A: Theory

The algorithm used on this assignment was the **averaged perceptron** learning algorithm (as described on *pg. 36, Sarkar 2011.*). The algorithm is a type of linear classifier that makes predictions based on a predictor function that combines a set of weights with the feature vector and outputs the averaged weight parameter 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 $tagset$ and # of epochs of the
    algorithm $numepochs$
Output: Featured weight vector
initialize $feat\_vector$ and $cumulative\_feat\_vec$ with zeroes
initialize $epoch$ and $count$ with zero
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, $postag$ to $postags$ list and $chunktag$ to $truetags$ list
        initialize $tagset$ with $tagsetfile$ and $default\_tag$ 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 $feat\_vec[wrongkey]$ and 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 $correct$ and increment $i$. Then, 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 $feat\_vec[wrongkey]$ and increment value of $feat\_vec[rightkey]$
            increment i
        for $key$ in $feat\_vec.keys$ do
            set $cumulative\_feat\_vec[key]$ to the value of $cumulative\_feat\_vec[key] + feat\_vec[key]$
        increment count
    increment $epoch$
for $key$ in $cumulative\_feat\_vec$ do
    set $cumulative\_feat\_vec[key] = cumulative\_feat\_vec[key]/count$
return $cumulative\_feat\_vec$
```