# Image Captioning in Italian

COL 774: Assignment 4

Sem II, 2019-20

**Due Date: Sept 5, 11:50 pm**

**Notes:**

- This assignment has two parts - Non Competitive and Competitive.
- You should submit all your code (including any pre-processing scripts written by you) and any graphs that you might plot.
- Do not submit the datasets. Do not submit any code that we have provided to you for processing.
- Include **a write-up (pdf) file**, which includes a brief description for each question explaining what you did. Include any observations and/or plots required by the question in this single write-up file.
- **You should use Python as your programming language and PyTorch as the deep learning framework.**
- **Your code should have appropriate documentation for readability.**
- You will be graded based on what you have submitted as well as your ability to explain your code.
- This assignment is supposed to be done in **teams of 2**. You should carry out all the implementation by yourself.
- We plan to run Moss on the submissions. We will also **include submissions from the internet** to maintain integrity. Any cheating will result in a zero on the assignment and possibly much stricter penalties (including a **fail grade** and/or a **DISCO**).

1. **Dataset Links**

   (a) **Train Images:** Link

   (b) **Train Captions:** Link

   (c) **Public Test Images:** Link

   (d) **Public Test Captions:** Link

   (e) **Private Test Images:** *Coming Soon*

2. **Non Competitive Part**

   You are given a dataset of images with 5 possible captions for each image. The images are named as `image_id.jpg` where id is the image index. The `captions.tsv` file contain the captions of all the images, with each line (tab separated) format having image id and the 5 corresponding captions. You will use the Encoder-Decoder architecture for modelling this problem. An encoder is used to encode the input into a vector representation and a decoder is applied on this vector representation to generate the sequence auto-regressively (one word at a time). You have to implement the encoder for the images and a decoder for text captions in this part along with beam search to find the most optimal sequence. You can use this as a starting point.

   (a) **Encoder**: Design a CNN based encoder that handles the variable sized images.

(b) **Decoder**: Design a RNN / LSTM based decoder which generates the captions given the encoded image input.

(c) **Training Setup**: Use cross-entropy as the loss function and teacher-forcing for training the decoder. Don't forget to use START and END tokens to allow variable length caption outputs in the decoder.

(d) **Inference at test time**: Instead of using the token with maximum probability at each step (Greedy Decoding) for generating the tokens (words) in the caption, you will use Beam Search for generating your captions. It is a Dynamic Programming method to get better sequences than simple Greedy Decoding. Read Section 4 of this pdf document. We also recommended you to read Sections 1.2, 1.3, and 1.4 for better understanding.

(e) **Evaluation:** You have to generate the top-5 captions for each image using Beam Search as described above. We will use BLEU (BiLingual Evaluation Understudy) scores for evaluating your model performance. You can read more about it in Section 5.3 of this pdf document.

3. **Reference**

You can refer to this paper to get an in-depth understanding of Image Captioning Encoder-Decoder framework. We also provide the boiler plate code with base classes to make it easier to implement the assignment. You can download it from this link.

4. **Competitive Part**

(a) Instructions *coming soon.*

5. **Submission Instructions**

Make .tsv files `[EntryNumber1]_[EntryNumber2]_[TYPE].tsv` similar to `test_captions.tsv` for each of the test sets (TYPE = public/private). Submit these 2 files along with your code and report in a single zip format. Only **one of you** should make the submission.