CS431: CÁC KĨ THUẬT HỌC SÂU VÀ ỨNG DỤNG

IMAGE CAPTIONING & APPLICATIONS

GVHD: Nguyễn Vĩnh Tiệp

Nguyễn Hoàng Long
20520239@gm.uit.edu.vn

00

NỘI DUNG BÀI THUYẾT TRÌNH

01

GIỚI THIỆU ĐỀ TÀI

01

GIỚI THIỆU

Image Captioning là gì?

Image Captioning (Chú thích hình ảnh) là quá trình tạo mô tả văn bản của một hình ảnh. Sử dụng cả Xử lý ngôn ngữ tự nhiên và Thị giác máy tính để tạo phụ đề cho ảnh.

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

Tập dữ liệu sẽ ở dạng **[image → captions]**. Bộ dữ liệu bao gồm các hình ảnh đầu vào và chú thích đầu ra tương ứng của chúng.

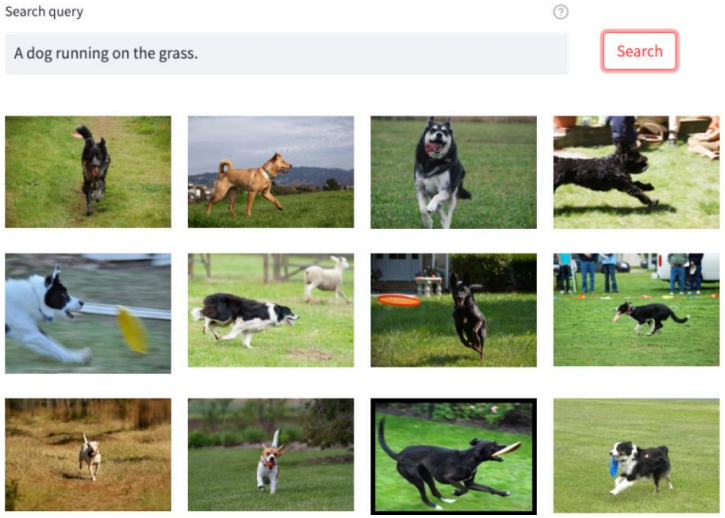
01

ỨNG DỤNG

Các ứng dụng của Image Captioning?

Ta có thể thấy ngay 2 ứng dụng của **Image captioning**:

- Để giúp những người già mắt kém hoặc người mù có thể biết được cảnh vật xung quanh hay hỗ trợ việc di chuyển. Quy trình sẽ là: **Image -> text -> voice.**
- Giúp google search có thể **tìm kiếm** được hình ảnh dựa vào caption.



02

BỘ DỮ LIỆU

02

FLICKR8K

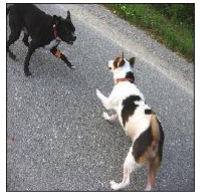
Dataset sử dụng?

Dữ liệu dùng trong bài toán là ***Flickr8k Dataset.*** Dữ liệu gồm **8000** ảnh, **6000** ảnh cho traning set, **1000** cho dev set (validation set) và 1000 ảnh cho test set.

Ví dụ các ảnh sau có **5 captions** ➔

Như vậy traning set sẽ có 6000 * 5 = **40000** datasets

a little girl in a pink dress going into a wooden cabin .
a little girl climbing the stairs to her playhouse .
a little girl climbing into a wooden playhouse .
a girl going into a wooden building .
a child in a pink dress is climbing up a set of stairs in an entry way .



two dogs on pavement moving toward each other .
two dogs of different breeds looking at each other on the road .
a black dog and a white dog with brown spots are staring at each other in the street .
a black dog and a tri-colored dog playing with each other on the road .
a black dog and a spotted dog are fighting

young girl with pigtails painting outside in the grass .
there is a girl with pigtails sitting in front of a rainbow painting .
a small girl in the grass plays with fingerpaints in front of a white canvas with a rainbow on it .
a little girl is sitting in front of a large painted rainbow .
a little girl covered in paint sits in front of a painted rainbow with her hands in a bowl .

man laying on bench holding leash of dog sitting on ground
a shirtless man lies on a park bench with his dog .
a man sleeping on a bench outside with a white and black dog sitting next to him .
a man lays on the bench to which a white dog is also tied .
a man lays on a bench while his dog sits by him .

the man with pierced ears is wearing glasses and an orange hat .
a man with glasses is wearing a beer can crocheted hat .
a man with gauges and glasses is wearing a blitz hat .
a man wears an orange hat and glasses .
a man in an orange hat starring at something .

THUẬT TOÁN

03 TỔNG QUAN VỀ BÀI TOÁN

Phân tích bài toán

Input là ảnh và **output** là text, ví dụ ***"man in black shirt is playing guitar".***

Nhìn chung các mô hình machine learning hay deep learning đều **không xử lý trực tiếp** với text như 'man', 'in', 'black',... mà thường phải **quy đổi (encode)** về dạng số. Từng từ sẽ được **encode** sang dạng **vector** với độ dài cố định, gọi là **word embedding.**

Input là ảnh thường được extract feature qua pre-trained model với dataset lớn như ImageNet và model phổ biến như VGG16, ResNet, quá trình được gọi là embedding và output là 1 vector.
Nhìn thấy **output** là text nghĩ ngay đến **RNN** và sử dụng mô hình **LSTM**.

03 TỔNG QUAN VỀ BÀI TOÁN

Phân tích bài toán

➔ **Ý tưởng** sẽ là dùng embedding của ảnh và dùng các từ phía trước để dự đoán từ tiếp theo trong caption.

Ví dụ:

Embedding vector + A -> girl

Embedding vector + A girl -> going

Embedding vector + A girl going -> into

Embedding vector + A girl going into -> a.

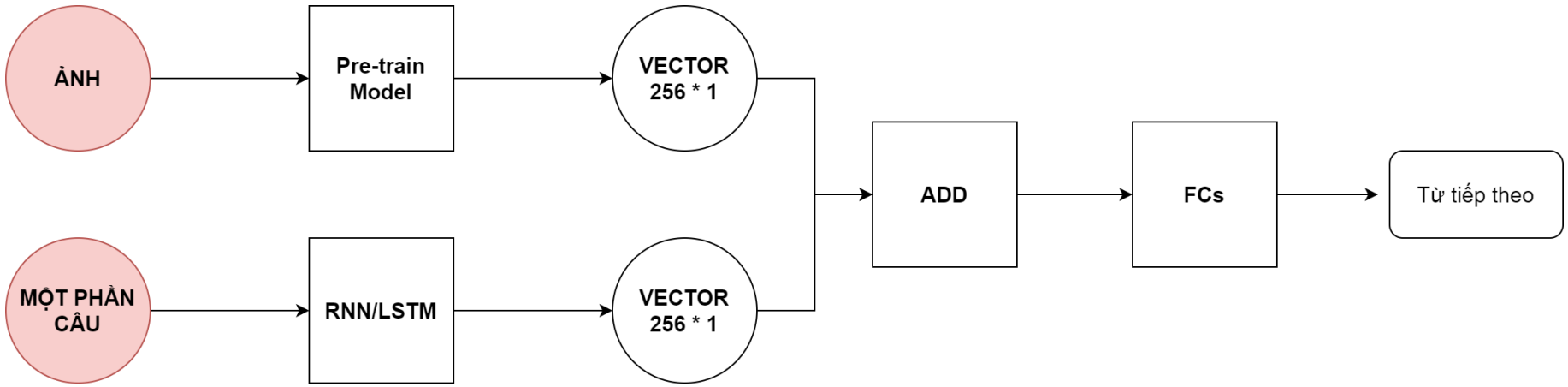
Embedding vector + A girl going into a -> wooden building .

Embedding vector + A girl going into a wooden -> building .

03

TỔNG QUAN VỀ BÀI TOÁN

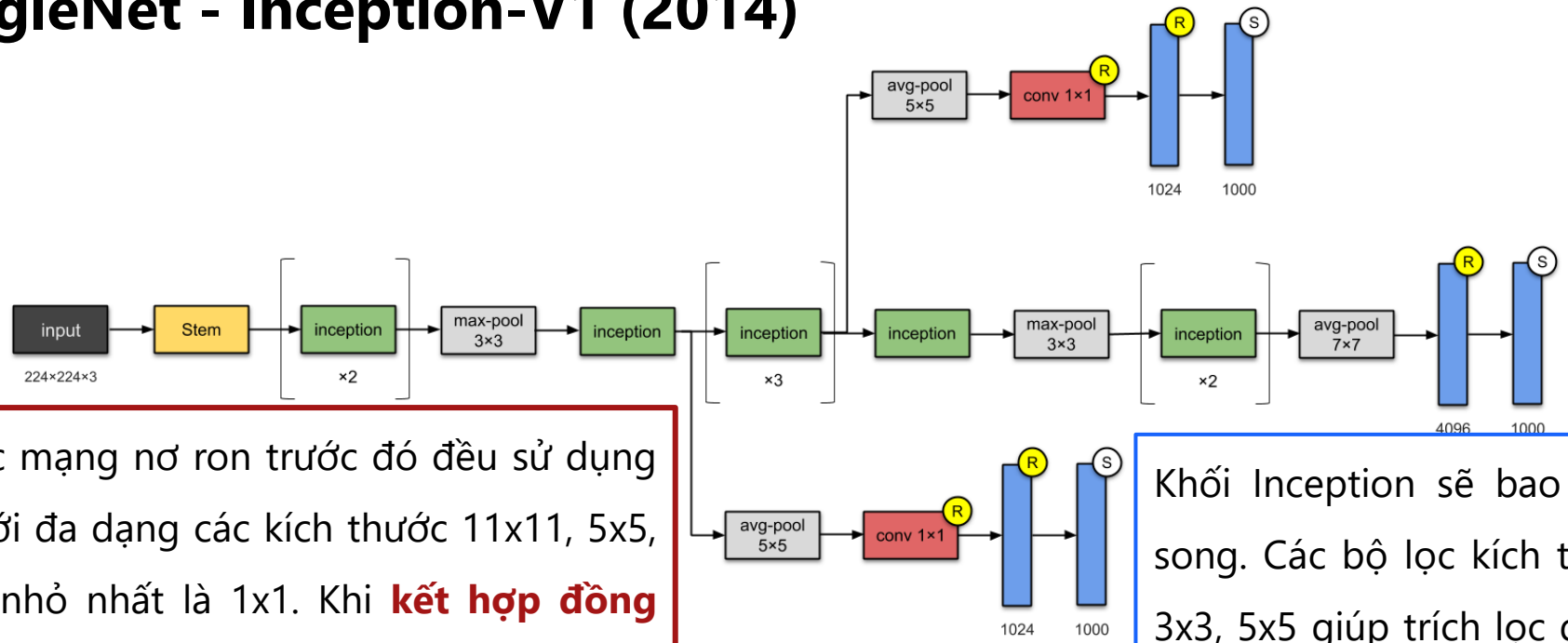
Phân tích bài toán



Để dự đoán từ tiếp theo ta sẽ xây dựng từ điển các từ xuất hiện trong training set (ví dụ 2000 từ) và bài toán trở thành bài toán phân loại từ, xem từ tiếp theo là từ nào, khá giống như bài phân loại ảnh.

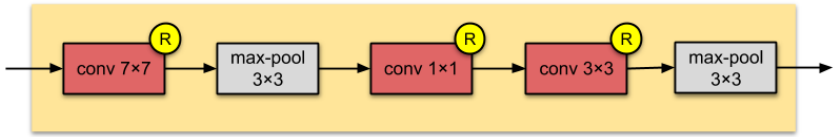
03

GoogleNet - Inception-V1 (2014)

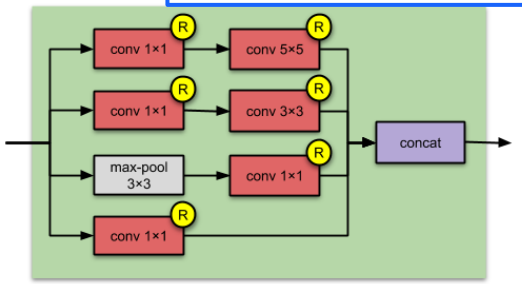


Các kiến trúc mạng nơ ron trước đó đều sử dụng các bộ lọc với đa dạng các kích thước 11x11, 5x5, 3x3 cho tới nhỏ nhất là 1x1. Khi **kết hợp đồng thời** các bộ lọc này vào cùng một block có thể mang lại hiệu quả đó chính là kiến trúc khối Inception.

Khối Inception sẽ bao gồm 4 nhánh song song. Các bộ lọc kích thước lần lượt là 1x1, 3x3, 5x5 giúp trích lọc đặc trưng trên những vùng nhận thức có **kích thước khác nhau.**



Stem

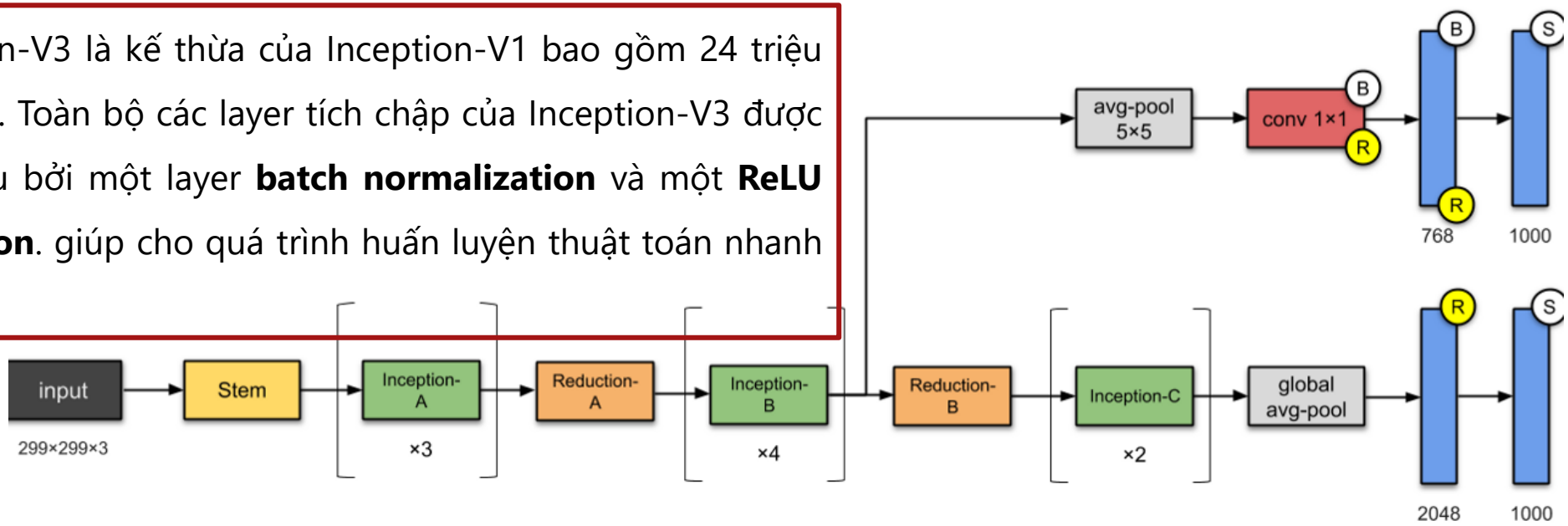


Inception module

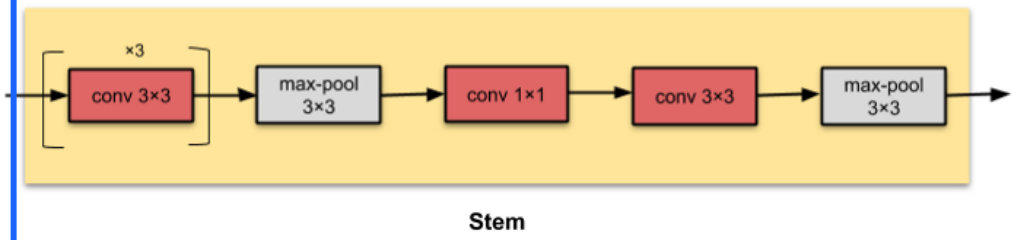
03

GoogleNet - Inception-V3 (2015)

Inception-V3 là kế thừa của Inception-V1 bao gồm 24 triệu tham số. Toàn bộ các layer tích chập của Inception-V3 được theo sau bởi một layer **batch normalization** và một **ReLU activation**. giúp cho quá trình huấn luyện thuật toán nhanh hơn.

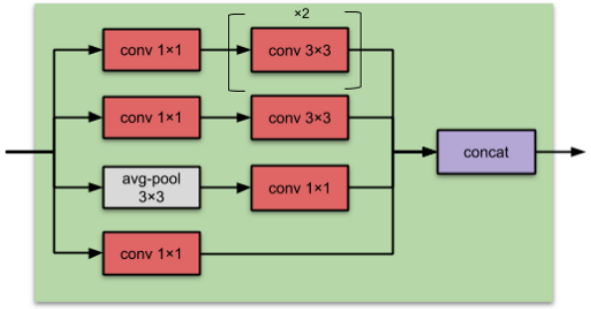


Inception-V3 giải quyết được vấn đề thắt cổ chai (representational bottlenecks. Đồng thời Inception-V3 có một cách tính toán hiệu quả hơn

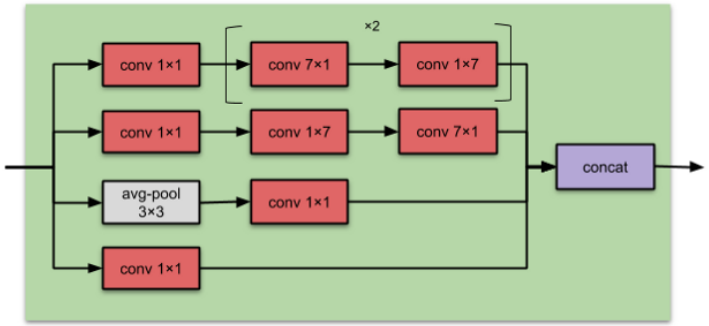


03

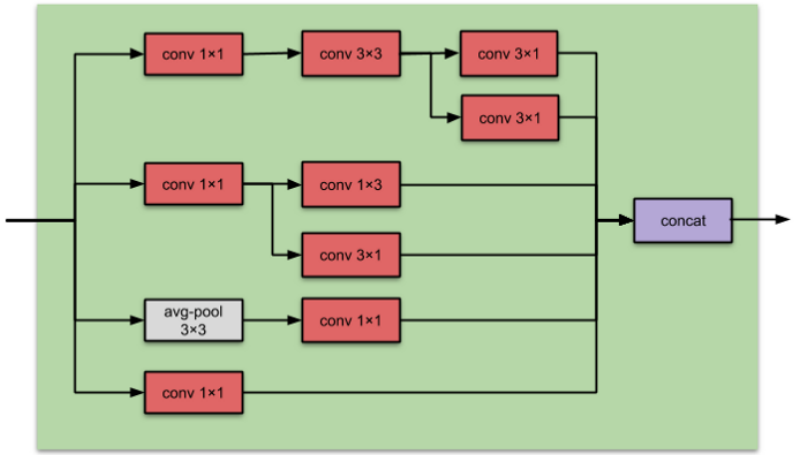
GoogleNet - Inception-V3 (2015)



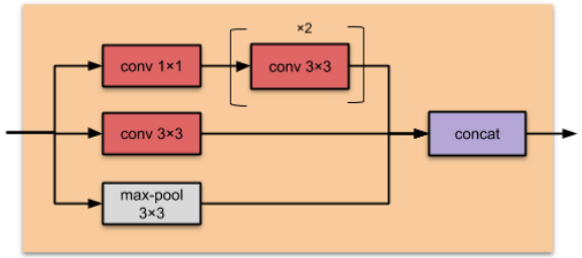
Inception-A



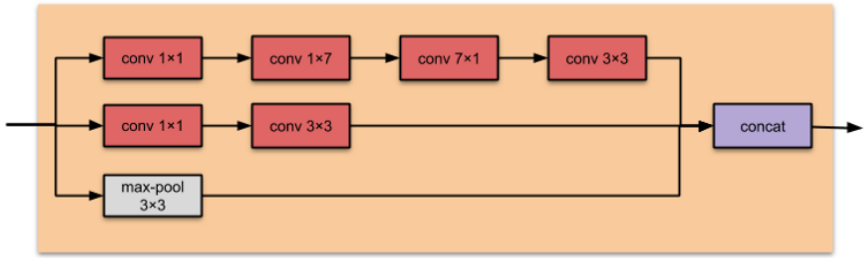
Inception-B



Inception-C



Reduction-A



Reduction-B

Hiện tại Inception module bao gồm **3 version** là Inception-A, Inception-B và Inception-C. Ngoài ra ở Inception-V3 còn sử dụng **2 kiến trúc giảm chiều dữ liệu** là Reduction-A và Reduction-B.

03

CÁC BƯỚC CHI TIẾT

Ta sẽ sử dụng pre-trained model Inception-V3 với dataset Imagenet. Do là pre-trained model yêu cầu ảnh đầu vào là 229*229 nên ra sẽ resize ảnh về kích thước này. Sau khi qua pre-trained model ta sẽ lấy được embedding vector của ảnh, kích thước 256*1

Text preprocessing

Ta xử lý text qua một số bước cơ bản.

- Chuyển chữ hoa thành chữ thường, “Hello” -> “hello”
- Bỏ các kí tự đặc biệt như “%”, “$”, “#”
- Loại bỏ các chữ có số như hey199

Sau đó ta sẽ thêm 2 từ “startseq” và “endseq” để biểu thị sự bắt đầu và kết thúc của caption.

Ví dụ: “startseq a girl going into a wooden building endseq“. “endseq” dùng khi test ảnh thì biết kết thúc của caption.

03

CÁC BƯỚC CHI TIẾT

Text preprocessing

Không quan tâm những từ mà chỉ xuất hiện 1 vài lần, vì nó giống như là nhiễu vậy và không tốt cho việc học và dự đoán từ của model, nên ta chỉ giữ lại những từ mà xuất hiện trên 10 lần trong số tất cả các caption.

Word embedding

Pre-trained GLOVE Model được sử dụng cho quá trình word embedding.

Từng dòng trong file sẽ lưu text và encoded vector khích thước 200*1

03

MODEL

```
Model: "model_1"
_________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #     Connected to
=================================================================================================
 input_3 (InputLayer)        [(None, 33)]              0           []

 input_2 (InputLayer)        [(None, 2048)]            0           []

 embedding (Embedding)       (None, 33, 200)           326400      ['input_3[0][0]']

 dropout (Dropout)           (None, 2048)              0           ['input_2[0][0]']

 dropout_1 (Dropout)         (None, 33, 200)           0           ['embedding[0][0]']

 dense (Dense)               (None, 256)               524544      ['dropout[0][0]']

 lstm (LSTM)                 (None, 256)               467968      ['dropout_1[0][0]']

 add (Add)                   (None, 256)               0           ['dense[0][0]',
                                                                    'lstm[0][0]']

 dense_1 (Dense)             (None, 256)               65792       ['add[0][0]']

 dense_2 (Dense)             (None, 1632)              419424      ['dense_1[0][0]']

...
Total params: 1,804,128
Trainable params: 1,804,128
Non-trainable params: 0
```

04

ĐÁNH GIÁ

04

BLEU SCORE

Phương pháp tính

Cách tính của BLEU cũng khá đơn giản. Phương pháp đếm số matching n-grams 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 BLEU không sử dụng word order. Càng match nhiều tức là càng tốt.

Khi đếm matching n-grams cần chú ý cả số lần xuất hiện của từ trong reference, một từ trong reference khi được match rồi thì không nên match nữa.

04

BLEU SCORE

Ví dụ

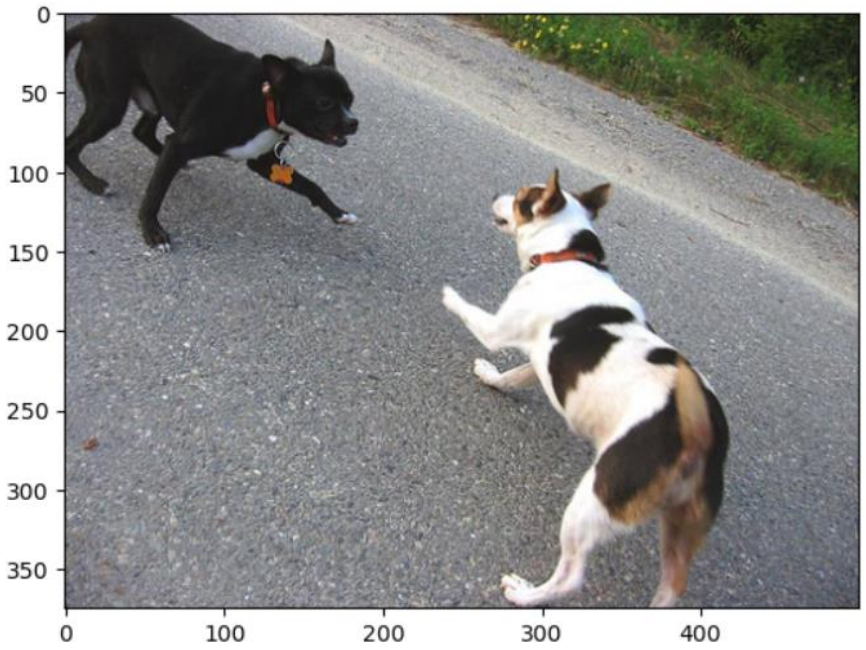
```
from nltk.translate.bleu_score import corpus_bleu

reference = [['this', 'is', 'the', 'test'], ['that', 'do', 'not', 'train']]
candidate = [['this', 'is', 'the', 'train']]
print('Individual 1-gram: %f' % corpus_bleu(reference, candidate, weights=(1, 0, 0, 0)))
print('Individual 2-gram: %f' % corpus_bleu(reference, candidate, weights=(0, 1, 0, 0)))
print('Individual 3-gram: %f' % corpus_bleu(reference, candidate, weights=(0, 0, 1, 0)))
print('Individual 4-gram: %f' % corpus_bleu(reference, candidate, weights=(0, 0, 0, 1)))
>>> Individual 1-gram: 1.000000
>>> Individual 2-gram: 0.666667
>>> Individual 3-gram: 0.500000
>>> Individual 4-gram: 0.000000
```

weights là 1 tuple thể hiện trọng số tương ứng với từng i-gram score ở vị trí i-th.

04 KẾT QUẢ

man in red helmet is riding bike through the woods

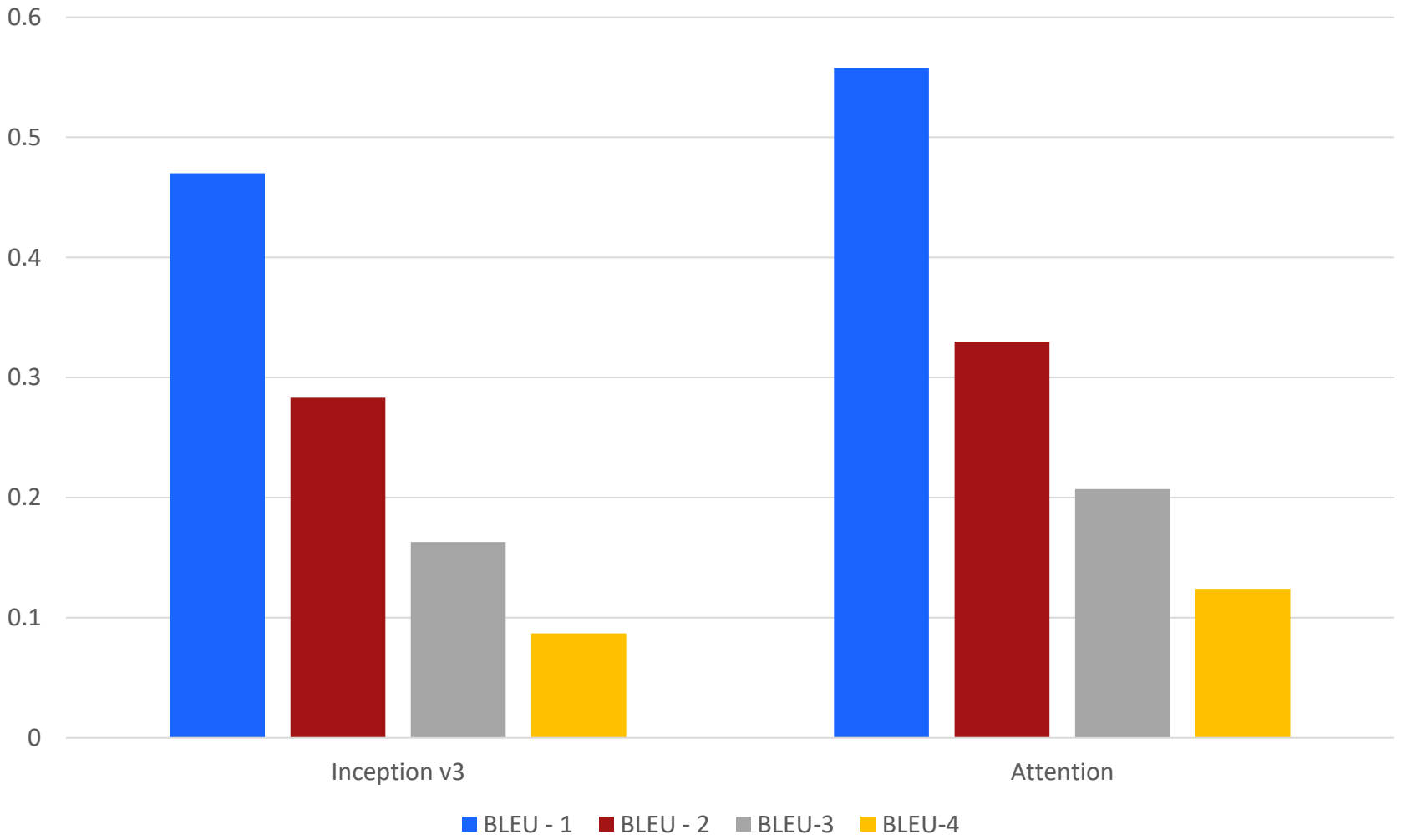
```
---------------------Actual---------------------
startseq black dog and spotted dog are fighting endseq
startseq black dog and tri-colored dog playing with each other on the road endseq
startseq black dog and white dog with brown spots are staring at each other in the street endseq
startseq two dogs of different breeds looking at each other on the road endseq
startseq two dogs on pavement moving toward each other endseq
--------------------Predicted--------------------
startseq two dogs are playing with ball in the snow endseq
```



04

ĐÁNH GIÁ

KẾT QUẢ CỦA CÁC THUẬT TOÁN



05

HƯỚNG PHÁT TRIỂN

HƯỚNG PHÁT TRIỂN

05

- Xây dựng thêm Web và App để hỗ trợ cho người dùng bị khiếm thị
- Huấn luyện trên nhiều bộ dữ liệu hơn
- Fine-tuning các mô hình end-to-end như LAVIS để xây dựng ứng dụng

DEMO