



图像Caption-看图说话

指导教师：张华平

汇报人：刘云皓，李桐，邱小尧，刘天锐，王中琦



北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

德以明理 学以精工



目录

- 一、Image Caption 简介
- 二、数据集和评价指标
- 三、经典综述
- 四、前沿技术1
- 五、前沿技术2
- 六、技术展示



Image Caption 简介

汇报人：刘云皓



- 篮球场上有三个小人，一个人在灌篮，两个人在对抗

看图说话！！



CV+NLP



What color are her eyes?
What is the mustache made of?



How many slices of pizza are there?
Is this a vegetarian pizza?



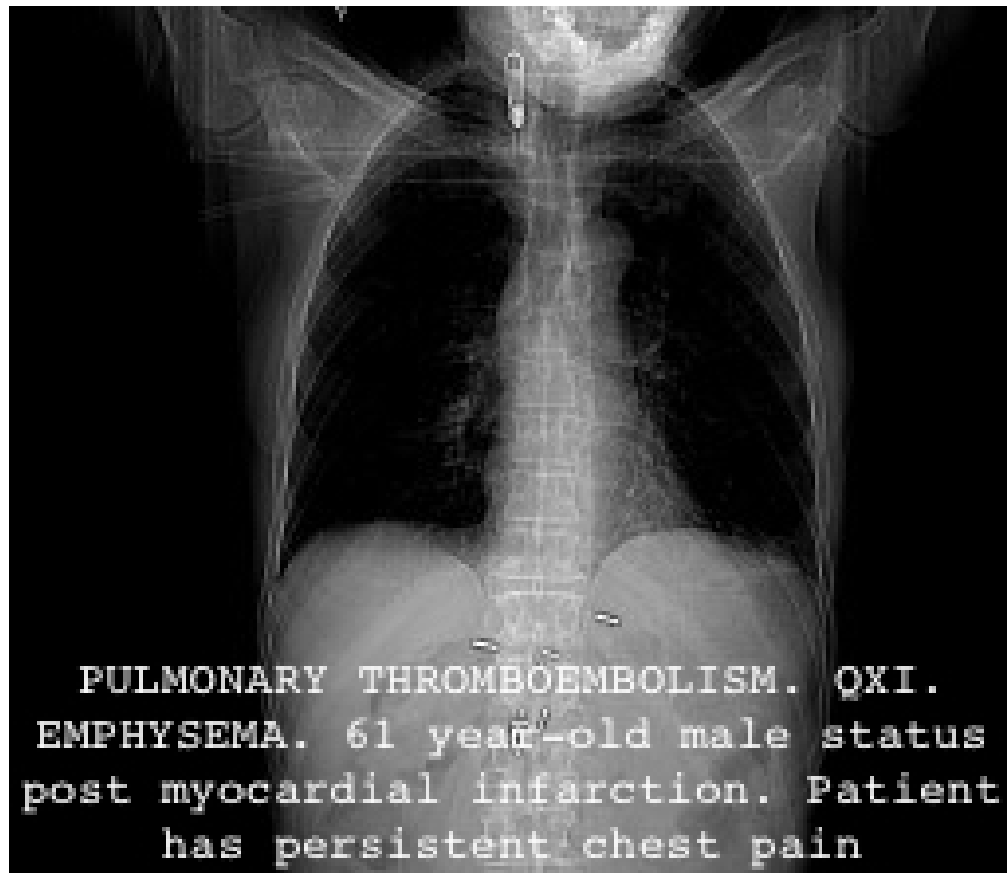
Is this person expecting company?
What is just under the tree?



Does it appear to be rainy?
Does this person have 20/20 vision?

视觉问答 (VQA)

图片源自 [VQA: Visual Question Answering](#)



医疗图像描述

图片源自 [A survey on biomedical image captioning](#)

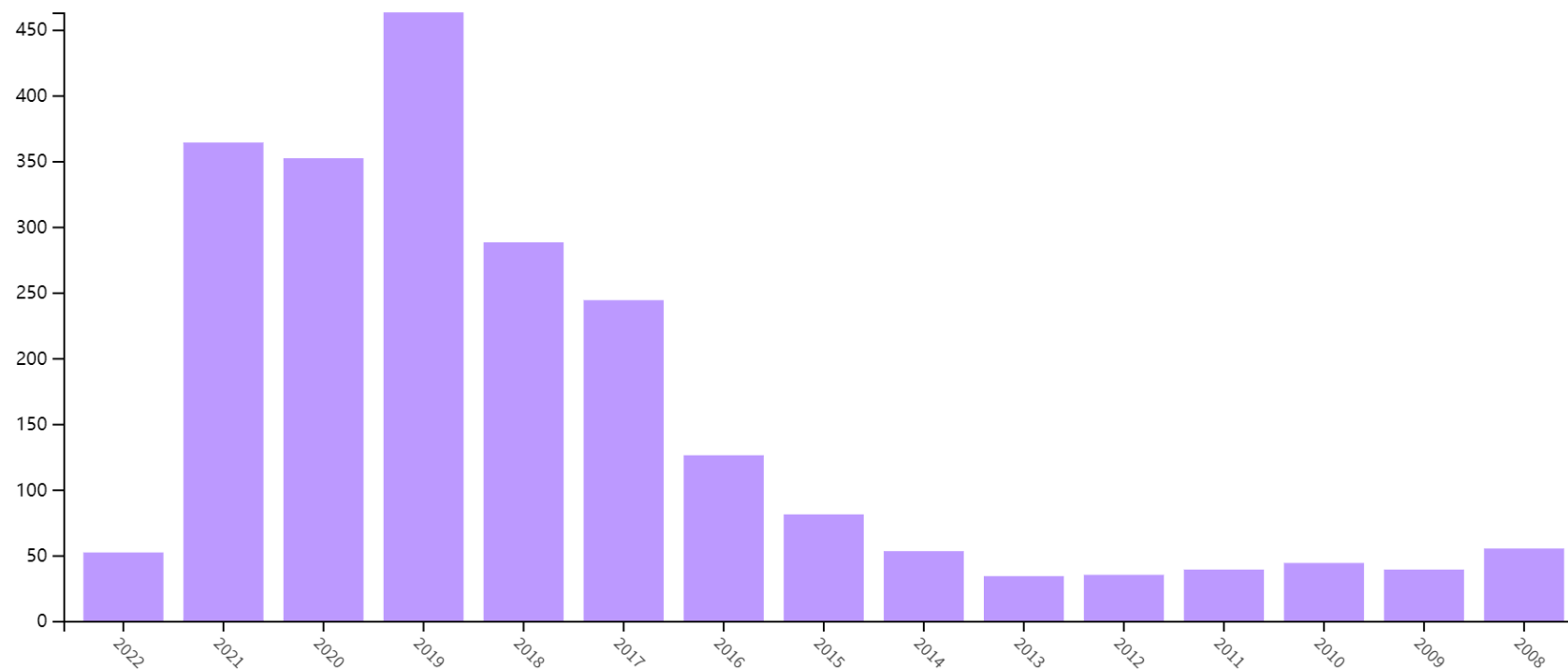


A digital thermometer resting on a wooden table, showing 38.5 degrees Celsius.

针对视障人群的辅助技术

图片源自数据集VizWiz Captions

面对视障人群的应用：[SeeingAI](#)



数据来源: [web of science](#)



Viewpoint variation



Intra-class variation

图片源自付莹的“计算机视觉”课程PPT



Background clutter



Occlusion

图片源自付莹的“计算机视觉”课程PPT



早期：不符合语法、语义的问题

深度学习时代：

1. 曝光偏差
2. 损失评估不匹配
3. 梯度消失
4. 梯度爆炸
5. 幻觉



数据集和评价指标

汇报人：刘云皓



- Standard captioning datasets: 标准数据集
- e.g. MS COCO, Flickr8k, Flickr30k

- Pre-training datasets: 大型数据集。用做预训练
- e.g. SBU-Captions, CC3M, CC12M, 悟空

- Domain-specific datasets: 专用数据集。用于某一特定任务
- e.g. VizWiz Captions, CUB200, Oxford-102, Fashion Captioning, BreakNews, GoodNews, Loc. Narratives



- 328000 images
- 91 objects
- 5 captions/image

应用最广泛！！



Table 1: An overview of datasets for VLP model pre-training.

Dataset	Language	Availability	Image-text pairs
Flickr30k (Young et al., 2014)	English	✓	31,783
CxC (Parekh et al., 2020)	English	✓	247,315
SBU Captions (Ordonez et al., 2011b)	English	✓	1,000,000
Product1M (Zhan et al., 2021)	Chinese	✓	1,000,000
CC12M (Changpinyo et al., 2021)	English	✓	12,000,000
YFCC100M (Thomee et al., 2016)	English	✓	99,200,000
WIT (Srinivasan et al., 2021)	multilingual	✓	11,500,000
LAION-400M (Schuhmann et al., 2021)	English	✓	400,000,000
JFT-300M (Sun et al., 2017)	English	✗	300,000,000
JFT-3B (Zhai et al., 2021a)	English	✗	3,000,000,000
IG-3.5B-17k (Mahajan et al., 2018)	English	✗	3,500,000,000
M6-Corpus (Lin et al., 2021)	Chinese	✗	60,500,000
Wukong (Ours)	Chinese	✓	101,483,885

开源、亿级、中文



A computer screen with a Windows message about Microsoft license terms.



A can of green beans is sitting on a counter in a kitchen.



A photo taken from a residential street in front of some homes with a stormy sky above.



A blue sky with fluffy clouds, taken from a car while driving on the highway.



A hand holds up a can of Coors Light in front of an outdoor scene with a dog on a porch.



A digital thermometer resting on a wooden table, showing 38.5 degrees Celsius.



A Winnie The Pooh character high chair with a can of Yoohoo sitting on it in front of a white wall.



A cup holder in a car holding loose change from Canada.



Flickr8k: <https://academictorrents.com/details/9dea07ba660a722ae1008c4c8afdd303b6f6e53b>

Flickr30k: <https://pan.baidu.com/s/1r0RVUwctJsIOiNuVXHQ6kA> code: hrf3

COCO: <https://pan.baidu.com/s/1QT-s0iwVY1ClMThVySzuTQ> code: 2pyr

SBU Captions: <http://www.cs.virginia.edu/~vicente/sbucaption/>

CC3M: <https://ai.google.com/research/ConceptualCaptions/download>

CC12M: <https://github.com/google-research-datasets/conceptual-12m>

VizWiz: <https://vizwiz.org/tasks-and-datasets/image-captioning/>

CUB 200: <https://pan.baidu.com/s/10L3s7XmzoaYmbBocYJjYyQ> code: 7jeb

Oxford-102: <https://s3.amazonaws.com/fast-ai-imageclas/oxford-102-flowers.tgz>

Fashion Cap.: https://github.com/xueyang/Fashion_Captioning

GoodNews: <https://github.com/furkanbiten/GoodNews>

Text Caps: <https://textvqa.org/textcaps/>

Localized Narratives: <https://google.github.io/localized-narratives/>

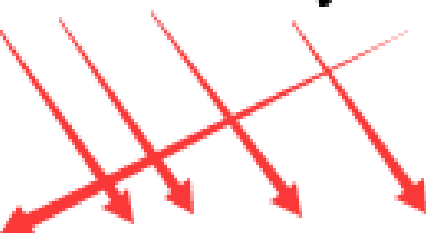
悟空: <https://wukong-dataset.github.io/wukong-dataset/download.html>



- 人工评价：
 - 表达的流畅度
 - 与图片的相关度
 - 表达的多样化程度
- 基于规则的自动化评价方法：
 - BLEU（机器翻译）
 - METEOR（机器翻译）
 - ROUGE（文本自动摘要）
 - CIDEr（图像描述）
 - SPICE（图像描述）

It is a nice day today.

Today is a nice day.



$$1\text{-gram} = 5/6$$

It is a nice day today.

Today is a nice day.



$$3\text{-gram} = 2/4$$

C: the the the the

S: The cat is standing on the ground

1-gram=1???



C: the the the the

S: The cat is standing on the ground

1-gram=1???

$\text{Count}_{\text{clip}} = \min(\text{Count}, \text{Max_Ref_Count})$

$$P_N = \frac{\sum_i \sum_k \min(h_k(c_i), \max_{j \in m} h_k(s_{ij}))}{\sum_i \sum_k \min(h_k(c_i))}$$

$$\text{BLEU} = \text{BP} * \exp\left(\sum_{n=1}^N W_n \log P_N\right) \quad \text{BP} = \begin{cases} 1 & \text{if } l_c > l_s \\ e^{1 - \frac{l_s}{l_c}} & \text{if } l_c \leq l_s \end{cases}$$

优点：使用方便、快捷，评价结果接近人类评价

缺点：不考虑语法，也没有涉及同义词或相似表达，长语句表现不佳



- BLEU的改进版，更关注召回率而非精确率。
- ROUGE-N 将BLEU的精确率优化召回率（常用）
- ROUGE-L 将BLEU的n-gram优化为最长公共子序列（常用）
- ROUGE-W 将ROUGE-L的连续匹配给予更高的奖励
- ROUGE-S 允许n-gram出现跳词(skip)

$$R_{LCS} = \frac{LCS(C, S)}{\text{len}(S)}$$

$$P_{LCS} = \frac{LCS(C, S)}{\text{len}(C)}$$

$$F_{LCS} = \frac{(1 + \beta^2) R_{LCS} P_{LCS}}{R_{LCS} + \beta^2 P_{LCS}}$$

$$ROUGE - N = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)}$$



$$METEOR = (1 - pen) \times F_{means}$$

$$F_{means} = \frac{PR}{\alpha P + (1 - \alpha)R}$$

$$P = \frac{m}{c}$$

$$R = \frac{m}{r}$$

$$Pen = \frac{\#chunks}{m}$$

← 相邻匹配的个数

C: the president spoke to the audience.

S: the president then spoke to the audience.

#chunks=2

优点：召回率和精确率的调和平均；增加了基于WordNet的同义词库，解决同义词匹配

缺点：参数过多，计算复杂；需要外部知识源；非语义对应



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

$$CIDEr_N(c_i, S_i) = \frac{1}{m} \sum_j \frac{g_n(c_i) * g_n(s_{ij})}{\|g_n(c_i)\| * \|g_n(s_{ij})\|}$$

TF-IDF+余弦相似度

$$g_n(\cdot) = \frac{h_n(\cdot)}{\sum_{\omega_l \in \Omega} h_l(\cdot)} \log\left(\frac{|I|}{\sum_{I_p \in I} \min(1, \sum_q h_n(\cdot))}\right)$$

优点：对出现频率不同的单词赋予不同的权重，区分了重要性

缺点：非语义对应



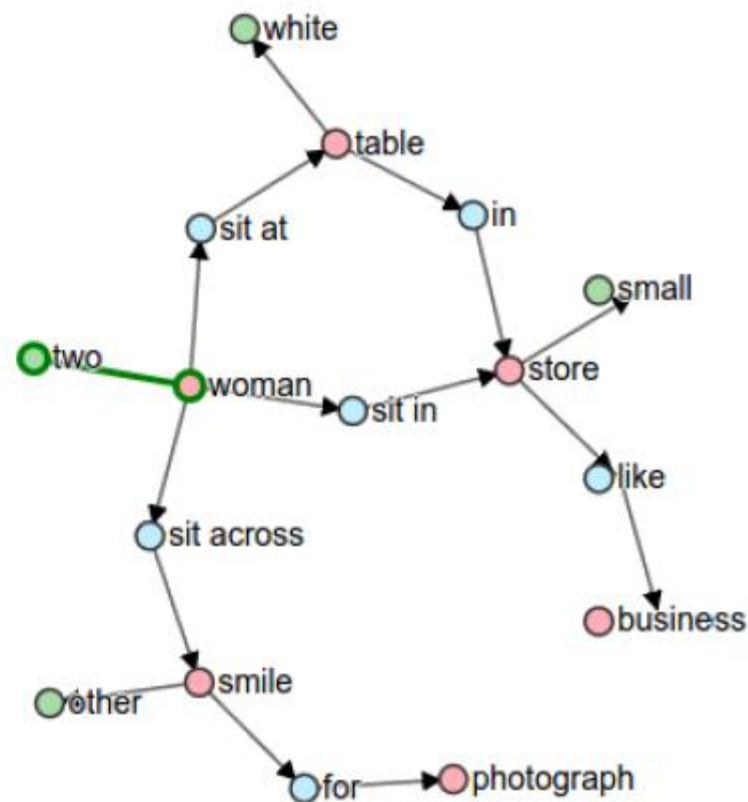
"two women are sitting at a white table"

"two women sit at a table in a small store"

"two women sit across each other at a table smile for the photograph"

"two women sitting in a small store like business"

"two woman are sittina at a table"



https://blog.csdn.net/csdn_tclz



$$SPICE(c, S) = F_1(c, S) = \frac{2 \cdot P(c, S) \cdot R(c, S)}{P(c, S) + R(c, S)}$$

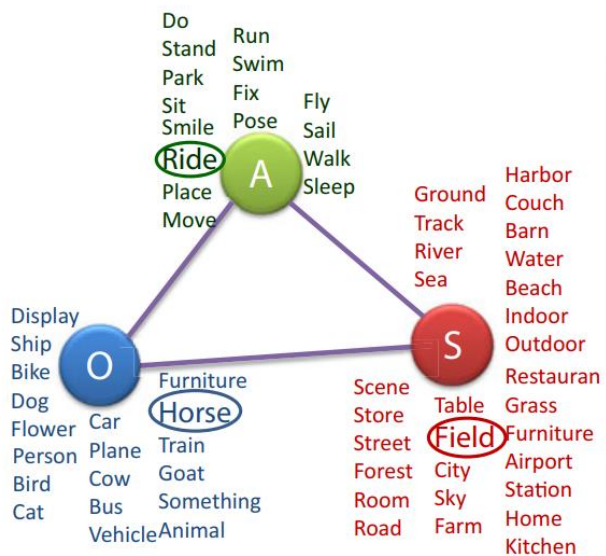
$$P(c, S) = \frac{|T(G(c)) \otimes T(G(S))|}{|T(G(c))|}$$

$$R(c, S) = \frac{|T(G(c)) \otimes T(G(S))|}{|T(G(S))|}$$

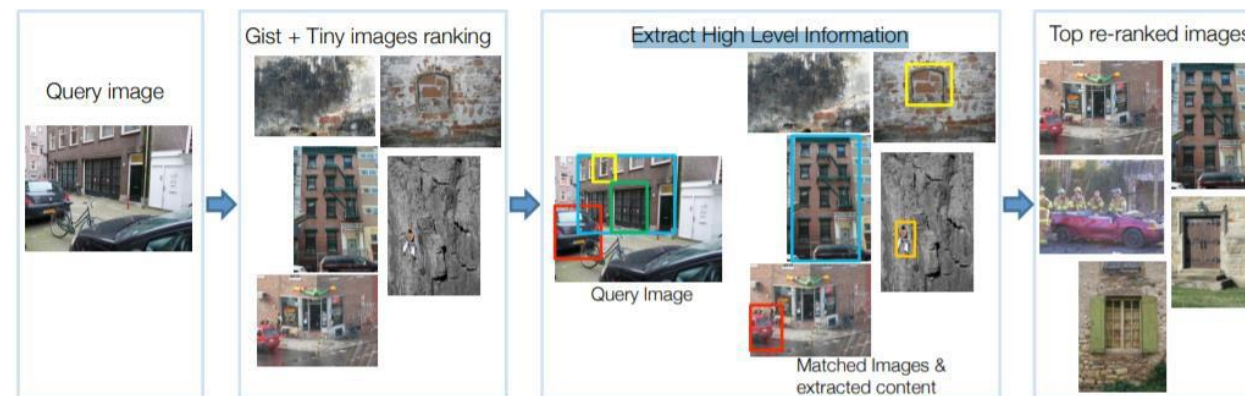
优点：基于图的语义表示

缺点：不能准确判断语法结构错误

基于模板的图像描述方法



基于检索的图像描述方法



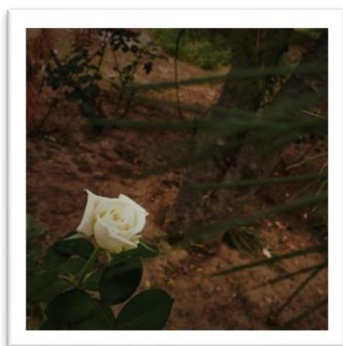
存在种种问题，表现并不好.....



经典综述

汇报人：李桐

bottom-up模型

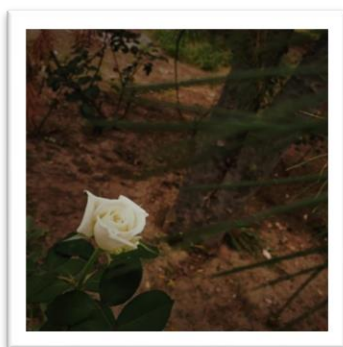


bench
WOODEN
sitting
Tree



A wooden bench
sitting net to a tree.

top-down模型



Deep
learning

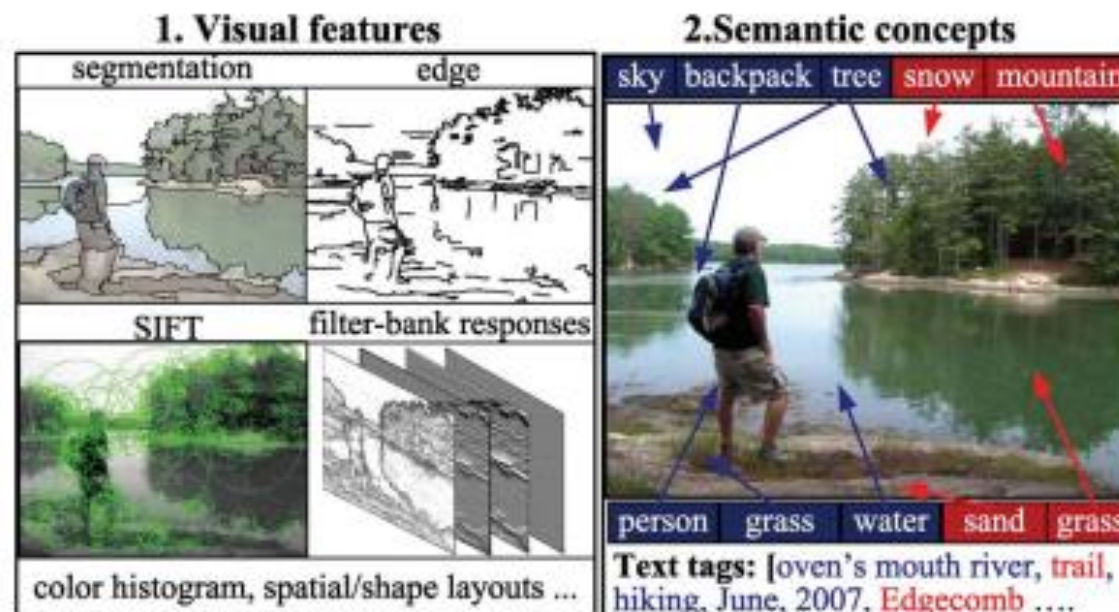


A wooden bench
sitting net to a tree.

bottom-up模型

传统手工设计

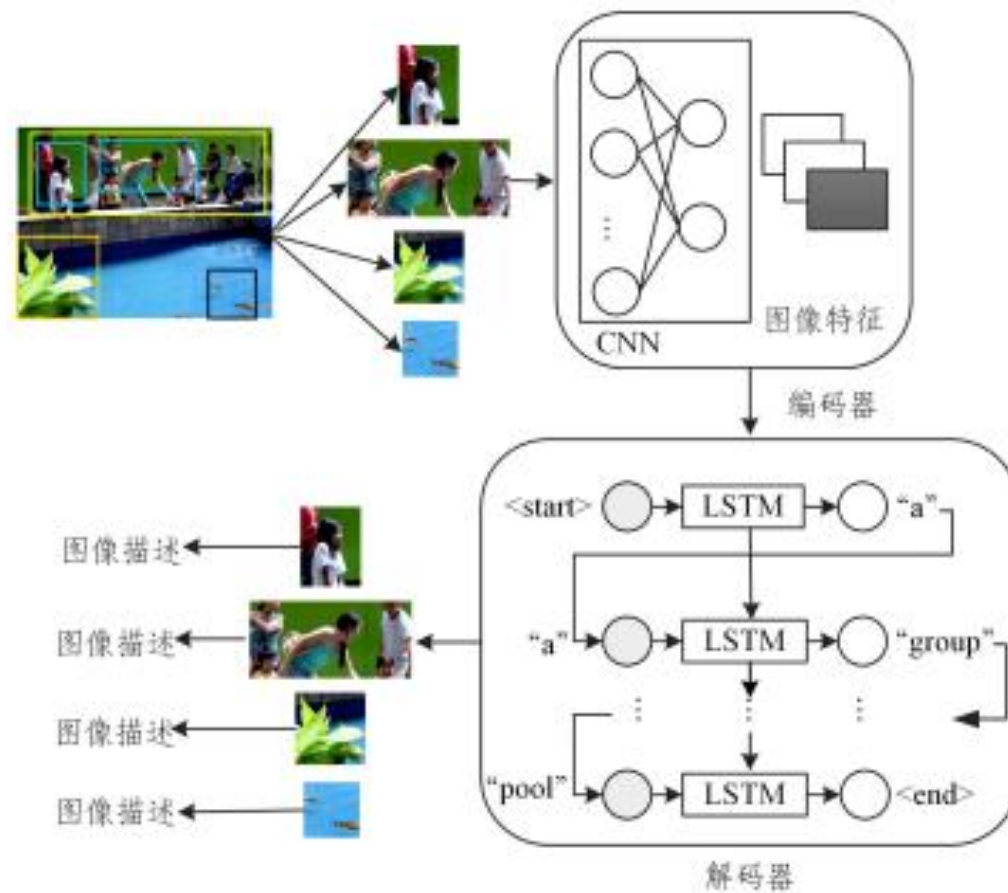
早期深度学习

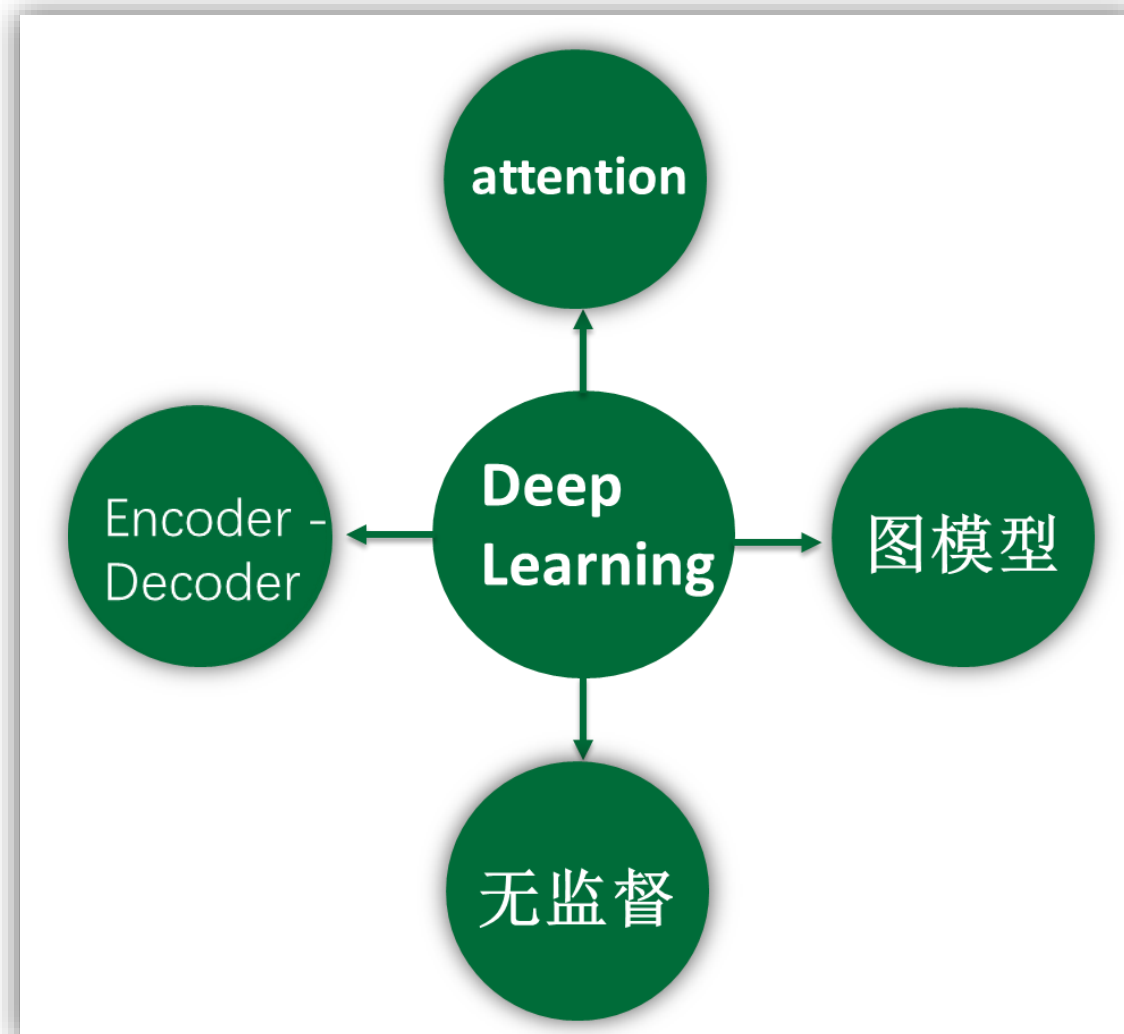
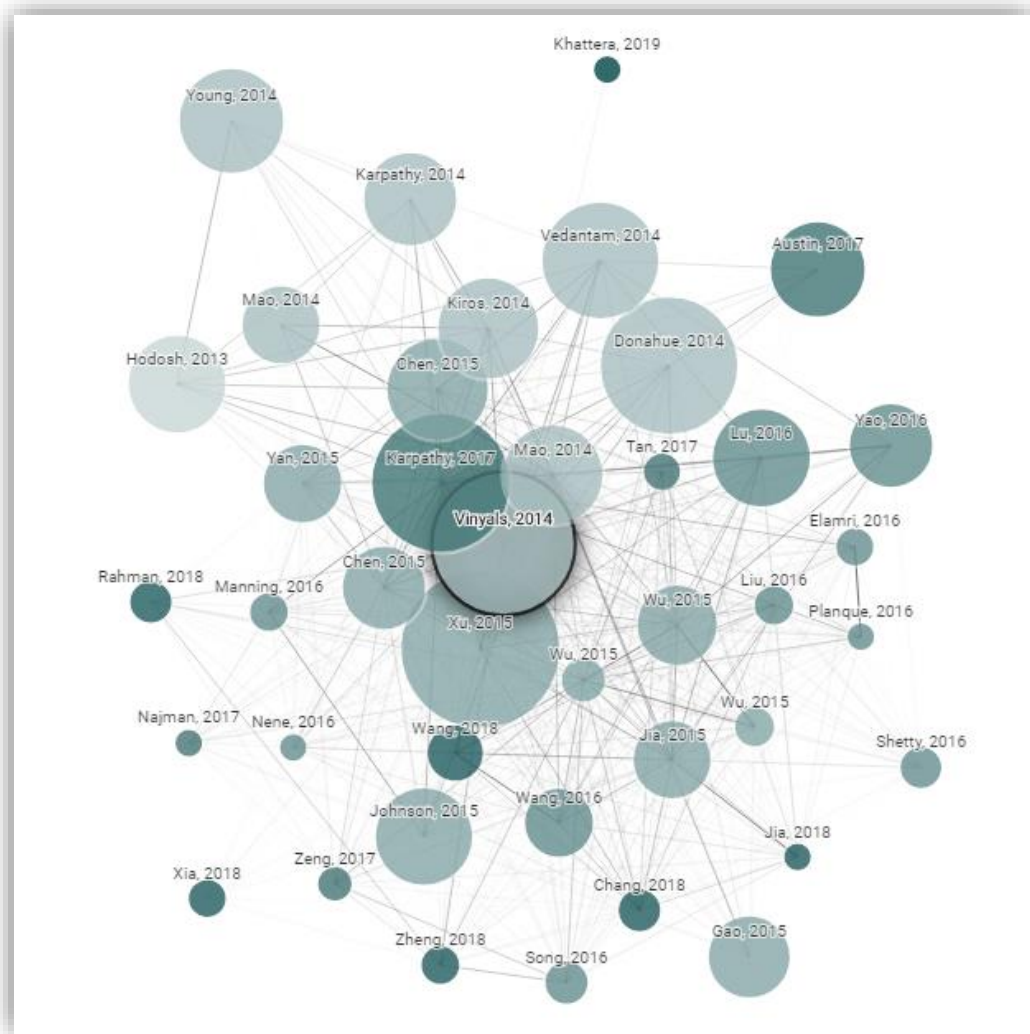


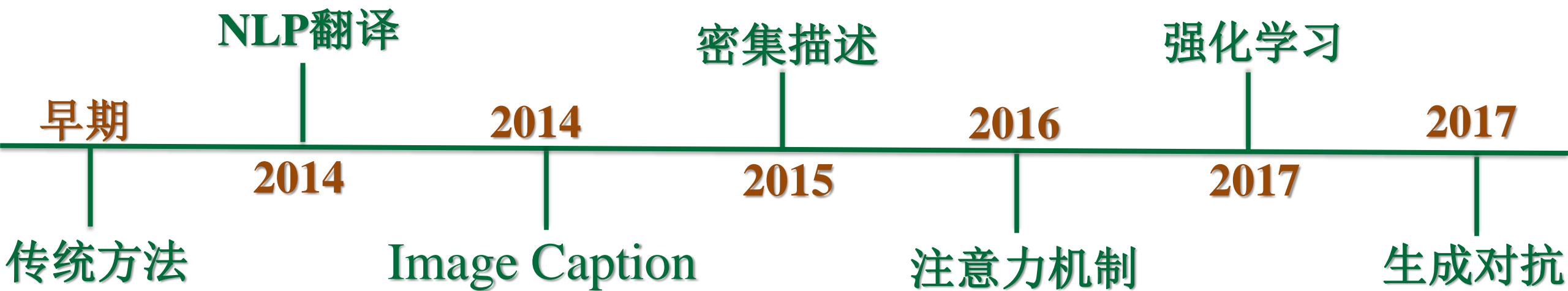
top-down模型

端到端模型

密集描述方法







2012 — Alexet

2014 — VGG

2014 — Translation

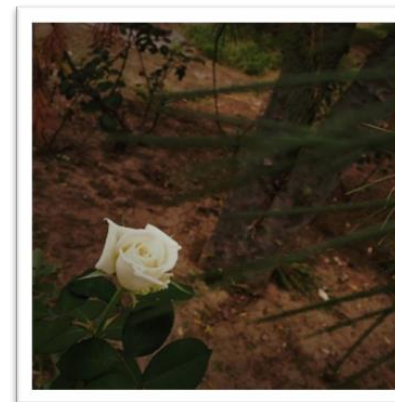
2014 — Image Caption

A wooden bench
sitting net to a tree.

encoder

decoder

一张木凳坐在
一棵树旁边

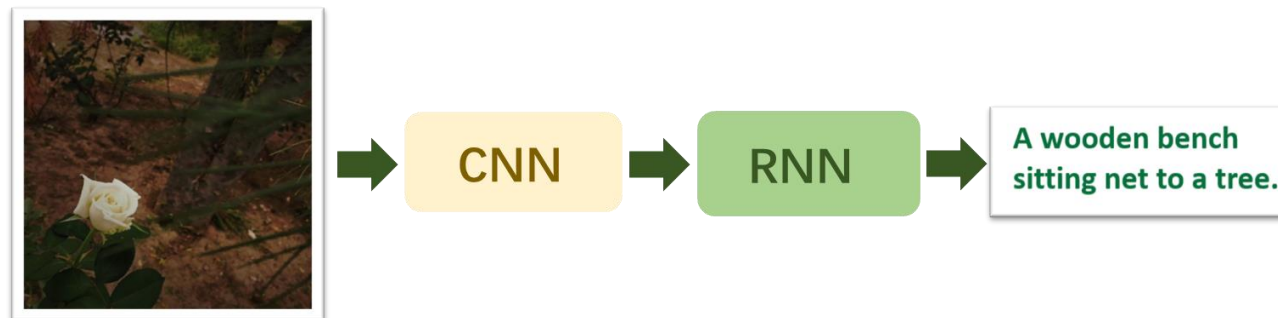


encoder

decoder

A wooden bench
sitting net to a tree.

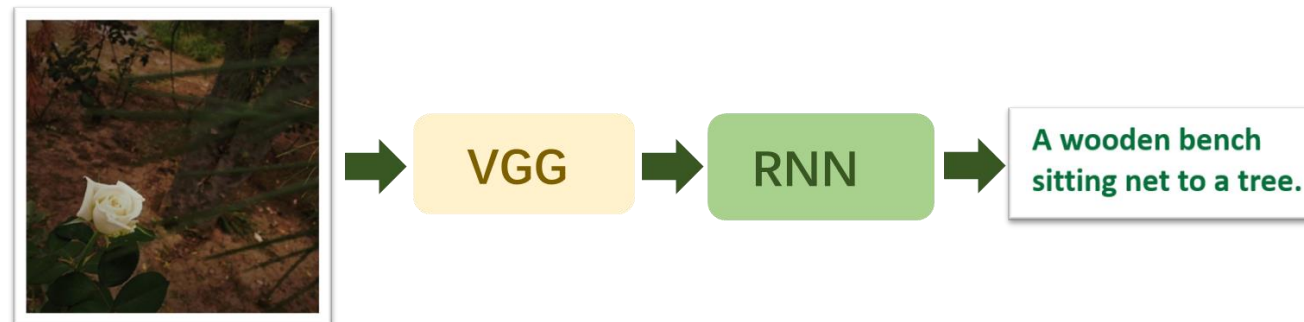
2014 — Image Caption



2014 — Show and Tell

Show and Tell

2014 — Neural Talk



Neural Talk

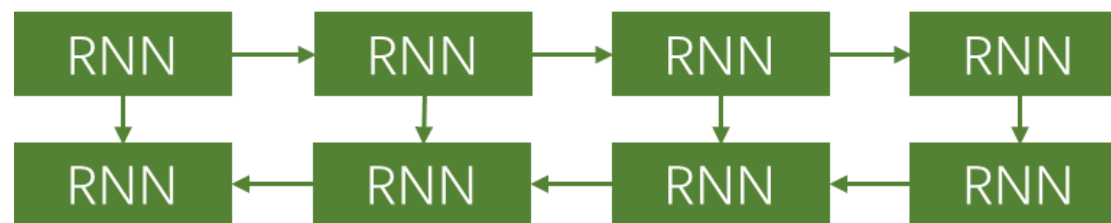
2014

Encoder

Decoder

2015

改善

双向RNN_[1]双向双层LSTM_[2]phi-LSTM_[3]

... ..

[1]:Mind's Eye: A Recurrent Visual Representation for Image Caption Generation

[2]:Image Captioning with Deep Bidirectional LSTMs

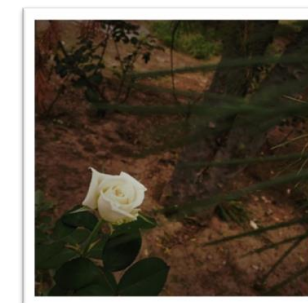
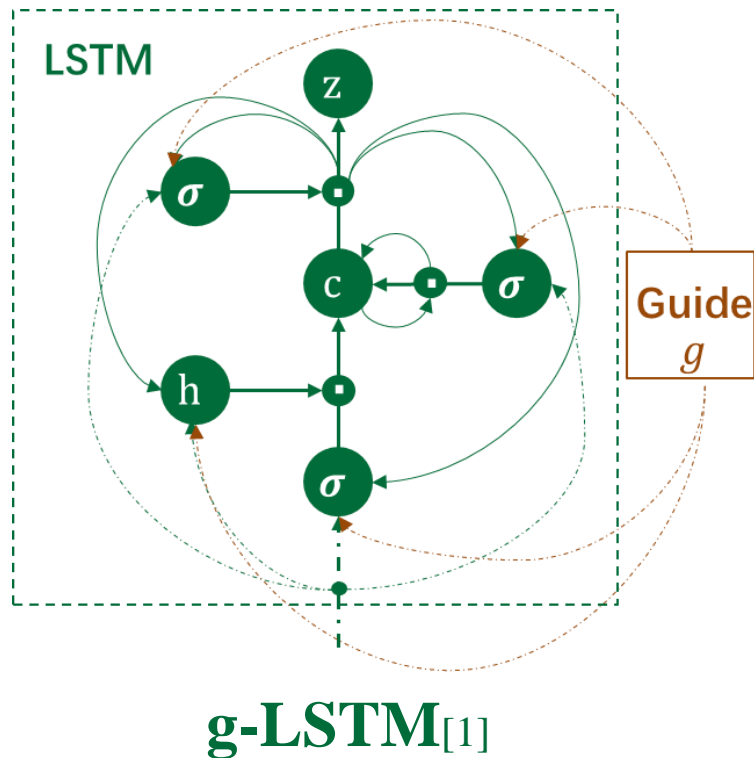
[3]:A Phrase-based Hierarchical LSTM Model for Image Captioning

2014

Encoder
Decoder

2015

改善



CNN

CNN

A wooden bench
sitting net to a tree.CNN解码模型^{[2][3]}

[1]: Guiding the Long-Short Term Memory Model for Image Caption Generation

[2]: Convolutional image captioning

[3]: CNN+ CNN: Convolutional Decoders for Image Captioning

2015

DenseCap

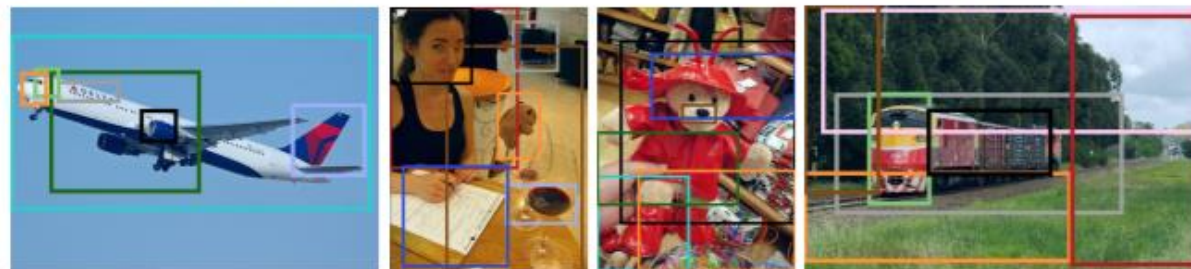
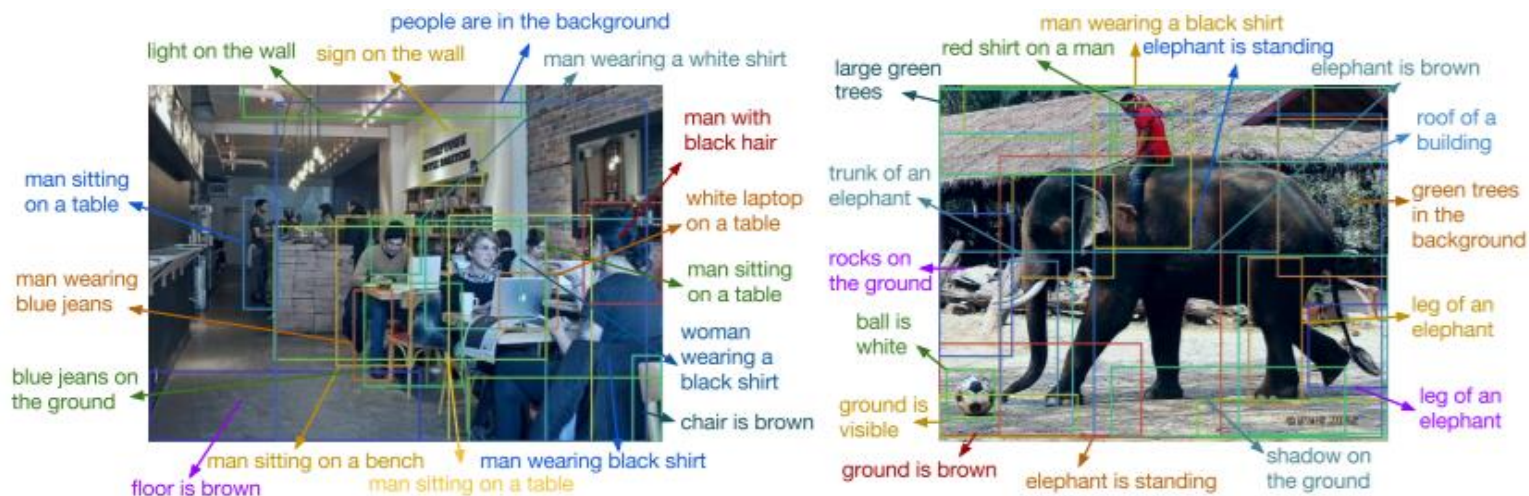
2017

joint inference

2019

Dense relational
captioning

DenseCap



Our Model: plane is flying, tail of the plane, red and white plane, plane is white, engine on the plane, windows on the plane, nose of the plane.

Full Image RNN: A large jetliner flying through a blue sky.

woman wearing a black shirt, teddy bear is brown, chair is black, glass of wine, table is brown, woman with brown hair, paper on the table.

A man and a woman sitting at a table with a cake.

teddy bear is wearing a red shirt, red and white teddy bear, bear is wearing a red hat, red and white shirt, side is brown, black nose of a bear.

A teddy bear with a red bow on it.

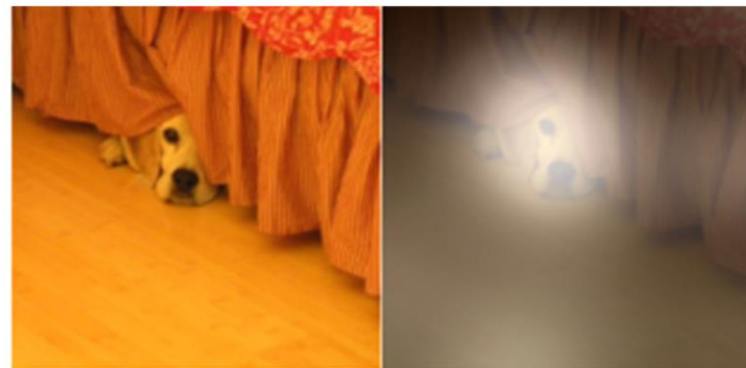
train on the tracks, trees are green, front of the train is yellow, grass is green, green trees in the background, photo taken during the day, red train car.

A train is traveling down the tracks near a forest.

DenseCap: Fully Convolutional Localization Networks for Dense Captioning CVPR2015

Show, Attend and Tell

2015

Attention +
TranslationA dog is standing on a hardwood floor.

2016

Attention +
Image CaptionA group of people sitting on a boat
in the water.

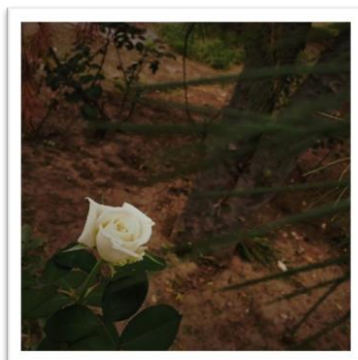
2015

Attention +
Translation

2016

Attention +
Image Caption

Show and Tell



CNN

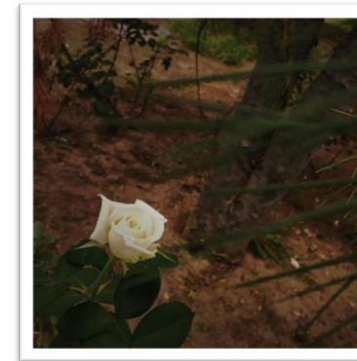


RNN

A wooden bench
sitting net to a tree.

2014 → 2016

Show, Attend and Tell



CNN

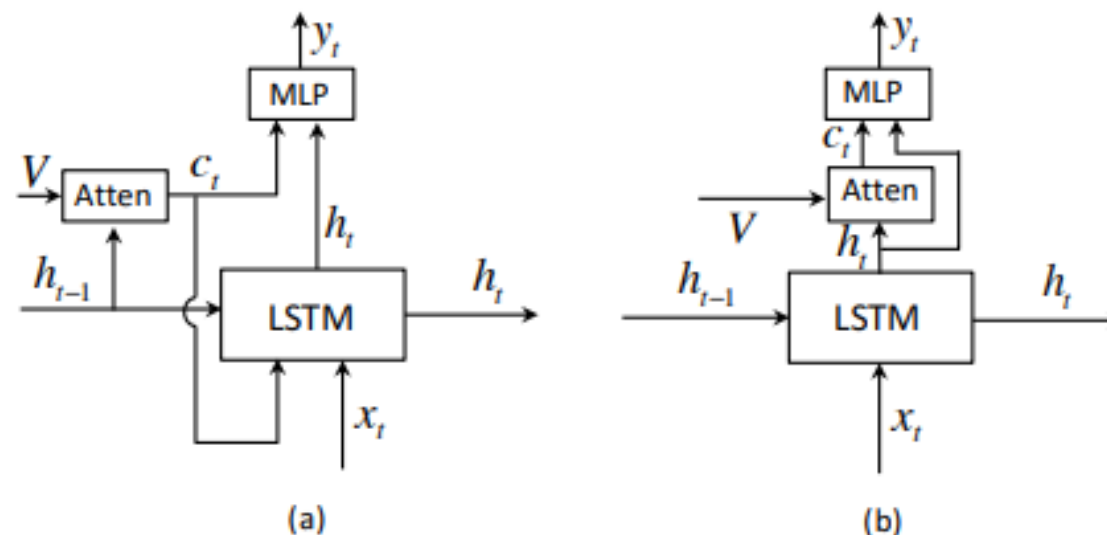
RNN +
AttentionA wooden bench
sitting net to a tree.

2016

Attention 改善

2017

Knowing When to Look (CVPR 2017)



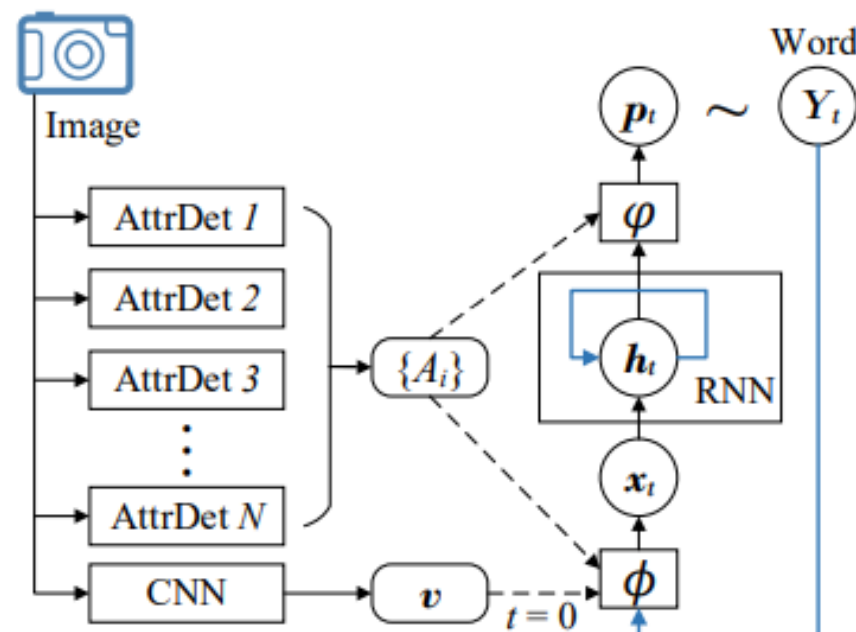
2016

Attention 改善

2017

Image Captioning with Semantic Attention

(CVPR 2016)

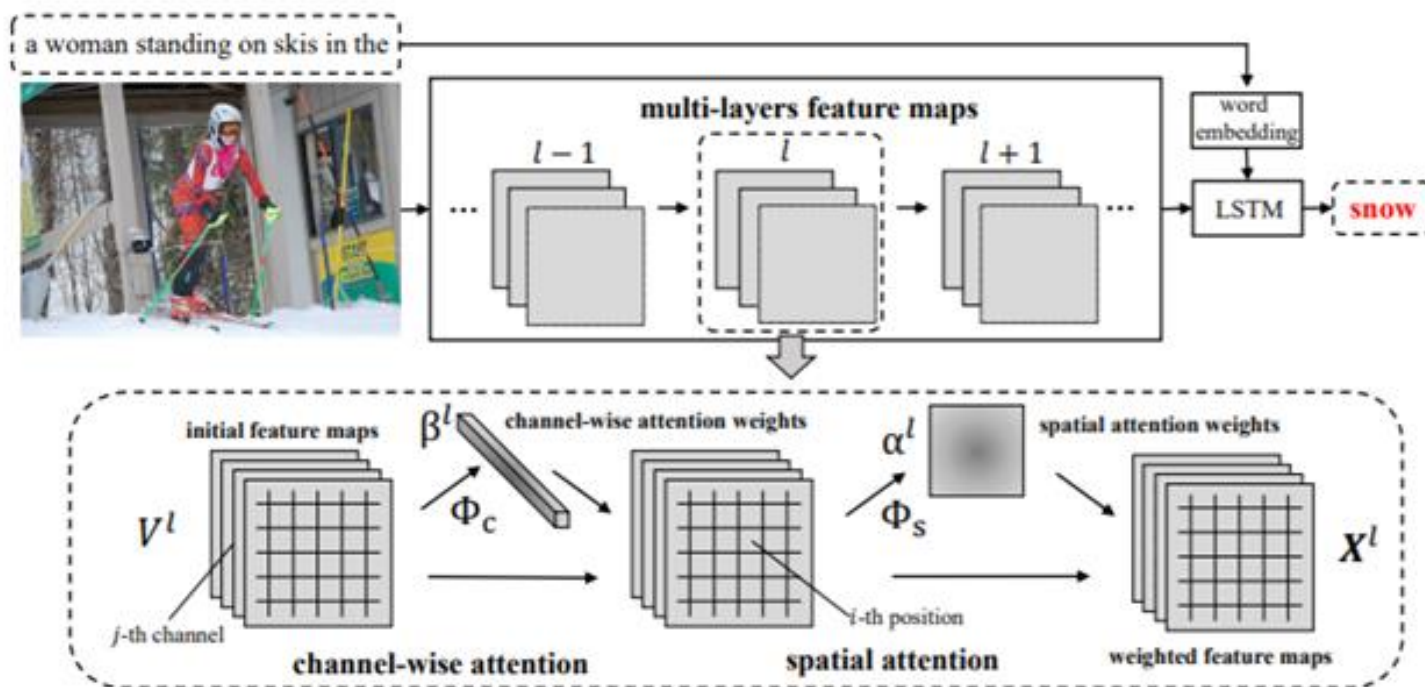


2016

Attention 改善

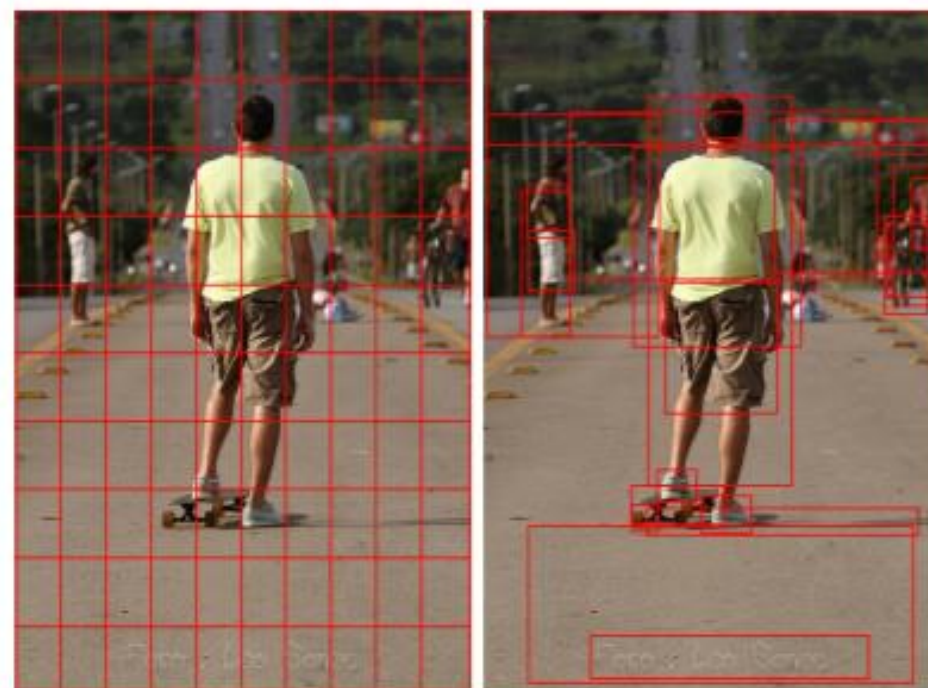
2017

SCA-CNN(CVPR2017)



Bottom-Up and Top-Down Attention

(CVPR 2017)



2016

Attention 改善

2017

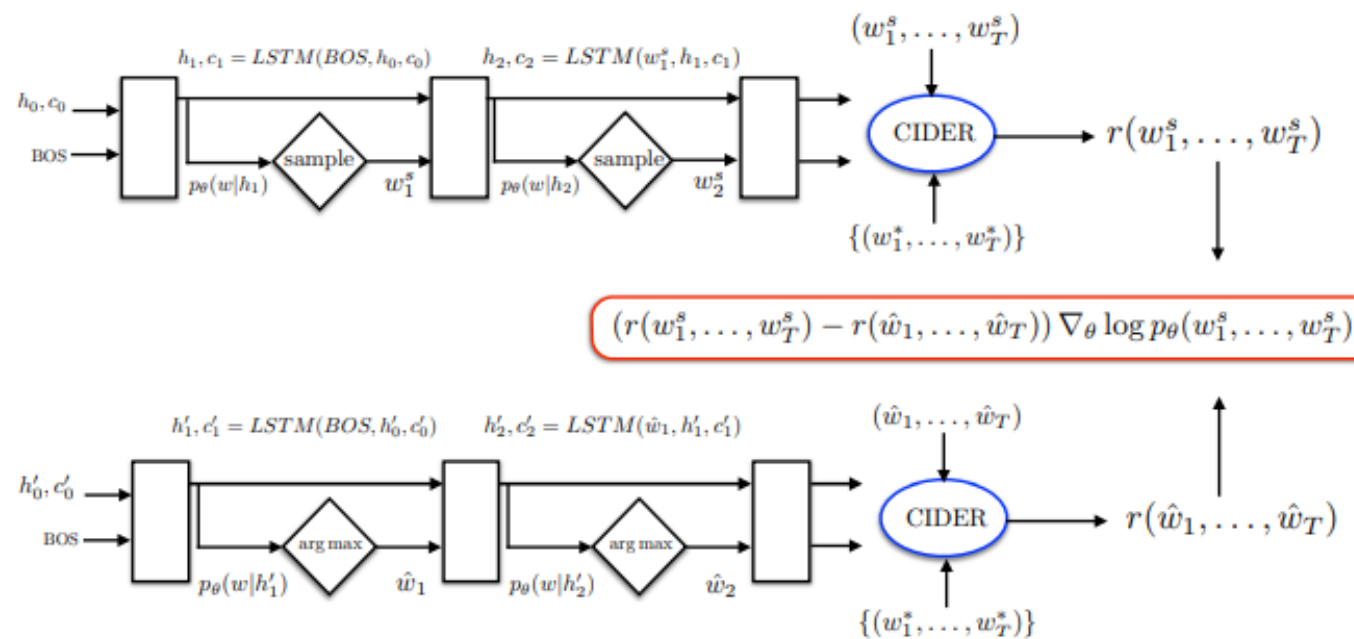
2015

强化学习
Translation

2017

强化学习
Image Caption

Self-critical sequence training for image captioning (CVPR 2017)



2014

GAN

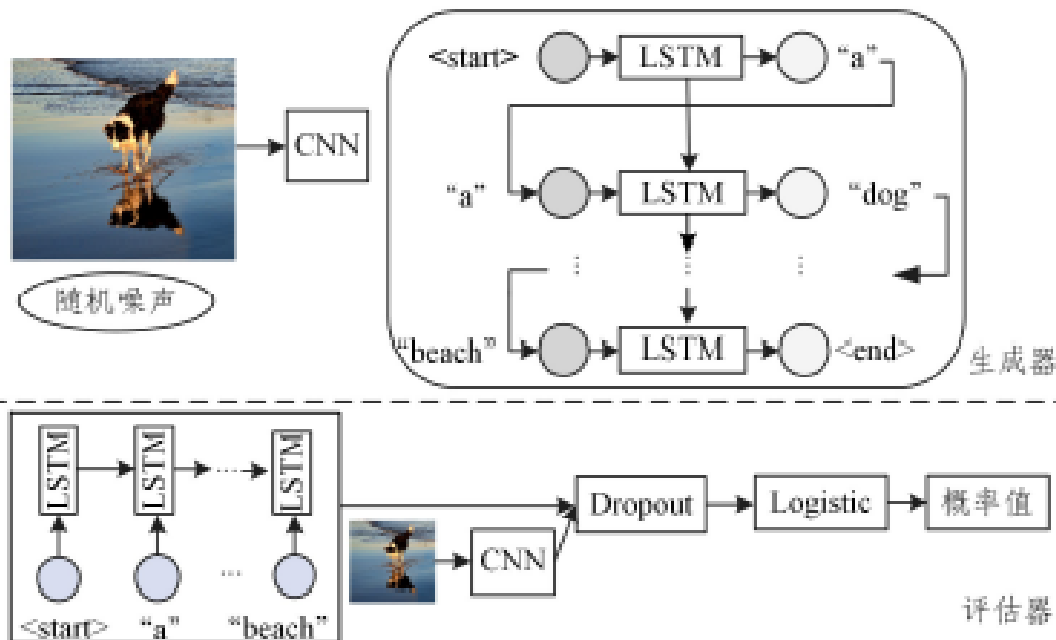
Generative Adversarial Networks

2017

Image Caption

Towards Diverse and Natural Image Descriptions via a Conditional GAN

(ICCV 2017)





Attention编解码器方法

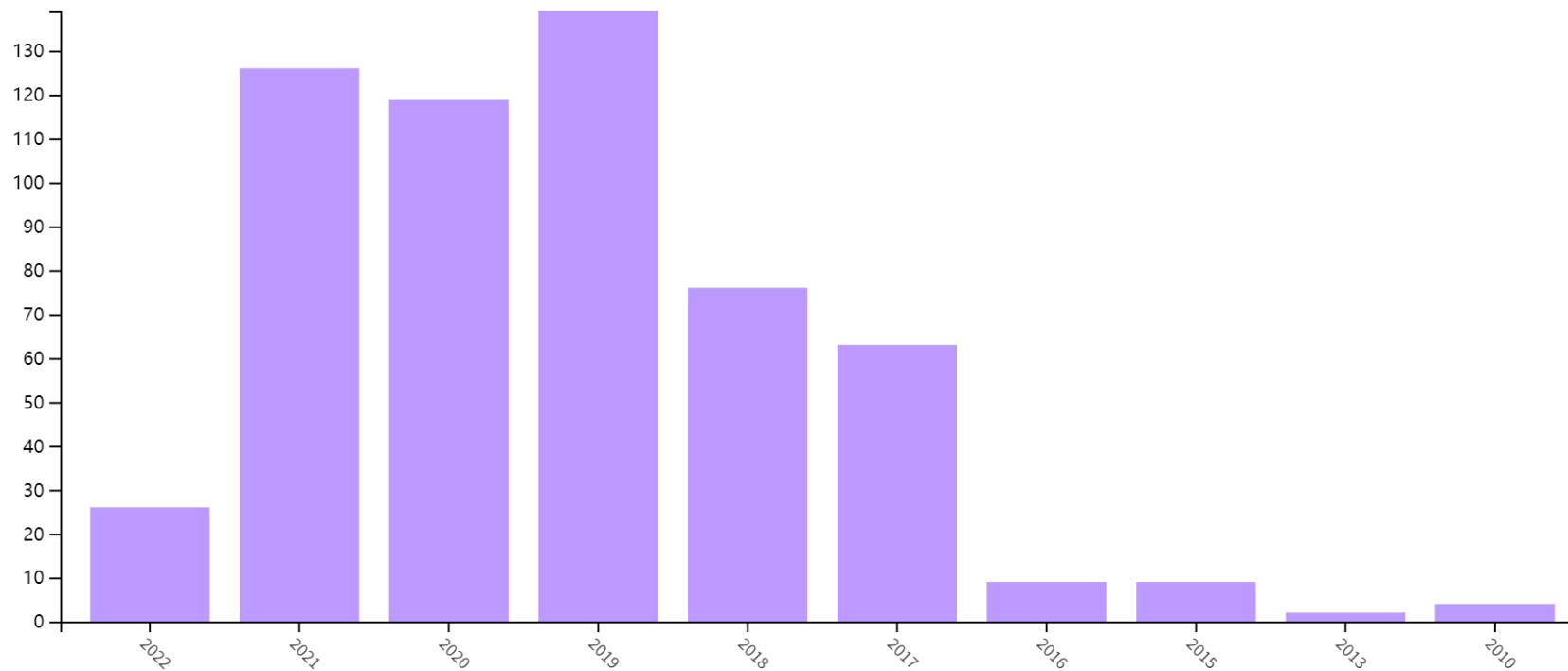
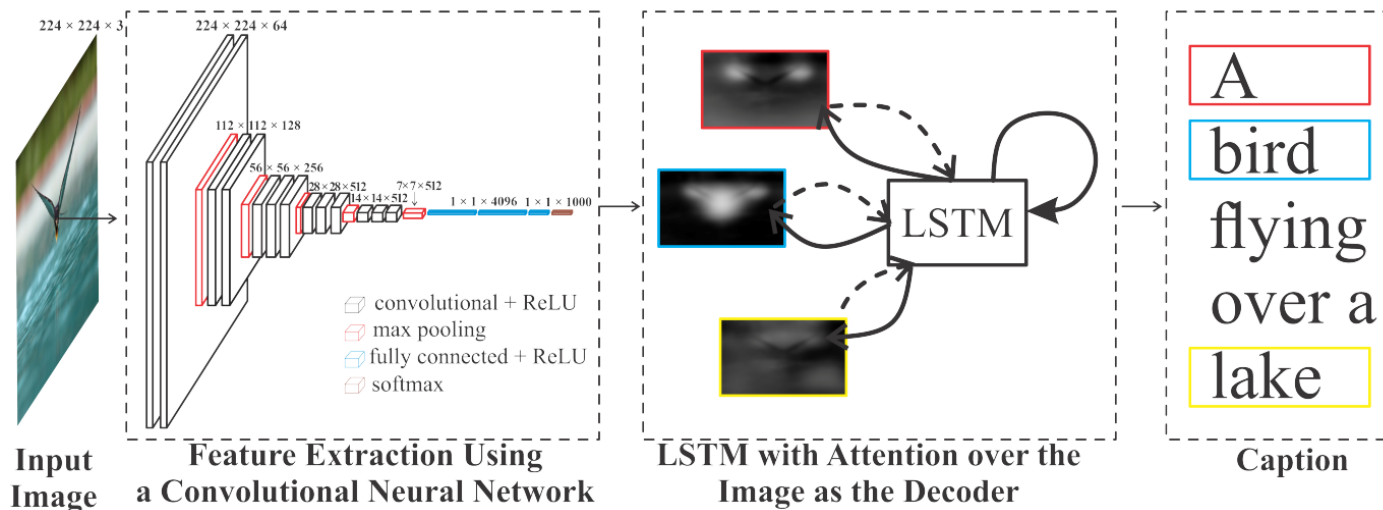


Image caption+Attention



通过CNN做编码器，然后传给RNN，经过RNN对于每张图逐字生成caption向量。

在RNN中添加attention机制

[Show, Attend and Tell: Neural Image Caption Generation with Visual Attention]

2015

Attention
+
Translation

2016

Attention +
Image
Caption

Show and Tell



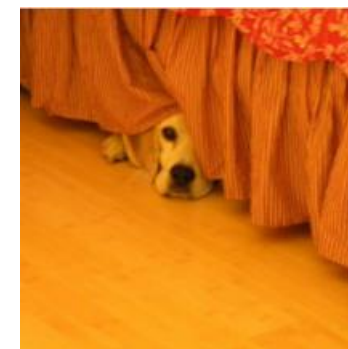
CNN

RNN

A small white dog
laying on a bed

2014 → 2016

Show, Attend and Tell



CNN

RNN +
Attention

A dog is standing on
a hard wood floor

hard-attention



A man and a woman playing frisbee in a field.

soft-attention



A woman is throwing a frisbee in a park.

整张图作为输入，并未考虑目标region信息。

[Show, Attend and Tell: Neural Image Caption Generation with Visual Attention]

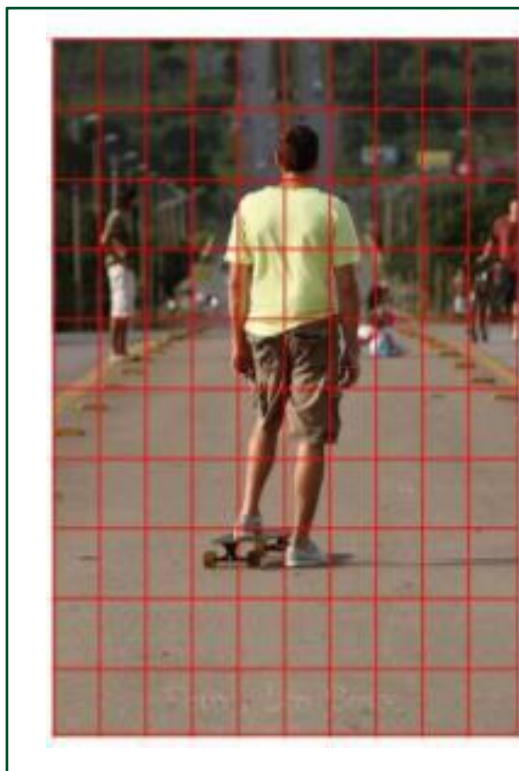


Attention Is All You Need

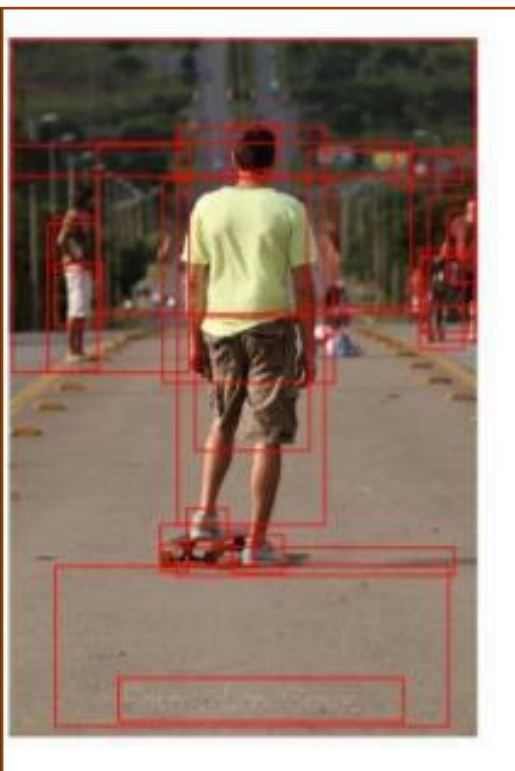
WHAT IS ATTENTION ?

[NeuralPS-2017 Attention Is All You Need]

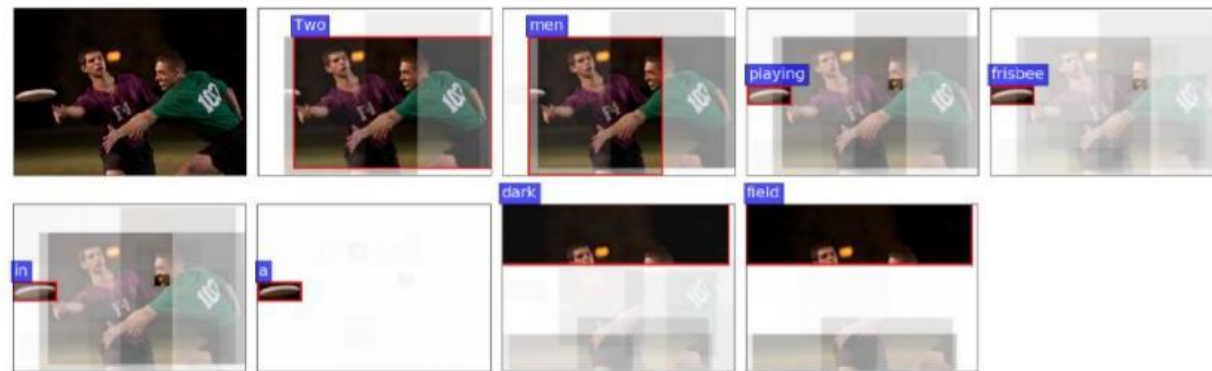
Top-Down



Bottom-Up



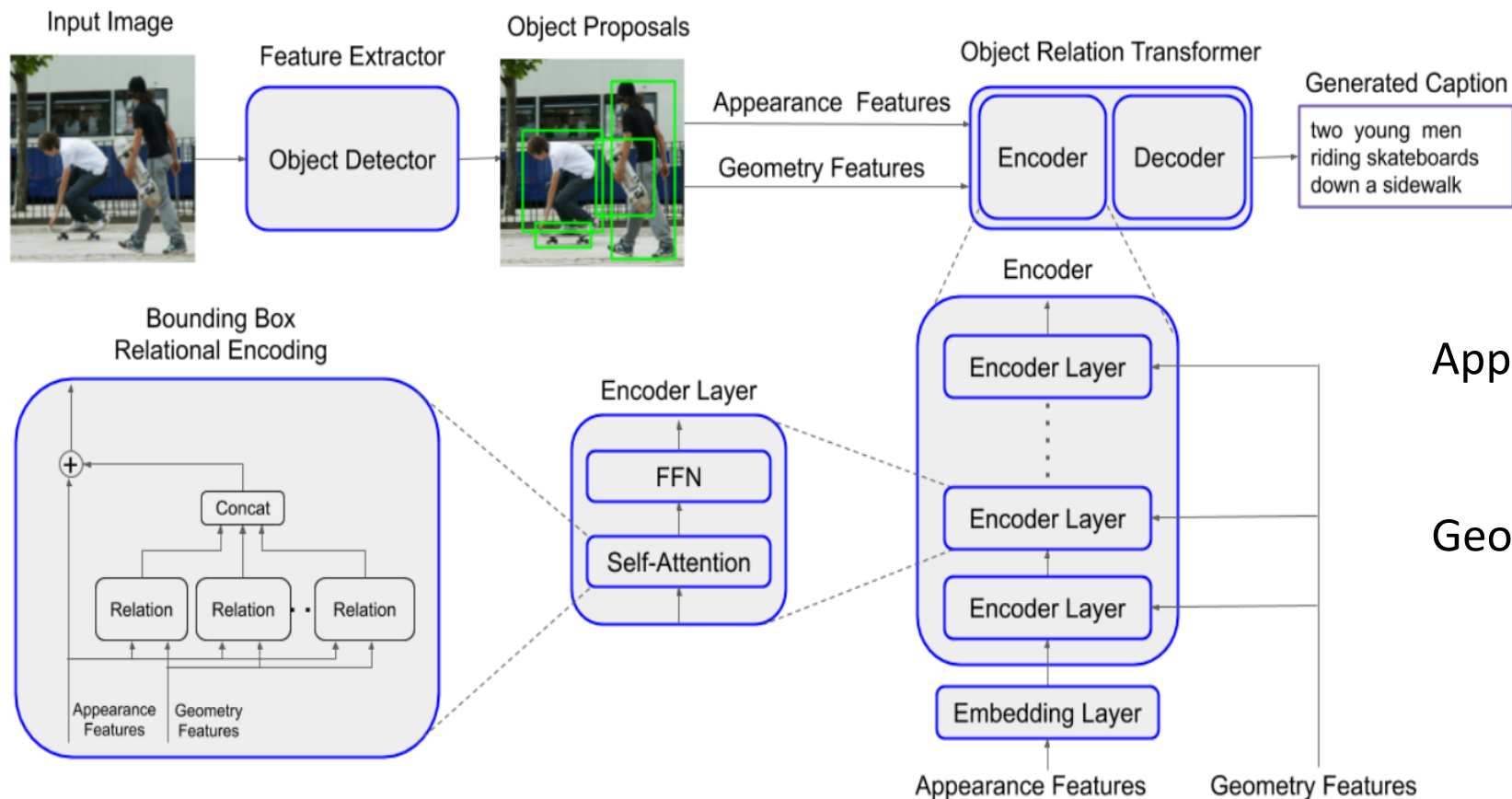
利用目标检测Faster-Rcnn获得自底向上的注意力特征



具体来说，bottom-up机制基于Faster R-CNN，即基于目标来计算attention。得到图片中每个目标或显著区域的特征向量表示。

[2018-CVPR Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering]

Attention加入位置空间信息



在模型中加入了
Geometry 信息，适
于模型提取空间位
置特征

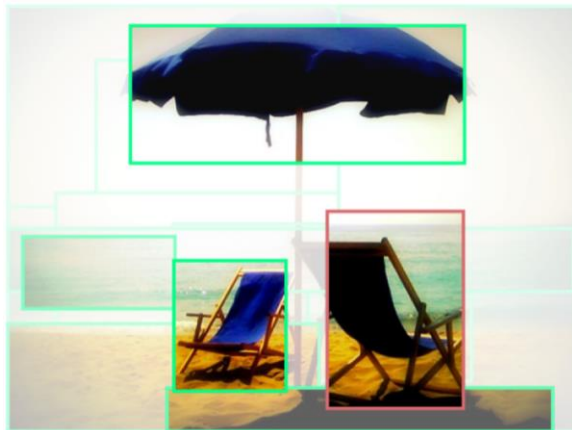
Appearance Features: Resnet-101

Geometry Features: Faster-RCNN

加入了空间信息，
但未考虑多区域多
细粒度的交互

[NeuralPS-2019 Transforming Objects into Words]

Attention加入位置空间信息



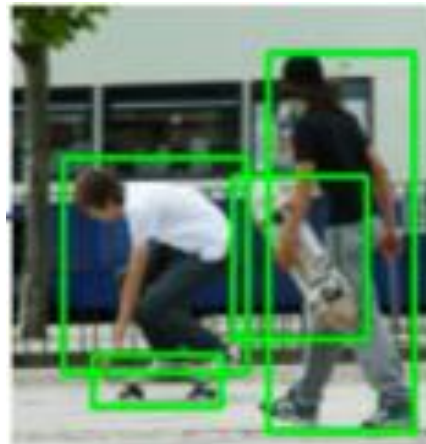
two chairs and an umbrella on a beach

two beach chairs under an umbrella on the beach

Standard: two chairs and an umbrella on a beach

ORT: two beach chairs **under an umbrella** on the beach

[NeuralPS-2019 Transforming Objects into Words]

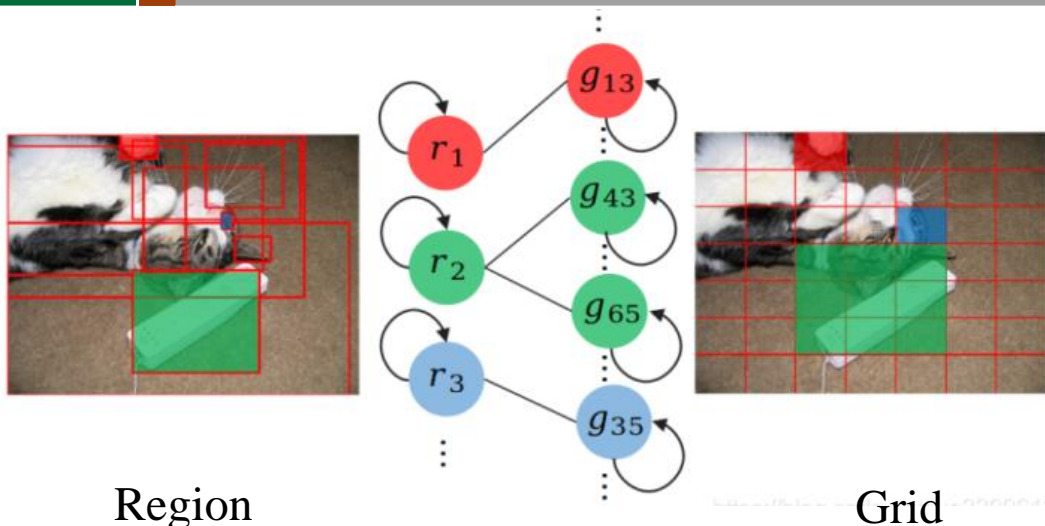


a group of young men riding skateboards down a sidewalk

two young men riding skateboards down a sidewalk

Standard: **a group of** young men riding skateboards down a sidewalk

ORT: **two** young men riding skateboards down a sidewalk

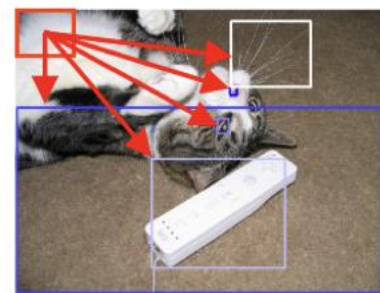


顶层区域，交互性好，包含上下文信息，但细粒度低

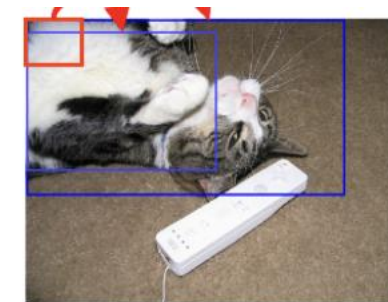
底层特征，交互性差，不包含上下文信息但是细粒度高

作者提出了位置约束交叉注意模型来模拟区域和网格之间复杂的相互作用，实现层间融合。为了避免引入语义噪声，首先创建一个几何对齐图，所有区域和网格特征都表示为独立的节点，当且仅当网格和区域的有交点时，该网格节点与区域节点才相连。有交集的区域和网格(用相同颜色突出显示)由无向边连接，以消除语义上的不相关信息。注意每个节点都有一个自连接的边。

[2021 Dual-Level Collaborative Transformer for Image Captioning]



Transformer



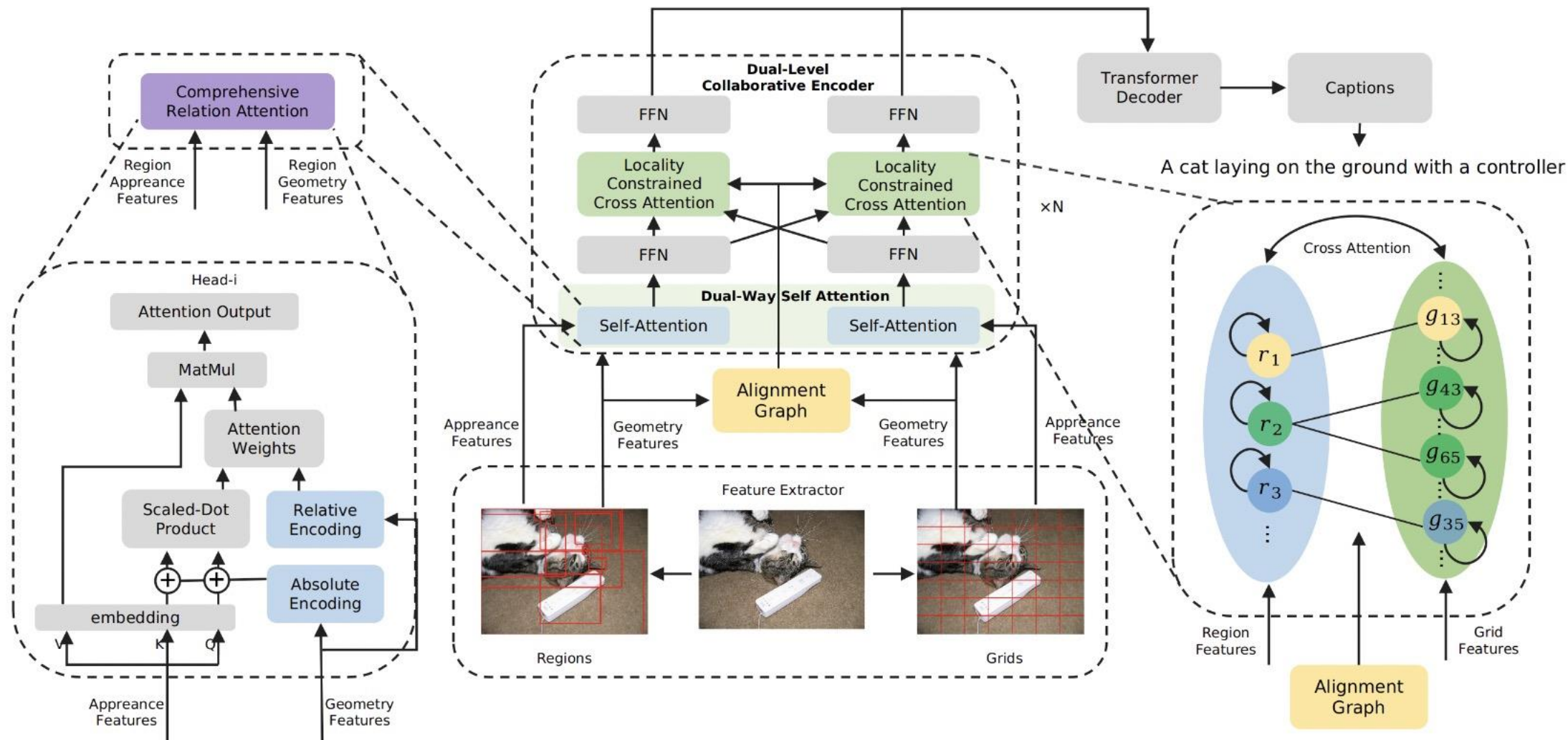
DLCT

Transformer 注意力机制考虑了图像中任意region之间的交互，而DLCT选取特定的region和特定grid进行交互。



Transformer: A brown horse grazing in a field

DLCT: Two horses grazing in a field of grass



[2021 Dual-Level Collaborative Transformer for Image Captioning]

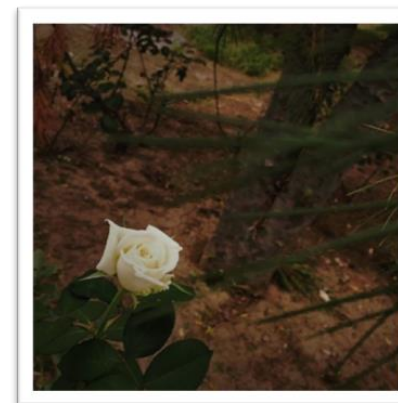
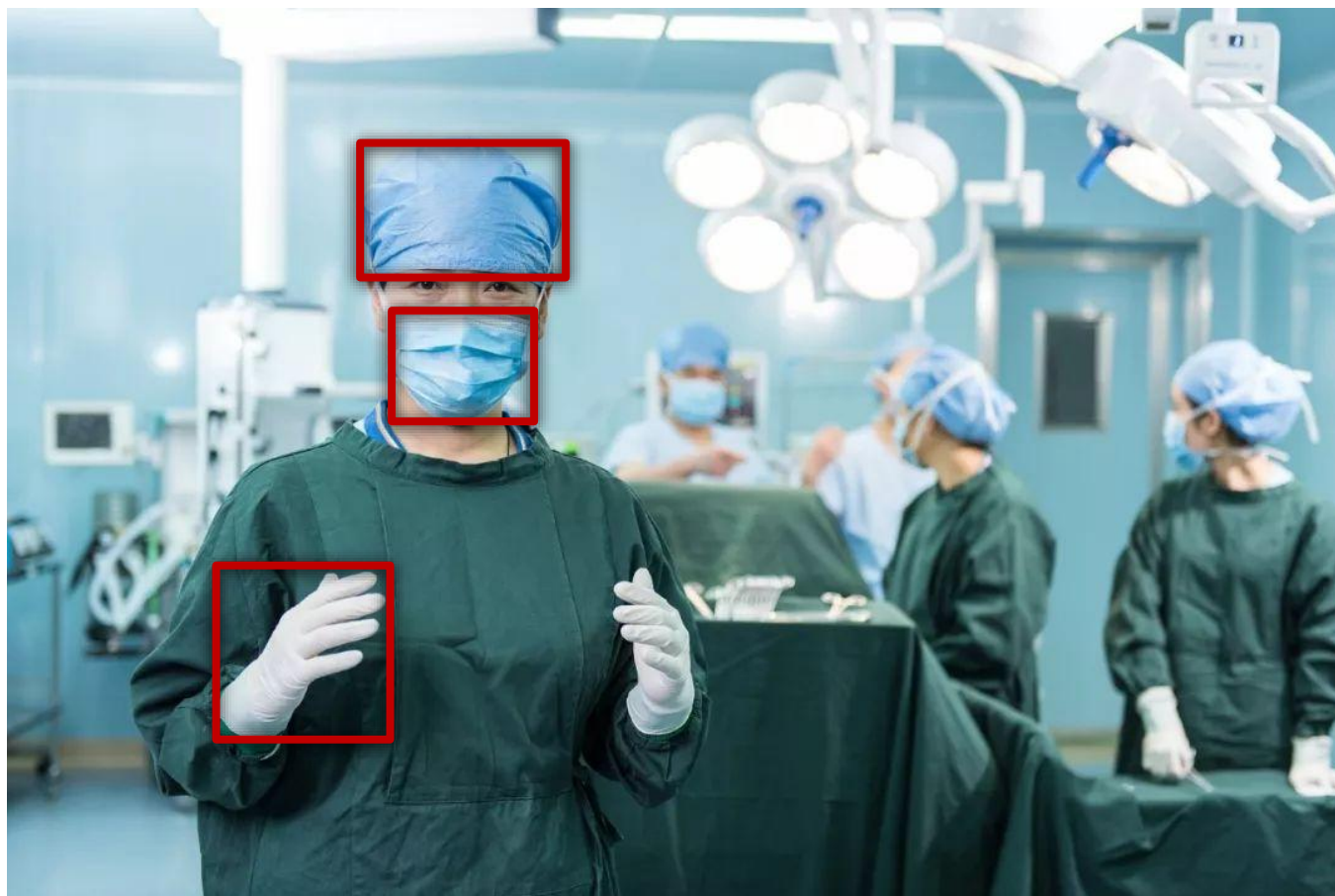


Model	B-1		B-2		B-3		B-4		M		R		C	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
SCST (ResNet-101)	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.7
Up-Down (ResNet-101)	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
HAN (ResNet-101)	80.4	94.5	63.8	87.7	48.8	78.0	36.5	66.8	27.4	36.1	57.3	71.9	115.2	118.2
GCN-LSTM (ResNet-101)	80.8	95.2	65.5	89.3	50.8	80.3	38.7	69.7	28.5	37.6	58.5	73.4	125.3	126.5
SGAE (ResNet-101)	81.0	95.3	65.6	89.5	50.7	80.4	38.5	69.7	28.2	37.2	58.6	73.6	123.8	126.5
AoA (ResNet-101)	81.0	95.0	65.8	89.6	51.4	81.3	39.4	71.2	29.1	38.5	58.9	74.5	126.9	129.6
HIP (SENet-154)	81.6	95.9	66.2	90.4	51.5	81.6	39.3	71.0	28.8	38.1	59.0	74.1	127.9	130.2
M2 (ResNet-101)	81.6	96.0	66.4	90.8	51.8	82.7	39.7	72.8	29.4	39.0	59.2	74.8	129.3	132.1
X-Transformer (ResNet-101)	81.3	95.4	66.3	90.0	51.9	81.7	39.9	71.8	29.5	39.0	59.3	74.9	129.3	131.4
X-Transformer (SENet-154)	81.9	95.7	66.9	90.5	52.4	82.5	40.3	72.4	29.6	39.2	59.5	75.0	131.1	133.5
DLCT (ResNeXt-101)	82.0	96.2	66.9	91.0	52.3	83.0	40.2	73.2	29.5	39.1	59.4	74.8	131.0	133.4
DLCT (ResNeXt-152)	82.4	96.6	67.4	91.7	52.8	83.8	40.6	74.0	29.8	39.6	59.8	75.3	133.3	135.4



基于图、强化学习的方法

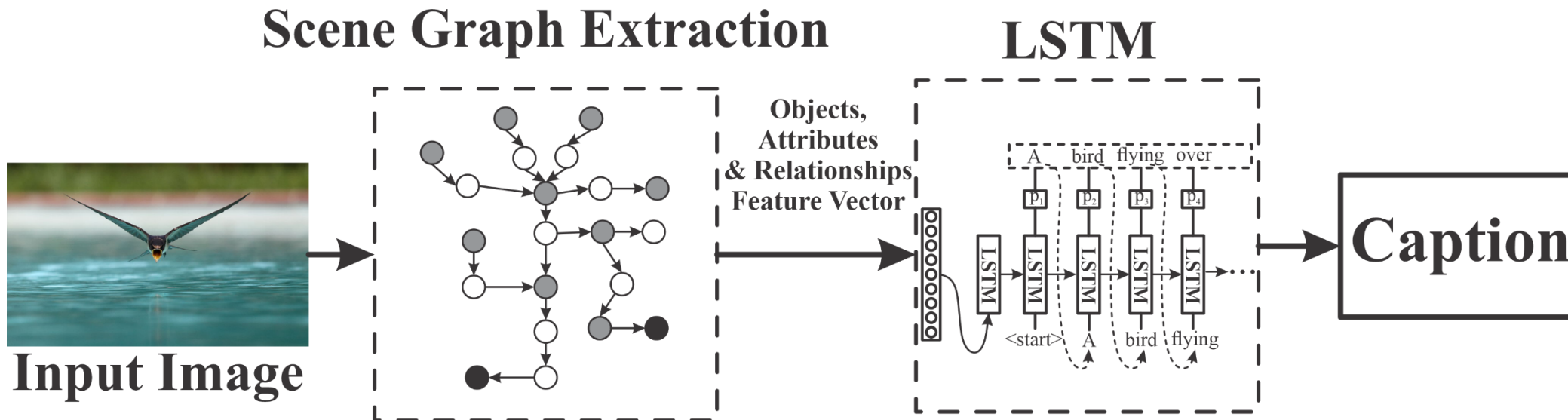
汇报人：刘天锐



encoder

decoder

A wooden bench
sitting net to a tree.



图卷积网络（GCNs）用来编码场景图中的区域和关系
获得的特征向量被传递给LSTM解码器生成标题

Intuition:

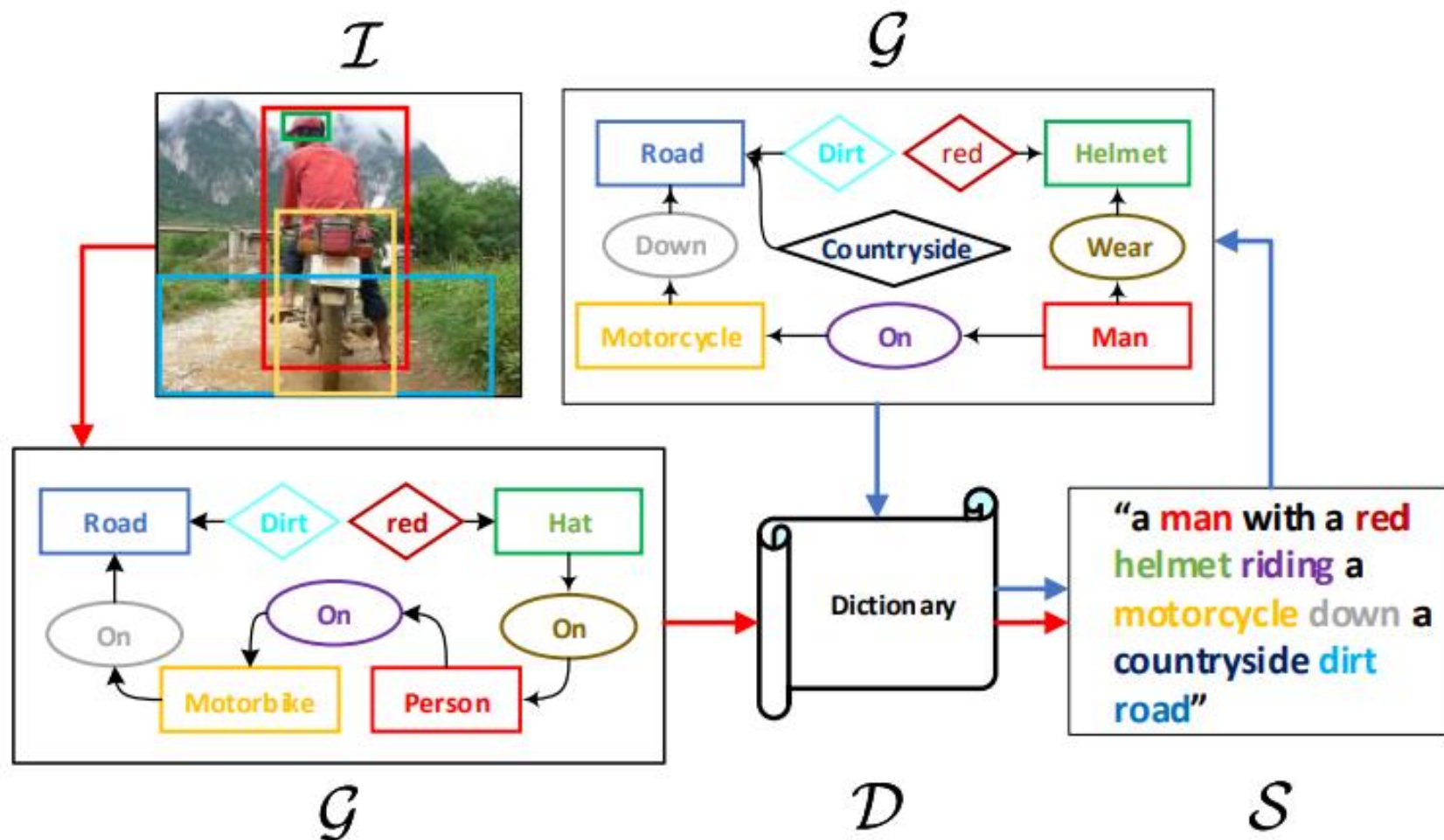
推理互补端到端，
知识 \rightarrow 视觉语言
连接实体、属性、关系

编码器: $V \leftarrow I$

映射: $G \leftarrow V$

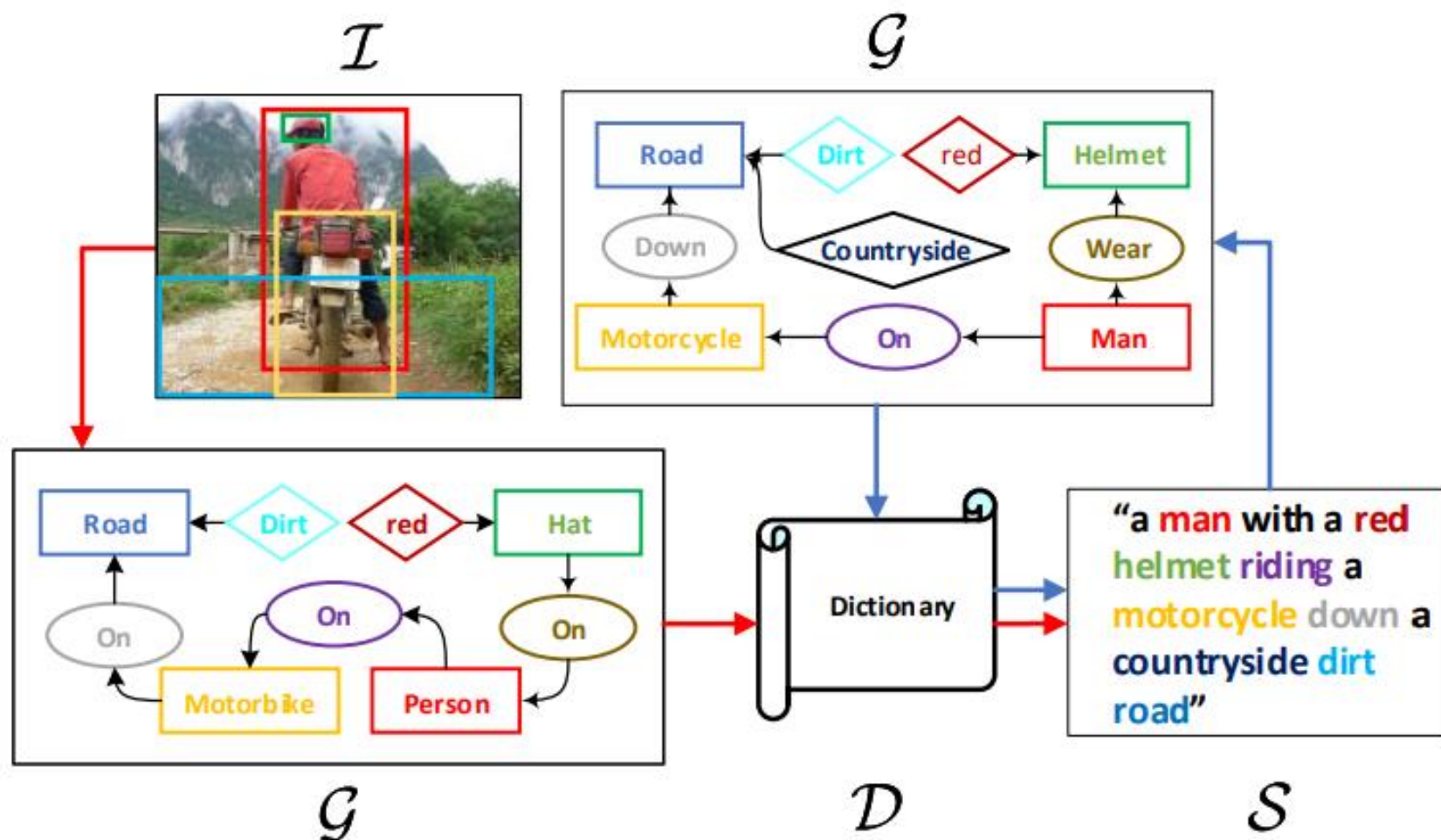
$V' \leftarrow R(V, G; D)$

解码器: $S \leftarrow V'$

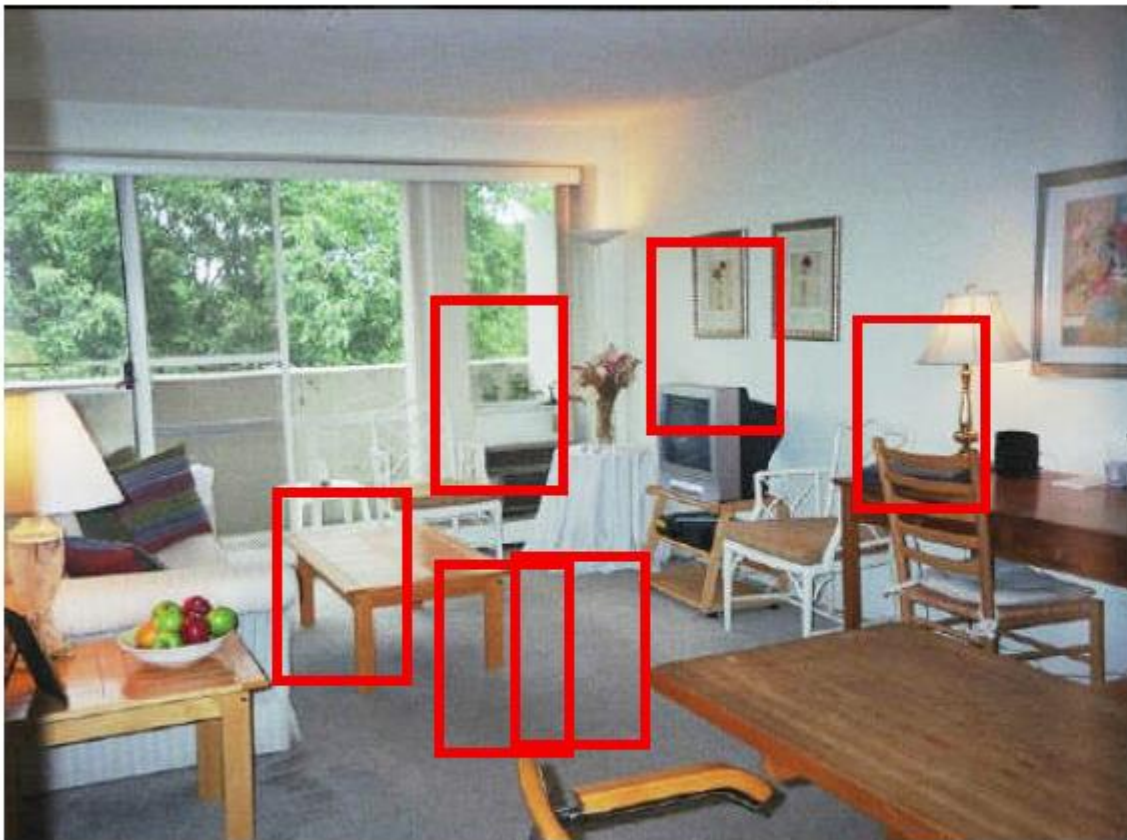


Contribution:

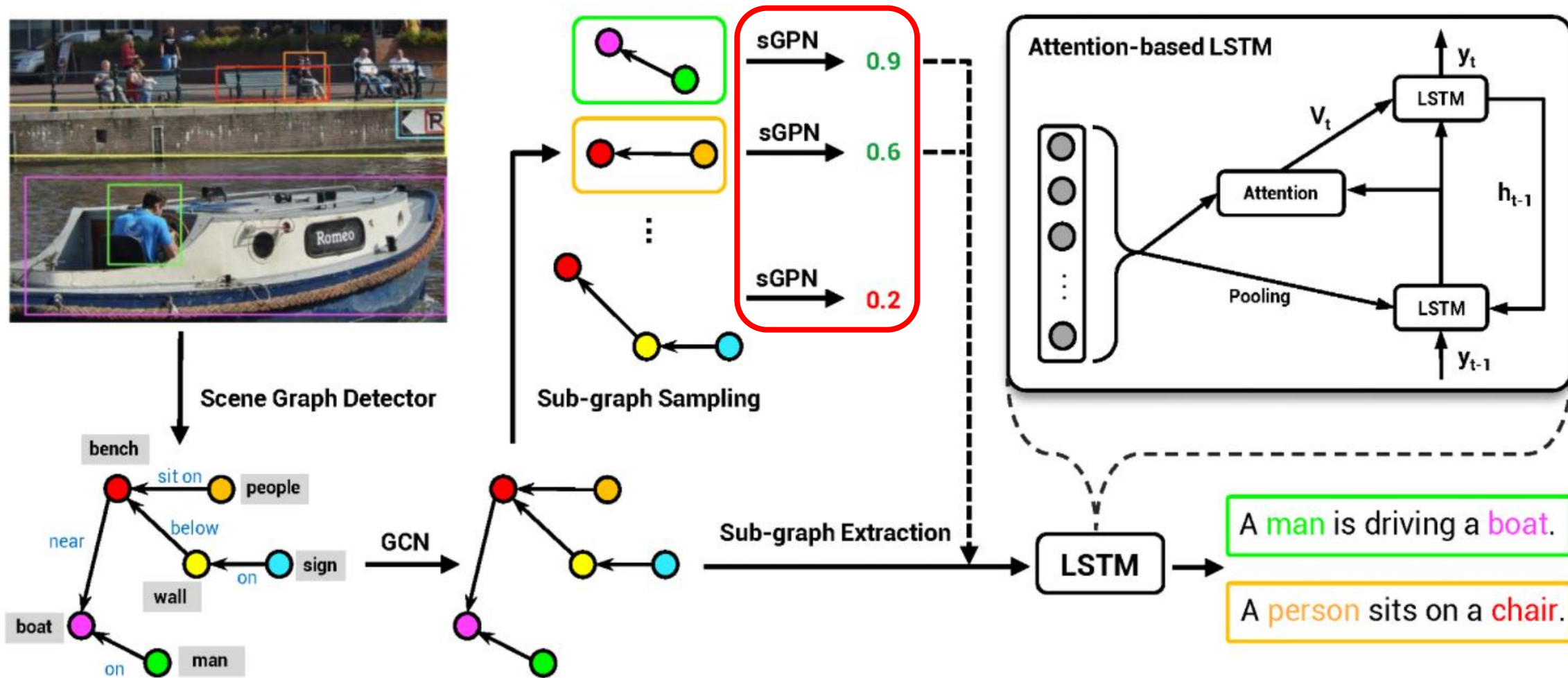
- SGAE (右半部分)
- 多模态图卷积网络
(补充视觉特征)
- 共享词典 D
(指导生成)



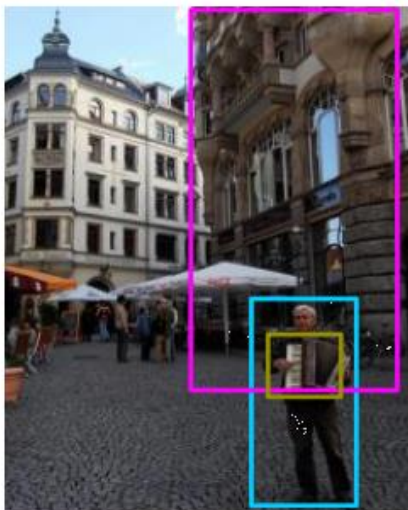
存在的问题: grounded_captioning



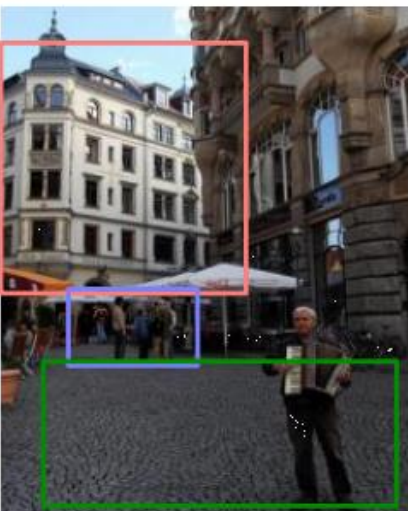
Pretty much garbage



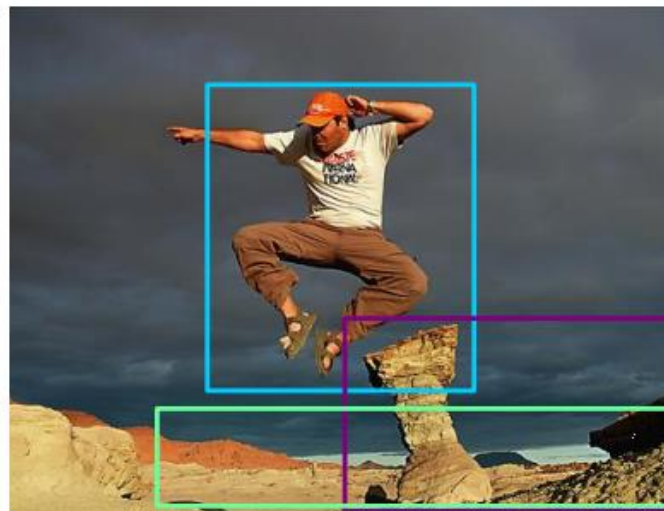
A **man** is playing a **accordion** in front of a **building**.



People walking down a **street** in a **city**.



A **man** in a **orange hat** and **brown pants** is jumping off a **rock**.



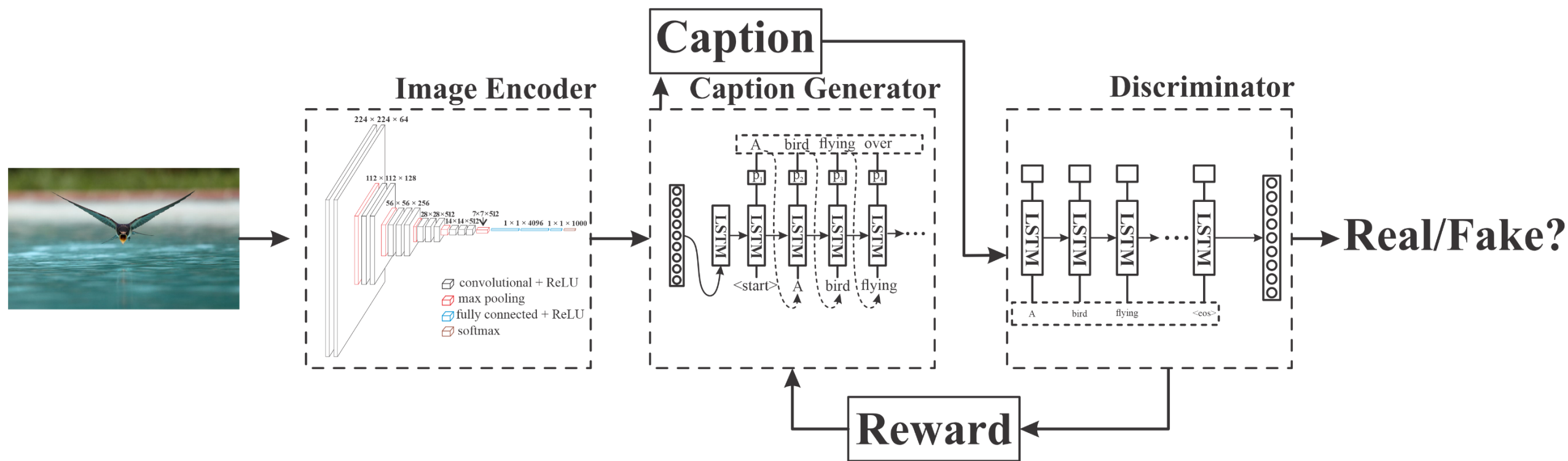
A **man** is jumping off a **rock** in a **rocky area**.



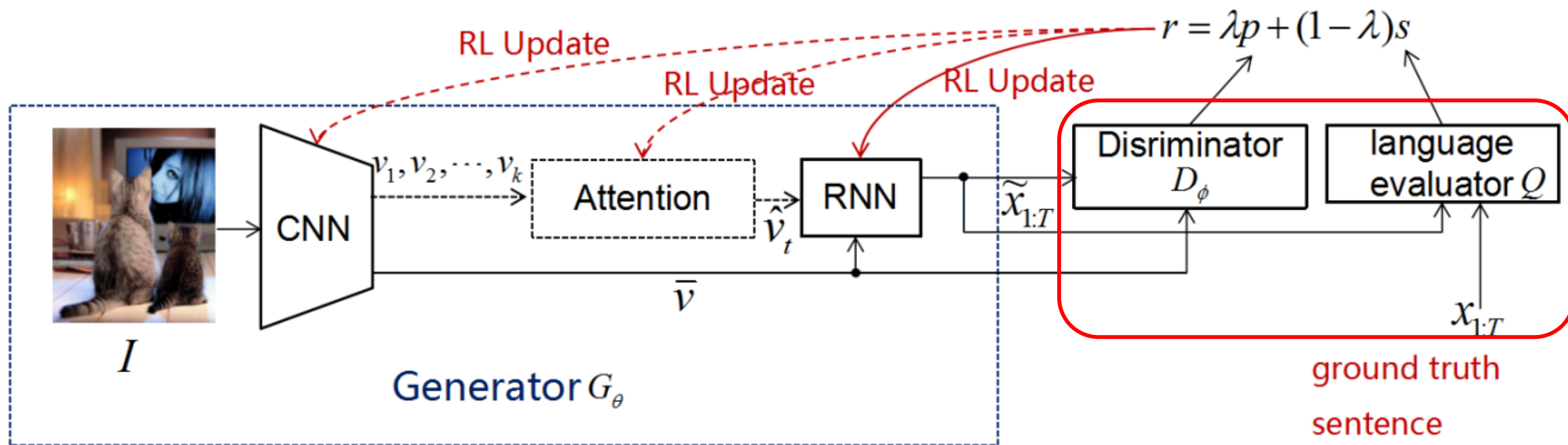
ImageNet: 1400W+张样例图片, 27大类、2W+小类

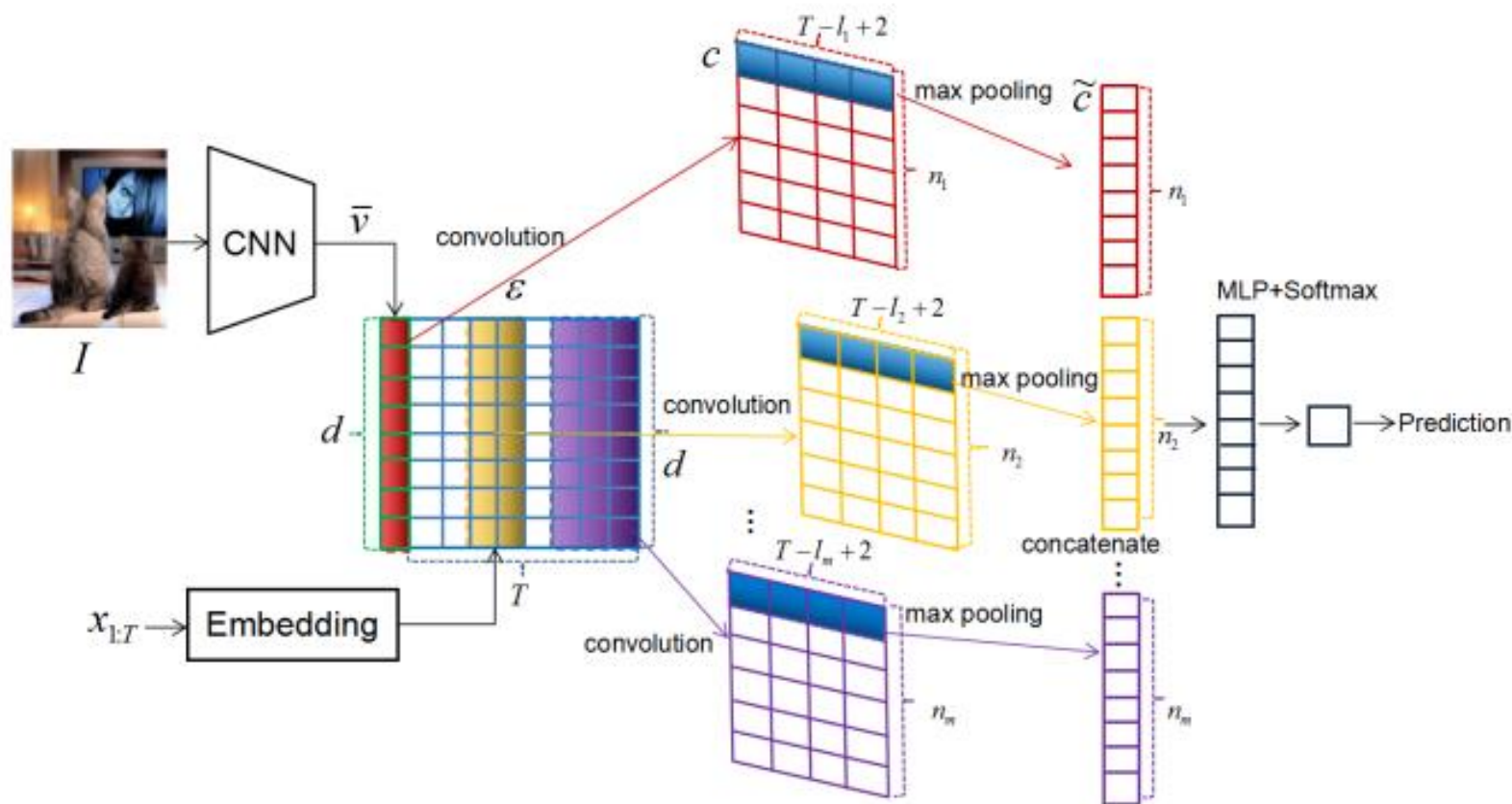
Open Images: 900万张图像, 7800种类别

Microsoft COCO: 100类, 训练11w+, 验证5k和测试4w+, 总计20w+
(很小+多种语言标题+新类别图片 → 耗时耗力)

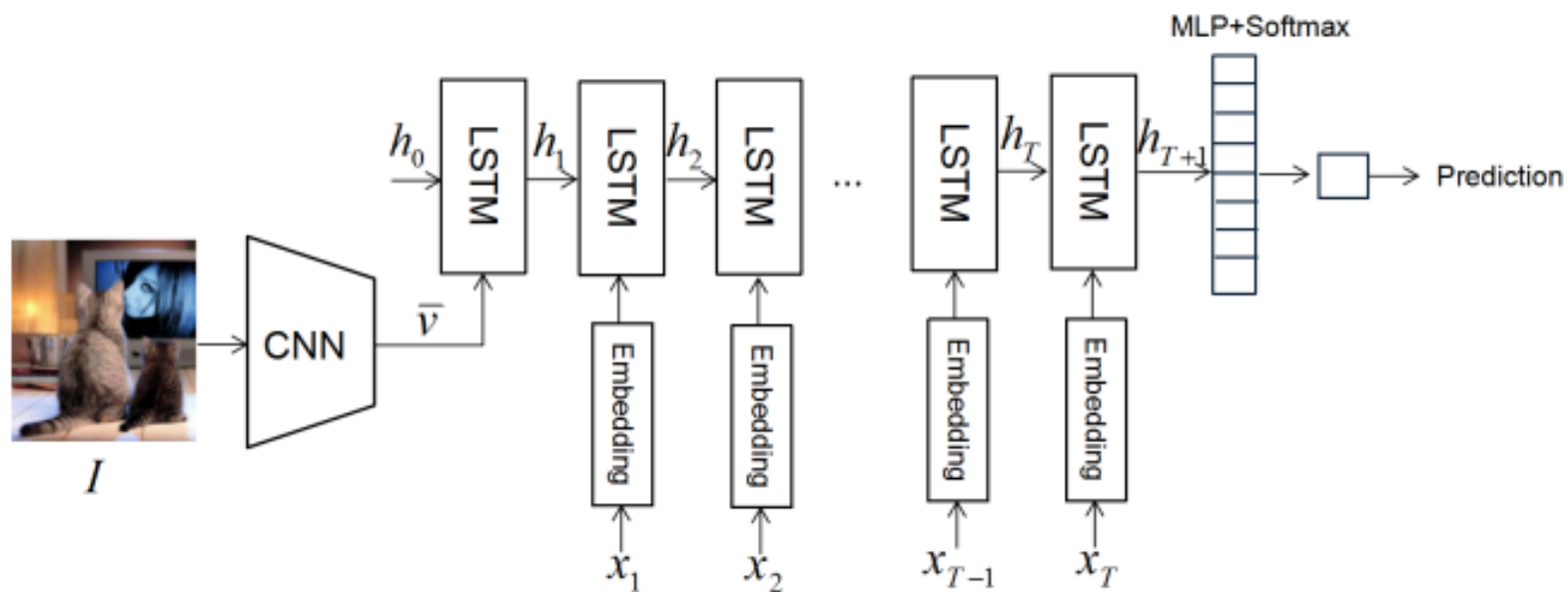


VGGNet 图像编码器，LSTM 字幕生成器，鉴别器也是LSTM，鉴别器对生成器进行相应的奖励，决定给定的标题是真实的还是由模型生成的。





(a) CNN-based discriminator



(b) RNN-based discriminator

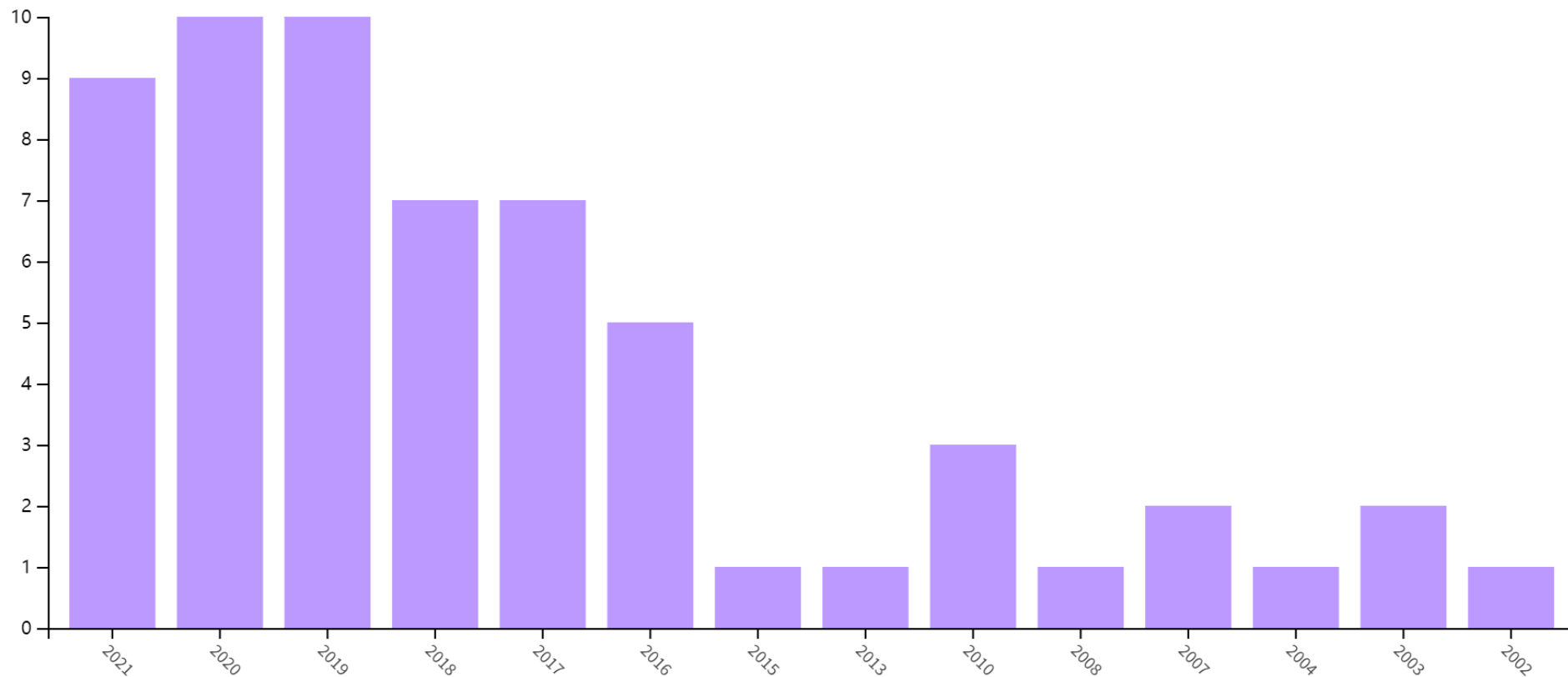


Image caption+无监督

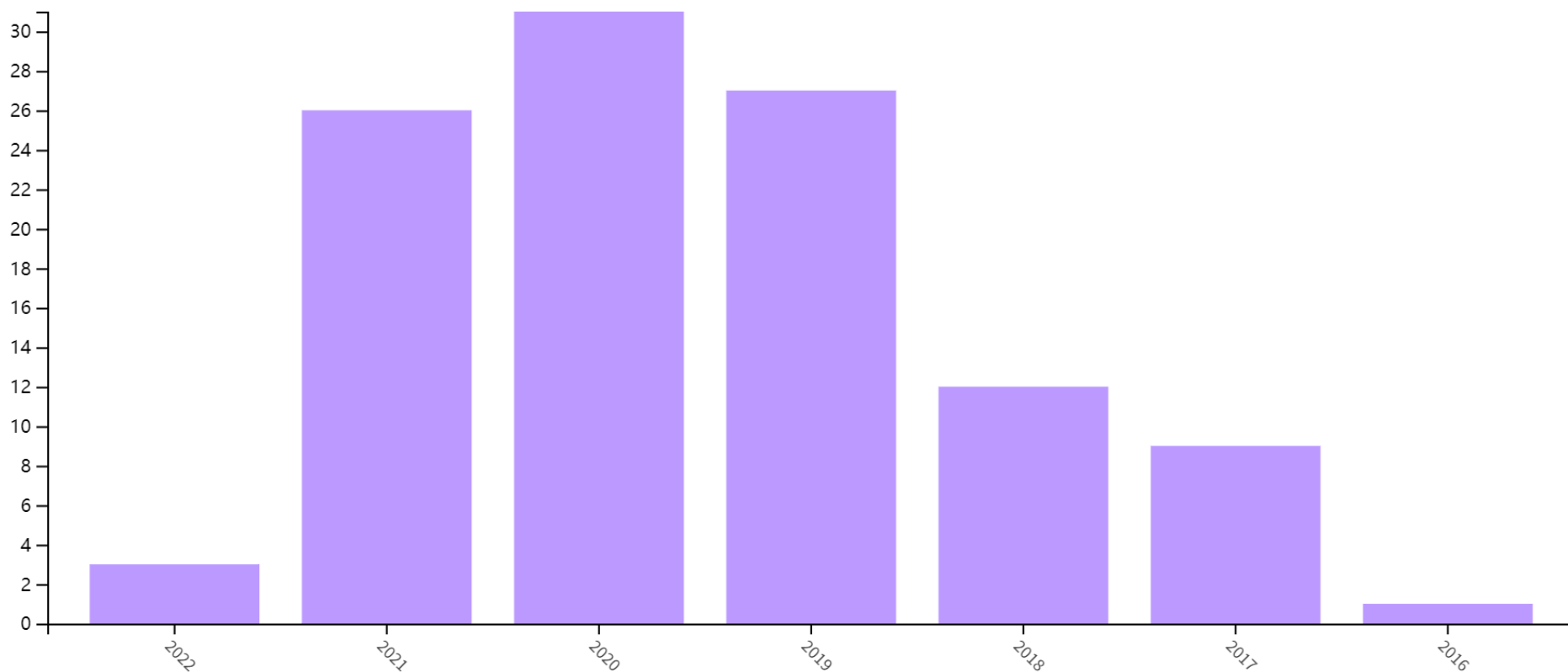


Image caption+强化学习

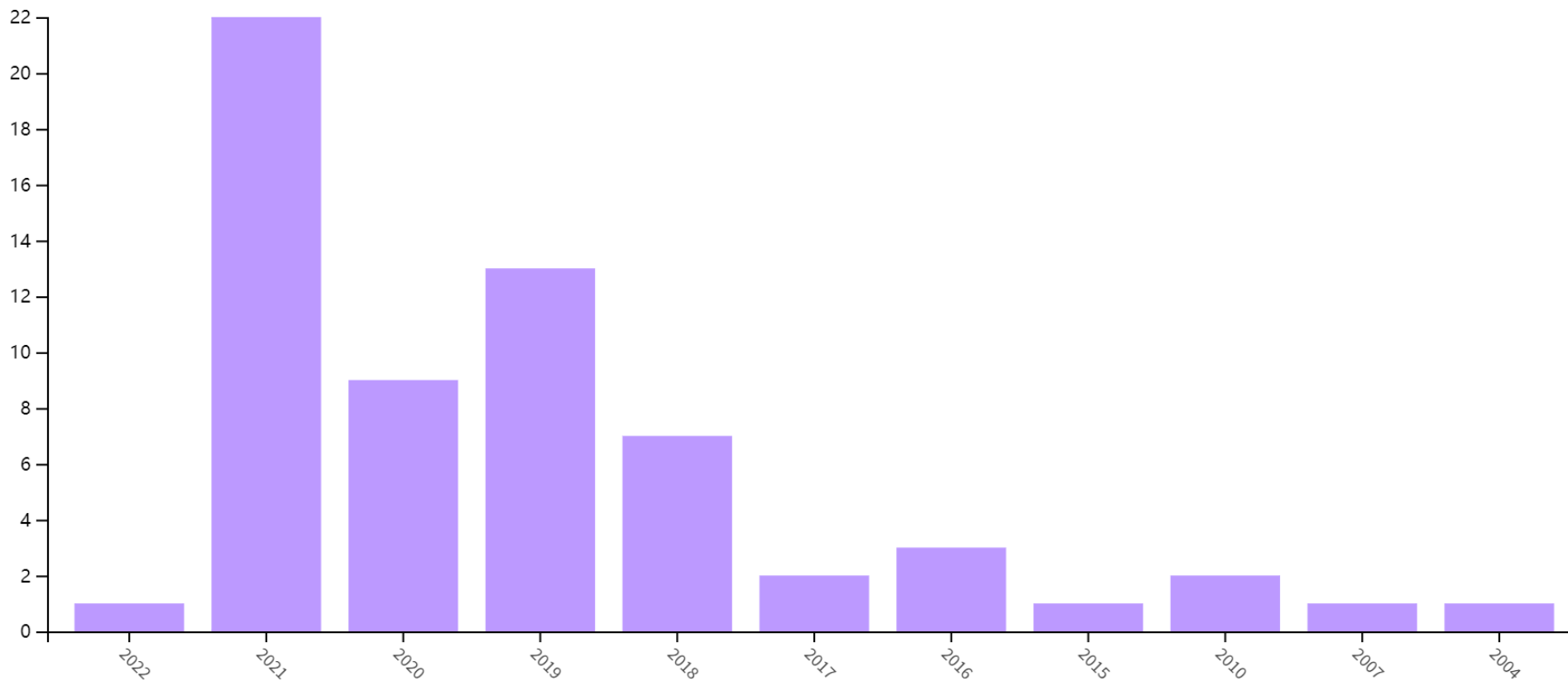


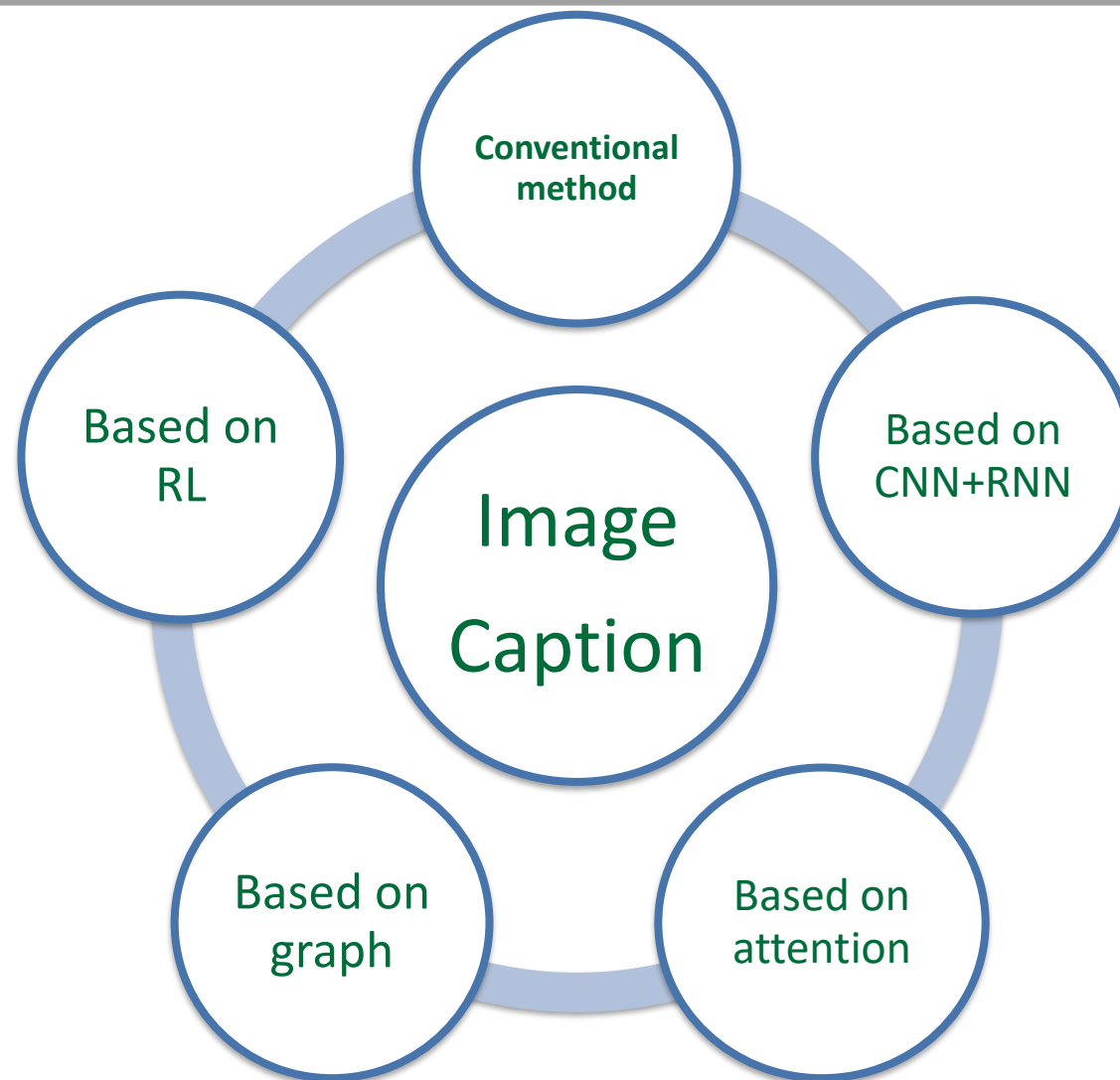
Image caption+场景图



技术展示

汇报人：王中琦

- 复现环境配置：
 - Windows 11
 - Geforce RTX 3080*1
 - torch1.10.2+cu113





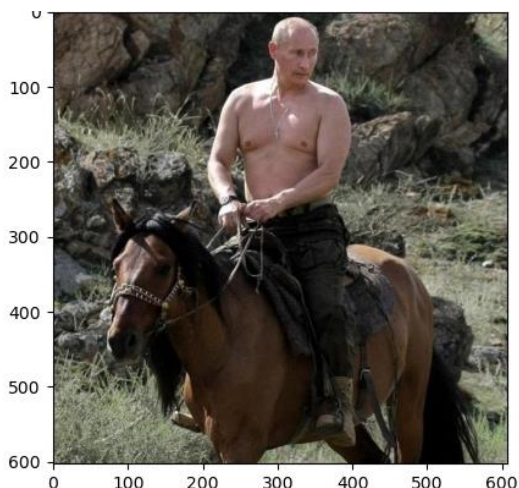
- 没有开源代码
- 国内关于这方面的技术博客几乎为0（百度、必应）
- 所创立的社区目前仍然很活跃
 - [WordsEye](http://www.wordseye.com)(<http://www.wordseye.com>)
- Github上以“Image Caption”为话题，2014年以前仅创建代码库114个



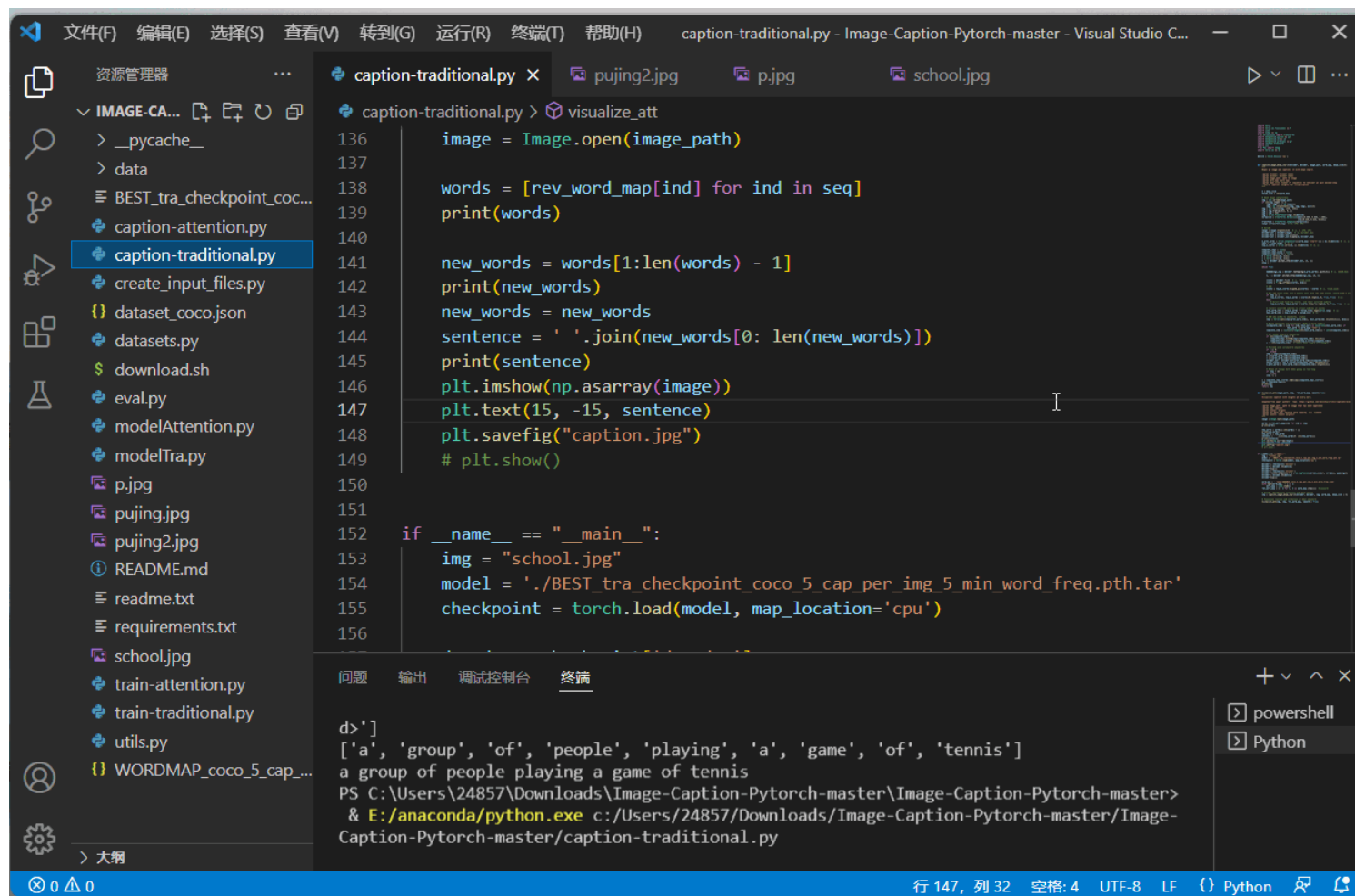
- “encoder-decoder”结构，代码量偏少
- 实验环境比较古老：python2.x、torch0.2、caffe、Lua.....
- 最热门的几个开源代码库：
 - [karpathy/neuraltalk2](https://github.com/karpathy/neuraltalk2): Efficient Image Captioning code in Torch, runs on GPU (github.com)
 - [karpathy/neuraltalk](https://github.com/karpathy/neuraltalk): NeuralTalk is a Python+numpy project for learning Multimodal Recurrent Neural Networks that describe images with sentences. (github.com)
 - [KranthiGV/Pretrained-Show-and-Tell-model](https://github.com/KranthiGV/Pretrained-Show-and-Tell-model): This repository contains pretrained Show and Tell: A Neural Image Caption Generator implemented in Tensorflow. (github.com)



a group of people standing next to each other



a man is standing next to a horse

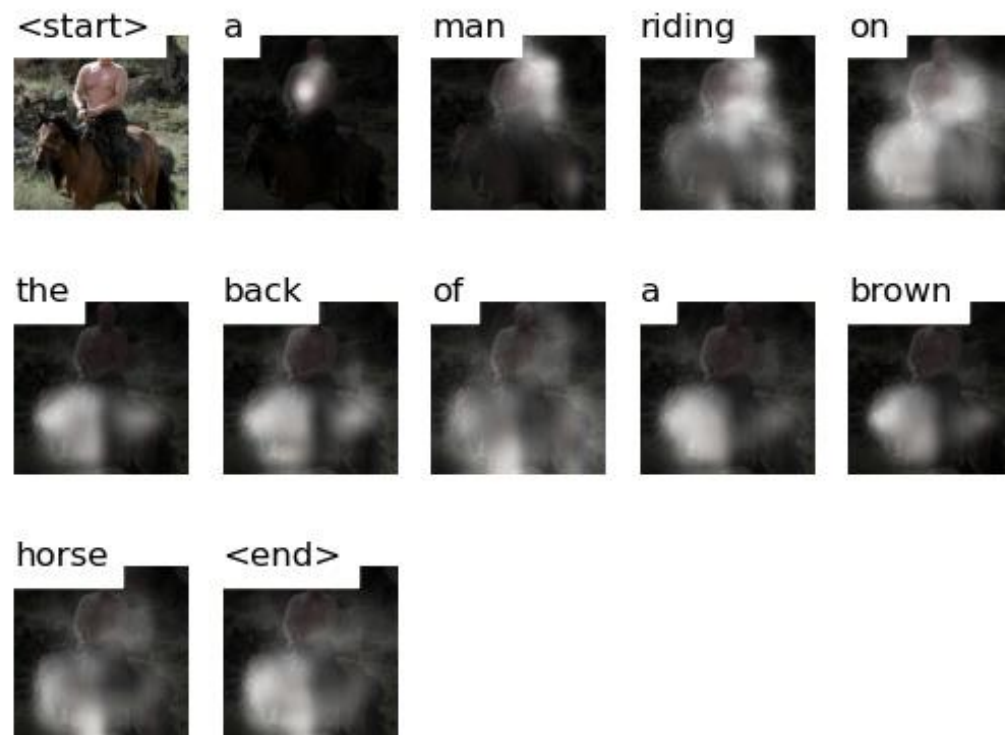


```

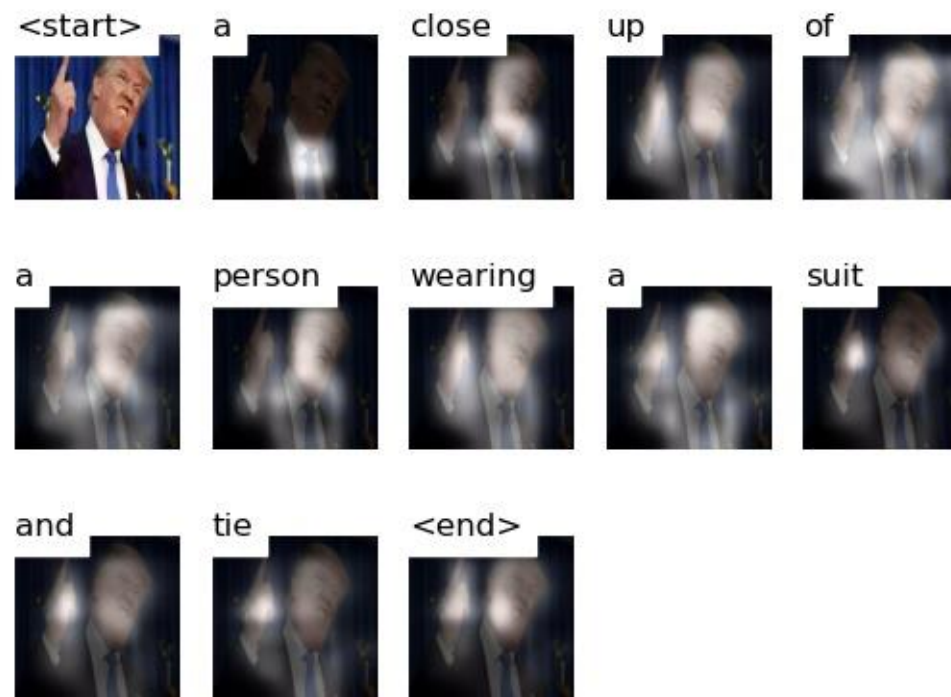
caption-traditional.py
136 image = Image.open(image_path)
137
138 words = [rev_word_map[ind] for ind in seq]
139 print(words)
140
141 new_words = words[1:len(words) - 1]
142 print(new_words)
143 new_words = new_words
144 sentence = ' '.join(new_words[0: len(new_words)])
145 print(sentence)
146 plt.imshow(np.asarray(image))
147 plt.text(15, -15, sentence)
148 plt.savefig("caption.jpg")
149 # plt.show()
150
151
152 if __name__ == "__main__":
153     img = "school.jpg"
154     model = './BEST_tra_checkpoint_coco_5_cap_per_img_5_min_word_freq.pth.tar'
155     checkpoint = torch.load(model, map_location='cpu')
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2
```



- 实验环境多样，python2.x/python3.x，torch/tenserflow
- 相关资源最丰富
- 项目维护得最好，常作为教程被引用
 - pytorch官方教程 [sgrvinod/a-PyTorch-Tutorial-to-Image-Captioning: Show, Attend, and Tell | a PyTorch Tutorial to Image Captioning \(github.com\)](https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Image-Captioning)
 - 陈云《深度学习框架Pytorch: 入门与实践》 [pytorch-book/chapter10-image_caption_at_master · chenyuntc/pytorch-book \(github.com\)](https://github.com/masterchenyuntc/pytorch-book)



A men riding on the back of a brown horse



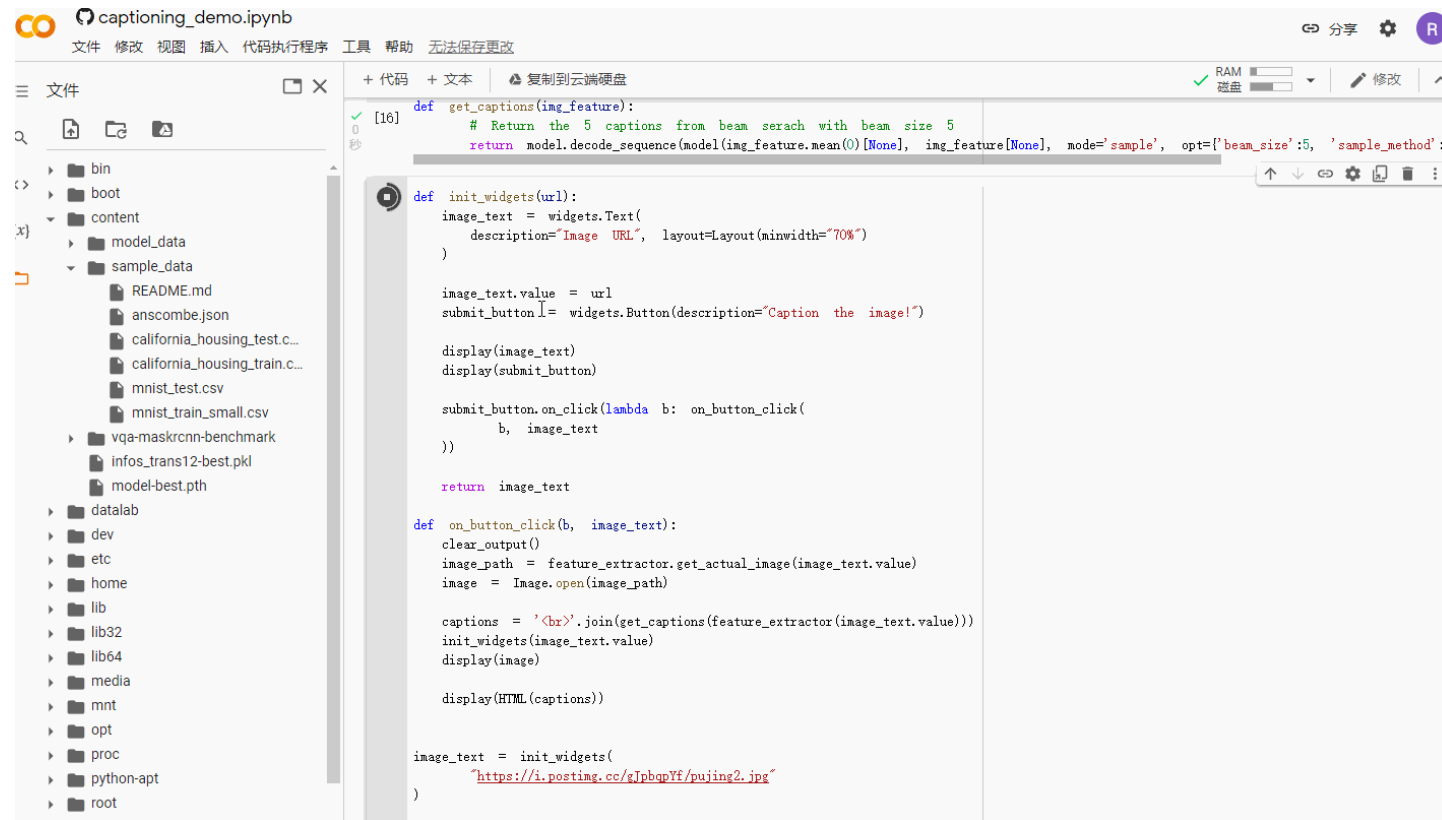
A close up of a **person** wearing a **suit** and **tie**



- 近四年出现的方法，但开源数目相对少，且复现难度较大
- 环境基本定型：pytorch1.3+、python3、GPU
- 诞生了目前最具影响力的代码（祖传）
 - [ruotianluo/self-critical.pytorch: Unofficial pytorch implementation for Self-critical Sequence Training for Image Captioning. and others. \(github.com\)](https://github.com/ruotianluo/self-critical.pytorch)
- 范式：MSCOCO2014 + Bottom up feature + Karpathy's split



a man riding on the back of a brown horse
a man riding a horse in the grass
a man riding a brown horse in a field
a man riding a horse in a grassy field
a man riding a brown horse through a field



```

captioning_demo.ipynb
文件 修改 视图 插入 代码执行程序 工具 帮助 无法保存更改

+ 代码 + 文本 复制到云端硬盘
RAM 磁盘 修改

[16] def get_captions(img_feature):
      # Return the 5 captions from beam search with beam size 5
      return model.decode_sequence(model(img_feature.mean(0))[None], img_feature[None], mode='sample', opt={'beam_size':5, 'sample_method':

def init_widgets(url):
    image_text = widgets.Text(
        description="Image URL", layout=Layout(minwidth="70%")
    )

    image_text.value = url
    submit_button = widgets.Button(description="Caption the image!")

    display(image_text)
    display(submit_button)

    submit_button.on_click(lambda b: on_button_click(
        b, image_text
    ))

    return image_text

def on_button_click(b, image_text):
    clear_output()
    image_path = feature_extractor.get_actual_image(image_text.value)
    image = Image.open(image_path)

    captions = '<br>'.join(get_captions(feature_extractor(image_text.value)))
    init_widgets(image_text.value)
    display(image)

    display(HTML(captions))

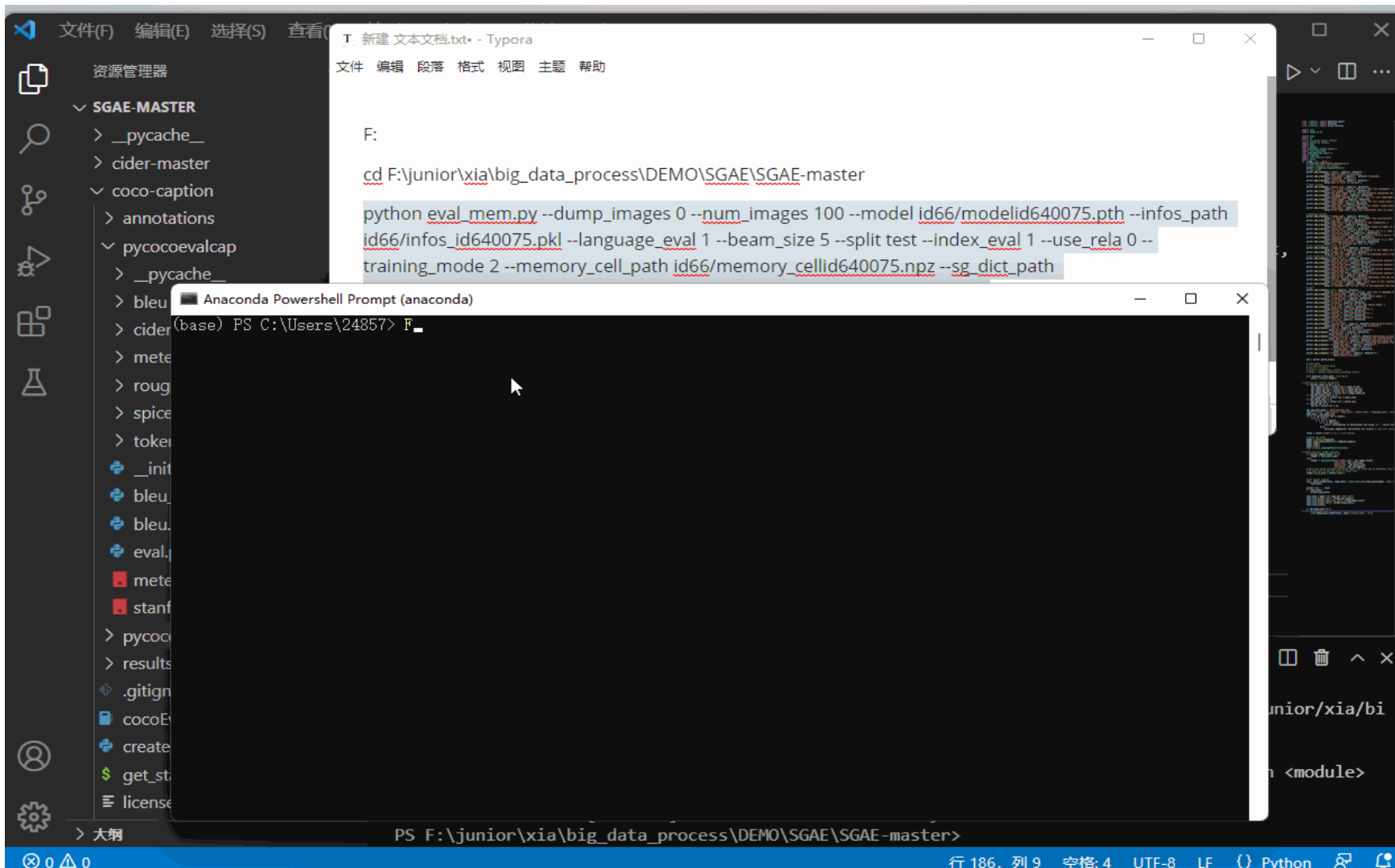
image_text = init_widgets(
    "https://i.postimg.cc/gIpbqYf/pujiang2.jpg"
)

```

By Google colab



- 同样，近年诞生的方法，开源生态逐渐丰富，对环境配置要求较高，复现难度较大
- 维护非常好的代码库：
 - [microsoft/scene_graph_benchmark: image scene graph generation benchmark \(github.com\)](https://github.com/microsoft/scene_graph_benchmark)
 - [google/sg2im: Code for "Image Generation from Scene Graphs", Johnson et al, CVPR 2018 \(github.com\)](https://github.com/google/sg2im)
 - [KaihuaTang/Scene-Graph-Benchmark.pytorch: A new codebase for popular Scene Graph Generation methods \(2020\). Visualization & Scene Graph Extraction on custom images/datasets are provided. It's also a PyTorch implementation of paper "Unbiased Scene Graph Generation from Biased Training CVPR 2020" \(github.com\)](https://github.com/KaihuaTang/Scene-Graph-Benchmark.pytorch)



文件(F) 编辑(E) 选择(S) 查看(V) T: 新建 文本文档.txt - Typora

文件 编辑 段落 格式 视图 主题 帮助

F:

```
cd F:\junior\xia\big_data_process\DEMO\SGAE\SGAE-master
python eval_mem.py --dump_images 0 --num_images 100 --model id66/modelid640075.pth --infos_path
id66/infos_id640075.pkl --language_eval 1 --beam_size 5 --split test --index_eval 1 --use_rela 0 --
training_mode 2 --memory_cell_path id66/memory_cellid640075.npz --sg_dict_path
```

Anaconda Powershell Prompt (anaconda)

```
(base) PS C:\Users\24857> F:
PS F:\junior\xia\big_data_process\DEMO\SGAE\SGAE-master>
```

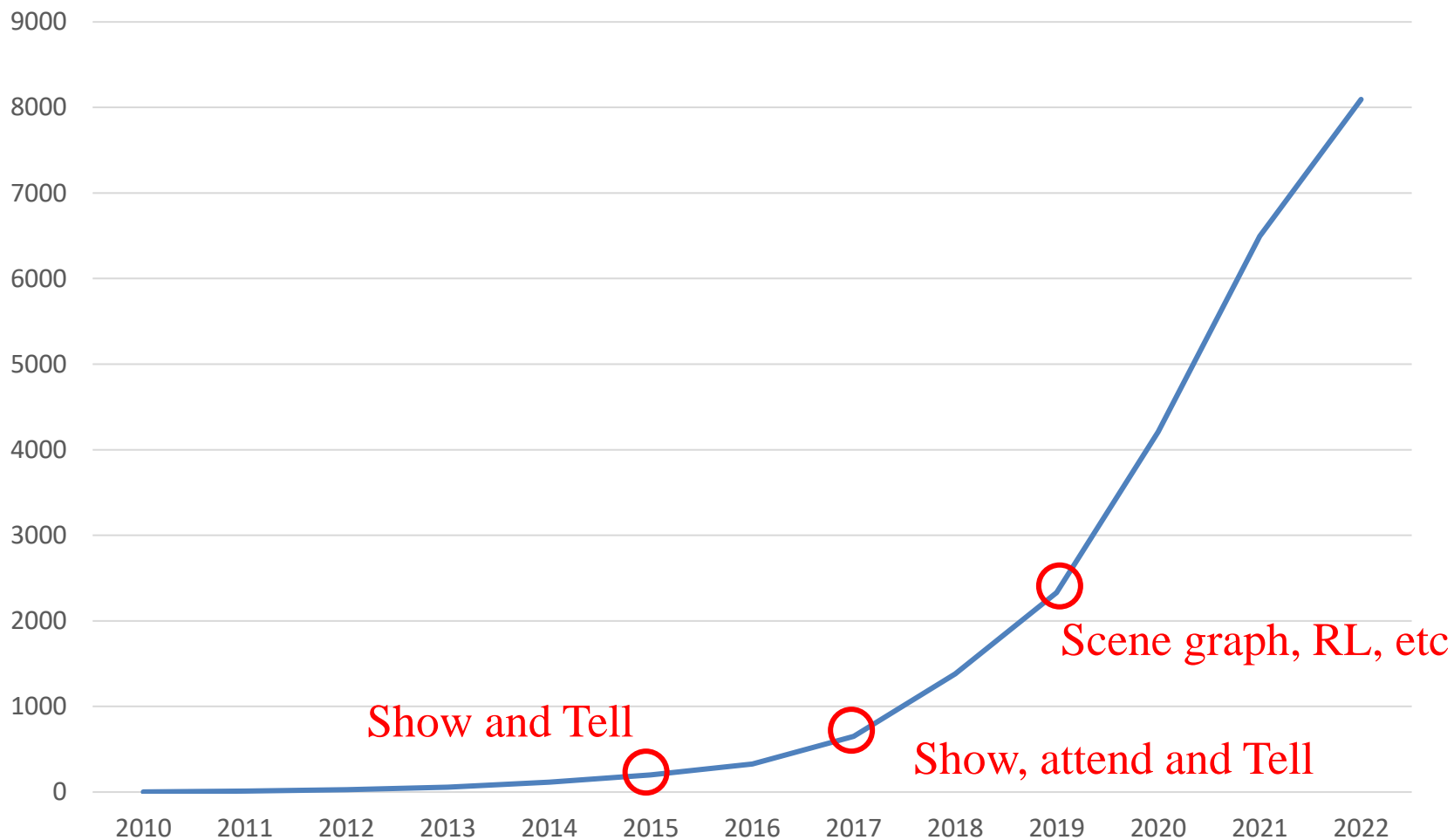
资源管理器

- SGAE-MASTER
 - __pycache__
 - cider-master
 - coco-caption
 - annotations
 - pycocoevalcap
 - __pycache__
 - bleu
 - cider
 - meteor
 - rouge
 - spice
 - token
 - __init__
 - bleu_
 - bleu_
 - eval_
 - mete
 - stanf
 - pycoc
 - results
 - .gitign
 - cocoE
 - create
 - get_st
 - license

> 大纲

行 186, 列 9 空格: 4 UTF-8 LF () Python

随年份代码提交总量 (Github)





现场演示



谢谢观看

答辩人：刘云皓，李桐，邱小尧，
刘天锐，王中琦
时间：2021-3-8

学以精工
德以明理