



```

decoder = MeshedDecoder(len(text_field.vocab), 54, 3, text_field.vocab.stoi['<pad>'])
model = Transformer(text_field.vocab.stoi['<bos>'], encoder, decoder).to(device)

data = torch.load('meshed_memory_transformer.pth')
model.load_state_dict(data['state_dict'])

dict_dataset_test = test_dataset.image_dictionary({'image': image_field, 'text': RawField()})
dict_dataloader_test = DataLoader(dict_dataset_test, batch_size=10, num_workers=0)

pred, true, scores = predict_captions(model, dict_dataloader_test, text_field)

```

Meshed-Memory Transformer Evaluation

Evaluation: 100%|██████████| 1/1 [00:00<00:00, 4.96it/s]

In [ ]:

```

# load and display image with annotations

I = io.imread('http://images.cocodataset.org/val2014/COCO_val2014_000000'+ str(img_id) + '.jpg')
plt.imshow(I); plt.axis('off')
ax = plt.gca()
annIds = coco.getAnnIds(imgIds=img_id, iscrowd=None)
anns = coco.loadAnns(annIds)
coco.showAnns(anns)

```

A woman eating vegetables in front of a stove.  
 A woman forks vegetables out of a bowl into her mouth.  
 Woman eating an assortment of mixed vegetables in a bowl.  
 A young woman standing in a kitchen eats a plate of vegetables.  
 A woman eating fresh vegetables from a bowl.



In [ ]:

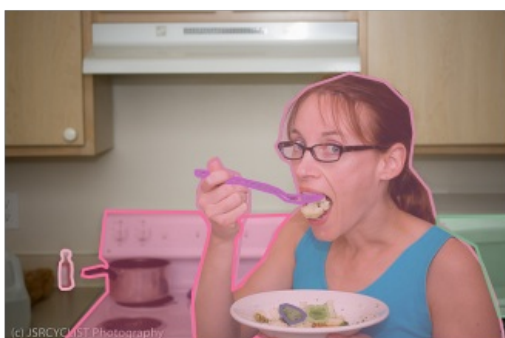
```

# load and display image with instance annotation
annFile = 'annotations/instances_val2014.json'
coco_caps=COCO(annFile)
annIds = coco_caps.getAnnIds(imgIds=img_id);
anns = coco_caps.loadAnns(annIds)

plt.imshow(I); plt.axis('off')
annIds = coco_caps.getAnnIds(imgIds=img_id, iscrowd=None)
anns = coco_caps.loadAnns(annIds)
coco_caps.showAnns(anns)

```

loading annotations into memory...  
 Done (t=4.82s)  
 creating index...  
 index created!



In [ ]:

```
print("Predicted Annotation: {}".format("".join(list(true.values())[0])))  
  
print("\nScores:\n{}".format(scores))
```

Predicted Annotation: a woman eating a bowl of food in a kitchen

Scores:

```
{'BLEU': [0.5999999999400001, 0.36514837163156744, 0.25543647743589953, 3.9281465085347674e-05], 'METEOR': 0.2332  
8211650703048, 'ROUGE': 0.5313588850174217, 'CIDEr': 0.0}
```

Loading [Mathjax]/extensions/Safe.js