Research on the image captioning metric method of CIDEr-D

Introduction

Computer vision image captioning is a task that combines computer vision and natural language processing, aiming to generate accurate, fluent, and readable text descriptions to describe the content of images. This problem has broad application prospects, such as speech assistants, smart homes, unmanned driving, and other fields.

When evaluating the quality of computer vision image captions, we need to use various evaluation metrics. These metrics can help us measure aspects such as the accuracy, fluency, and readability of image captions. Through these metrics, we can quickly compare the quality of image captions generated by different algorithms and find the best way to generate them.

Currently, there are many popular image captioning evaluation metrics widely used. The most classic ones include BLEU, ROUGE-L, METEOR, CIDEr, and SPICE.

The BLEU metric is a precision-based metric based on n-gram, which measures translation accuracy by calculating the overlap between n-grams in the reference captions and those in the machine-generated captions. The BLEU metric is widely used in machine translation, and it is easy to implement. In image captioning evaluation, the BLEU metric usually needs to consider multiple reference captions.

The ROUGE-L metric is an l-based metric based on the longest common subsequence, which uses the longest common subsequence to measure the similarity between machine-generated captions and reference captions. The ROUGE-L metric has also been widely used in text summarization and natural language processing. Unlike the BLEU metric, the ROUGE-L metric only needs a single reference caption.

The METEOR metric is a word-matching-based metric that uses word matching to measure translation accuracy. The METEOR metric also considers factors such as synonyms, inflections, and word order to improve the accuracy of evaluation results. The METEOR metric is suitable for scenarios where there are multiple reference captions and can help evaluate translations that are similar but not identical to the reference captions.

The CIDEr metric is a TF-IDF n-gram vector cosine similarity-based metric that is suitable for scenarios with multiple reference captions and can help evaluate translations that are similar but

not identical to the reference captions. Unlike the BLEU metric, the CIDEr metric not only considers the precision of n-grams but also the importance of word frequency.

The SPICE metric is a scene graph similarity-based metric that uses the relationships between concepts in the scene graph to calculate the similarity between reference captions and machine-generated captions. The SPICE metric works best when evaluating captions containing complex scene descriptions. Unlike other metrics, the SPICE metric can evaluate the semantic accuracy of caption generation.

All these metrics have their own characteristics and advantages, and choosing the appropriate metric depends on specific application scenarios and goals. By comparing the results of different metrics, we can better evaluate the quality of image captions, improve the effectiveness and accuracy of image captioning.

CIDEr-D method:

The CIDEr metric is a measure based on the cosine similarity of TF-IDF n-gram vectors and is suitable for scenarios with multiple reference captions. It can help evaluate translated results that are similar but not identical to the reference captions. Unlike the BLEU metric, the CIDEr metric considers not only the precision of n-grams, but also the importance of word frequency. Its emergence greatly improves the accuracy of image caption evaluation.

However, the CIDEr metric also has some limitations. Firstly, because the CIDEr metric only considers word frequency and not more advanced semantic information, its performance is not ideal in cases where there are multiple correct answers. Secondly, repeating a word or having an excessively long sentence that contains all words can obtain a high CIDEr score. These issues pose significant problems for training deep learning models.

To improve the CIDEr method, researchers have proposed considering the influence of sentence length and repeated words on top of improving the CIDEr metric. To this end, CIDEr-D adjusted the formula by introducing a Gaussian penalty term and a mechanism to restrict repeated scores, effectively avoiding situations where disadvantageous sentences receive high scores.

Below is a comparison data set.

Ground Truth:

A man with a red helmet on a small moped on a dirt road.

Man riding a motor bike on a dirt road on the countryside.
A man riding on the back of a motorcycle.
A dirt path with a young person on a motor bike rests to the foreground of a verdant area with a bridge and a background of cloud-wreathed mountains.
A man in a red shirt and a red hat is on a motorcycle on a hill side.

Different sample data:

character	caption	CIDEr	CIDEr-D
Repeated	A man with a red helmet helmet on a small moped moped moped along a dirt road through the countryside.	3.1	2.1
Too long	On a scenic countryside dirt road, a man wearing a red helmet skillfully navigates his small moped while passing a man riding a motorbike and another riding on the back, all while a young	3.7	2.2

	person on a motorbike rests in the foreground of a picturesque verdant area with a bridge and cloud-wreathed mountains in the background.		
normal	A man with a red helmet riding a motor bike on a dirt road on the countryside	2.9	2.7
worse	Man with a red hat is on a small moped on a hill side	1.5	1.3

Through the comparison of the above four examples, it is found that the CIDEr-D evaluation score is generally lower, but it has a good distinction for repeated word scores and long sentences. Moreover, it has a greater difference in evaluating poorer sentences, making it more suitable for evaluating image captions generated by models.

Due to its ability to adapt to different types of datasets and higher accuracy, the CIDEr-D method has become one of the most popular evaluation metrics in the current field of computer vision image captioning.

Research direction

Characteristics of CID-D evaluation method

The CIDEr-D evaluation metric has good characteristics. Firstly, it can distinguish repeated word scores and long sentences well. This is because CIDEr-D introduces a Gaussian penalty to punish long sentences based on the difference in length between candidate sentences and reference sentences. At the same time, the metric further suppresses the

repeated occurrence of specific n-grams by adding clipping to the n-gram counts, effectively avoiding the problem of "gaming" the evaluation metric.

Secondly, unlike other common image captioning evaluation metrics such as Bleu and Rouge, CIDEr-D does not use stemming techniques but retains the original form of all words, avoiding information loss due to language morphology. Therefore, CIDEr-D more accurately evaluates the similarity between the generated captions of the model and the reference captions.

In addition, CIDEr-D changes the way CIDEr is calculated by adding a factor of 10, making its evaluation value similar in magnitude to other metrics. The presence of this factor does not affect the evaluation results but facilitates the comparison and comprehensive consideration of the results of multiple metrics.

Construct very high CIDEr-D captions based on features

As shown in the figure, this formula is used to compute the CIDEr-D evaluation metric. We can approach it from a mathematical perspective to identify possible methods:

$$\text{CIDEr-D}_n(c_i, S_i) = \frac{10}{m} \sum_j e^{\frac{-(l(c_i)-l(s_{ij}))^2}{2\sigma^2}} * \frac{\min(\boldsymbol{g^n}(c_i), \boldsymbol{g^n}(s_{ij})) \cdot \boldsymbol{g^n}(s_{ij})}{\|\boldsymbol{g^n}(c_i)\| \|\boldsymbol{g^n}(s_{ij})\|}, \quad (4)$$

Where $l(c_i)$ and $l(s_{ij})$ denote the lengths of candidate and reference sentences respectively. We use $\sigma = 6$. A factor of 10 is added to make the CIDEr-D scores numerically similar to other metrics.

The final CIDEr-D metric is computed in a similar manner to CIDEr (analogous to Equation 3):

$$\text{CIDEr-D}(c_i, S_i) = \sum_{n=1}^{N} w_n \text{CIDEr-D}_n(c_i, S_i), \quad (5)$$

Gaussian penalty:

The length of the sentence should fall between the shortest and longest ground truth sentences that have CIDEr values close to the highest (the top five sentences before taking the average). Therefore, the construction function should find an appropriate length within the range of the shortest and longest lengths so that the penalty score loss caused by length is less than the final word chosen by the construction.

min() function to avoid repetition:

Repeating words in a sentence not only prevents obtaining higher scores but also increases the length of the sentence. Therefore, the construction function should avoid repeating words and consider the structure of the sentence.

Same n-gram situations:

When constructing new captions, the selected words should come from all grams from 1-gram to 4-gram. If the optimal length for the next sentence is determined to be 2-gram, then among the same length grams, the gram that contributes the most to the CIDEr-D value of the sentence should be selected.

Different n-gram situations:

When constructing new captions, if it is necessary to determine which length of word (1-gram, 2-gram, 3-gram or 4-gram) will bring the highest increase in CIDEr-D value, then the potential impact of different length grams needs to be compared. Too long may result in penalties, while too short may result in low scores, so comparisons need to be made based on different lengths.

Local optimum and global optimum:

If the sentence construction process solely relies on the current CIDEr-D value to select grams, the resulting final sentence will always be a local optimum, rather than the true global optimum. The local optimum depends on the initial sentence and has uncertainty, making it impossible to determine the optimal initial word selection. Therefore, the global optimum needs to be considered by selecting the best initial gram, and then continually constructing grams based on the global length, repetition, and potential scores. The potential score includes the score of the 4-gram, as well as the 3-gram, 2-gram, and 1-gram contained within it. Therefore, when choosing, the constructed part of the sentence needs to be considered, and both ends need to be matched to prevent situations such as "have a" and "a meal" being constructed into "have a a meal". In addition, the construction should also consider the repeating vocabulary at both ends of the sentence, and which end is optimal for the global impact when both can be matched.

Re-training the model using constructed captions:

Through testing on a dataset and statistical analysis, I confirmed that my method can improve data quality. After designing the algorithm, the coco2014 training set needs to be tested. Assuming each group of captions (5 in total) has the highest score c and sentence s, 44,000 sentences were constructed using the algorithm and compared with the original ground truth set to calculate the CIDEr-D value. The goal is for 50% of the new captions to have scores higher than c and for 30% of the new captions to have a CIDEr-D score increase greater than 0.1. This is used to observe the change in evaluation values of the model after re-training.

Replacing the training set with the new data:

After obtaining high-quality new captions that meet the criteria, the highest-scoring sentence in each group of captions from the original training set is replaced with a new caption. Then, using the replaced training set, the self-critical model is trained, and finally, the Karpathy's test split is used to evaluate the model, comparing the changes in CIDEr-D. Under the same self-critical model and feature extraction model, the effectiveness of the new training set on model training is determined.

Confirmation of the effect of constructed captions on model training:

Through Karpathy's test split evaluation, the results of the original model and the model trained with the new training set are compared based on the CIDEr-D values obtained. This determines whether the model trained with the new training set has achieved better performance under the same fc+self-critical model conditions.

Background

Implementation of CIDEr-D method

The implementation process of the CIDEr-D model roughly involves the following steps:

1.Confirm the format of the dataset being read and divide it into corresponding input data.

2.Input the overall data into the CIDEr-D model and convert it into a gram frequency record dictionary.

3.Calculate the score for each group of ground truth and corresponding captions using the mathematical formula, which requires operations such as TF-IDF calculation, normalization, deduplication, and vector multiplication.

4.Average the overall score and output the final evaluation value.

Training the Self-critical model

1.Data preparation: replace the training set data with our own data while ensuring that the format is correct.

2.Begin 30 epochs of fc training.

3.Begin 60 epochs of self-critical model training.

4.Generate captions and evaluate based on Karpathy's test split.

Caption Data for Transformer Models

For subtitles generated by meshed-memory-transformer models, an algorithm is used to generate new subtitles. The improvement in CIDEr-D score and the average improvement percentage compared to the statistics-based comparison are used to determine the improvement that the new subtitles bring compared to the model-generated subtitles.

TF-IDF Definition

The TF-IDF value is used as a weight to measure the importance of each word. The word bank comes from all ground truth, which refers to reference subtitles. By segmenting the reference subtitles, the frequency of each word is counted, and then the TF-IDF weight of each word is calculated. The generated subtitles also need to be segmented and then the TF-IDF weight of each word is calculated. Finally, the TF-IDF value of each word in the generated subtitles is matched with the corresponding TF-IDF value of the word in the reference subtitles, and their similarity is calculated.

Cosine Distance Definition

The cosine similarity algorithm is used to measure the semantic similarity between the generated subtitles and the ground truth. The cosine similarity algorithm is a common text similarity measurement method. Its basic idea is to represent text as a vector and calculate the cosine value of the angle between vectors to obtain the similarity between texts. Specifically, in the CIDEr-D evaluation, first, for each ground truth and generated subtitle, their n-gram frequency vectors are calculated separately. Then, these vectors are normalized to a length of 1 to eliminate the influence of vector length on similarity calculation. Finally, based on the cosine similarity formula, the similarity score between each ground truth and generated subtitle is calculated.

Gaussian Penalty Definition

Gaussian penalty is a commonly used technique in natural language generation that prevents sentences from being too long or too short, thus making the model biased towards generating sentences of appropriate length. This technique adds a Gaussian penalty term to control the length of the generated text.
In the specific implementation, the Gaussian penalty restricts the length of the generated sentence by introducing a random number from a normal distribution. The random number is adjusted based on the length of the text, so that longer or shorter sentences are punished to varying degrees. This makes the generated sentence length better match the distribution characteristics of the original dataset text length, thereby improving the model's generalization ability and robustness.
In addition to controlling the sentence length, Gaussian penalty can also further control the diversity and readability of sentences by adjusting parameters such as the variance and mean of the normal distribution.

Definition of Ground Truth:

Ground truth refers to the standard reference answer. In the task of caption generation, ground truth is usually one or more captions manually written by human experts based on the image content. These captions are considered as the "true" answers that correctly and completely describe the given image.
In the CIDEr-D evaluation model, five ground truths are typically used as references in order to cover multiple perspectives for describing the same image. Each ground truth is described from a different language style or cultural perspective. This not only improves the accuracy of evaluation results but also provides useful information for related tasks such as machine translation and cross-language generation.
It is important to note that different collections of ground truth can have a significant impact on the quality of generated captions. If the used collection of ground truth is not comprehensive enough or does not match the generated captions, it can lead to distorted system evaluation. In addition, the quantity and quality of the ground truth collection also directly affect the results of system evaluation.

Definition and source of dataset:

N-gram definition: N-gram is a commonly used text representation method. N-gram refers to the method of representing and processing all consecutive n words or characters in a piece of text as a basic unit.
COCO dataset:COCO (Common Objects in Context) is a large-scale object detection, segmentation, and captioning dataset. It was first introduced in 2014 and has since become a standard benchmark for many computer vision tasks.
In caption evaluation, 1-gram to 4-gram are usually used for evaluation. 1-gram refers to the frequency of occurrence of individual words, 2-gram refers to the frequency of occurrence of adjacent two words, 3-gram refers to the frequency of occurrence of adjacent three words, and so on. Using the n-gram model, generated captions can be evaluated, and their similarity with the standard captions generated by humans can be calculated.

Similar References

1.GitHub - ruotianluo/self-critical.pytorch: Unofficial pytorch implementation for Self-critical Sequence Training for Image Captioning. and others.
2.GitHub - aimagelab/meshed-memory-transformer: Meshed-Memory Transformer for Image Captioning. CVPR 2020
I referenced specific implementation procedures and evaluation data, but for the generation of new captions using our algorithm, no references were used as this was a completely original contribution.

Problem Definition

How to implement CIDEr-D method

1. Algorithm Design

a)Core concept design

The CIDEr-D algorithm evaluates the scores of all n-grams (1-gram to 4-gram) based on their local and global importance, taking into account both local coherency (such as word choices and connections) and the ultimate global score. In each iteration, the algorithm selects the top 10 n-grams with the highest scores, randomly choosing one from among them, and constructs a sentence based on the previously selected n-grams. This process is repeated until the constructed sentence's CIDEr-D score decreases. By repeating this method around 100 times, the algorithm can generate better captions for 50% of the sentences.

b)Local design

Special evaluation:

1.Estimation of scores for n-grams of the same length

Scores are estimated for n-grams of the same length, such as "on a" and "on the," as well as "a red helmet" and "on the road." This can be simplified using a formula:

$$\text{CIDEr-D}_n(c_i, S_i) = \frac{10}{m} \sum_j e^{\frac{-(l(c_i)-l(s_{ij}))^2}{2\sigma^2}} * \frac{\min(\boldsymbol{g^n}(c_i), \boldsymbol{g^n}(s_{ij})) \cdot \boldsymbol{g^n}(s_{ij})}{\|\boldsymbol{g^n}(c_i)\| \|\boldsymbol{g^n}(s_{ij})\|}, \quad (4)$$

In the case of equal length, the modulus is equal, so the comparison score is mainly based on min(g n(ci), g n(sij)) · g n(sij).

2.For different length gram scores estimation:

Because the word selection is based on existing sentences, the difference in Gaussian penalty between 1-gram and 4-gram is small and can be almost ignored. Adding with the longer length of 4-gram, |g n(ci)||g n(sij)| is larger, and higher length grams can be balanced by multiplying them with a parameter of 1.0-1.2. Therefore, the evaluation score can be simplified to min(g n(ci), g n(sij)) · g n(sij)*weight.

3.To calculate potential scores

For grams with length greater than 1, such as "a red helmet," all its 2-grams and 1-grams need to be added, such as "a red," "red helmet," "a," "red," and "helmet." Because CIDEr-D calculation is based on all grams, shorter grams' potential scores need to be considered when selecting 2, 3, 4-grams.

4.Elimination of Duplicate Scores

When selecting words for a sentence, if the selected sentence is "a red helmet on the road," and the next selection is "on the countryside," then the scores for the words "on" and "the" need to be subtracted to avoid duplicate scoring and ensure that the evaluated score conforms to the combined pattern.

5.Edge Score

For example, in the sentence "a red helmet on," if the selected words are "on the road," then the word "on" has already been selected, and the score brought by "on" should not be evaluated again because after edge combination, the vocabulary "on" is equivalent to not being selected.

Random Selection:

1.Based on Score Random Selection:

For all grams, random selection is based on the proportion of scores assigned to each possibility. For example, if the total score for all grams is 100, and the score for "red helmet" is 3, then the probability of selecting it is 3%. If the score for "a red helmet" is 2.5, then the probability of selecting it is 2.5%. Therefore, "red helmet" has a higher chance of being selected than "a red helmet", but both have a chance of being selected in multiple repetitions. Grams randomly selected based on scores are used for the next step of sentence concatenation.

Sentence Combination:

1.Edge Matching:

If the sentence "a red helmet" has already been constructed, and the next gram selected is "a red helmet on" or "red helmet on" or "helmet on", then the final result will be "a red helmet on". To avoid repeating words when linking sentences, the repeated edges need to be identified and trimmed before concatenation.

2.Edge Selection:

If the already constructed sentence is "a red helmet on" and the next selected gram is "on a", then there are two ways to concatenate the sentence to form either "a red helmet on a" or "on a red helmet on". There are three methods to resolve this: 1. Randomly select an edge that can be concatenated. 2. Choose the position where the CIDEr-D score after concatenation is the highest. 3. List all concatenation possibilities as a queue and try to form complete sentences with the highest scores for each possibility. I use methods 2 and 3 considering sufficient randomization and infrequent occurrence. In actual algorithm processing, method 2 is used to select the concatenation position based on the highest CIDEr-D value.

c)Overall Design

Repeated Algorithm:

Since random selection based on score proportions is used for sentence selection, multiple repetitions are necessary to obtain good results. Random selection avoids the time-consuming and unnecessary computations caused by using depth-first trees for permutations and combinations. Repeating 50-100 times can yield better results than trying 20,000 times with depth-first search. The time for each algorithm using depth-first search ranges from 10-20 minutes, while each random selection only takes about 0.1 seconds. Repeated 50-100 times, it takes about 7 seconds, which is more efficient and has a minimal effect on quality. The CIDEr-D error is not more than 0.1.

Segmented Repetition:

Based on the scores obtained from randomly generated subtitles, evaluate whether to continue generating them. If the score is higher than 1.3 times the expected value, run a maximum of 50 times. If it is higher than 1.1 times, run a maximum of 100 times. If it is higher than 1.0 times, run a maximum of 200 times. A limit of 500 times is set to prevent continuing the algorithm after finding very good subtitles. Through prior data analysis, terminating the repetition process early can still find very good results.

Retain the Best:

During the repeated algorithm process, the CIDEr-D value is calculated for each generated subtitle. The subtitle with the highest CIDEr-D value is retained.

2.Algorithm optimization

a)Data reading

1.For the process of calculating TF-IDF, it is possible to calculate it in advance and save it as a file, then read it directly without recalculating every time when computing the CIDEr-D value. Since the amount of data is too large, the calculation takes a long time. Reading it once can speed up the process, and the overall data only needs to be read once. The advantage is that only one reading is required for hundreds of thousands of subtitles, regardless of the amount of data, significantly improving the speed, but it requires more memory consumption.

2.When estimating the evaluation value of ground truth, it needs to be modified, so an additional global cider model is saved for calculation purposes. Another cider is dynamic and based on ground truth that can be modified, which can reduce the number of cider instances and improve computation speed.

3.Exporting data files after each calculation will affect the overall speed, so it is better to export and save them after every 100 pieces of data are completed. Each time, the saved data is combined with the previous data, reducing the frequency of saving and improving overall computational efficiency.

b)Multi-process optimization

Using multi-processes can parallelize the computation of the dataset and improve efficiency. For data ordering issues, multiple files can be saved and then merged according to the order of the original dataset. This can improve computational efficiency.

How to construct new subtitles with excellent CIDEr-D value:

Test statistics comparison + algorithm improvement: For each designed algorithm, use 1000 groups of subtitles to test the average improvement rate. Only about 50% of subtitle groups that achieve improvements can have significant comparisons. If not up to standard, the weight of word selection needs to be redesigned to balance local and global optimality and find a better algorithm that can generate new subtitles with high CIDEr-D values.

How to generate test data:

The algorithm processes the data in batches. The well-organized 44,000 test dataset is divided into 10 equal parts. For each group of subtitles, the algorithm generates corresponding new subtitles and replaces the subtitle with the highest CIDEr-D value within the original subtitle group. If the newly generated subtitle is not better than the original one, it will not be replaced.
When generating new subtitles, the algorithm generates 50 results for each group of subtitles and selects the highest score. If the score is more than 1.3 times the highest score, the algorithm stops, assuming that a good enough subtitle has been found. If not, it generates 100 times and stops if the score is 1.1 times higher than the highest score. The same process is followed for 200 times, and if the score is at least 1.0 times higher, it stops.

Otherwise, it runs a maximum of 500 times to ensure that the majority of subtitle groups produce better subtitles.

How to prove that the model trained on constructed data can achieve better evaluation scores:

Comparative experiments are conducted by training the model only by replacing the original training set. Using the same features and training epochs, the final model is evaluated on Karpathy's test split, and the scores are compared.

Implementation

Implement new algorithm

```
--------------------------------------max--------------------------------------------------------
man riding on the back of a motorcycle on a dirt road                  score=2.32434
--------------------------------------max--------------------------------------------------------
man riding on a dirt road on the back of a motorcycle                  score=2.36557
--------------------------------------max--------------------------------------------------------
red man riding a motor bike on a dirt road on the countryside          score=2.34881
--------------------------------------max--------------------------------------------------------
a dirt road a man riding on the back of a motorcycle                   score=2.08997
--------------------------------------max--------------------------------------------------------
a man riding on the back of a motorcycle                               score=2.21810
--------------------------------------max--------------------------------------------------------
man riding back of a motorcycle on a dirt road on the countryside      score=2.13427
--------------------------------------max--------------------------------------------------------
a man riding on the back of a motorcycle on a dirt road                score=2.32130
--------------------------------------max--------------------------------------------------------
a red helmet on a small moped on a dirt road on the countryside        score=2.52428
--------------------------------------max--------------------------------------------------------
motor man with a red helmet on a small moped on a dirt road on the     score=2.26973
--------------------------------------max--------------------------------------------------------
motor bike man a red helmet on a small moped on a dirt road on         score=2.16245
```

Result from ground truth set:

A man with a red helmet on a small moped on a dirt road.
Man riding a motor bike on a dirt road on the countryside.
A man riding on the back of a motorcycle.

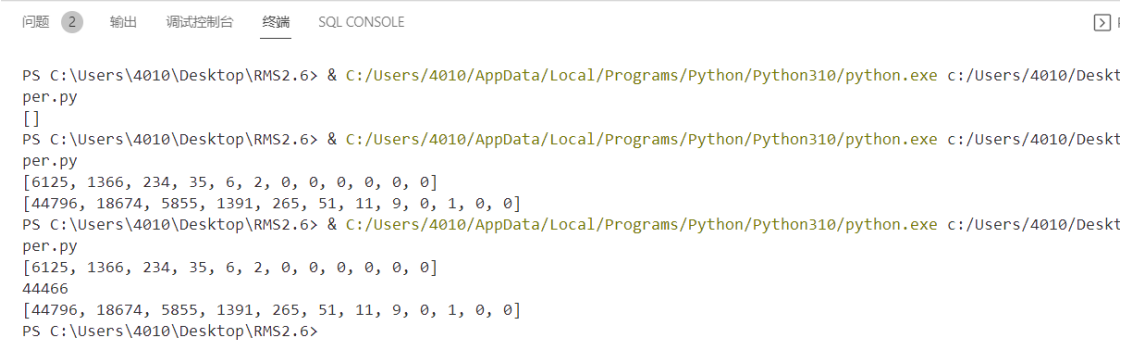
A dirt path with a young person on a motor bike rests to the foreground of a verdant area with a bridge and a background of cloud-wreathed mountains.
A man in a red shirt and a red hat is on a motorcycle on a hill side.

where best score is 2.52428

The test data of 120k from training set

The following data is the captioning generated by the new algorithm, and compared to the original highest score caption.

```
for i in id_all:
        entry = dataset1[i]
        if entry["Re_CIDEr"]<entry["Gt_CIDEr"]:
                delta = int((entry["Gt_CIDEr"]-entry["Re_CIDEr"])*10)
                cider_decrease[delta] = cider_decrease[delta]+1
        elif entry["Re_CIDEr"]==entry["Gt_CIDEr"]:
                cider_nochange+=1
        elif entry["Re_CIDEr"]>entry["Gt_CIDEr"]:
                delta = int((entry["Re_CIDEr"]-entry["Gt_CIDEr"])*10)
                cider_increase[delta] = cider_increase[delta]+1
```



All 123287 records, 44466 records' highest Cider-D no changes, like (not directly replace, is the result from 100 times running)


```
"538018":{
    "Re_caption":"two dogs in the street near two cars",
    "Re_CIDEr":2.503307896228929,
    "Gt_caption":"two dogs in the street near two cars",
    "Gt_CIDEr":2.503307896228929
},
```

Increase like:


```
"580983":{
    "Re_caption":"skateboarder riding a rail on a skate board doing a trick",
    "Re_CIDEr":2.9853530086655233,
    "Gt_caption":"a skateboarder riding a rail on a skateboard",
    "Gt_CIDEr":2.6990380564180527
},
```

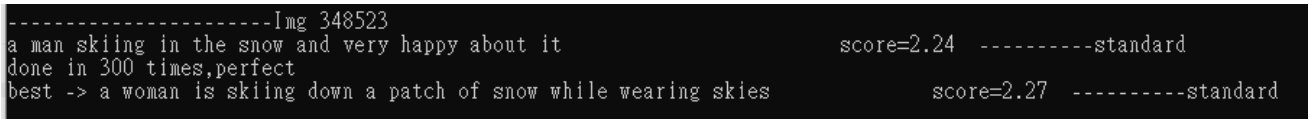

Decrease like:


```
"400044":{
    "Re_caption":"a snowboard into the air next to a man on skis' doing tricks",
    "Re_CIDEr":2.367528077001402,
    "Gt_caption":"a person riding a snowboard into the air next to a tall building",
    "Gt_CIDEr":2.379258858680829
},
```


Statistics of all the data:

Cider change	0-0.1	0.1-0.2	0.2-0.3	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	sum
Decrease	6125	1366	234	35	6	2	0	0	0	0	7768
Increase	44796	18674	5855	1391	265	51	11	9	0	1	71053

Lucky draw 523 entries from 52200+ bad processed sentence, and then process for 500 times (early ended at 300 times once better than standard)



Only a little sentence changes a lot and cider-d value increases, and almost 8% can be improved within 0.1 cider value.

```
[40, 4, 0, 0, 0, 0, 0, 0, 0, 0]
432
[36, 7, 1, 0, 0, 0, 0, 0, 0, 0]
```

(Scale 0.1 CIDEr value gap -> 1 index,like 40 for 0.1 value increase, 4 for 0.2 increase, 36 for 0.1 decrease)

with replacement, final dataset:

Cider change	0-0.1	0.1-0.2	0.2-0.3	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	sum
Decrease	6125	1366	234	35	6	2	0	0	0	0	7768
Increase	44836	18678	5855	1391	265	51	11	9	0	1	71097

Comparison model evaluation

	original model	new training model
cider after 30 epochs of fc	0.918	0.918
cider-d after 60 epochs of self-critical	1.045	1.055

After the self-critical training, the performance of the model is improved around 0.01 in CIDEr value evaluation of Evaluation on Karpathy's test split.

Conclusion

1. My method currently improves the evaluation score for self-critical networks, and it is a significant improvement, essentially surpassing the tolerance error.

   Processing the training set data with this algorithm can improve the evaluation score of the model, and replacing all training set data with the top five subtitles generated using this algorithm can bring even greater improvement, effectively improving the model training. The main function of the algorithm is to preprocess the training set data into sentences with suitable lengths and relevant keywords, making the description of the training set data more appropriate and better suited for the model training, and more in line with the evaluation rules.

2. My method can provide more high-quality subtitles for limited captions.

   For limited subtitle descriptions of images, my algorithm can generate more similar high-quality subtitles, increase the quantity of data, provide more reference subtitles, and generate new test set data that can be used to fully test the quality of the model evaluation.

3. My method can also test the quality of subtitle sets and identify subtitle sets that are not conducive to model training.

   It is noteworthy that out of the 120k data points, only 44k saw improvements after using my algorithm, and the remaining 76k data points are very difficult to generate better subtitles. This means that using such data for deep learning model training will limit the model's creativity because the generated subtitles will be infinitely close to the highest evaluated subtitles in the dataset rather than attempting new ones. On the other hand, datasets that can produce higher-quality subtitles are more likely to give positive feedback to deep learning models, resulting in positive effects. Instead of limiting themselves to imitating existing sentences, these datasets actively learn from the strengths of the top five sentences to construct sentences that are closer to the overall theme.

Reflection in RMS

In this RMS activity, I have learned many important skills and knowledge. Firstly, I have recognized the importance of the research process. By designing research, collecting data, determining research directions, conducting experiments and other related processes, the entire research process can be more organized and efficient, which further enhances the accuracy and credibility of the research results.

In addition, I also acquired knowledge in areas such as computer vision, natural language processing, and deep learning. These skills are crucial in the current scientific research field, which can help us better handle and analyze large-scale data, thus leading to more accurate conclusions. At the same time, these skills also provide me with more options and possibilities for future research.

During this activity, I also realized the importance of scientific collaboration. Only through sufficient communication and cooperation with other researchers, can we mine valuable information and insights from the data. Therefore, I will pay more attention to teamwork in my future research endeavors and actively participate in relevant scientific forums and seminars to expand my academic and social circle.

Most importantly, through this activity, I have gained a deeper interest in scientific research. I found that by maintaining curiosity, actively learning, and constantly exploring, we can continuously develop and achieve better results in the scientific research field. Therefore, I will focus more on cultivating interests and exploring new fields and directions in my future research.

In conclusion, through this RMS activity, I have obtained valuable experience and knowledge that will undoubtedly inspire and guide my future research. I would like to express my gratitude again to Prof. Antoni B. Chan, Dr. Wang Jiuniu, and the organizers of the RMS activity for providing me with this opportunity and support. This will definitely be a precious experience for my future scientific research.

References

CIDEr-D:

Vedantam, R., Lawrence Zitnick, C., & Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4566-4575).

CIDEr-D github project:

GitHub - ruotianluo/cider at e9b736d038d39395fa2259e39342bb876f1cc877
GitHub - ruotianluo/coco-caption at ea20010419a955fed9882f9dcc53f2dc1ac65092

Self-critical github project:

GitHub - ruotianluo/self-critical.pytorch: Unofficial pytorch implementation for Self-critical Sequence Training for Image Captioning. and others.

Meshed-memory-transformer github project:

GitHub - aimagelab/meshed-memory-transformer: Meshed-Memory Transformer for Image Captioning. CVPR 2020

Information:

Name- JIANG XIN, SID-56643608, Major-Computer Science
Supervising professor: **Prof. Antoni B. Chan**
Supervising PHD: WANG JIU NIU