Image Captioning

Yuan Feng, 521030910369
Shanghai JiaoTong University

1. Introduction

Images often contain very rich content, including backgrounds, people, people's actions, objects next to people, and so on. A lot of useful information can be unlocked from a single image.

In the academic field, researchers use various figures and charts in scientific papers to convey complex information. The captions of these figures are critical for conveying effective messages. However, low-quality figure captions commonly occur in scientific articles, which is detrimental to people's understanding of the meaning of the images and the comprehension of the article.

Therefore, whether it is everyday images or figures in academic papers, we hope to use a model, an end-to-end neural network framework, to achieve the goal of inputting an image and automatically generating a sentence to summarize the content of the image.

For example, the image shown in Figure 1 below.

Figure 1. Image Captioning

2. Experiment Tasks Overview

In this project, I primarily conducted the following experiments:

1. I understand how cross-modal models process different modality data, and the overall framework of the encoder-decoder model.

2. I implement the Scheduled Sampling in three different ways and compare the results.

3. I understand the purpose of different evaluation metrics for cross-modal generation tasks.

4. For different metrics, I print the generated samples with the best and worst metric scores, and compare the objective & subjective evaluations.

5. Try modifying the code and hyperparameters to fine-tune the model and improve its performance.

3. Model Framework

We use the classic Encoder-Decoder framework. The Encoder extracts features from the input image, and the Decoder, based on the Encoder's output features and the current word embedding, decodes to generate the next word.

In particular, we use the **CNN+LSTM** [1] architecture as the baseline model. ResNet is used as the image encoder to represent the image as a 512-dimensional vector, i.e., obtaining a $14 \times 14 \times 512$ feature map. Then, this image vector is fed into a fully connected layer to match the dimension of the word embedding and the LSTM decoder. Meanwhile, a global attention mechanism is added to the LSTM decoder to better model the context [2]. The LSTM decoder takes the image vector as the initial state and generates the captions.

The overall framework is illustrated in Figure 2.

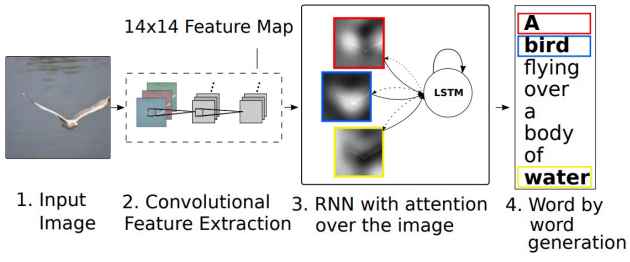


Figure 2. The Framework

The detailed schematic of the Decoder is shown in Figure 3.

4. Metrics

We use several metrics to evaluate this model, including Bleu-n, Rouge, METEOR, CIDEr, SPICE, and SPIDEr.

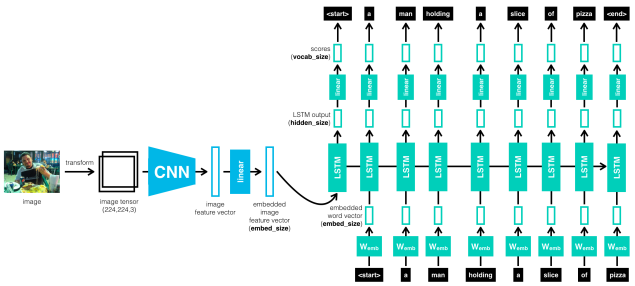


Figure 3. Decoder

It's worth noting that the original code in `evaluate.py` did not include the SPICE evaluation metric, which needs to be manually added.

1. **Bleu**: Bilingual Evaluation Understudy. This metric was originally proposed in the context of machine translation, to evaluate the match between machine-generated translations and human translations. Here the key idea is to utilize N-gram to perform one-to-one matching between the generated sentence and the reference sentence. Bleu-{1,2,3,4} correspond to the 1-gram, 2-gram, 3-gram, and 4-gram, respectively.

2. **Rouge**: It is also used to compare the similarity between the generated text and the reference text, which can be seen as an improved version of BLEU, focusing on recall rather than precision. In other words, it examines how many n-gram word groups in the reference translations appear in the output.

3. **METEOR**: Compared to the two standards mentioned above, METEOR pays attention to generated text that is accurately translated but still does not match the reference text, such as when the generated text uses synonyms of the reference text words. METEOR requires WordNet to expand the set of synonyms and perform word form conversions. In the calculation, it uses the harmonic mean of precision and recall as the evaluation criterion.

4. **CIDEr**: CIDEr is a combination of BLEU and the vector space model. It treats each sentence as a document and calculates the cosine similarity between the TF-IDF vectors (where the terms are n-grams). The similarity of different length n-grams is averaged to obtain the similarity between the generated text and the reference text. Different n-grams have different weights based on their TF-IDF values, reflecting the amount of information they contain. This allows for better evaluation of the match between the two texts.

5. **SPICE** [3]: Semantic Propositional Image Caption Evaluation. This method first requires converting the text into a dependency graph, as shown in Figure 4, which indicates the objects (in red), attributes (in green), and relations (in blue) that are present. We can then obtain the several tuples shown in Figure 5, and the quality of the generated text is determined by the F1 score calculated over the tuples in the reference scene graph.

6. **SPIDEr** [4]: SPICE and CIDEr metrics are better correlated, but have traditionally been hard to optimize for. We use a policy gradient method to directly optimize a linear combination of SPICE and CIDEr (i.e., SPIDEr): the SPICE score ensures our captions are semantically faithful to the image, while the CIDEr score ensures our captions are syntactically fluent.

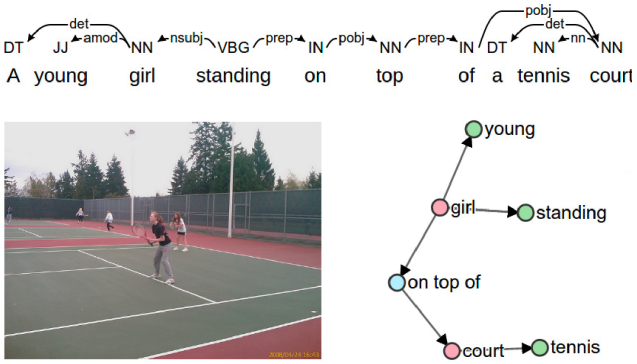


Figure 4. The Dependency Graph

{ (girl), (court), (girl, young), (girl, standing)
(court, tennis), (girl, on-top-of, court) }

Figure 5. Tuples

5. Experiments

5.1. Baseline

Results:

Table 1. Baseline Results

Bleu-1	Bleu-2	Bleu-3	Bleu-4
0.586	0.396	0.264	0.173

Rouge	METEOR	CIDEr	SPICE	SPIDEr
0.403	0.192	0.474	0.138	0.306

5.2. Analysis

We can obtain the highest and lowest scoring examples from the dataset, along with their corresponding image samples, and conduct further analysis.

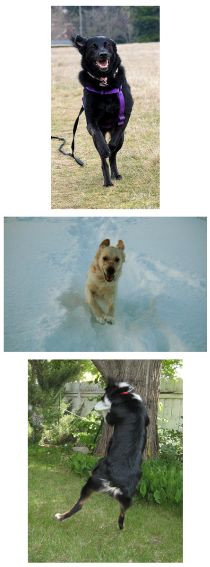
Table 2. Best and Worst Scores

	Bleu-1	Bleu-2	Bleu-3	Bleu-4
Best	1.000	1.000	1.000	1.000
Worst	0.000	0.000	0.000	0.000

Rouge	METEOR	CIDEr	SPICE	SPIDEr
1.000	0.615	4.592	0.737	2.664
0.000	0.019	0.000	0.000	0.000

We can extract specific images and examine them, and find that the examples with high metric scores indeed have comprehensive and accurate captions from a subjective perspective; the examples with low scores lack effective inference of the information in the image. Here are a few representative examples that we have analyzed in more detail. See Figure 6 and 7.

5.2.1 Best



Bleu1,CIDEr,SPICE,SPIDEr-**Best**

Reference:
- a black dog with a purple collar and a black leash runs in the grass
- a black dog with a purple collar is running
- a black dog with a purple collar running
- a black dog with purple collar running
- the black dog is wearing a purple collar with a black leash as he is running

Prediction:
- **black dog with a purple collar is running in the grass**

Rouge-**Best**

Reference:
- a dog is playing in the deep snow
- a dog running through deep snow pack
- a dog runs through the deep snow
- a white dog is running through the snow
- white dog running through snow

Prediction:
- **white dog is running through the snow**

METEOR-**Best**

Reference:
- a black and white dog jumping in the air to catch a frisbee
- a black and white dog jumping in a yard
- a black dog jumps for a frisbee in a backyard with a white fence
- a dog catches a flying disk in a yard with a tree
- a jumping dog in motion

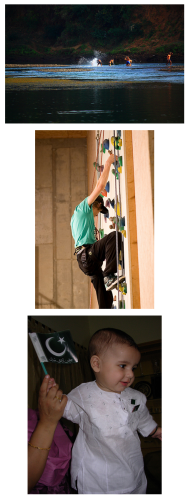
Prediction:
- **black and white dog jumping in the air to catch a frisbee**

Figure 6. Best Cases

We can find that the images with higher scores are mostly of small animals, indicating that the model generates better summaries for images with a single object.

The predictions here generally state the name of the animal in the image, correctly infer the background, and describe the animal's action, which is why they achieve the highest scores.

5.2.2 Worst



Bleu1,Bleu2,Bleu3,Bleu4,Rouge,CIDEr,SPIDEr-**Worst**

Reference:
- a group of friends play in a lake
- five people are in a natural water source
- many children play in the water
- people in bathing suits bend down and reach into the water of a lake
- several people are playing outside in the water

Prediction:
- **man is surfing**

METEOR-**Worst**

Reference:
- a man in a green shirt climbs an indoor climbing wall
- a man in a green shirt is climbing an obstacle course
- a teenage boy climbs an indoor climbing wall
- a teenager climbs a rock wall
- someone climbs an indoor climbing wall

Prediction:
- **skateboarder in the air**

SPICE-**Worst**

Reference:
- a baby in a white garment holds a flag with crescent moon and star
- a baby is holding a small black flag with a moon and a star on it
- a baby wearing a white gown waves a muslim flag
- a little toddler dressed in white is smiling while a lady helps him wave a flag
- baby in white outfit holding black and white flag

Prediction:
- **young boy wearing a white jacket**

Figure 7. Worst Cases

In these images, we can find that the predictions generally fail to infer the correct subject (i.e., the people) - for example, the first image does not indicate there are multiple people, the second image completely fails to identify the person, and the third image describes the baby as a 'young boy'. Therefore, these low-scoring captions are indeed quite poor from a subjective perspective as well.

5.3. Scheduled Sampling [5]

Recurrent Neural Networks can be trained to produce sequences of tokens given some input, and the current approach to training them consists of maximizing the likelihood of each token in the sequence given the current (recurrent) state and the previous token. However, at inference, the unknown previous token is then replaced by a token generated by the model itself. This discrepancy between training and inference can yield errors that can accumulate quickly along the generated sequence.

Therefore, we propose a curriculum learning strategy to gently change the training process from a fully guided scheme using the true previous token, towards a less guided scheme which mostly uses the generated token instead. Experiments show that this approach yields significant improvements.

Specifically, for each token y_t to predict in the i^{th} batch of the training algorithm, we use the true previous token with probability ϵ_i, and the currently generated token with probability $1 - \epsilon_i$.

When $\epsilon_i = 1$, the model is trained exactly as before, while when $\epsilon_i = 0$ the model is trained in the same setting as inference.

We need to come up with a way to gradually decrease ϵ_i from 1 down to 0 or a value close to 0. This way, we can

start the training with a high probability of using the true previous token, and then towards the later stages of training, use the generated token more often.

The gradual decrease of ϵ_i is the essence of the Scheduled Sampling approach.

I have tried three Scheduled Sampling methods.

1. **Linear decay**: $\epsilon_i = max(\epsilon, k - ci)$ where $0 \leq \epsilon < 1$. Here, we consider k to be fixed at 1 and ϵ to be fixed at 0.1, while c is a hyperparameter that can be tuned, for which we experimented with values of 0.05 and 0.02 in our experiments. (Note that ϵ and ϵ_i are different here. ϵ is a small value used to prevent $k - ci < 0$)
2. **Exponential decay**: $\epsilon_i = k^i$ where $k < 1$. Here, k is a hyperparameter for which we tried two different values: 0.95 and 0.8.
3. **Inverse sigmoid decay**: $\epsilon_i = k/(k+exp(i/k))$ where $k \geq 1$. Similarly, k is a hyperparameter, and in this case we tried k=9.

The experimental results are shown in Table 3.

Table 3. Scheduled Sampling Results

	/	L0.05	L0.02	E0.8	E0.95	I9
B1	0.59	**0.63**	0.60	**0.63**	0.62	0.61
B2	0.40	**0.43**	0.41	0.39	0.42	0.42
B3	0.26	**0.28**	0.27	0.24	**0.28**	**0.28**
B4	0.17	**0.18**	**0.18**	0.15	**0.18**	**0.18**
R	0.40	**0.42**	0.41	0.40	**0.42**	0.41
M	0.19	0.19	0.19	0.17	**0.20**	**0.20**
C	0.47	0.50	0.49	0.40	**0.51**	**0.51**
SC	**0.14**	**0.14**	**0.14**	0.12	**0.14**	**0.14**
SD	0.31	0.32	0.31	0.26	**0.33**	0.32

In this table, B1,B2,B3,B4 represents Bleu-{1,2,3,4}, R, M, C, SC, SD represent Rouge, METEOR, CIDEr, SPICE, SPIDEr separately. L, E, I represent three methods of decay, that is, Linear decay, Exponential decay, and Inverse sigmoid decay. The numbers following represent the values of the constant terms in the formula of that method.

As can be seen from the table, the **Linear Decay** with $c = 0.05$ and **Exponential Decay** with $k = 0.95$ methods achieved the best metric scores the most often.

5.4. Different Decoder Dimensions

Finally, I tried changing the dimension of the Decoder, which was originally 256, and attempted to change it to 512. Building upon the **Linear Decay** with $c = 0.05$, I compared the results obtained with the 512-dimensional Decoder to the 256-dimensional one. The comparison is shown in the Table 4 below:

It can be observed that when $dim = 512$, more metrics outperform the case when $dim = 256$. Therefore, increasing the dimension of the Decoder can improve the model performance.

Table 4. Scheduled Sampling Results

	256-dim	512-dim
Bleu-1	**0.632**	0.622
Bleu-2	**0.429**	0.426
Bleu-3	0.284	**0.286**
Bleu-4	0.183	**0.185**
Rouge	**0.417**	0.415
METEOR	0.194	**0.197**
CIDEr	0.501	**0.507**
SPICE	0.139	**0.144**
SPIDEr	0.320	**0.326**

6. Best Result

In summary, considering that metrics such as CIDEr, SPICE, and SPIDEr are more aligned with subjective judgment, I have chosen to use the method with a **Decoder dimension of 512** and **Linear Decay** with $c = 0.05$. The configuration file can be found at `config/best.yaml`.

Results are shown in Table 5.

Table 5. Best Results

Bleu-1	**Bleu-2**	**Bleu-3**	**Bleu-4**
0.622	0.426	0.286	0.185

Rouge	**METEOR**	**CIDEr**	**SPICE**	**SPIDEr**
0.415	0.197	0.507	0.144	0.326

References

[1] Ting-Yao Hsu, C. Lee Giles, and Ting-Hao 'Kenneth' Huang. Scicap: Generating captions for scientific figures, 2021. 1

[2] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation, 2015. 1

[3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation, 2016. 2

[4] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, October 2017. 2

[5] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks, 2015. 3