A: Theory

The algorith used on this assignment was the **Perceptron** algorithm for Hidden Markov Model (as described on Homework 2). The perceptron algorithm is a type of linear classifier that makes predictions based on a function that combines a set of weights with the feature vector. The details of our implementation is given on the section below.

B: Algorithm

The algorithm for the weight vector update is given below:

Algorithm 1 Phrasal Chunking

Input: Tagset file $tagsetfile$
Train data $train\_data$,
Set of tags $tagset$,
Number of epochs of the algorithm $numepochs$
Output: Featured weight vector

initialize $feat\_vector$ with zeroes
while *epoch* is less than *numepoch* **do**
  initialize *mistakes* and *correct* with zero
  for *sentence_data* in *train_data* **do**
    initialize *words*, *postags* and *truetags* as empty lists
    initialize *label_list* as *sentence_data*[0] and $feat\_list$ as *sentence_data*[1]
    for *label* in *label_list* **do**
      split *label* by *spaces* and assign it to a triple $(word, postag, chunktag$
      append *word* to *words* list
      append *postag* to *postags* list
      append *chunktag* to *truetags* list
    initialize *tagset* with content of $tagsetfile$
    initialize $default\_tag$ as *tagset*[0]
    initialize *argmaxtags* as the result of *perc_test* passing $feat\_vec$, *label_list*, $feat\_list$, *tagset* and $default\_tag$ as params.
    initialize $feat\_index$ and i as zero
    for *word* in *words* **do**
      get feats c for word with params $feat\_index$ and $feat\_list$
      initialize arg_max with $argmaxtags[i]$ and tru with $truetags[i]$
      if argamax equal tru **then**
        increment i and go back to next loop iteration
      for f in $feats\_for\_this\_word$ **do**
        initialize *wrongkey* with tuple $(f, argmax)$ and *rightkey* with tuple (f, tru)
        decrement value of $feat\_vec[wrongkey]$
        increment value of $feat\_vec[rightkey]$
      increment i
    set i to zero
    for *word* in *words* **do**
      initialize arg_max with $argmaxtags[i]$ and tru with $truetags[i]$
      if argamax equal tru **then**
        increment *correct*
        increment i and go back to next loop iteration
      else
        increment *mistakes*
      initialize *argmaxprev* and *truprev* with *"B:"*
      if i equal zero **then**
        concatenate *argmaxprev* and *truprev* with *"B -1"*
      else
        concatenate *argmaxprev* with $argmaxtags[i-1]$ and *truprev* with $truetags[i-1]$
      initialize *wrongkey* with tuple $(f, argmax)$ and *rightkey* with tuple (f, tru)
      decrement value of $feat\_vec[wrongkey]$
      increment value of $feat\_vec[rightkey]$
      increment i
  increment *epoch*
return $feat\_vec$