Image Captioning Based on Deep Neural Networks

Xter: Hà Quốc Huy - huyhqFX04371
Mentor: Nguyễn Qúy

Mở đầu: Với sự phát triển của deep learning, tổ hợp của computer vision và natural language process đã thu hút sự chú ý trong những năm gần đây. Image captioning là đã diện cho trường lĩnh vực này. Image captioning: khiến máy tính học cách sử dụng một hoặc nhiều câu để hiểu nội dung trực quan của hình ảnh. Qúa trình nhận biết ngữ nghĩa của hình ảnh không chỉ dừng lại ở việc nhận biết đối tượng và khung cảnh mà còn là khả năng phân tích trạng thái, mối quan hệ phụ thuộc của các đối tượng này. Vì vậy image caption là trường nghiên cứu khó khăn và phức tạp đến nay đã có nhiều kết quả tích cực cho mô hình nghiên cứu. Trong paper này tôi tập chung đi sâu vào sử dụng deep neural network: CNN-RNN based , đi kèm là giới thiệu, đề cập các metrics, vấn đề trong mô hình.

1 – Cơ sở lý thuyết

Trong vài năm gần đây, thị giác máy tính đã có sự tiến triển vượt bật, nổi bật là hai lĩnh vực image classification [1] và object detection [2]. Từ việc phát triển này, việc nghiên cứu cho máy tính có khả năng tự động tạo ra 1 hoặc 2 câu hiểu nội dung đằng sau bức ảnh để hình thành trường image captioning. Việc khởi tạo được mô tả dưới dạng text cho nội dung của một bức ảnh là hữu ích, có thể thấy những ứng dụng phổ biến của image captioning như: mô tả thông tin cho hình ảnh y tế, trích xuất ảnh dựa trên văn bản, tiếp cận thông tin dành cho người mù và tương tác giữa robot với con người.

Input là một hình ảnh mới khi đó thuật toán image captioning sẽ cho output là bản mô tả về hình ảnh ở cấp độ ngữ nghĩa. Ví dụ trong hình ảnh Ptc. 1 dưới đây

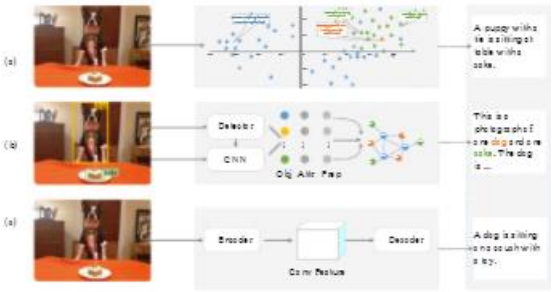
Ptc. 1 . A man is waving his hands on the mountain

input là hình ảnh bao gồm: con người, khung cảnh thì output sẽ là mô tả cho hành động của con người cũng như khũng cảnh được mô tả trong trường text.

Đối với nhiệm vụ image captioning, con người dễ dàng nắm bắt nội dung của hình ảnh theo dạng và thể hiện nó trong khuôn mẫu NLP. Đối với máy tính để làm được như vậy, nó yêu cầu tiến trình kết hợp sử dụng của: image processing, computer vision, natural language processing và một số trường nghiên cứu khác để cho ra kết quả. Thử thách đối với image captioning là khởi tạo ra model hiệu quả nhất trong việc phát sinh nhiều bản mô tả trường text với ý nghĩa khớp với suy nghĩa của con người. Nhìn chung vẫn chưa có cái nhìn rõ ràng về việc làm thế nào bộ não hiểu hình ảnh và tổ chức trình phân tích thông tin và đưa ra nội dung của hình ảnh. Image captioning liên quan đến một cái nhìn sâu về thế giới và đâu là phần nổi trội trong tổng thể.

Mặc dù vậy, image captioning đã có những cải tiến đáng kể trong những năm gần đây. Thuật toán image captioning có thể được chia thành 3 loại. Loại thứ nhất Ptc. 2, sử dụng phương pháp dựa trên truy hồi – the retrieval-based methods, phương pháp này đầu tiên sẽ truy vấn các hình ảnh phù hợp nhất và chuyển đổi bản mô tả của chúng thành mô tả cho hình ảnh truy vấn. Phương pháp này có thể tạo ra trường text đúng ngữ pháp nhưng không tạo ra được chú thích cho hình ảnh mới. Loại thứ 2 Ptc. 2 dựa trên mẫu – the template-based methods . Phương pháp này xây dựng bản mô tả dựa trên cấu trúc ngữ pháp đã được quy định và tiến hành chia nhỏ các câu thành nhiều phần. Phương pháp này dựa trên trình phân loại đối tượng và sau đó áp dụng mẫu câu thô cho hình ảnh truy vấn.

Mặc dù nó có thể tạo ra một câu mới, nhưng các phương pháp này không thể diễn đạt ngữ cảnh trực quan một cách chính xác hoặc tạo ra các câu linh hoạt và có ý nghĩa.



Ptc.2. 3 Phương thức của image captioning

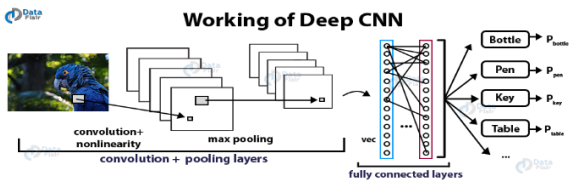
Loại cuối cùng cũng là phương thức phổ biến đó là phương thức dựa trên mạng nerual network – the nn-based methods Ptc.2 . Lấy cảm hứng từ kiến trúc bộ mã hóa-giải mã của máy học [5], những năm gần đây hầu hết các phương pháp tạo phụ đề hình ảnh đều sử dụng Mạng thần kinh kết hợp (CNN) làm bộ mã hóa và Thần kinh tái tạo. Mạng (RNN) làm bộ giải mã, đặc biệt là Bộ nhớ ngắn hạn dài (LSTM) [6] để tạo phụ đề [7], với mục tiêu tối đa hóa khả năng của một câu với các đặc điểm trực quan của hình ảnh. Một số phương pháp đang sử dụng CNN làm bộ giải mã và học tập củng cố làm mạng ra quyết định.

Phần tiếp theo tôi sẽ nói tiệm cận về CNN-RNN based framework và hướng ứng dụng LSTM để xử lý bài toán image captioning.

2 – CNN-RNN based framework

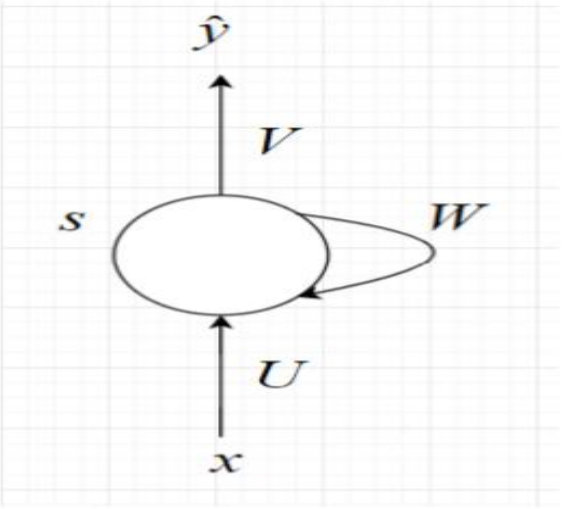
CNN là dạng neural sâu chuyên biệt có thể xử lý dữ liệu có hình dạng đầu vào giống như ma trận 2D. Hình ảnh có thể dễ dàng đại diện dưới dạng một ma trận 2D và CNN hữu ích trong việc xử lý với hình ảnh này.

CNN sử dụng một trình phân loại hình ảnh và xác định hình ảnh là chim, máy bay hay siêu nhân...



Nó sẽ quét hình ảnh từ trái qua phải từ trên xuống dưới để lấy ra các tính năng sau đó kết hợp chúng lại để phân loại các hình ảnh. Nó có thể xử lý các hình ảnh đã được dịch, xoay, thu nhỏ và thay đổi góc nhìn. Tuy nhiên điểm hạn chế của CNN nằm ở việc chỉ xử lý được input tương ứng với 1 image. Bài toán : Cần phân loại hành động của người trong video, input là video 30s, output là phân loại hành động, ví dụ: đứng, ngồi, chạy, đánh nhau, bắn súng,...Ta giả sử 1s tương ứng 1 img cần xử lý như vậy 30s tương ứng với 30 ảnh. Ta có thể dùng CNN để phân loại 1 ảnh trong 30 ảnh trên, nhưng rõ ràng là 1 ảnh không thể mô tả được nội dung của cả video. Như vậy cần mô hình mới có thể giải quyết được bài toán với input là sequence (chuỗi ảnh 1-30) .Khi đó có sự xuất hiện của RNN (Recurrent Neural Network)

RNN xử lý input là sequence hoặc time-series data. Mô hình RNN rút gọn Ptc. 3

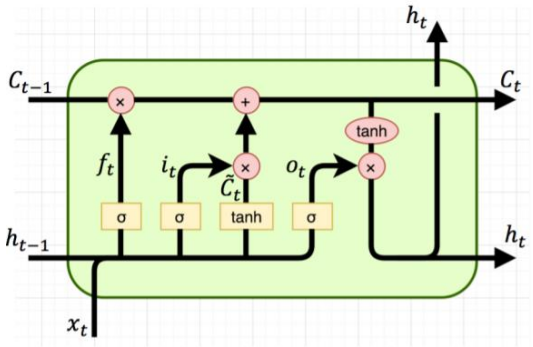


Ptc.3. Mô hình rút gọn RNN

Mô hình RNN tận dụng từng thành phần của quá trình phân đoạn input làm đầu vào cho mỗi state cùng với đó kết hợp với output của state trước đó làm nghiệm cho state tiếp theo. Cấu trúc mô hình tuần tự cho đến state cuối thông qua activation function trả về output cuối cùng. Các hệ số cần tối ưu hóa trong mô hình như : V, W ptc.3 được xử lý qua backpropagation through time (BPTT). Tuy nhiên thì điểm bất lợi của RNN là xuất hiện vanishing gradient đối với các state càng ở xa trước đó, các hệ số không được update. Hay nói cách khác RNN không đọc được các thông tin ở trước đó xa do vanishing gradient. Như vậy về lý thuyết

RNN có thể mang thông tin từ các layer trước đến các layer sau, nhưng thực tế là thông tin chỉ mang được qua một số lượng state nhất định, sau đó thì sẽ bị vanishing gradient, hay nói cách khác là model chỉ học được từ các state gần nó => short term memory. Vấn đề được giải quyết khi có sự có mặt của LSTM.

LSTM. Tổng quan về cấu trúc mô hình. Output: ct ,ht , ta gọi c là cell state, h là hidden state. Input: ct−1,ht−1, xt . Trong đó xt là input ở state thứ t của model. ct−1, ht−1 là output của layer trước. h đóng vai trò khá giống như s ở RNN, trong khi c là điểm mới của LSTM, như trong hình Ptc.4



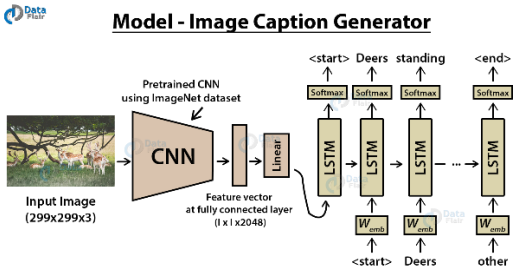
Ptc.4. Mô hình LSTM

Cách đọc biểu đồ trên: ta thấy kí hiệu σ, tanh ý là bước đấy dùng sigma, tanh activation function. Phép nhân ở đây là element-wise multiplication, phép cộng là cộng ma trận. So khớp với mô hình RNN có sự trùng khớp về W, U , output gate (ht). Trong khi đó ct giống như một băng chuyền ở trên mô hình RNN vậy, thông tin nào cần quan trọng và dùng ở sau sẽ được gửi vào và dùng khi cần như vậy có thể mang thông tin từ đi xa. Do đó mô hình LSTM có cả short term memory và long term memory.

Nguyên nhân dẫn tới vansishing gradient trong RNN $\frac{\partial s_{t+1}}{\partial s_t} = (1 - s_t^2) * W$, trong đó $s_t, W < 1$. . Tương tự trong LSTM ta quan tâm đến $\frac{\partial c_t}{\partial c_{t-1}} = f_t$. Do $0 < f_t < 1$ về cơ bản thì LSTM vẫn bị vanishing gradient nhưng bị ít hơn so với RNN. Hơn thế nữa, khi mang thông tin trên cell state thì ít khi cần phải quên giá trị cell cũ, nên ft ≈ 1 dẫn tới tránh được vanishing gradient.

3 – Phân tích, lựa chọn và xây dựng models cho bài toán image captioning

Trong project này tôi sẽ kết hợp hai cấu trúc CNN và LSTM để xử lý bài toán image captioning. Nó cũng được gọi là CNN-RNN model. CNN đảm nhiệm việc trích xuất các features từ ảnh . Tôi sẽ sử dụng pre-trainded model InceptionV3. LSTM sẽ sử dụng thông tin từ CNN để khởi tạo mô tả của hình ảnh, như hình Ptc.5

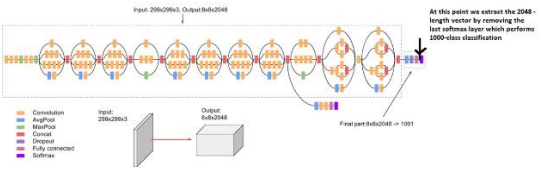


Ptc.5 . Model – Image Caption Ge

a.Phân tích cơ sở dữ liệu và tiền xử lý
- Training set: 6000 images
- Dev set: 1000 images
- Test set: 1000 images

Mỗi hình ảnh chứa 5 captions.
Cấu trúc mỗi dòng dữ liệu theo định dạng: <image name>#i<caption>, với 0<=i<=4
Xử lý raw caption:
+ Loại bỏ ký tự đặc biệt (“%, \”, etc.)
+ Loại bỏ các ký tự số
+ Lower-casing các từ

Khởi tạo bộ từ vựng đơn nhất của 5*8000 captions. Kết quả thực nghiệm thu được bộ từ vựng đơn nhất với số lượng 8763 từ tuy nhiên có một số từ vựng xuất hiện với tuần xuất thấp điều năng tăng cường model và ở đây tôi lấy bộ từ vựng với tần xuất xuất hiện ít nhất 10 lần. Kết quả thu được 1651 từ + 0’s (zero padding) có kết quả 1652. Bước tiếp theo tôi xử lý clean captions theo định dạng ‘startseq + caption+ ‘endseq’ là input cho mô hình LSTM.
Xử lý image:
Model tôi sử dụng cho việc trích xuất feature cho mỗi image là InceptionV3. Mô hình này perform trình phân loại 1000 lớp khác nhau. Tuy nhiên mục đích của chúng ta là feature vector bởi vậy layer cuối được loại bỏ. Mô hình InceptionV3, như hình Ptc. 6:



Ptc.6. Mô hình InceptionV3

Xử lý captions:

Tôi sẽ dự đoán caption qua hình thức word by word. Input vào mô hình cần ở dạng số nên mỗi từ sẽ được chuyển đổi thành dạng vector với kích cỡ cố định. Tiến trình chuyển đổi này được tôi sử dụng kĩ thuật word embedding được nhắc tới ở phần sau. Trong phần này tôi chuyển đổi mỗi caption thành vector cố định với mỗi thành phần là index các từ , kĩ thuật được sử dụng là Tokenizer() từ thư viện keras. Khởi tạo một biết max_length tìm độ dài caption lơn nhất nhằm xác định cấu trúc tham số cho mô hình LSTM.

b.Chuẩn bị dữ liệu cho mô hình
Vấn đề đặt ra làm thế nào để kết hợp bộ dữ liệu bao gồm image feature, caption feature vào trong mô hình của tôi? Một hàm khởi tạo được sử dụng. Giả sử ở đây tôi có hai hình ảnh đi cùng với caption, như hình Ptc. 7:

(Train image 1) Caption -> The black cat sat on grass

(Train image 2) Caption -> The white cat is walking on road

Ptc. 7 . Ví dụ

Chuyển đổi “image-1”, ”image-2” lần lượt thành hai vector có 2048 feature. Tiếp theo tôi xây dựng bộ từ vựng bằng cách thêm <start>, <end> vào mỗi caption. Định dạng cụ thể:
Cap1: “’startseq the black cat sat on grass endseq”
Cap2: “startseq the white cat is walking on road endseq”
Vocab={black, car, endseq, grass, is, on, road, sat, startseq, the, walking, white}
Tôi tiến hành gán index cho mỗi từ :
Black – 1, cat-2, end – 3, grass – 4, is – 5, on – 6, road – 7, sat – 8, start – 9, the – 10, walking – 11, white – 12
Tôi có tập hợp điểm dữ liệu D ={Xi, Yi}. Xi là feature vector của điểm dữ liệu I, Yi là gán nhãn dữ liệu. Bảng ma trận dữ liệu của tôi ,như hình Ptc.8:

	Xi		Yi	
i	Image feature vector	Partial Caption	Target word	
1	Image_1	startseq	the	data points corresponding to image 1 and its caption
2	Image_1	startseq the	black	
3	Image_1	startseq the black	cat	
4	Image_1	startseq the black cat	sat	
5	Image_1	startseq the black cat sat	on	
6	Image_1	startseq the black cat sat on	grass	
7	Image_1	startseq the black cat sat on grass	endseq	
8	Image_2	startseq	the	data points corresponding to image 2 and its caption
9	Image_2	startseq the	white	
10	Image_2	startseq the white	cat	
11	Image_2	startseq the white cat	is	
12	Image_2	startseq the white cat is	walking	
13	Image_2	startseq the white cat is walking	on	
14	Image_2	startseq the white cat is walking on	road	
15	Image_2	startseq the white cat is walking on road	endseq	

Data Matrix for both the images and captions

Ptc.8

Tuy nhiên đã sẵn sàng cho việc gán index cho mỗi từ. Chuyển đổi bảng dữ liệu trên theo index, như hình Ptc.9:

	Xi		Yi
i	Image feature vector	Partial Caption	Target word
1	Image_1	[9]	10
2	Image_1	[9, 10]	1
3	Image_1	[9, 10, 1]	2
4	Image_1	[9, 10, 1, 2]	8
5	Image_1	[9, 10, 1, 2, 8]	6
6	Image_1	[9, 10, 1, 2, 8, 6]	4
7	Image_1	[9, 10, 1, 2, 8, 6, 4]	3
8	Image_2	[9]	10
9	Image_2	[9, 10]	12
10	Image_2	[9, 10, 12]	2
11	Image_2	[9, 10, 12, 2]	5
12	Image_2	[9, 10, 12, 2, 5]	11
13	Image_2	[9, 10, 12, 2, 5, 11]	6
14	Image_2	[9, 10, 12, 2, 5, 11, 6]	7
15	Image_2	[9, 10, 12, 2, 5, 11, 6, 7]	3

Data matrix after replacing the words by their indices

Ptc.9

Cần đảm bảo cho mỗi sequence có chiều dài bằng nhau. 0’S được sử dụng lúc này, như hình Ptc.10:

	Xi		Yi
i	Image feature vector	Partial Caption	Target word
1	Image_1	[9, 0, 0 ...., 0]	10
2	Image_1	[9, 10, 0, 0 ...., 0]	1
3	Image_1	[9, 10, 1, 0, 0 ...., 0]	2
4	Image_1	[9, 10, 1, 2, 0, 0 ...., 0]	8
5	Image_1	[9, 10, 1, 2, 8, 0, 0 ...., 0]	6
6	Image_1	[9, 10, 1, 2, 8, 6, 0, 0 ...., 0]	4
7	Image_1	[9, 10, 1, 2, 8, 6, 4, 0, 0 ...., 0]	3
8	Image_2	[9, 0, 0 ...., 0]	10
9	Image_2	[9, 10, 0, 0 ...., 0]	12
10	Image_2	[9, 10, 12, 0, 0 ...., 0]	2
11	Image_2	[9, 10, 12, 2, 0, 0 ...., 0]	5
12	Image_2	[9, 10, 12, 2, 5, 0, 0 ...., 0]	11
13	Image_2	[9, 10, 12, 2, 5, 11, 0, 0 ...., 0]	6
14	Image_2	[9, 10, 12, 2, 5, 11, 6, 0, 0 ...., 0]	7
15	Image_2	[9, 10, 12, 2, 5, 11, 6, 7, 0, 0 ...., 0]	3

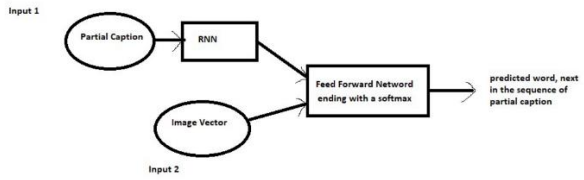
Appending zeros to each sequence to make them all of same length 34

Ptc.10

Tính toán kích cỡ ma trận dữ liệu n*m. Với n là số lượng điểm dữ liệu , m là chiều dài mỗi điểm dữ liệu. Cụ thể: m = chiều dài vector(2048) + chiều dài của caption (X); m = 2048 + x. Với x = max_length caption * chiều dài vector của mỗi từ sau khi nhúng bằng kĩ thuật word – embedding.

c.Word embeddings
Kỹ thuật embedding được sử dụng để map mỗi từ tới một vector có độ dài 200 sử dụng pre-trained GLOVE model.

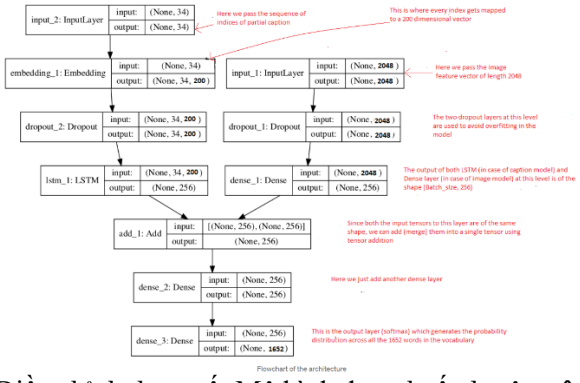
d.Kiến trúc mô hình
Input gồm image vector và từng phần của catpion. Sử dụng Functional API để tiến hành khởi tạo mô hình merge, như hình Ptc.11:



High level architecture

Ptc.11

Kiến trúc thực nghiệm đốc rút từ kết quả huấn luyện mô hình:



Flowchart of the architecture

Điều chỉnh tham số: Mô hình được huấn luyện trên 30epochs, learning_rate = 0.001, 3 pictures mỗi batch. Tuy nhiên sau 20 epochs thì cắt giảm learning_rate = 0.0001 và train model 6 pictures mỗi batch. Thực nghiệm càng về cuối epochs thì mô hình có xu hướng hội tụ, việc cắt giảm learning_rate nhằm đưa mô hình từng bước nhỏ tiến về điểm tối ưu toàn cục.
Kết quả của việc thực thi trước và sau điều chỉnh learning_rate = 0.001:

```
Epoch 1/1
2000/2000 [==============================] - 134s 67ms/step - loss: 4.1161
Epoch 1/1
2000/2000 [==============================] - 131s 65ms/step - loss: 3.4084
Epoch 1/1
2000/2000 [==============================] - 121s 60ms/step - loss: 3.1949
Epoch 1/1
2000/2000 [==============================] - 122s 61ms/step - loss: 3.0587
Epoch 1/1
2000/2000 [==============================] - 118s 59ms/step - loss: 2.9650
Epoch 1/1
2000/2000 [==============================] - 119s 59ms/step - loss: 2.8900
Epoch 1/1
2000/2000 [==============================] - 118s 59ms/step - loss: 2.8345
Epoch 1/1
2000/2000 [==============================] - 118s 59ms/step - loss: 2.7878
Epoch 1/1
2000/2000 [==============================] - 118s 59ms/step - loss: 2.7484
Epoch 1/1
2000/2000 [==============================] - 118s 59ms/step - loss: 2.7090
```

```
Epoch 1/1
1000/1000 [==============================] - 68s 68ms/step - loss: 2.4589
Epoch 1/1
1000/1000 [==============================] - 68s 68ms/step - loss: 2.4495
Epoch 1/1
1000/1000 [==============================] - 68s 68ms/step - loss: 2.4380
Epoch 1/1
1000/1000 [==============================] - 68s 68ms/step - loss: 2.4300
Epoch 1/1
1000/1000 [==============================] - 68s 68ms/step - loss: 2.4200
Epoch 1/1
1000/1000 [==============================] - 68s 68ms/step - loss: 2.4085
Epoch 1/1
1000/1000 [==============================] - 67s 67ms/step - loss: 2.3994
Epoch 1/1
1000/1000 [==============================] - 67s 67ms/step - loss: 2.3957
Epoch 1/1
1000/1000 [==============================] - 67s 67ms/step - loss: 2.3877
Epoch 1/1
1000/1000 [==============================] - 67s 67ms/step - loss: 2.3766
```

e.Truy vấn dữ liệu
Một hình ảnh mới được sử dụng, như hình Ptc12:

Test image

Caption: the black cat is walking on the grass
Tiến trình thực hiện truy vấn hình ảnh mới để trả về một caption. Vocab = {startseq, the, black, cat, is, walking, on, the, grass, endseq}. Model cho ra vector có độ dài 12 cho mỗi từ những bộ dữ liệu ban đầu chúng ta có mỗi từ đại diện bởi một vector có độ dài là 1652

f.Đánh giá mô hình

Để kiểm nghiệm kết quả mô hình thì một bộ test hình ảnh tương thích với tập train được đưa vào để kiểm nghiệm. Tương thích ở đây là hình ảnh về chủ thể của tập train và tập test phải tương đồng , giả sử : hình ảnh huấn luyện thuộc về con người thì tập test cũng thuộc về con người. Phương thức được sử dụng để đánh giá là BLUE score. **Cơ chế của BLUE score** hoạt động trên tiêu chí đếm số n-gram matching của candidate và reference (hoặc match trên bất kỳ reference nào nếu như có nhiều reference), kết quả sẽ là số match chia cho số từ của candidate. Các match này không phụ thuộc vào vị trí ,do vậy BLUE score không sử dụng word-order. Càng match – nhiều tức là càng tốt. Phạm vi đánh giá nằm trong khoảng [0.0, 0.1].
Kết quả của mô hình được đánh giá thoong qua sử dụng sequence_socre kết quả trả về 0.6 cho mỗi caption so khớp từ predict cho tới bộ dữ liệu có sẵn trên tập test.

g.Cải tiến mô hình
Tôi đề xuất phương pháp cải tiến độ chính xác mô hình thông qua việc tăng kích thước bộ dữ liệu cụ thể là ảnh và corpus tương ứng với mỗi hình ảnh.