CSC449-Report
Zhenfei Ji

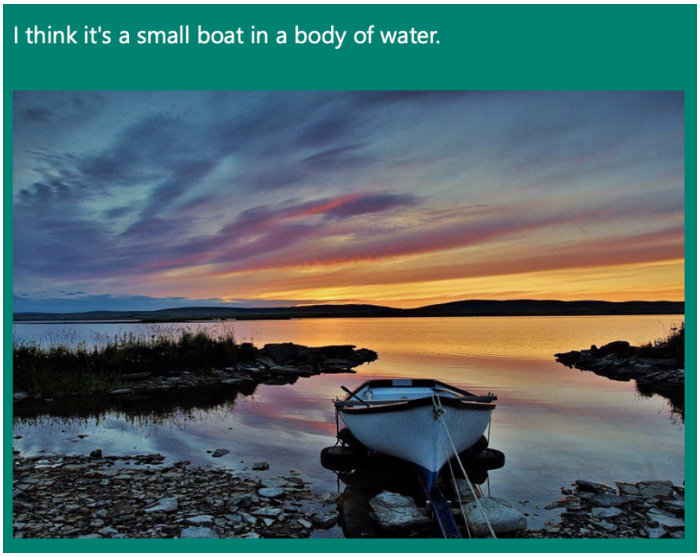
1. **Evaluation scores**

```
Bleu_1: 0.639
Bleu_2: 0.459
Bleu_3: 0.315
Bleu_4: 0.211
computing METEOR score...
METEOR: 0.211
computing Rouge score...
ROUGE_L: 0.489
computing CIDEr score...
CIDEr: 0.716
computing SPICE score...
Parsing reference captions
Parsing test captions
SPICE evaluation took: 3.47
SPICE: 0.134
```

2. **Compete against Microsoft's AI**

(1) Image one

The result of Microsoft's AI:



The result of my model:

```
<start> a beach with a bunch of people on the beach and a large body of water . <end>
```

(2) Image two

The result of Microsoft's AI

The result of my model:

```
<start> a small dog is laying on a green field . <end>
```

(3) Image three

The result of Microsoft's AI:

The result of my model:

```
<start> a woman is sitting on a bench with her dog . <end>
```

(4) Analysis

It should be obvious that Microsoft's AI defeats my model utterly.

Although Microsoft's AI makes some mistakes during captioning, it basically captures the main object of the image, which means the accuracy of object recognition of Microsoft's AI is very high. On the contrary, my model is even not able to capture the main content of an image, it looks like my model is making up the sentence without observing the corresponding image.

3. New ideas

First of all, the accuracy of image classification of model should be improved. In the project, transfer learning is used. Specifically, ResNet-152 is adapted in this model. If I have enough data, it's better to train our model from scratch. Whenever we use transfer learning, our training data should have two options. First, the distribution of the training data which our pre-trained model has used should be like the data that we are going to face during test time don't vary too much. Second, the number of

training data for transfer learning should be in a way that we don't overfit the model. If you have a few numbers of labeled training data, and your typical pre-trained model, like ResNet has millions of parameters, you have to be aware of overfitting by choosing appropriate metrics for evaluation and good testing data which represent the real distribution of the real population.

The low accuracy of the model may be caused by overfitting. So, we can change the architecture of current CNN a little. That is, reduce the number of parameters. Basically, the number of layers is a hyper-parameter that there is no consensus on how to be chosen. We may try different number of layers and measure the accuracy of the model on test data. Besides, there are some other hyper-parameters worth trying, such as learning rate and number of workers in this project.

Some other optimization tools should also be considered. In this project, the SGD with momentum is used. We can try other methods, such as Adam, dropout and RMSProp.

4. Evaluation Metrics for Image Captioning

BLEU from Papineni

Bleu can be defined as an evaluation method for generated sentences with respect to human written sentences. This is also defined as an algorithm for evaluation the quality of text which has been machine-translated (in our case, machine-generated) from one natural language (from image information in our cases) to another by Wikipedia. The result of this method can be mapped between 0 to 1. The result of 1 is human written referential results. The BLEU formula is shown below:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log P_n\right)$$

From which, BP is the brevity penalty and its formula is shown below:

$$\text{BP} = \begin{cases} 1 & if\ c > r \\ e^{(1-r/c)} & if\ c \leq r \end{cases}$$

Where c is the length of the candidate translation (generation) and r is the effective reference corpus length.

And P_n is formulated as:

$$P_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n-gram \in C} Count_{clip}(n-gram)}{\sum_{C' \in \{Candidates\}} \sum_{n-gram' \in C'} Count(n-gram')}$$

n-gram refers to n unigram precision, count means how many the target matches the reference.

One of the main drawbacks of this evaluation method is the weights of words should be different in a sentence. For example, the action, verb word should be the most important word in a sentence. And word like quantity or definite article should be the least in a sentence. But for this method, all words will have the same weights, which might make mistakes about the evaluation.

CIDEr Metric

CIDEr is to automatically evaluate for image I_i how well a candidate sentence c_i matches the consensus of a set of image descriptions $S_i = \{s_{i1}, \dots, s_{im}\}$. And its calculate formula is shown below:

$$\mathrm{CIDEr}(c_i, S_i) = \sum_{n=1}^{N} w_n CIDEr_n\,(c_i, S_i)$$

From which the $CIDEr_n(c_i, S_i)$ can be obtained by:

$$CIDEr_n(c_i, S_i) = \frac{1}{m}\sum_{j} \frac{g^n(c_i) \cdot g^n(s_{ij})}{\|g^n(c_i)\|\|g^n(s_{ij})\|}$$

Where $g^n(c_i)$ is vector formed by all n-grams of length n and $\|g^n(c_i)\|$ is the magnitude of it. This equation is calculation of cosine value of two sentence vectors. And $g_k(s_{ij})$ can be computed by:

$$g_k(s_{ij}) = \frac{h_k(s_{ij})}{\sum_{w_l \in \Omega} h_l(s_{ij})} \log\left(\frac{|I|}{\sum_{I_p \in I} \min\left(1, \sum_q h_k(S_{pq})\right)}\right)$$

From the equation above, Ω is the vocabulary of all n-grams and I is the set of all images in the dataset.

One of the drawbacks of this Metrix is that sometimes there might be unbalanced tf-idf weighting over n-grams. Then this will cause some unimportant words assigned with higher weights, which consequently will cause ineffective and worse caption evaluation.