1. LSTM-based Language Decoder

After training, based model can produce a caption for a bird image.

The output of the bird image is: **a bird sitting on a branch of a tree.**

Run eval.py, the output is:

```
{'reflen': 972, 'guess': [986, 886, 786, 686], 'testlen': 986, 'correct': [610, 262, 104, 38]}
ratio: 1.01440329218
Bleu_1: 0.619
Bleu_2: 0.428
Bleu_3: 0.289
Bleu_4: 0.191

METEOR: 0.196

ROUGE_L: 0.456

CIDEr: 0.671

SPICE: 0.133
```

2. **Implement My LSTM**

After training, my model can produce a caption for a bird image.

The output of the bird image is: **a close up of a bird perched on a tree branch.**

Run eval.py, the output is:

```
{'reflen': 953, 'guess': [960, 860, 760, 660], 'testlen': 960, 'correct': [612, 267, 94, 32]}
ratio: 1.0073452256
Bleu_1: 0.637
Bleu_2: 0.445
Bleu_3: 0.290
Bleu_4: 0.186
```

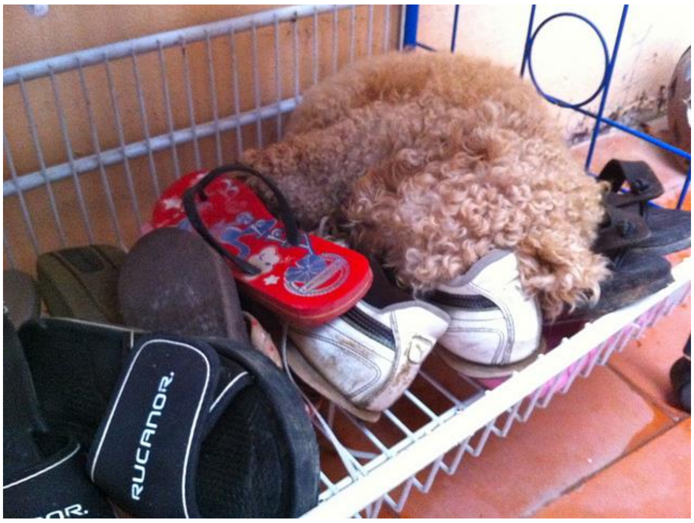
METEOR: 0.192

ROUGE_L: 0.456

CIDEr: 0.682

SPICE: 0.126

3.**Compute against Microsoft's AI**

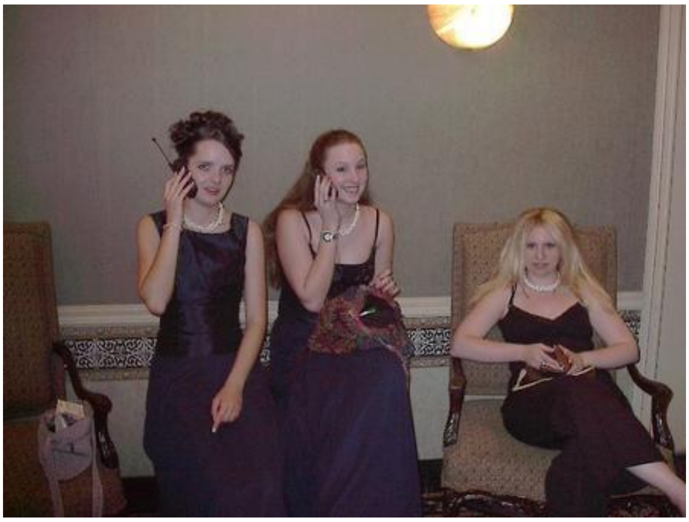


Test image 1: COCO_val2014_000000000042.jpg

Microsoft's AI: **a dog sitting in a basket.**

My model: **a teddy bear sitting on a couch with a teddy bear.**

Explanation: In this example, Microsoft's AI performs better. My model recognizes a teddy dog as a teddy bear.

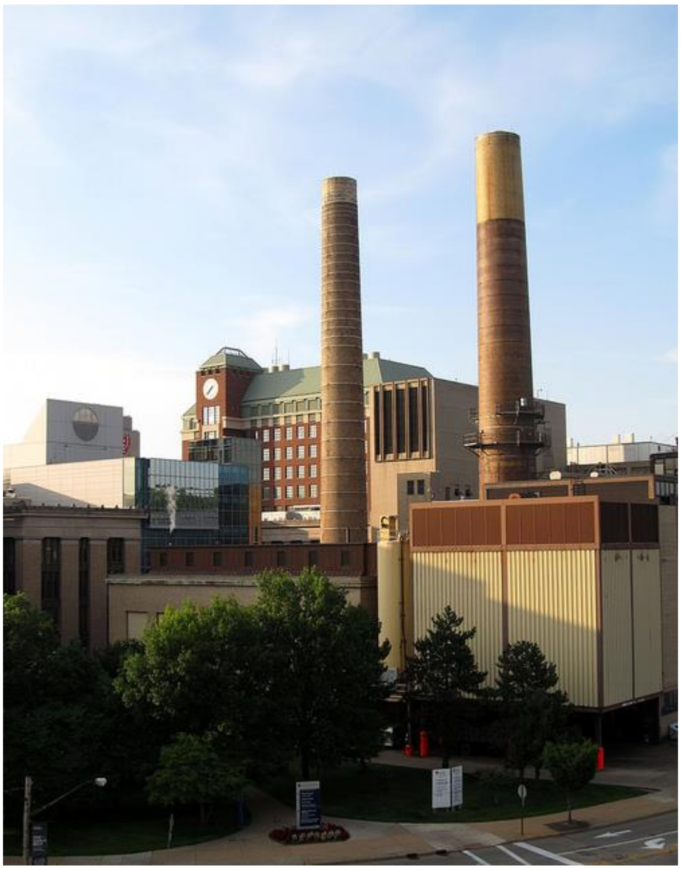


Test image 2: COCO_val2014_000000000536.jpg

Microsoft's AI: **a couple of people that are standing in a room.**

My model: **a woman is holding a cell phone in her hand.**

Explanation: In this example, however, I guess my model performs better. First of all, not all the people are standing, so Microsoft's AI makes a mistake. Second, my model recognizes "woman" while Microsoft's AI just recognizes "people". Moreover, my model recognizes the woman holds a cell phone, while Microsoft's AI doesn't.



Test image 3: COCO_val2014_000000000873.jpg

Microsoft's AI: **it's a tall building in a city.**

My model: **a group of people walking around a city street.**

Explanation: In this example, Microsoft's AI performs better. My model is totally wrong, because there is no people in the image, but building and street.

4.**New ideas**

We can use Adversarial Training.

Define a discriminator D. We have can define original captioning model as a generator G.

First of all, we can pre-train G and D.

We set the label of generative sentences as 0, and set the label of ground truth sentences as 1 for discriminator. In this step, we just update the parameter of discriminator.

When training G, we set the label of generative sentences as 1 and feed them into discriminator. We backpropagate the loss of discriminator to update the parameter of G (but keep parameters of D).

Evaluation Metrics for Image Captioning

1. BLEU (bilingual evaluation understudy) is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. It uses a modified form of precision to compare a candidate translation against multiple reference translations.

   Limitation of BLEU:

   1) There is no guarantee that an increase in BLEU score is an indicator of improved translation quality.
   2) The approach of comparing by how much a computer translation differs from just a few human translations is flawed.

2. METEOR (Metric for Evaluation of Translation with Explicit Ordering) is a metric for the evaluation of machine translation output. The metric is based on the harmonic mean of unigram precision and recall, with recall weighted higher than precision.

   Limitation of METEOR:

   1) Compare the translation with each reference separately and select the reference with the best match, which uses multiple reference translations in a weak way.
   2) Once all the stages have been run, unigrams mapped through different mapping modules are treated the same.