# Natural Language Processing Unit 1 Evaluation

Date: 19th Sep 2015

Time: 3:30 pm to 6:00 pm IST

## Problem # 1: Develop a MaxEnt classifier for the Mobile Reviews Corpus for Named Entity tagging

You are provided with a dataset that has the mobile phone reviews as a text data. In this assignment you will first run the NLTK's NE tagger to perform baseline tagging on the given dataset. Then you will build a MaxEnt (also known as Log-Linear) Model classifier that can classify the input in to one of the Named Entity tags, where the training data with tags are obtained using NLTK tagget.

Specifically, you are required to perform the following:

1. Tag the given dataset using NLTK NE tagger.
2. Look at NLTK and identify the list of supported tagset: Y
3. The space of all tags is the output space Y.
4. Build the history h records that constitute the input space
5. Features are at the heart of the MaxEnt classification process. For each of the tags to be supported you are required to identify one or more features. A feature in MaxEnt model is a function $f_k(x, y)$, where x is an element in X and y is an element in Y. In our case, x is a history tuple and y is a tag. You are required to implement this function which should return 0 or 1 as the value of the feature. The total number of features is the dimensionality of the input to MaxEnt model. NOTE: Please start with a feature set not exceeding 10. Once you get the complete classifier working and based on the time it takes for training, you can increase the dimensionality. For a dimensionality of 10, you will need to write 10 such functions. These functions are called indicator functions when they return binary value.
6. Design and implement the class: MyMaxEnt() with the following methods:
    a. __init__ should accept history list and the list of feature functions as input. You can use a method create_dataset() that takes the history and produces the training examples for the MaxEnt. The training examples are feature vectors where each element of this vector is a binary.
    b. Initialize the model (v in our slides) to a vector of zeros. Note that model will have same dimensionality as feature vector. For each feature there is a model parameter. ($f_k(x, y)$, $v_k$)
    c. cost(model): Given the model, compute the cost as per the equations given in the slides.

d. train() should train the MaxEnt model using 80% of the tuples selected as above and transformed to a dataset using the feature functions. Once you have written the cost function you can invoke for example:
   from scipy.optimize import minimize as mymin
   def train(self):
       params = mymin(self.cost, self.model, method = 'L-BFGS-B')
       self.model = ….
e. The method p_y_given_x(h, tag) should take the history tuple and the required tag as the input and return the probability.
f. You may also (optionally) write the function gradient() that maximizes the log-likelihood (or minimizes the negative of this) and set jac = gradient when you invoke the minimization function of scipy
g. The method classify(h) performs the classification by determining the tag that maximizes the probability. That is call p_y_given_x(h, t) for various values of t and select the t that returned the maximum value.

2. Run the classifier classify() method with the test samples and record the result

**Deliverables:**

1. Source code of your program
2. The NLTK tagged dataset
3. A brief write up on the performance of your program in the Facebook – Deadline Sunday 6 pm
4. Any other observations (optional)

Your submissions should be made to nlp2015 mail id ☺

Best of luck!