# Assignment 3

**Deadline: Monday, 3rd June (23:59)**

## Question 1: Sequence Classification (10pts)

### Task 1.1: Document-level Sentiment Classification

Build a Bidirectional Recurrent Neural Network (RNN) model for multi-class sentiment classification. Compare the performance with a Unidirectional RNN model. Your model (each) shall include:

- RNN network that learns sentence representation from input sequences.
- Fully connected network that predicts sentiment label, given the learnt state representation.

Train the model by using data iterator and batch generator. Evaluate the trained model on the provided test set.

### Task 1.2: Aspect-level Sentiment Classification

Build an attention-based aspect-level sentiment classification model with RNN. Your model shall include:

- RNN network that learns sentence representation from input sequences.
- Attention network that assigns attention score over a sequence of RNN hidden states based on aspect terms embedding representation.
- Fully connected network that predicts sentiment label, given the representation weighted by the attention score.

Train the model by using data iterator and batch generator. Evaluate the trained model on the provided test set.

### Task 1.3: Discussion

- What is the motivation of incorporating an "attention mechanism" in a Machine Translation task? What is the main issue that this attention trying to solve? Mention the advantage(s) as compared to the model without attention.
- What is the motivation of adding an "attention" network in aspect-level sentiment classification? What is the main issue that this attention trying to solve? Mention the advantage(s) as compared to the model without attention.

### Data Description

Document-level and aspect-level data sets can be accessed in `/home/datasets/recsys`. The raw data set contains two domains: (1) Restaurant reviews; and (2) Electronics reviews. Choose one domain to train the models in task 1.1. and 1.2.

# Question 2: Image Caption Generation (10pts)

Build an image caption generator model, as described in [1]. The model shall consist of:

1. Image encoder (feature extractor)

   The encoded features from a pretrained model and all required preprocessed data (i.e. vocabulary dictionary, training set) are provided. Use preprocessing code if necessary (preprocessing.zip).

2. Caption generator

   Most of the parts in the Caption generator are uncompleted. Your main task is to finalize the whole encoder-decoder architecture. Be alert to the dimension (output shape) of each tensor and how tensors (Input, Output, intermediate tensors) are connected to each other.

## Data Description

Flickr8$k$ dataset [2] and preprocessed sets can be accessed in `/home/datasets/recsys`. This data set consists of 8.000 images and their corresponding text descriptions (captions). Each image corresponds to five (5) captions. The preprocessed data consists of 6.000 images in training set (total 30.000 pairs of image-caption for training the model), 1.000 validation set, and 1.000 test set.

## Task 2.1: Encoder - Decoder Model

Complete the provided model skeleton (*.ipynb) to build an **encoder-decoder** model for Image Caption Generator. Train the model by using data iterator and batch generator. Stop the training once your validation error loss starts to increase.
*Note*: By using an early stopping method, the training will cost approximately ±2 hours in Zirconium CPU (20 epochs, early stopping with patience=10, Adam optimizer with learning rate *lr=0.001*).

## Task 2.2: Decoder Model

Build **a decoder model** for generating captions in inference stage (i.e. after done training the model) by using two approaches:

1. Greedy search **)* Example is shown in Practical.
2. Beam search **)* Your own implementation.

*Note*: Choose one or two image(s) from the provided test set.

## Task 2.3: Discussion

- Briefly describe one of the preceding works on modeling Image-Caption according to the paper and its limitation. Name the advantage(s) of the current Image-Caption generator as compared to the previous work?
- How does the model extract image features from raw images? (inspect the provided code *preprocessing1.0.py*, *preprocessing2.0.py*)
- How are the train descriptions represented into the model? Why do we need to add "starting" and "ending" token of every caption in a preprocessing stage?

[1] Vinyals, Oriol, et al. "Show and tell: A neural image caption generator." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
[2] provided by University of Illinois at Urbana-Champaign for educational purpose. Do not redistribute.

- What is the motivation of incorporating Beam Search in Sequence-to-Sequence learning? Briefly explain how the method works in an inference stage.

## Question 3: Peer review (0pt)

Finally, each group member must write a single paragraph outlining their opinion on the work distribution within the group. Did every group member contribute equally? Did you split up tasks in a fair manner, or jointly worked through the exercises. Do you think that some members of your group deserve a different grade from others?