Convolutional Neural Network with Recurrent Neural Network: From Image to Story

Changhao Shi[1], Kunyu Shi[2], Shuo Xu[2],
Xinyue Zhang[2], Wenrui Liu[3] and Yang Xu[3]
[1]Department of Electrical and Computer Engineering
[2]Department of Computer Science and Engineering
[3]Department of Mathematics
University of California, San Diego, La Jolla, CA 92122
{cshi,kshi,s3xu,xiz013,wel031,yax088}@ucsd.edu

Abstract

In the past few years, generating text from images and videos has gained a lot of attention in the Computer Vision (CV) and Natural Language Processing (NLP) communities, and several related tasks have been proposed [1]. In this project we will utilize neural network structure to generate stories based on a single image. The need of generating more narrative texts from images which may reflect experiences, rather than just listing objects and their attributes, has given rise to tasks such as visual storytelling[1, 2]. Compared to a single image, the combination of a detailed story and a image can present a segment of story in a more vivid and informative way. The main idea is first use CNN and RNN to embed image and caption into same vector space, and create several captions for a image using nearest neighbor method, then we use decoder to decode these captions to a skip thought vector[3] as a generic representation of these sentences. By doing style transfer, we transfer the skip thought vector from caption style to romantic novel style, and after this an RNN decoder is used to generate story conditioned on this vector.

1 Introduction

A quick glance at an image is sufficient for human to point out and describe an immense amount of details about the visual scene[4]. However, for visual recognition model, it has been regarded as a challenging task to generate descriptions for images since it involves integrating vision, learning and language understanding. The model not only needs to correctly recognize what objects are in images but also incorporate spatial relationships and interactions between them. Even with all these information, it is not easy to generate a relevant and grammatically correct description[5].

Besides a brief description, human may also imagine a lot according to the different objects and backgrounds in the image. Such imaginations can be fairly impressive, creative and various. For the same image, a fantacy novel writer, a comedy actor and a romantic poet would have totally distinct stories. We can say, every picture tells different stories. Commonly, a computer is a machine without imagination. However, it can learn to be imaginative enough to tell stories with any genre due to the help of a well-trained recurrent neural network (RNN).

As for text generation, a wide variety of applications rely on it, including machine translation, video/text summarization, question answering, among others. From a machine learning perspective, text generation is the problem of predicting a syntactically and semantically correct sequence of consecutive words given some context [6].

Preprint. Work in progress.

In common image caption models, people use captions of images as target to train CNN and RNN jointly. While our motivation is to let model tell stories without any image labeled with story. In this project, we will implement a convolutional neural network (CNN) to extract semantic information in images and generate meaningful stories from upstream information. The first stage of our model is a image captioning model, and we let visual information flow to sentence information. In the second state, we use skip thought vector encoder to encode generated captions to a vector representation, and conditioned on such vector we use an RNN to generate passages. Instead of creating a brief description accompanying an illustration, our goal is let the model tell stories by understanding images.

2 Related work

Linking languages and images has a rich history. A simple and natural method is to divide images into several parts and predict a word for each sub-image. Mori *et al* first uniformly divide each image into sub-images with key words and then calculate a vector quantization of the feature vector of sub-images, where the results of the vector quantization is treated as the voting probability of each word[7]. Duygulu *et al* continues this method with machine translation, a mapping between region types and keywords supplied with the images is learned using a method base around EM[8]. Later, a wide range of methods are discussed[9, 10]. The methods perform fairly well, but still find difficulty placing annotations on the correct regions. Also, we find it not sufficient to describe images with only a set of words.

Sentences are richer and more complex than lists of words, because they describe activities, properties of objects, and relations between entities (among other things)[11]. Older methods are based on fixed visual representations and translated them into textual descriptions. Farhadi *et al* describe a system that can compute a score linking an image to a sentence. Thus, given a image, this score can be used to attach an accurate descriptive sentence. They map the image and the sentence into a meaning space and calculate the score by comparing an estimate of those two meanings obtained from the image and the sentence. Each of the estimates of meaning comes from a discriminative procedure that can be learned via data[11]. Kulkarni *et al* present a system to automatically generate natural language descriptions from images that exploits both statistics gleaned from parsing large quantities of text data and recognition algorithms from computer vision. They try to make a tight connection between the particular image content and the sentence generation process. This is completed by detecting objects, modifiers (adjectives) and spatial relationships (prepositions), smoothing these detections with respect to a statistical prior obtained from descriptive text and generating sentences using the smoothed results as constraints[12].

Recently, several approaches based on RNNs emerge, generating captions via a learned joint image-text embedding. Kiros *et al* introduce an encoder-decoder pipeline that learns (a): a multi-modal joint embedding space with images and text and (b): a novel language model for decoding distributed representations from their space. Their pipeline effectively unifies joint image-text embedding models with multi-modal neural language models. They also introduce the structure-content neural language model that disentangles the structure of a sentence to its content, conditioned on representations produced by the encoder. The encoder allows one to rank images and sentences while the decoder can generate novel descriptions from scratch[5]. Vinyals *et al* present a generative model based on a deep recurrent architecture that combines recent advances in computer vision and machine translation and that can be used to generate natural sentences describing an image. Their model takes an image I as input, and is trained to maximize the likelihood $p(S|I)$ of producing a target sequence of words $S = \{S1, S2, ...\}$ where each word S_t comes from a given dictionary, that describes the image adequately. Then an "encoder" RNN reads the target sequence and transforms it into a rich fixed-length vector representation, which in turn in used as the initial hidden state of a "decoder" RNN that generates the target sentence[13].

3 Dataset

3.1 MS COCO Dataset

MS COCO dataset is a large-scale object detection, segmentation, and captioning dataset that has been extensively used in computer vision research. We plan to use Microsoft COCO caption task

dataset to train our first stage network. Since MS COCO caption dataset consists of 330,000 images and each image is label with at least 5 captions, we are confident to have sufficient data to train even a very deep network.

The dataset MS COCO has five annotation types and we use the annotation type for image captioning. For each image to be better captioned, each image have at least 5 annotations. Figure 1 shows a demo of the image ad the following are its notations.

Figure 1: Example image from MS COCO Dataset

1. A man is skate boarding down a path and a dog is running by his side.
2. A man on a skateboard with a dog outside.
3. A person riding a skate board with a dog following beside.
4. This man is riding a skateboard behind a dog.
5. A man walking his dog on a quiet country road.

3.2 Caption to Story Dataset

The story data is obtained from the BookCropus dataset. We first got to know this dataset which contains word corpus of novels from [14]. The dataset link from its original project on GitHub is outdated but we managed to find a "Homemade BookCorpus" tool on GitHub as well as its generated BookCorpus txt file mirror. It's basically a sentence-to-vector dataset. We plan to use it to train our second stage encoder-decoder model and get reasonable small paragraphs according to the input captions.

The new word corpus we acquired is constructed from the romantic novels and these are parts of them. Figure 2 shows a part of the word corpus. From the image, we can see that every sentence is regarded as an individual part. In the training part, each sentence will be encoded into numerical vectors and decoded into passages.

```
i know it all and starlings is not the place where you want to be after dark .
the only reason why no one knows this is because jason , emily , seth and i have kept it
that way .
i walked along the empty road alone , occasionally waving to passing kids on bikes .
my backpack was slung over my shoulder , filled with my writing books and sketchpads .
i kept my eyes on the shadowed road , watching my every step .
usually i was more aware of my surroundings , but today i was tired and didnt care if i
rammed into a tree .
i kicked a rock into the grass .
the sun was starting to set , painting the sky in brilliant oranges and reds .
it slipped down the sky , allowing the first stars to peek out from behind the bright
curtain .
```

Figure 2: A small example from the word corpus

4 Method

In this project part, we divide our task into two stages: image to caption and caption to story. Section 4.1 lists the architectures of all the neural networks used in this project and we expand more details about what we do in Section 4.2.

4.1 Neural Networks

4.1.1 CNN: Inception v3

Inception v3 is a famous, widely used neural network designed by Szegedy *et al* [15, 16] and figure 4 shows its architecture.

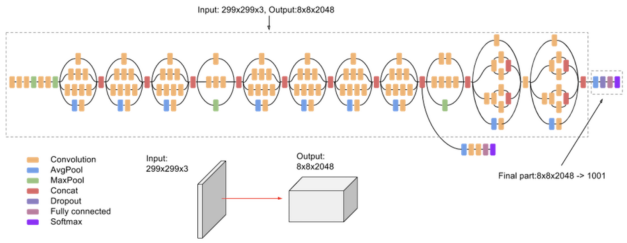


Figure 3: Architecture of Inception v3 [17]

4.1.2 CNN: VGG-19

VGG-19 is a convolutional neural network proposed by Simonyan and Zisserman [18], which has a wide range application.

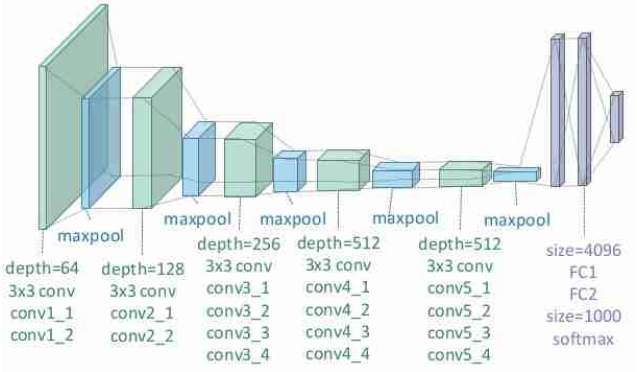
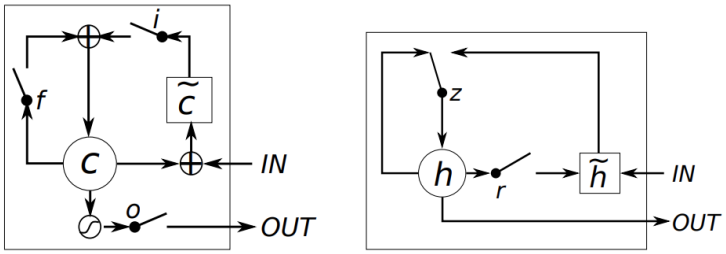


Figure 4: Architecture of VGG-19 [19]

4.1.3 RNN: LSTM

The standard RNN suffers the gradient vanishing problem and it lacks the ability in dealing with long-term dependencies [20]. The long short-term memory (LSTM) is a particular type of recurrent network designed by Hochreiter and Schmidhuber [21] to solve the long-term dependency shortcoming. The LSTM does so via input, forget, and output gates; the input gate regulates how much of the new cell state to keep, the forget gate regulates how much of the existing memory to forget, and the output gate regulates how much of the cell state should be exposed to the next layers of the network [22]. Figure 5 (a) shows the structure of LSTM.



(a) Long Short-Term Memory (b) Gated Recurrent Unit

Figure 5: Illustration of (a) LSTM and (b) gated recurrent units [23]

4.1.4 RNN: GRU

Unlike LSTMs which control the exposure of memory content (cell state), GRUs expose the entire cell state to other units in the network. The GRU operates using a reset gate and an update gate. The reset gate sits between the previous activation and the next candidate activation to forget previous state, and the update gate decides how much of the candidate activation to use in updating the cell state [22]. The structure of GRU is displayed in Figure 5 (b).

Both LSTMs and GRUs have the ability to keep memory/state from previous activations rather than replacing the entire activation like a vanilla RNN, allowing them to remember features for a long time and allowing backpropagation to happen through multiple bounded nonlinearities, which reduces the likelihood of the vanishing gradient [22].

4.2 Architecture of Image to Story

The image-to-story project consists of two parts. The first part embeds images and sentences into the same vector space and predicts most possible captions based on the given image. The second part is a encoder-decoder module which is responsible for encoding the promising captions from previous sub-module into the skip-thought vector form and decode the vector into a short story. Figure 6 shows the overall process and figure 7, 8 show the process in more details.

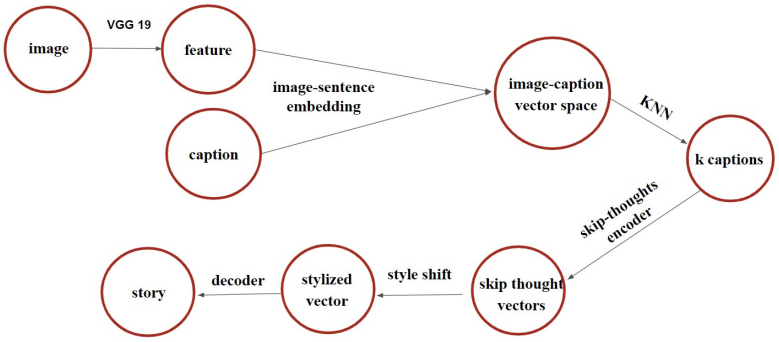


Figure 6: Total process

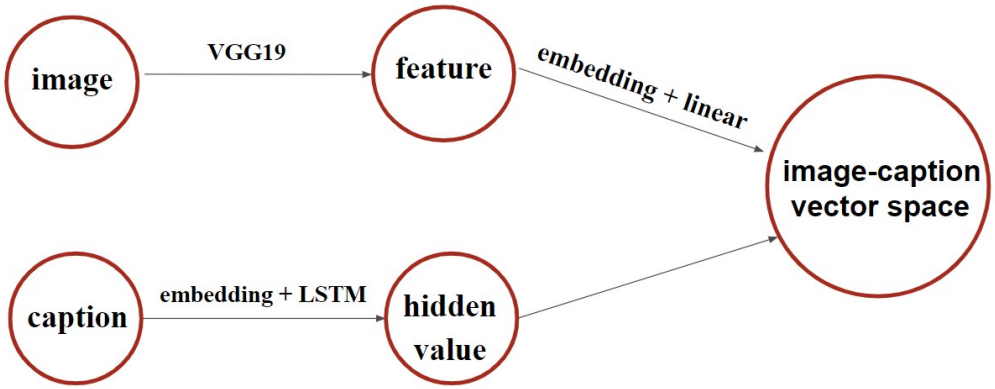


Figure 7: First Part of Training

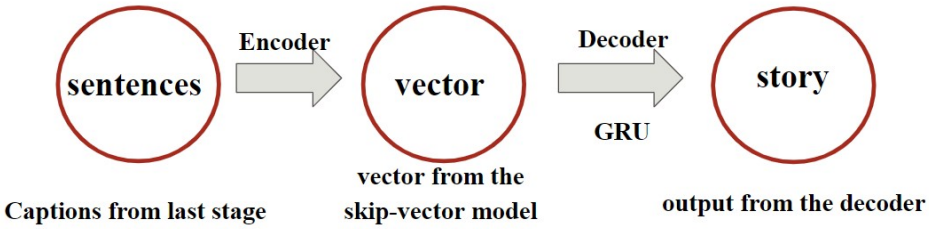


Figure 8: Second Part of Training

4.2.1 Image to Captions

As for the first part, we pre-processed training dataset using Inception v3, with its fully-connected layer removed at the very end, to extract features from input images as the upper-stream input of the embedding sub-module. These generated features are stored on hard disk for future convenience. We adopted transfer learning on a pretrained Inception v3 for convenience. This CNN is pretrained on ImageNet. This choice came from its comparison with VGG-19 and Resnet-50. VGG-19 took longer than Inception v3 and Resnet-50 to extract features. Inception v3 has a better accuracy than Resnet-50 and almost the same number of parameters.

Then, we embed the image feature with a embedding layer and a linear forward layer, and a embedding layer and LSTM layer for caption embedding. Thus, both images and captions are projected into the same image-caption vector space. The goal of training is to both narrow the distance between corresponding images and captions and maximize the distance between irrelevant images and captions.

To complete such a task, we used a pairwise ranking loss function inspired by [5] as is shown below.

$$\min_{\boldsymbol{\theta}} \sum_{\mathbf{x}} \sum_{k} \max\left\{0, \alpha - s(\mathbf{x}, \mathbf{v}) + s\left(\mathbf{x}, \mathbf{v}_k\right)\right\} + \sum_{\mathbf{v}} \sum_{k} \max\left\{0, \alpha - s(\mathbf{v}, \mathbf{x}) + s\left(\mathbf{v}, \mathbf{x}_k\right)\right\}$$

where x and v stands for embedded image vector and caption vector, respectively. Moreover, v_k is a contrastive (non-descriptive) sentence for image embedding x, and vice-versa with x_k.

4.2.2 Captions to Story

This part of task is based on a skip thought encoder-decoder model from [3]. The main idea of skip thought model is to generate generic vector representation for sentences. The author used large corpus text data and trained this model for 28 days, and we transferred encoder part of this weight from theano to pytorch.

By fixing the trained encoder, we encode captions generated from last stage, and calculate the mean skip thought vector of these encoded vector and denote it as d. We calculate the mean skip thought vectors over all the captions in coco dataset and denote it as c, and calculate the mean skip thought vector over all the paragraphs in our romantic novel dataset and denote it as r. Here we do style transfer to transfer the encoded vector from coco image caption style to romantic style:

$$d := d - c + r \tag{1}$$

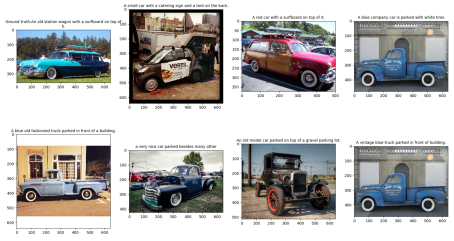
After we get the vector representation of captions, we use GRU model to generate story conditioned on this representation. We do this by initialize the hidden state of GRU to the this skip thought vector, and give GRU a start input and let it to generate stories.

The training of the GRU decoder is done on a large romantic novel dataset, we use paragraph consisting many sentences as input sequence to encoder using teacher forcing to train the decoder. Softmax cross entropy loss is used at the end of the network.

5 Results

5.1 Image to Caption

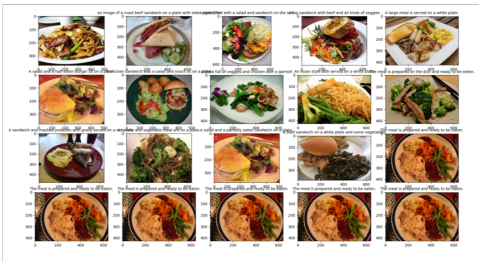
To test the performance of this part, we first extract the feature of a random image from validation dataset, and embed the feature into the image-sentence space using upper-stream depicted in Figure 4, and compute the cosine similarity between the embedded image vector and all the embedded caption vector in training dataset. Then, we plot top-K most similar captions to show the performance of our embedding network. It's a straightforward method to exhibit the similarity between target image and predicted sentence by visualizing both captions and images. The result is displayed in Figure 9 and it proves that our model performs well.



(a) Cars

(b) Kitchens

(c) People



(d) Food

Figure 9: Top K Similar Images and Captions

Figure 10 (a) is one of our test image and Figure 10 (b) shows the corresponding top 10 captions.

'**a group of people standing in a field with a frisbee** . <eos>',
'a group of people on a field with a frisbee . <eos>',
'a group of people on a field playing with a frisbee . <eos>',
'a group of people standing around a field catching a frisbee . <eos>',
'a group of young people playing with frisbee on field . <eos>',
'a group of people standing on a field with a frisbee . <eos>',
'a group of people with uniforms playing with a flying disc . <eos>',
'people standing around a field while one holds onto a frisbee <eos>',
'a bunch of people playing frisbee together in a field . <eos>',
'four people playing a game with a frisbee in a grassy area . <eos>',

(a) Test Image (b) Top 10 Captions

Figure 10: Top 10 Captions

5.2 Caption to Story

Condition on the obtained captions in Figure 11 (b), we generate several stories with different temperature hyperparameter values.

You know that area , and a lot of **a couple of a couple of a couple of** the same team , and I ' s **a couple of a couple of a couple of a couple of a couple of** the same team . We park and I saw a couple of the guys , and **a couple of a couple of** the team , and I was **a couple of** the same as **a couple of** the other people around town , and I spotted a couple of the same team , and I was **a couple of** the same as a couple of the same team , and I was **a couple of** a few yards off the same as a couple , and I guess that she was **a couple of** the team . I could be **a couple of a couple of** the game , and I ' s attention . We were **a couple of** the team . I was **a couple of** the other guys in the girls , and **a couple of** the boys , and **a couple of a couple of** the way to see how to see **a couple**.

(a) T=0.2

I dance alone on the park and falls to see and Jerry against the leash . No turn there , which winked at the band crowds as we have been too many stories of **boys hang out of a mile away** from kids weren ' s team . We stuffed animals , as **a couple kids** here for Lauren scene and the road with a bit up resigned team , **Joe and everyone watching Travis** . It's a couple into sports adults off cliff at a couple yards up to be in the most of consternation , a lot at the team **a few students** to join us to Sam for everyone in line and someone else to be **lurking around a squad** around Duncan mob of jogging at the night Moore neared Tessa looks at least that player knew everyone and Matt , **the game** , and **they were performing an athlete** directed toward Becky swings a dozen students and these people watch to be friends in the theatre to play game to planet . " Boys match if it was something else around , and barged into the guys discussed the crowd.

(b) T=0.7

I ' s Golden tails Driving college , electronic at nine o ' s lighter word . Then Rocky listens to help win giggling laughing . Nobody else Austin & Probably if it would be impressed Ellen over and figure Charlie hear about Cal but teacher boulder playing games with her goofy guys Ca not protesting animals the school moves to overhear our fastest strayed from El player led to wake up and it in Sam , who void set tall followers . A dart participated games wherever they could have to see two mic rain takes a pool 's guide off signal and ok outside of St a fascinating distance apart from Minton up in appearances of Germany with Marcy about forty-five Sound rowing ours hunted around town standing at school with speed jumping ahead of tequila run around . Now that glistening mischievous near Eden knows fast playground watch doing a Police attitude anniversary . We smiling down fast march runs through self-defense atmosphere...

(c) T=1.0

Figure 11: Generated Stories with Different Temperature Hyperparameters

5.3 Image to Story

Compared the quality of the generated stories under various temperature hyperparameters (Figure 10), we finally choose $T = 0.7$ and Figure 12 shows the result of another test image.

I recognize someone seriously . Before even though at the night , that someone else , a woman who am there is dressed in the phone tags over thirty minutes at the one of the bride, who notices the least a woman 's friend who am dressed in the recognized being a woman who are wrong and the old woman dressed in the way to see a lot of a street in the sign of a woman ' s in the girl who hits a sight : A woman dressed in the woman dressed in a woman who lets out of sparkling blues enchanted off the girl 's voice mail into this man , busy at least of an hour or someone else , having trouble all the girl whose door , a woman who makes it out of a woman who am dressed with a woman who am dressed a woman in the whole story from the girl 's face dressed in a city standing in the girl 's voice mail up and I knew it in a man who knows a stranger whom I 've been a woman dressed in a woman with a woman dressed

(a) Test Image (b) Generated Stories (T=0.7)

Figure 12: Test Result

6 Discussion

6.1 CNN for feature extraction

We tried out 2 types of CNN, namely VGG19 and Inception V3, which generates 4096 and 2048 dimensions of features, separately. However, in terms of image-sentence embedding and image-sentence similarities from K-Nearest Neighbor (KNN), the feature extracted via VGG19 leads to better performance. This might because that VGG19 keep track yields feature vector of more dimensions(4096) than Inception(2048), thus more useful information is left for feature mapping.

6.2 Representation of image and text in the same manifold

In image-sentence embedding model, we explored the possibility of uniforming heterogeneous format of information into the same manifold, and this simple idea worked well. The idea behind this model is intuitive. No matter given an image of a dog, a sentence of a dog, or the barking sound of a dog, human beings can easily mapping these different information forms into the same concept(dog). So there might be a manifold where all the heterogeneous but conceptually related formats corresponds. And this idea is supported by the nice result of k-NN image-sentence mapping.

6.3 Text generation methods

When using the text decoder to generate story, we unexpectedly found that stochastic generation method works better than beam search. Either argmax search or beam search tends to generate repeated word sequence, while the text generated by stochastic search with temperature makes more sense semantically and grammarly. We assume this is because that the decoder isn't perfectly trained to convergence, and the pre-processing (data washing) of book corpus isn't ideally design as well. Some previous work on this indicates the decoder should be able to generate fluent text via beam search. And this is a point which could be further exploit.

6.4 Shift text style

With the support of out experiment result, we conclude that text style can not only be encoded, but also shifted by adding and subtracting the skip-thought vector representing text style. The idea is intuitive but worked well. This inspired us that not only can the actual content can be encoded into the vector, but also the abstract "style". This fully inspired us that though some concepts seem hard to be described clearly, like style, there actually is such a way, even quantifiable, to identify it.

7 Conclusions

In this project, we train a hierarchical model to generate descriptive stories from images. We first use two CNNs (Inception v3 and VGG-19) and a LSTM respectively to map the images and the captions into the same vector space. Using KNN algorithms, we are able to find the k most descriptive captions for the image. From the captions to story, we use a separate encoder-decoder model. We pass those captions through a skip-vector model and perform style transfer by managing style vectors, then we feed this style shifted encoding vector of the image to a GRU decoder to generate the final stories.

Our results show that this hierarchical model is capable of generating meaningful stories from the images. The model can efficiently captions the images and expands these captions to stories in a reasonable way. Though the final stories still contain grammar mistakes and other violations of English language, the basic structure of the sentence and the target romantic style are sensible. This illustrates that our model has the potential to mimic the human writing style in a better way, and this hierarchical model has a promising prospect in real various fields of applications such as entertainment and education.

8 Restrictions

Few restrictions exist in our model as mentioned above. One restriction is that we lack a quantitative metric for the evaluation of our model. As far as we know, there is not a common metric for this kind

of generation task. Another restriction is that our final stories still contain unreasonable grammar and phrasing. Since the intermediate captions are well-generated, we suspect that the following encoder-decoder structure limits the overall performance of our model. A more advanced implementation like attention modules might be necessary for improving the performance. Additionally, a beam search sampling can replace the temperature sampling to achieve a better performance. Finally, at the very begining, the kNN algorithms can be problematic. The COCO dataset, though very inclusive, cannot be exhaustive, and this will limit the accuracy of our intermediate captions. This can cause poor performance on images with specific contents.

9 Future Work

As mentioned in previous section, we just pick up top-K most similar captions from the set of all training data captions, which means that we are probably unable to pick up proper captions for a fairly rare test image. Some other neural network based methods can be used to generate more accurate captions for the images. Furthermore, our decoder is trained via romantic novels to generate stories. However, other genre of texts such as fantasy, comedy and are also great resource to train a story-generate decoder. Besides images and texts, similar work of associating image and music can be done as well. The combination of images, texts and music will lead to a richer media.

10 Contributions

Changhao Shi:
Code: Implement and train a decoder to generate detailed stories of target style from captions.
Report: Conclusion, Restrictions

Kunyu Shi:
Code: Work on skip thought vectors and write corresponding report.
Report: Abstrat, methods for caption to story.

Shuo Xu:
Code: Implemented image-caption embedding sub-module; Crawled and cleaned BookCorpus dataset; Experimented on encoder-decoder and style shift with Changhao and VGG19 feature extraction with Xinyue.
Report: Discussion

Xinyue Zhang:
Code: Implement image to feature which uses VGG-19, InceptionV3 for image feature extraction.
Report: Image to caption.

Wenrui Liu:
Report: Dataset, Method, PPT preparation

Yang Xu:
Report: Introduction, Related Work, Method, Results, Future Work, PPT preparation

References

[1] D. Gonzalez-Rico and G. Fuentes-Pineda, "Contextualize, show and tell: a neural visual storyteller," *arXiv preprint arXiv:1806.00738*, 2018.

[2] T.-H. K. Huang, F. Ferraro, N. Mostafazadeh, I. Misra, A. Agrawal, J. Devlin, R. Girshick, X. He, P. Kohli, D. Batra, *et al.*, "Visual storytelling," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1233–1239, 2016.

[3] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Advances in neural information processing systems*, pp. 3294–3302, 2015.

[4] L. Fei-Fei, A. Iyer, C. Koch, and P. Perona, "What do we perceive in a glance of a real-world scene?," *Journal of vision*, vol. 7, no. 1, pp. 10–10, 2007.

[5] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *arXiv preprint arXiv:1411.2539*, 2014.

[6] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," *arXiv preprint arXiv:1511.06732*, 2015.

[7] Y. Mori, H. Takahashi, and R. Oka, "Image-to-word transformation based on dividing and vector quantizing images with words," in *First International Workshop on Multimedia Intelligent Storage and Retrieval Management*, pp. 1–9, Citeseer, 1999.

[8] P. Duygulu, K. Barnard, J. F. de Freitas, and D. A. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *European conference on computer vision*, pp. 97–112, Springer, 2002.

[9] R. Datta, J. Li, and J. Z. Wang, "Content-based image retrieval: approaches and trends of the new age," in *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pp. 253–262, ACM, 2005.

[10] D. Forsyth, T. Berg, C. O. Alm, and G. Wang, "Words and pictures: Categories, modifiers, depiction, and iconography," *Object Categorization: Computer and Human Vision Perspectives*, p. 167, 2009.

[11] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, "Every picture tells a story: Generating sentences from images," in *European conference on computer vision*, pp. 15–29, Springer, 2010.

[12] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, "Baby talk: Understanding and generating image descriptions," in *Proceedings of the 24th CVPR*, Citeseer, 2011.

[13] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.

[14] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[17] B. Raj, "A simple guide to the versions of the inception network." `https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202`, 2016. Last accessed 15 March 2019.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] "Transfer learning in tensorflow." `https://mc.ai/transfer-learning-in-tensorflow/`, 2018. Last accessed 15 March 2019.

[20] Y. Bengio, P. Simard, P. Frasconi, *et al.*, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] T. Shen, "Grus vs. lstms." `https://medium.com/paper-club/grus-vs-lstms-e9d8e2484848`, 2017. Last accessed 15 March 2019.

[23] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.