***QUESTION 1 PART 1***

First I converted all single inverted commas to double inverted commas. Then I converted text"text to text###text. Then I looked for words within double quotes that are not dialogues i.e. they don’t contain punctuation marks and converted them to single quotes. Then I converted text###text back to text’text.

The required files are:

test.txt : input

assn1\_1a.py : code

assn1\_1a.txt : result

***QUESTION 1 PART 2***

Marked ambiguous sentence terminators like Mr., G. K. C., Dr. etc with #. Then applied tags to the remaining sentence terminators including .,!,? and \n\n. Then I unmarked the previously marked #s.

The required files are:

test.txt : input

assn1\_1b.py : code

assn1\_1b.txt : result

***QUESTION 2***

Trained a neural network from scikit using an input vector constructed from the context window around a punctuation mark and output being 1 or 0 if the sentence is tagged or not. The input vector is a 14(7+7) sized vector where each half represented the context window before and after the punctuation mark. The value of an element of vector is 0 or 1 depending on whether it contains a particular element.

0 = a-z

1 = A-Z

2 = ' '

3 = '\n'

4 = '\''

5 = ','

6 = ';'

The learnt NN is then used to predict whether a punctuation mark is a sentence terminator or not and the results were compared with the text obtained after tagging. The file ‘fullTest.txt’ was divided into two parts for training and testing purposes. The resulting accuracy is about 98%.

The required files are:

train.txt : input for training

testq2.txt : input for testing

testres.txt : correct output for testing input

assn1\_2.py : code