CS-GY 9223 Project Report - CapVis: An Image Caption Visual Analytic System

Dongzi Qu
New York University
Brooklyn, NY
dq394@nyu.edu

Qi Yin
New York University
Brooklyn, NY
qy652@nyu.edu

Abstract

Image captioning, which can automatically generate the title for images, is becoming more and more mature and important in recent years. However, it's always obscure for people to really know about the internals of such a captioning model. In this paper, we design CapVis, a wonderful analytic visualization system for image caption task. Through this system, users can explore more features about image caption model, and thus obtain deeper insight.

1. Introduction

Millions of images emerge on the internet every day, their contents ranging from natural scenery to social event. Those images contain extensive underlying information which is hard to process. Image caption, which lies at the intersection of computer vision and natural language processing, perfectly bridges the gap of image and text. Automatically generating the description for image drastically decreases the complexity of information extraction from images, which is of great use.

Currently image caption's capability is still limited, there are still lots of unknown to explore in this area. Since the mainstream approach to this problem is using seq2seq models with deep neural network, what happens within the network is hard to interpret and analyze due to its BLACK-BOX nature. On the other hand, although image captioning has been a hot issue for several years, it seems that the related visualization system has lagged behind. Accordingly, researchers are still unaware of the internal of captioning model when generating the captions. So, in this work, we propose to take advantage of one state-of-the-art attention based image caption model[1] to build our own visualization system.

We propose CapVis, an analytic visualization system for image caption, which enables users to see internal structure of the computation model and to explore features of model base on cases. CapVis mainly contains two parts, covering the visualized and intuitive analysis for both internal and external of image captioning model. As for internals, we would like to present the structures and inner connections of the key weights inside Neural Network. In this part, providing users with an interactive tool to explore more about the BLACK-BOX, we aim at helping users figure out what exactly happened when a caption was generated. From the perspective of external analysis, we adopt case-based reasoning to reveal the underlying pattern of captions. For one aspect, we generate massive captions and then cluster those captions for further analysis. As for another, we offer users an interface so that they can cover the image and see the change inside corresponding caption. In this part, applying model to specific cases and doing statistic analysis on the result, we focus on the problem that why the model generate a certain caption.

The GitHub link of our project can be found here CapVis.

2. Related Work

In this section we provide relevant background on previous work both in image captioning model and visualization system. In order to implement our own insightful system, it's undoubtedly significant to know the recent research achievement of these areas.

Image Caption: The problem of generating natural language descriptions for visual data has been studied for a long time. Traditionally, people used template-base approaches generate caption whose slots are filled in based on outputs of object detection, attribute classification and scene recognition. The approach from [2] combines detection and recognition algorithms to extract content from image and then generate sentence based on statistics and predicted data. For recent years, inspired by the success of seq-2-seq frameworks in machine translation[3], neural based methods has made great progress. [4] is the first one who use an LSTM instead of vanilla RNN as the decoder. Later, attention mechanisms have been introduced to encoder-decoder neural frameworks of image captioning task[1], which incorporates an attention mechanism to learn a latent alignment from scratch when generating cor-

responding words. Our Visualization is is based on the computation model from[1].

Visualization System: Although the research on image captioning visualization system is rare, we can still get inspiration from visualizing the internals of neural networks. There were several talented works building the interactive system for visualizing and analyzing CNN or RNN in recent years. One talent work from them is [5], in which the authors proposed a tool for visual analysis of hidden layers dynamically in RNN. Undoubtedly, it's convenient for users to observe the working flow and cells connection between hidden states in LSTM and input, which is very instructive for our visualization system. The paper [6] introduced a structure called deconvolutional layer, with which users can take feature maps as input and project it back to the input image. This pattern allow us to visualize the different level of feature extraction intuitively. Moreover, [7] is a well-designed CNN analytic visualization system for domain expert, they propose a cube-style novel cube-style visualization to reveal the complex correlations of data and training weights. The representation of Deeper tracker is informative, but it might be a little bulky in terms of image captioning. As for designing of website, there are various of style, ranging from Card Based Design to Blog-like design. Finally, we choose to implement a flat web page with pull down menu, keeping things simple while organized. Codyhouse is a good example of flat website.

Baseline Model: Tutorial-to-Image-Captioning by Sagar Vinodababu is a talent implementation of original paper of [1] in Pytorch version. Based on that, we write a web demo to display the output of the model on the website as our baseline. Based on this, our final work expands the internal presenting parts and introduce image clustering to analyze on the captioning results. Moreover, we improve the effectiveness of the visualization to give more straightforward and better interpretation on model.

3. Method

3.1. Framework of System

In this subsection, we would like to introduce the structure of CapVis and specific tools we have used firstly, and then the logic and effectiveness of each visualization design will be discussed.

System Structure: We choose to implement our visualization system in forms of flat website. The total framework is shown as Figure 1, which contains three parts: rear-end, front-end and server-end. Inside the rear-end, we put prerequired files, including model trained on Flick30k, clustering result, feedback log and some record for visualization. All the interface for user are in front-end, where user can upload image, require to cover one specific area and provide their feedback for the system and caption quality. We Implemented front-end with HTML combined with CSS and Javascript. For server-end, we choose Flask to deploy our website frame work, which can receive and save the upload images and feedback from front end, and respond to user's request.

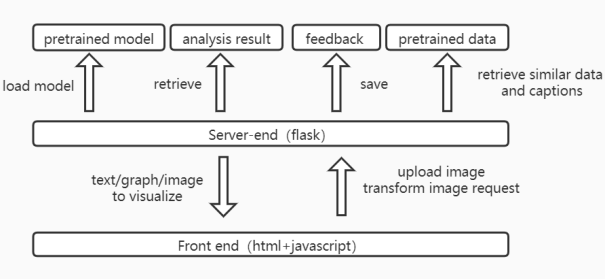

Figure 1. System Structure

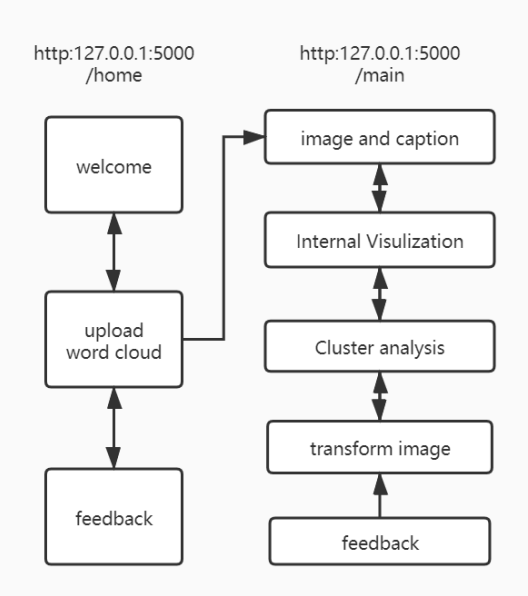

Figure 2. Website Frame

Website Frame Design: Specifically, the workflow of our visualization system on website is shown as Figure 2. Users can upload testing images from home page and then the web page will redirect them to main.html. Main.html contains core visualization components of our system. In this page, uploaded image and generated captions are displayed in section "image and caption". In "internal Visalization" section, output of encoder and attention weights are exhibited in format of pictures. Besides, user can see contribution of every pixel for each generated word. Moreover, by clicking "show cluster", users can retrieve similar images, captions and word distribution within the same cluster. In section "Cluster analysis", users are presented with

two graphs which shows statistic result of 8000+ generated captions. In section "transform image", users can select a box on the image to mask that area and get new caption. As for feedback part, users can mark a caption as "good" or "bad" in "image and caption" section and type in their thoughts in "feedback" section.

3.2. Visualization Design

In this subsection, we will expound the details of each part of our visualization system.

CNN Vis and RNN Vis: Given the computational model uses ResNet-101 as encoder to extract features and attention based LSTM as decoder to generate captions, we want to know how the decoder take the output from encoder to generate captions. Therefore, knowing the mechanism inside such a BLACK-BOX is essential for us. As for encoder, ResNet-101 aims to capture the key information from original images. Accordingly, the output of its last convolutional layer is 2048 features in shape of (14,14). Figure 3 shows the very first 16 features from the whole feature maps and each of them can be treated as abstraction of the input image. Bright parts represent more focus on the objects from the region of input. While for the decoder, attention based LSTM uses different weights on different parts of input images when generating caption word by word. As shown in Figure 4, the first line represents features learnt from encoder's output; the second line is after applying the features on original image; while the last line shows the corresponding words in each time step. Obviously, attention model mainly focus on the white plate when generating the word **frisbee** from this example.

Figure 3. CNN features

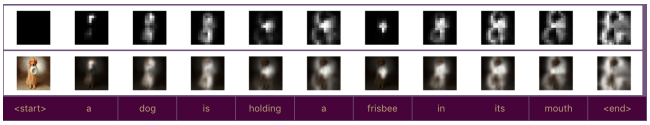

Figure 4. Attention features

Similar Image Retrieval: In order to facilitate users to know more about why unsuitable captions are generated sometimes, we implement this functionality providing users with several similar images with their corresponding captions to the original input image shown as Figure 5. To achieve clustering and selecting similar images, we take use of VGG16, a powerful model on classification task from keras, and Flickr-8K as our database where images are selected from. With the help of VGG16, each image from Flickr-8K is represented as a vector. After that, we can apply KMeans algorithm on all the 8k vectors to generate 100 clusters. Finally, VGG16 is used again on the input image to show the cluster it belongs to and then select samples from this cluster. We believe that users can explore more internal patterns of captioning model by comparing similar images and captions.

Figure 5. similar captions and images

Analysis on Clustering: As we mentioned in previous component of our system, we cluster more than 8000 images to 100 labels. In order to understand the external of model in terms of caption length and words in captions, we design some tools analyzing the images and captions after clustering. Figure 6 shows the statistic data of caption length for each cluster, where the x-axis represents the different clusters and caption length for y-axis. The larger circle means caption with this length has higher ratio in its corresponding cluster. And when user put mouse on the bubble, a promt box will show details of that data. For example, as shown by Figure 7, the frequency of captions with 7 words from the 57th cluster is 0.31. According to all these circles, we can find that the length of captions are mainly lying between 7 to 10 no matter which cluster it belongs to.

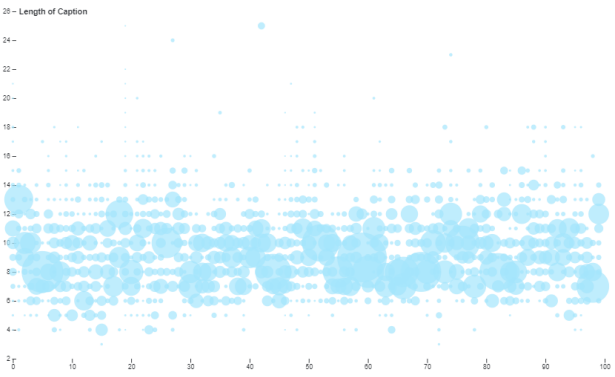

Figure 6. caption length for each cluster

Besides the analysis based on caption length, the category of used words from all captions is also crucial for our system. In each cluster, we divide words by their tags, in-

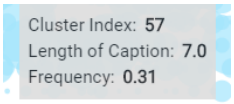

Figure 7. caption length for each cluster-detail

cluding NN: **noun**, JJ: **adjective**, IN: **preposition** and so on. Finally, we integrate all 100 clusters into one pie chart shown as Figure 8. Furthermore, from the perspective of used words among all 100 clusters, we construct a Word Cloud to present the high-frequency used words with larger font shown as Figure 9.

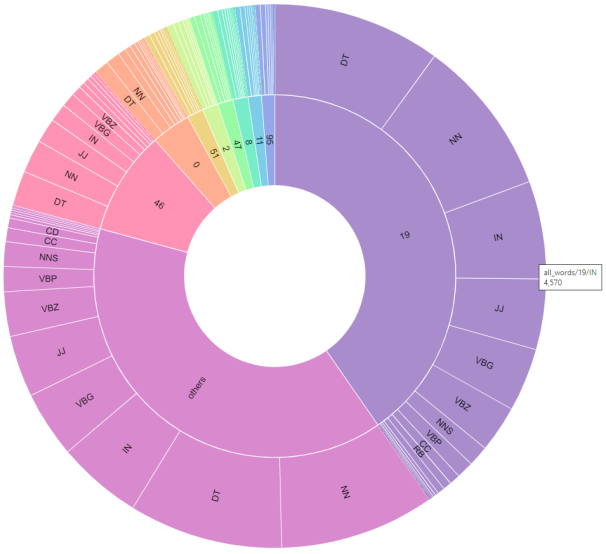

Figure 8. word tag for each cluster

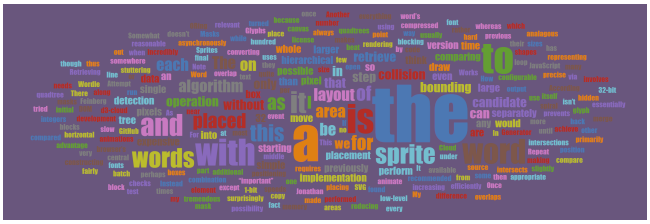

Figure 9. word cloud of generated captions

Image Masking: According to the previous RNN visualization, there is no doubt that the LSTM focuses on different parts of input image when generating different words. To demonstrate this characteristic more detailed, we design a interactive tool allowing users to cover some parts of input image. Then, users are shown with new updated caption for the covered image.

In order to mask an image, we need to implement two steps. First, provide interface for users to select the rectangle area on the image, which can be moved and resized, and collect corresponding coordinates. Then, pass them into the server-end, and request for process. Finally, server will set all pixels inside that rectangle area to 0, caption again, and send transformed image and new caption back, so corresponding area becomes black. One example is shown in Figure 10, we can see the frisbee area is covered.

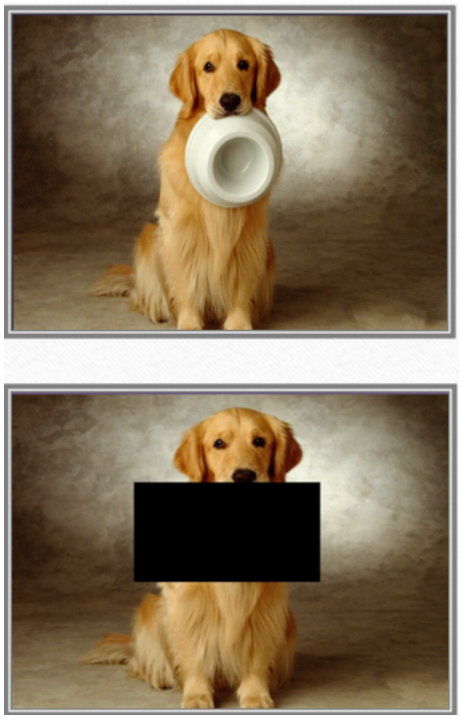

Figure 10. cover a rectangle area

4. Evaluation

To demonstrate the usefulness and evaluate performance of CapVis, we carry out several case studies and conduct a questionnaire with users.

4.1. Case study

In this subsection, two cases (two testing samples) are used for the evaluation of our system.

dogs: The original and covered images of dog is shown as Figure 10. Figure 11 shows the similar images to this testing sample, which is obvious well suited clustering. Besides of image clustering, we also list the different captions for original/covered images:

Original: *a dog is holding a frisbee in its mouth*

Masked: *dog that is sitting on a floor*

From this case we see that masking image is a good way to split part of image out of consideration.

little girl: As shown in the Figure 13, we see that the original caption is not much reasonable since there is no

Figure 11. similar images picture 10

toothbrush in the image. Accordingly, we start to think about why the caption contains "toothbrush". Firstly, we look back to the "RNN Vis" and find out that for the word "toothbrush", attention weights indicate higher points on her mother's hand. So, we intuitively speculate that the model regard mother's hand as a tooth brush. In order to prove it, we mask that area and generate a new caption again. We can see form Figure 14, new caption doesn't have the word "toothbrush" and become more reasonable. Even though we obtain more ideal caption by covering the input image, we still want to know why the caption make such a mistake on this part. Therefore, we observe on the similar images and captions. As Figure 12 shows, our model can recognize "girl" faultlessly. So, when generating caption for this picture, it naturally generates a caption about "girl". On the contrary, "hand" is not a salient feature for this picture. We intuitively guess that it is rough for the model to capture the the characteristic of hands because of its low appearance in training data.

Comparing these two cases, we find that the image clustering works better for dog's image than the little girl's. Our conjecture is that the effectiveness of image clustering will be more ideal for the images with simple objects and background. Furthermore, image covering really impact on the generating captions.

Figure 12. similar images of little girl

4.2. User feedback

As for the visualization system, interpret ability, interactivity, usability and simplicity should be scored by the users. Since these criterion are quite difficult to be quantified, we make one questionnaire survey to students from NYU Tandon School of Engineering and conduct interviews with several PhD students major in Computer Science. To sum up, the result shows that this visualization system is comprehensible enough for all testers. Moreover, students majoring in Computer Science can quickly master to explore the performance of captioning model using CapVis. On the other hand, some of the students reflect that the latency after uploading an image is not short enough.

Figure 13. original little girl image

Figure 14. masked little girl image

5. Conclusions

In this project, we design and implement an analytic visualization system for image caption to disclose the internals of Caption model and display our analysis results. Clustering on generated captions, we find the underlying pattern of captions, and we use pie charts, bubble charts and zoom -able donut charts to exhibit our result on words frequency, part-of-speech distribution and caption length and

give our explanation to the results. In addition, we provide users with interactive interface, with which users can retrieve similar images and corresponding captions, and transform image to see the difference of the output from the model.

CapVis interprets caption model in a different way and serves as a tool for user to explore the process of generating captions. For those people who are new to image caption, but want to quickly form an basic understanding of image captioning and those who want to identify defects of model, CapVis is one of their choices.

There are many avenues for future work. For one thing, the visualization of hidden state inside LSTM is still a challenging task. How to visualize the high-dimension matrix and explain the meaning of hidden state may be next step work. For the caption model, enriching the language diversity is also a tricky problem, which can be seen from our results: captions generated tend to converge on a specific expression pattern. We can ponder on this problem on both dataset and model directions.

References

[1] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *International conference on machine learning*, pages 2048–2057, 2015.

[2] 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 Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903, 2013.

[3] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[4] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator, 2014.

[5] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush. Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):667–676, 2018.

[6] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.

[7] Dongyu Liu, Weiwei Cui, Kai Jin, Yuxiao Guo, and Huamin Qu. Deeptracker: Visualizing the training process of convolutional neural networks. *ACM Trans. Intell. Syst. Technol.*, 10(1), November 2018.