How to make automated Image Captioning ?

Team
Thomas Chaton
Tran Thanh Toan

Introduction

Why deep learning to solve this ?

Lots of Deep Neural Networks have been recently introduced and shown impressive ac
curacy in vision and language modeling tasks, outperforming previous existing models.

Deep Neural Networks in the COCO (Common Objects in COntext) challenge have
come with great results on the task to describe the visual content of images and videos with words and sentences.

Our goal in this project ?

The goal of the project will be to use available resources on Internet as models, data, thesis papers, code in order to run a Deep Learning trained model on a standard and specialized data set of images.

Introduction

Two models caught our attention

NeuralTalk2/NeuralTalk

It is an open source software for Automatic Image Captioning with Text-Conditional Attention on GitHub. The project was based on ”Watch What You Just Said: Image Captioning with Text Conditional Attention ”

DenseCap

It is an open source software for Automatic Dense Captioning on GitHub. The project was based on ”DenseCap: “Fully Convolutional Localization Networks for Dense Captioning ”.

Introduction

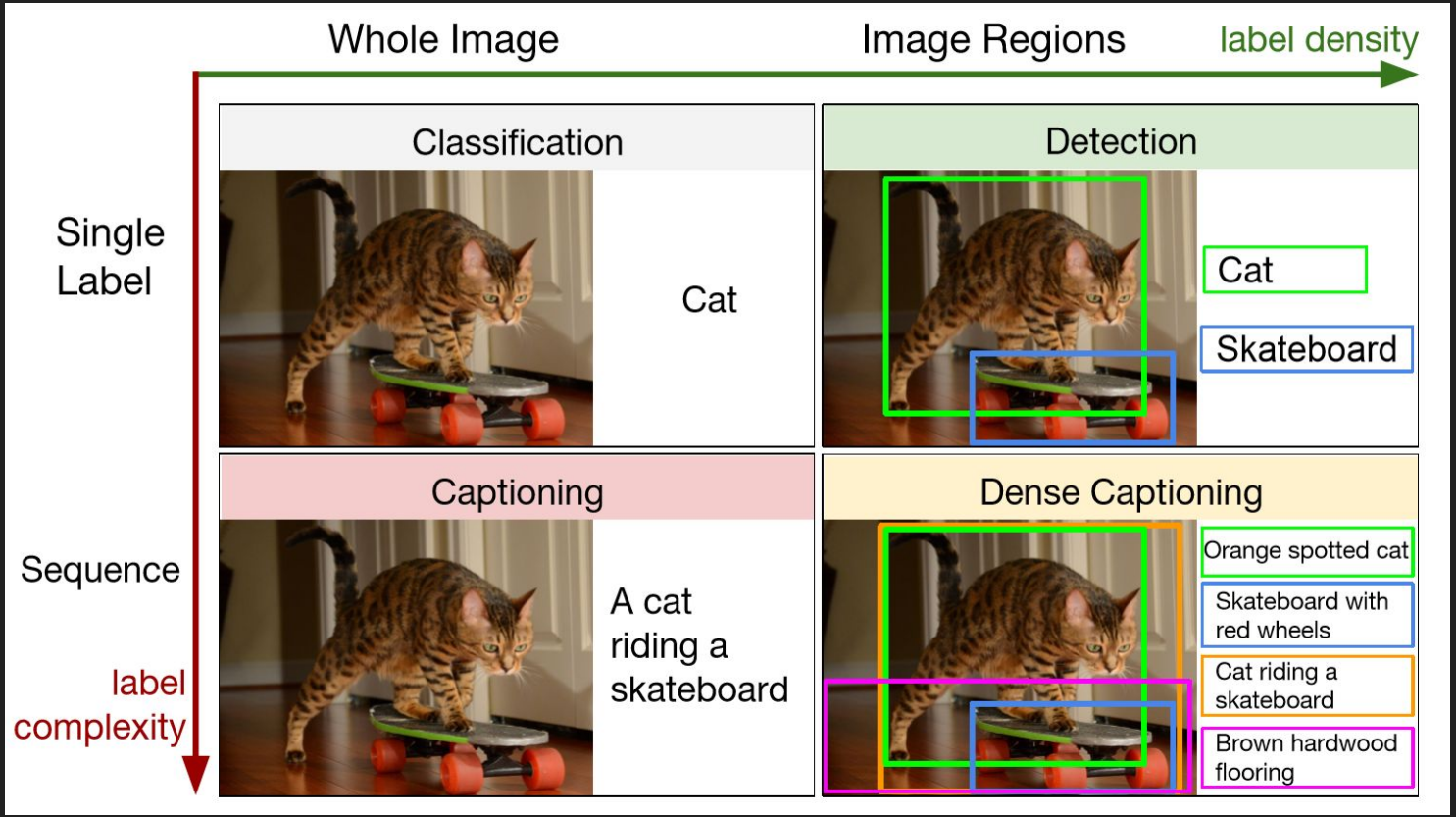


Table of Contents

Part I : Semantical Analysis

Word Embeddings

Lemmatizing with word2vec (library)

1)

- A reasonable dataset might have around 20000 distinct words
- The average sentence is 40 words long

→ 20000 x 40 matrix for a sentence ~ 3.2 megabytes in 32bits

The method word2vec allow to reduce it to 100*40 matrix ~ 16 kilobytes in 32bits so 200x less

2)

Embedding words in a vector space so that words that are semantically similar are near each other allow to use → allow word addition and subtraction

Example : king - man + woman = queen

Metrics : Evaluation of text generation

Main idea of metrics : "the closer a machine translation is to a professional human translation, the better it is"

BLEU (**bilingual evaluation understudy**):

METEOR (**Metric for Evaluation of Translation with Explicit ORdering**)

Correlation with human judgement at the same corpus level

METEOR with human judgement at the corpus level	0.964
BLEU with human judgement at the corpus level	0.817
MAXIMUM correlation with human judgement	0.403

Part II : Neuraltalk2

Presentation/Archicture

Neuraltalk2

- ***Neuraltalk2*** is an open source image captioning system by Karpathy, follows 2 paper

  +“Deep Visual-Semantic Alignments for Generating Image Descriptions”, Andrej Karpathy and Li Fei Fei
  +“Show and Tell: A Neural Image Caption Generator”, Vinyals et al.

- It is an enhanced version of Neuraltalk (written in Python). Neuraltalk2 is

  + batched,
  + written on on Lua in Torch using GPU
  + support CNN tuning

  => approximately 100 times faster.

Some examples of captioning

"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."

"boy is doing backflip on wakeboard."

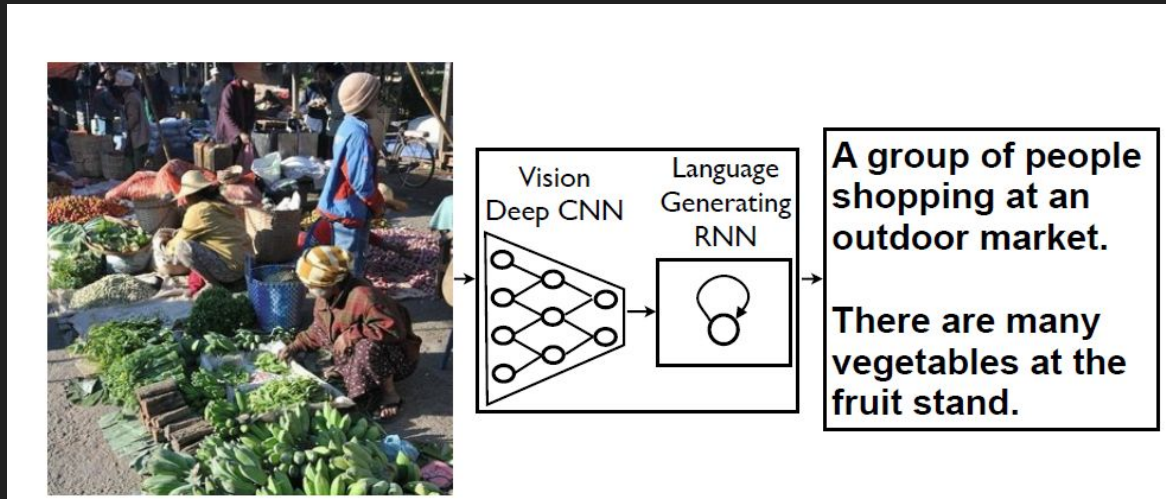
"a young boy is holding a baseball bat."

"a cat is sitting on a couch with a remote control."

"a woman holding a teddy bear in front of a mirror."

"a horse is standing in the middle of a road."

Basic Structure



Basic Structure

Convolutional neural network (CNN, or ConvNet)

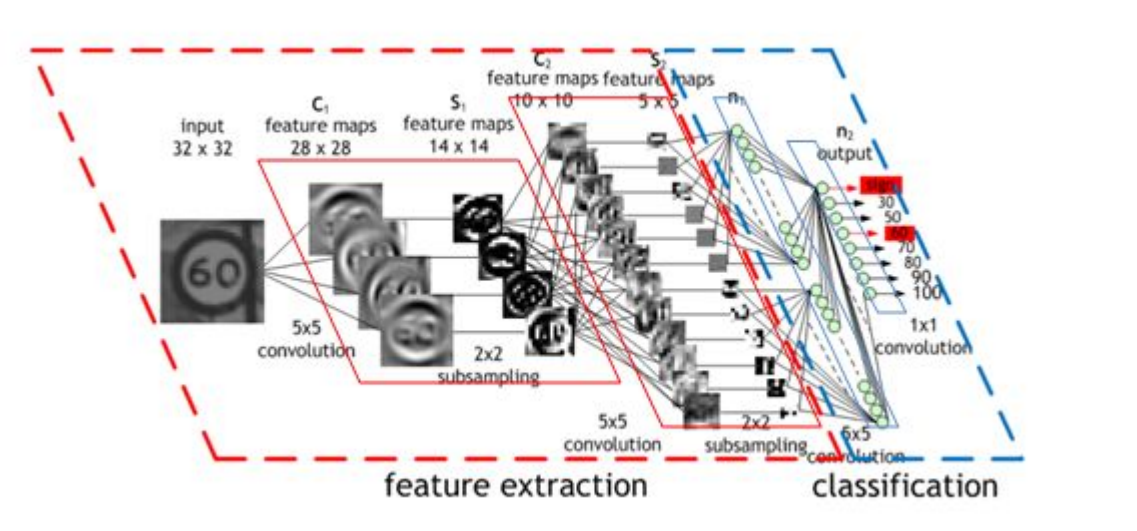
What is it ?

It is **feedforward artificial neural network** in which the connectivity pattern between its neurons is **inspired by the organization of the animal visual cortex**, whose individual neurons are arranged in such a way that they **respond to overlapping regions tiling the visual field**.

Performances

Convolutional networks were inspired by biological processes
and are variations of multilayer perceptrons **designed to use minimal amounts of preprocessing.**

Basic Structure



Basic Structure

Recurrent neural network (RNN)

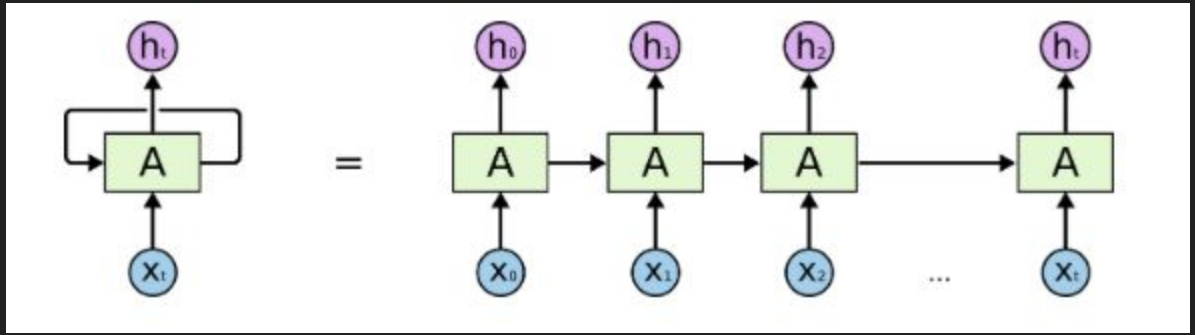
What is it ?

It is a class of artificial neural network where connections between units form a **directed cycle**. This creates an **internal state of the network** which allows it to exhibit **dynamic temporal behavior**.

What are they for ?

Unlike feedforward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs. This makes them applicable to tasks such as **unsegmented connected hand-writing recognition or speech recognition.**

Basic Structure



NeuralTalk2 Architecture

Global Architecture

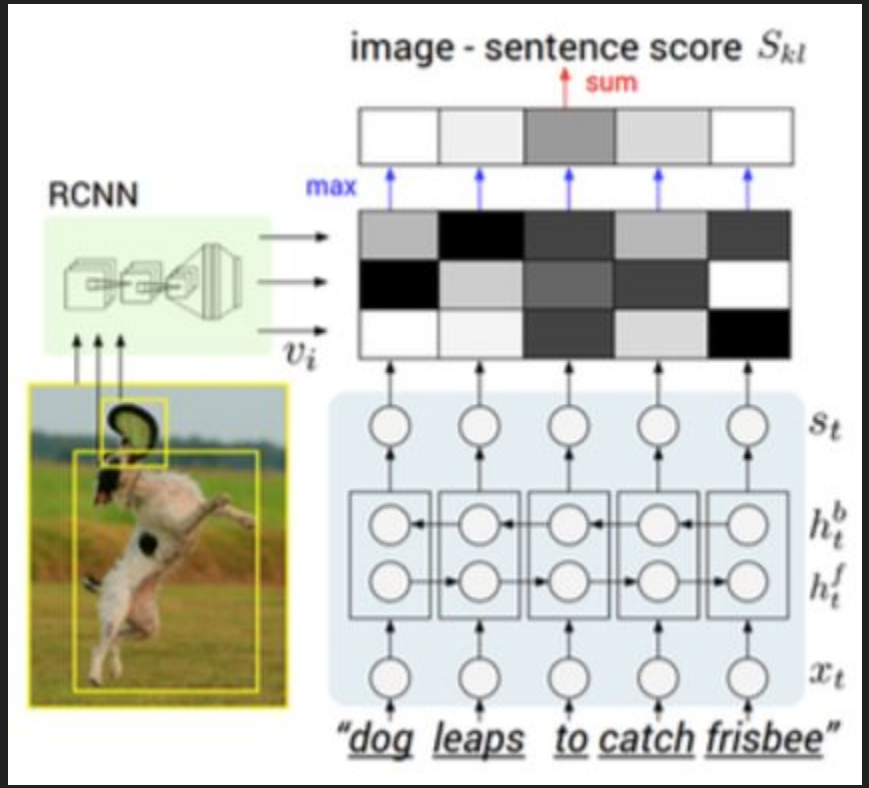
Vision Related

Region-based Convolution Neural Network (RCNN). It is a state-of-the-art visual object detection system that combines bottom-up region proposals with rich features extraction computed by a convolutional network.

Natural Language Processing

Bidirectional Recurrent Neural Networks is to split the neurones of a regular RNN into two directions, one for **positive time direction** (forward state), and another for **negative time direction (backward state)** . By using two time directions, input information from the past and future of the current time frame can be used.

NeuralTalk2 Architecture



Part II : Neuraltalk2

Setting up

Setting up enviroment

Python 3.4 We choose to use python for working since it is a language with supported libraries for machine learning. We use it for further investigation since systems may not in the same language

Jupyter Notebook it is a web application, which allows us to create documents with live code and explanatory text. It is an executable documents.

TensorFlow is an open source software library (based on Python) for numerical computation using data flow graphs.

Installation of NeuralTalk2 and its dependencies

Here is the list of libraries required:

- Torch
- LuaRocks *
- Lua JSON
- loadcaffe
- torch hdf5
- hdf5

*LuaRocks: install cuda for GPU can be quite tricky

How it works

Evaluation:

+ Download the pretrained checkpoints: for both GPU and CPU.
+ In the end, we can only use CPU for evaluation, since we do not have enough GPU power.

$ th eval.lua -model [/path/to/model] -image folder [/path/to/image/directory] -num images [number of images]

Visualization the results:

$ cd vis

$ python -m SimpleHTTPServer

Part II : Neuraltalk2

Bootstrapping

Bootstrapping

Idea:

+ Uses predictions of NeuralTalk2 and train it .

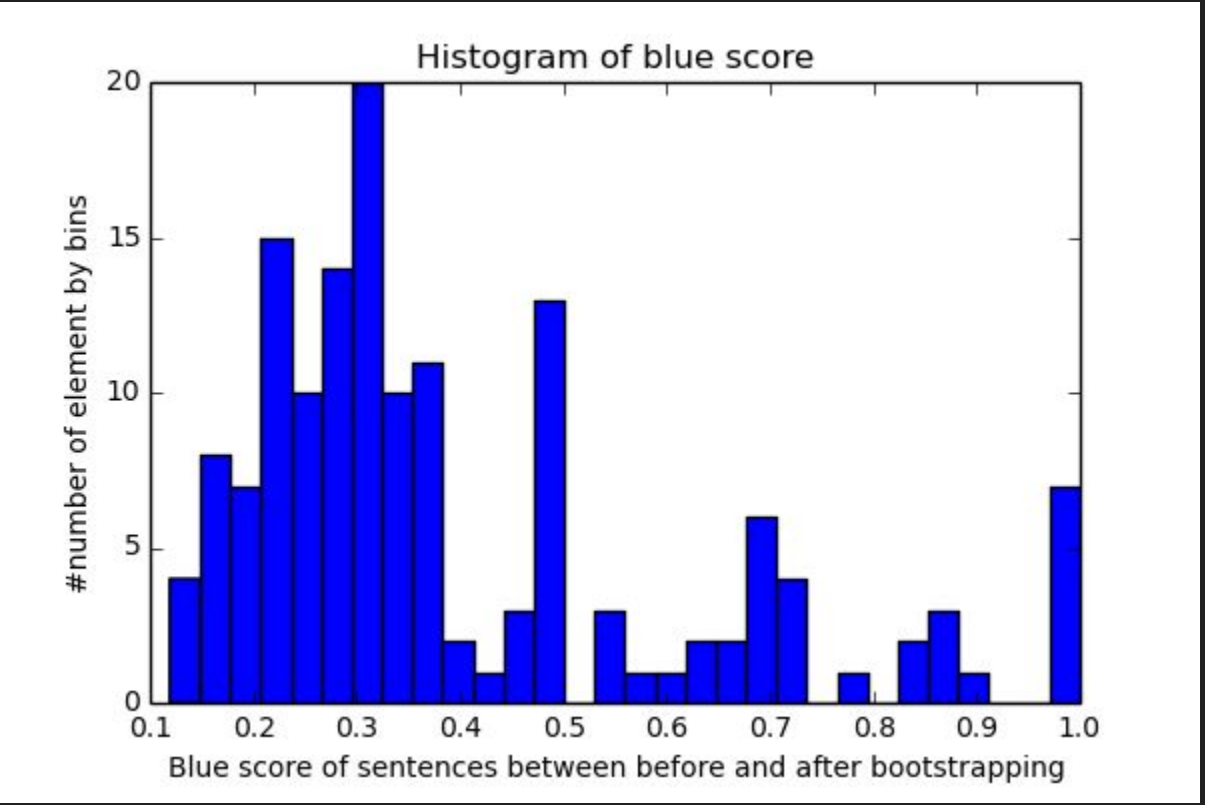
Difficulties:

- Unlabeled dataset, it was hard to tell what was good or not without looking into it.

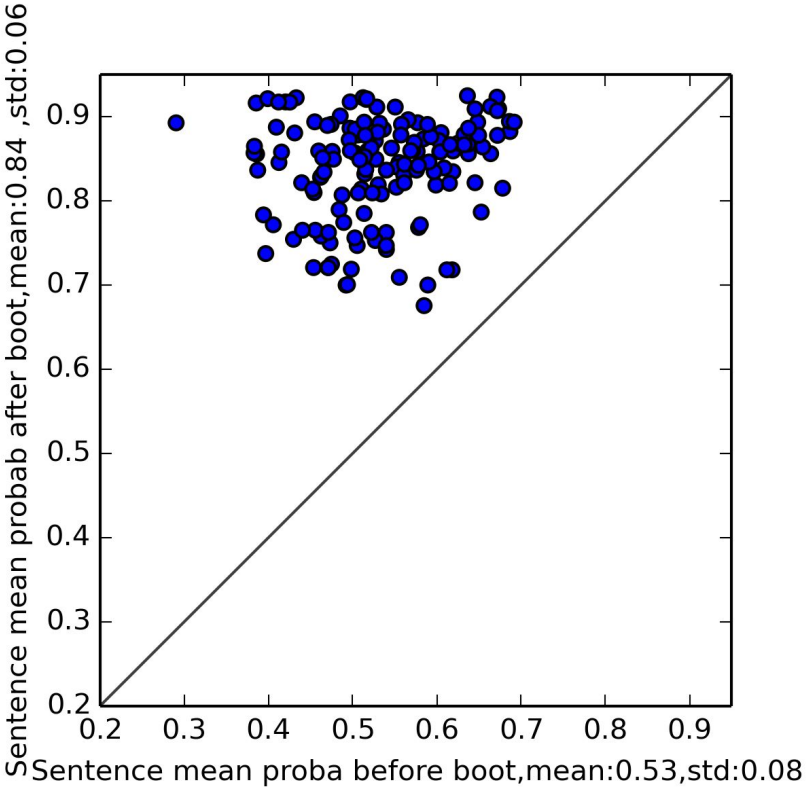
Objectives:

+ Clean the dataset from duplicate or wrong images.
+ Find a way to evaluate NeuralTalk2 performances before/after bootstrapping.
+ Find a way to evaluate its confidence.

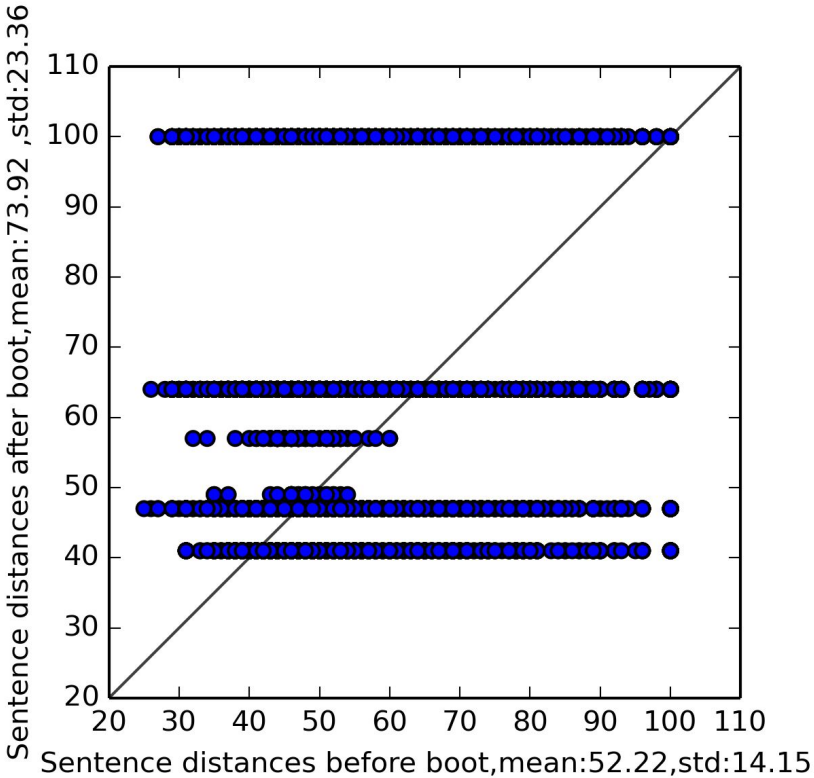
Bootstrapping on Cannes images



Bootstrapping on Cannes images



Bootstrapping on Cannes images



Part III : DenseCap

Presentation/Architecture

DenseCap

Densecap

It is another open source image detecting and captioning objects in images system by Karpathy,Johnson, Justin, Andrej Karpathy, and Li Fei-Fei based on the paper “DenseCap: Fully Convolution Localization Networks for Dense Captioning”

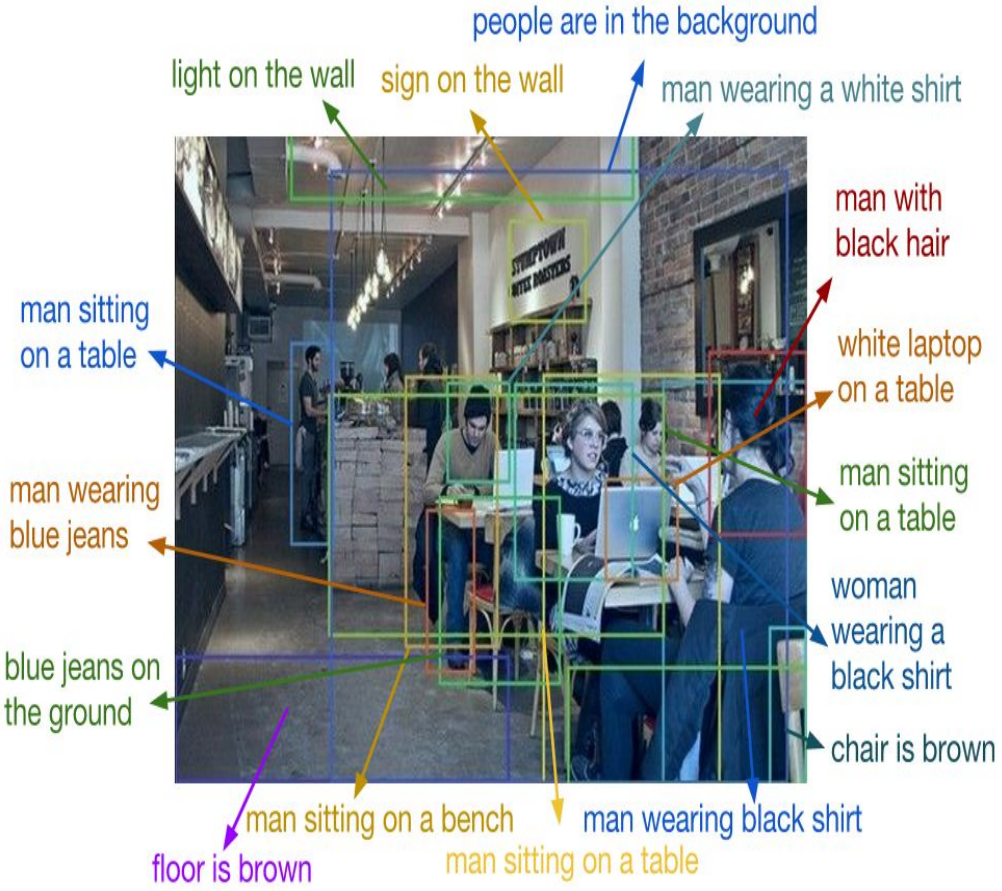
What is it for ?

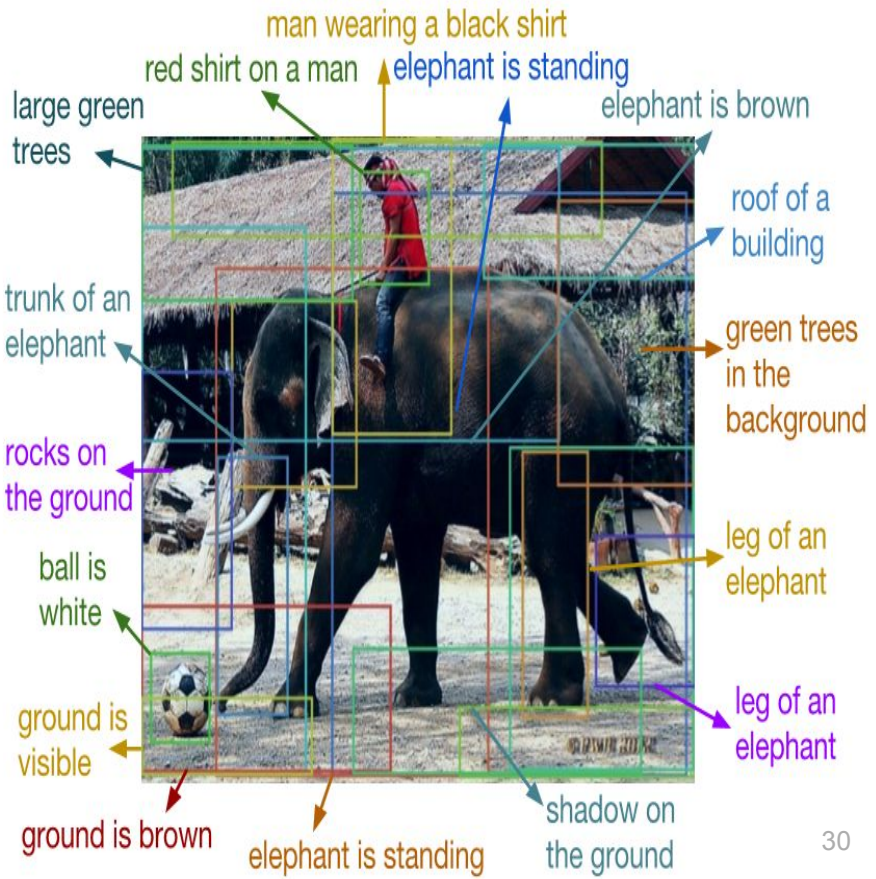
Detect objects in images and describes them in natural language.

Our assumption

=> Gives more information about the images to contribute to NT2.

Some examples of dense captioning





DenseCap Architecture

Three Core Components

Vision Related

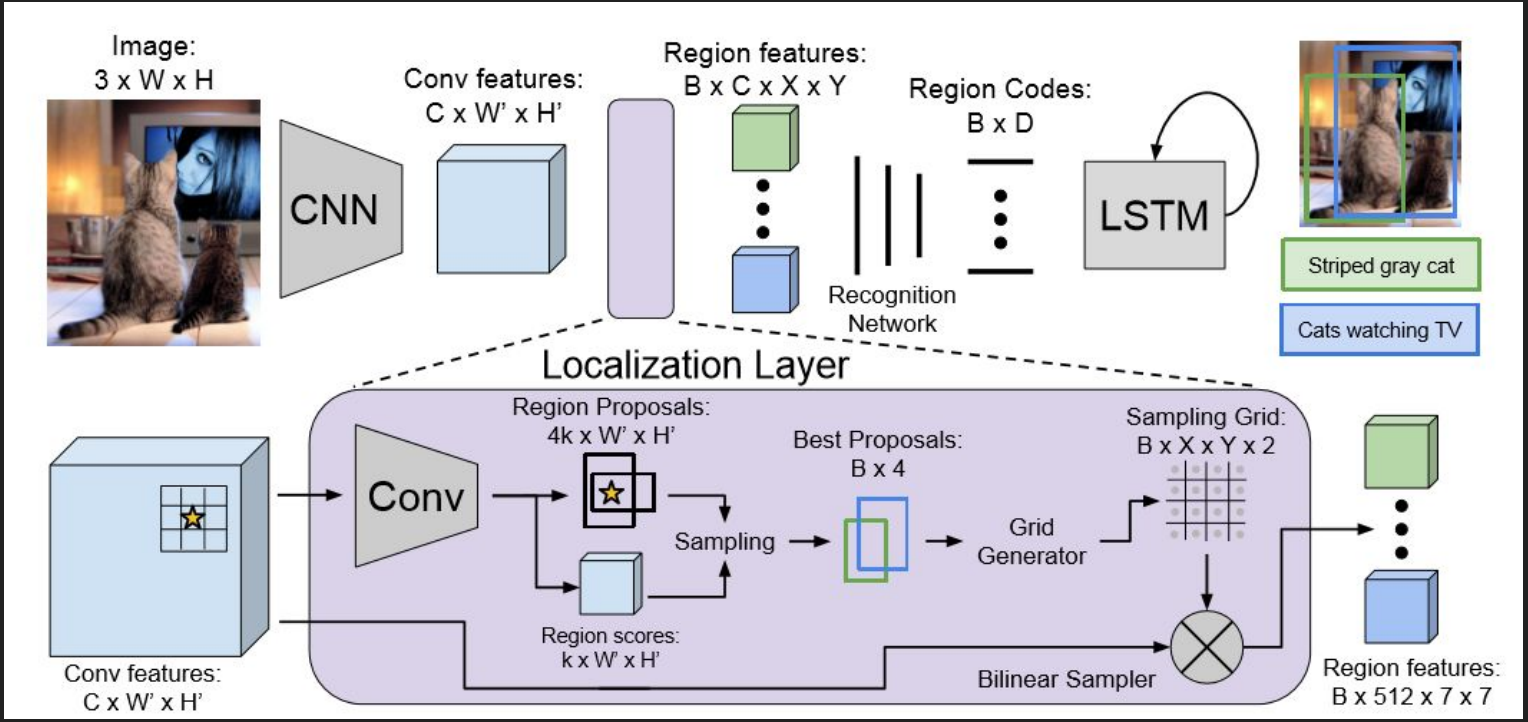
- A Convolutional Neural Network is used to extract core features of the image.They used for this task the VGG-16 architecture for its state-of-the-art performance.

- A Fully Convolutional Localization Layer receives an input tensor of activations, identifies spatial regions of interest and smoothly extracts a fixed-sized representation from each region

NLP Related

- A Long Short Term Memory (LSTM) which is a special kind of RNN.

DenseCap Architecture



Basic Structure

Neuraltalk vs. Densecap

- The main difference between DenseCap and Neuraltalk2 is the localization layer, which produces the interesting regions.

- While Neuraltalk2 applies the RNNs for the whole image convolution feature, DenseCap applies RNNs for the best region proposals features

Part III : DenseCap

Setting up

Installation of DenseCap and its dependencies

Mostly the same as the Neuraltalk2

Torch/Torch7

Torch/image

qassemoquab/stnbhwd

Torch/nn

lua-cjson

jcjohnson/torch-rnn

How it works

Evaluation:

+ Download the pretrained checkpoints $ sh scripts/download_pretrained_model.sh
+ Same as Neuraltalk2, we can only use CPU for evaluation, since we do not have enough GPU power.

$ th run_model.lua -input_dir [/path/to/image/directory]

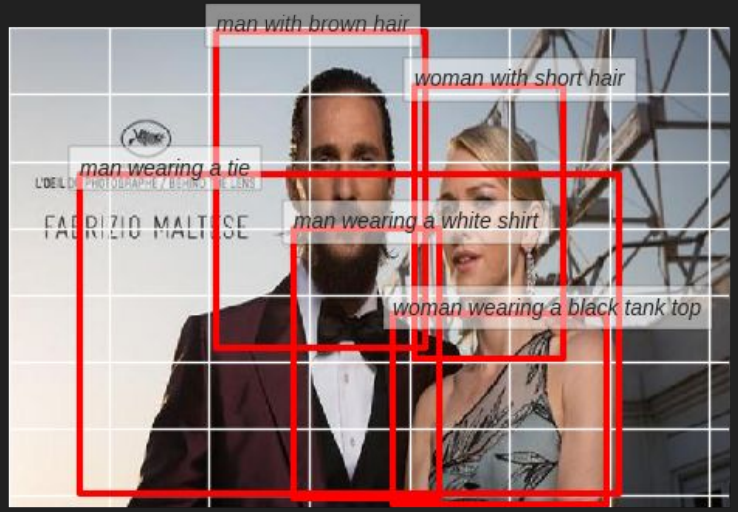
This task is extremely long, it required 1 week to run about 944 images.

Visualization the results:

$ cd vis

$ python -m SimpleHTTPServer 8181

Results and Comments



Much better than Neuraltalk2 in general => using it for NT2

Part III : DenseCap

Use DenseCap regions to train NeuralTalk2

Use DenseCap regions to train NeuralTalk

Ideas: Use DenseCap predictions for Training NeuralTalk2

+ Find best regions in DC captions/regions per images and feed them to NT2
+ Find best regions in DC captions/regions per images and associate them to the images before feeding it to NT2

Difficulties:

- Find a way to find those best regions.

Objectives:

+ Implement an algorithm to find those best regions.
+ When it is done, implement both ideas in order to train NT2.

Find best regions

Ideas: Use both word/sentence similarity , DC score and regions overlapping

1) Associate each words with its scores and see how they correlated with all the others weighted by their relative scores

2) Made a combination of all word pair and for each word of the pair we associated a new score which is the sum of exp[percentage of overlapping regions*α] * similarity between pair of words * DenseCap score of the sentence where the word was found.

3) Same as 2) using sentences instead of words.

Difficulties:

- Make it continuous.
- Extract stop words (don't bring any information)
- Make real best captions to have the higher score.

Algorithm 4.2 Second Approach

Require: *dataset* = The 80 Generated regions along with its score and caption per images

captionDict (defaultdict in python)
for each *data* of an image in *dataset* **do**
  wordDict (defaultdict in python)
  $\alpha = 2$ {α is used to make a distorsion inside the exponential. It helps to seperate the best words from others. We tried several values of α from 1 to 3}
  splitData = Split each sentences of the captions of *data* and associated each one of them to the respective score of the sentence and bounding box
  cleanedData = Get rid of the stop words and the UNK Token in *splitData*
  allCombi = All the pair of combinaison of word-score-boundingBox in *cleanedData*
  for each *pair* in *allCombi* **do**
    $word_1 = pair[0][0]$
    $score_1 = pair[0][1]$
    $boundingBox_1 = pair[0][2]$
    $word_2 = pair[1][0]$
    $score_2 = pair[1][1]$
    $boundingBox_2 = pair[0][2]$

    $wordDict[word_1] += score_1 *$
    similarity($word_1$,$word_2$)*
    exp |$OverLappingRegionsPercentage$(boundingBox$_1$,$boundingBox_2$)*$\alpha$|

    $wordDict[word_2] += score_2 *$
    similarity($word_1$,$word_2$)*
    exp |$OverLappingRegionsPercentage$(boundingBox$_1$,$boundingBox_2$*$\alpha$|)
  end for
end for
for each *data* of an image in *dataset* **do**
  for each *obj* in *data* **do**
    $captions = obj[0]$
    $score = obj[1]$
    $captionDict[captions]$ = matched_Word_Weighted_By_Scores($captions$,$wordDict$)*$score$
  end for
end for

Train NT2 with best regions

What we have done there:

+ We have implemented both proposed solutions . And when we evaluated on the test set . The results were catastrophic.

Ideas to solve the issues:

+ We thought it might come from the fact that NT2 create its vocab from given sentences (~100 words). We modified the code to give him all the words (~10 000 words).

Conclusions :

- After we tried everything to make it train, the results were terrible.
- We concluded it came from either a too little dataset, not enough power to run it or an error from us.

Part III : DenseCap

Use DenseCap to evaluate NeuralTalk2

Use Densecap to evaluate NeuralTalk2

Idea:

- Based on the same assumption that DC provides more accurate captions. We use DC's captions in the same image to evaluate NT2's results.

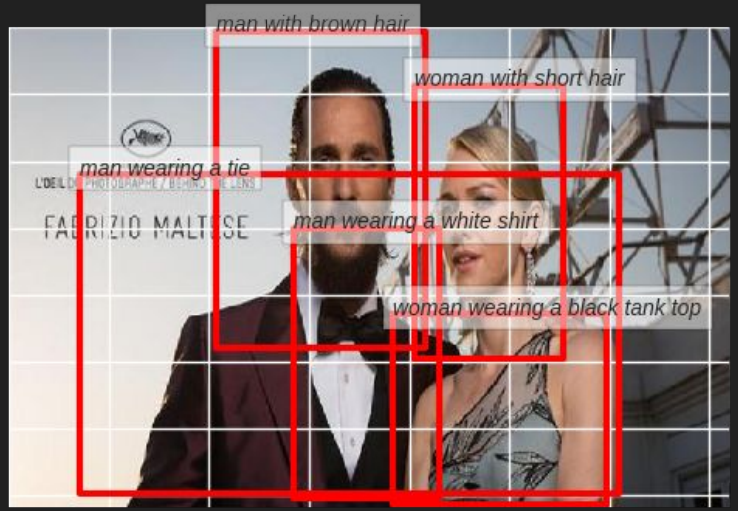
Difficulties:

- Neuraltalk2 copies and renames all the images in the captioning step, while Densecap just keeps the same name. We need to match those.

Objectives:

- Algorithm to use Densecap's caption to evaluate or classify Neuraltalk2's captions into good or bad ones.

Use Densecap to evaluate NeuralTalk2



a man and a woman standing next to each other

a man is standing in front of a microphone

Preprocessing

1. Removing stop words such as ‘a’, ‘an’, ‘the’, etc in the captions of both NT2 and DC. These words appear with high frequency but do not provide much information.

2. Stemming: reducing the inflated or derived words to their word stem. There are too many inflated and derived words, which can be infeasible for calculate the similarities.

   => Each caption becomes a set of keywords.

3. Mapping: map the NT2’s caption with the corresponding DC’s set of captions. The evaluation of NT2 copy the images and rename them so we lost the connection of 2 sets of results.

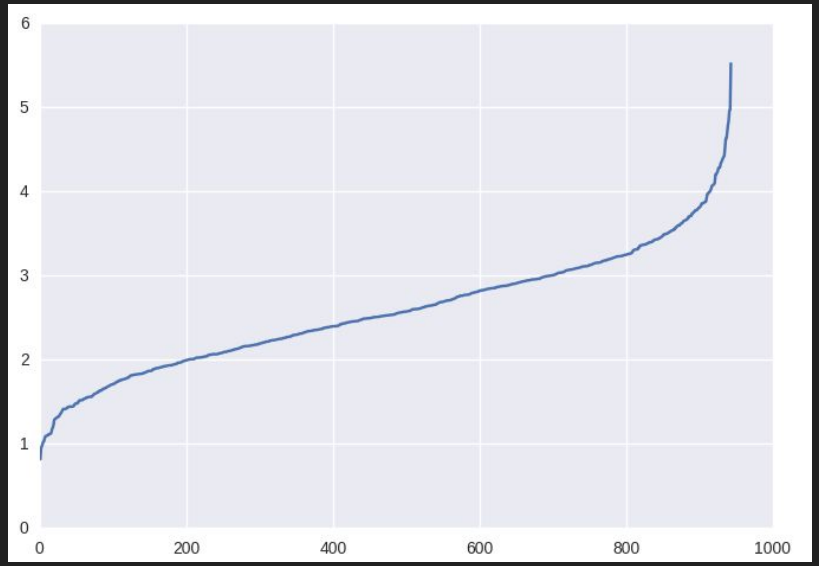
The proposed algorithm

Ideas:

The similarity of a caption in NT2 to the set of caption in DC is the weighted sum of all the similarities of each pair of keywords, one from NT2's caption and the other from DC's top k captions. The weights of the pair is the score of the corresponding Densecap's Caption

Our hypothesis is that higher score, the better the caption of NT2.

Result



Distribution of the scores

3 regions:

- Middle region: linear, score 1.5 < x < 3.5

- Left region: predicted bad captions

- Right region: predicted good captions.

Result (continue)

- No formal measure => evaluate by our perception.

- 3 classes: true, partially true, false

	TRUE	PARTIALLY TRUE	FALSE
PREDICTED GOOD	12	17	0
PREDICTED BAD	4	13	17

- There is no poster or sign images in the predicted good set.

- Most of the predicted bad set are poster or sign images.

Part IV : Applications

Applications

Applications

IDEAS OF APPLICATIONS :

- For blind people: better knowledge of their environment
- Google, Instagram queries: look in the generated caption space instead of the image space.
- Automatic image classification on camera based on complex labelisation.
- Give future IA an understanding of its surrounding.

Conclusions

First, we focus our work on trying to use bootstrap method to finetune NeuralTalk2 on a standard and specialized data set . But as it wasn't labeled, we were able to estimate/measure only two things. How much the evaluations before and after the boot-strapping method were different and how much the network was confident about it. To overcome the issue of unlabeled data set, we have chosen to focus on an other open source software DenseCap which is able to create dense captioning.
We had in mind, that it might be possible to use those informations either to train NeuralTalk2 or to evaluate it.

So secondly, we used DenseCap evaluations and find a way to get the best regions (most significative) per images to get them to be fed by NeuralTalk2. But unfortunately for us, it didn't work as expected. We consider several ways to make it work without success. We arrived to the conclusion that we hadn't enough data and power (all training were done on CPU) to make it work correctly)..

Lastly, we focused our work on a way to evaluate NeuralTalk2 predictions from the tremendous amount of information given by DenseCap and we were able to isolate the best labeled captions by NeuralTalk2

Thanks for your attention

Do you have questions ?