part with fine tuning. And we also try Adamax as our optimizer. In experiments, we compare our model with baseline models on Flicker30k dataset and the better performances validates the effectiveness of our model. We conduct detailed ablation analysis and also visualization and caption examples are listed to demonstrate our results. Moreover, through group work, we learned task division, time schedule and group discussion, which are as important as academic knowledge. In future, we can also try different word embeddings and beam search strategy which mentioned in original papers for further improvements.

6 Contributions

In this project, Xudong Sun mainly takes charge of detection model. Yi Luo contributes most in dataset replacement and incorporation of localization features and attention-based aggregation. Feihua Fang is charge of DenseNet model and fine-tuning it. HaoCheng Li is responsible for literature research and visualization of attention maps. We four completed the report together.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[2] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 91–99. MIT Press, 2015.

[3] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *14th European Conference on Computer Vision, ECCV 2016*. Springer Verlag, 2016.

[4] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1635–1643, 2015.

[5] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[7] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3156–3164. IEEE, 2015.

[8] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 6, 2017.

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

[10] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*, 2017.

[11] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 6298–6306. IEEE, 2017.

[12] Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. Deep reinforcement learning-based image captioning with embedding reward. *arXiv preprint arXiv:1704.03899*, 2017.

[13] Yufei Wang, Zhe Lin, Xiaohui Shen, Scott Cohen, and Garrison W Cottrell. Skeleton key: Image captioning by skeleton-attribute decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7272–7281, 2017.

[14] Cheng Wang, Haojin Yang, Christian Bartz, and Christoph Meinel. Image captioning with deep bidirectional lstms. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 988–997. ACM, 2016.

[15] Hongge Chen, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, and Cho-Jui Hsieh. Show-and-fool: Crafting adversarial examples for neural image captioning. *arXiv preprint arXiv:1712.02051*, 2017.

[16] Laurens van der Maaten Kilian Q. Weinberger Gao Huang, Zhuang Liu. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[17] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. *arXiv preprint arXiv:1511.06361*, 2015.

[18] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

[19] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[20] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

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