Question 1: Feature Engineering (30 points total)

1.1 Feature design explanation (10 points):

Gazetteer Feature:

a. I used a collection of lexicons, to take advantage of the lexicons given, I made a match of each token against the data inside each lexicon data. Any match was made as a feature of the token.
b. To implement, a dictionary was created for all the lexicon data with key being each word in the lexicon line and value being the name of the lexicon in the preprocessing function
c. Later, the same dictionary was used in the token2feature function where each input token was matched in the dictionary keys and if matched then the name of the lexicon was added as the feature. Token2feature function
d. There could be a scenario that, a token is matched from multiple lexicon, then all such lexicons were added as features in the preprocessing phase.
e. All the words added from the lexicons to the lexicon dictionary were made lower case and removed off punctuation.

Orthographic Feature (Word Shape):

a. Orthographic features related to the shape of the word were added as features in the feature list for each token.
b. For each token input, the word shape was calculated by taking the first two and last two characters of the word and replacing them with ‘X’ for capital letter and ‘x’ for small letter and ‘d’ for a digit.
c. For the tokens with length greater than 4, the remaining middle string excluding the first two and last two characters was encoded by doing the same encoding but reducing it to a set of replaced characters.
d. The obtained shape of the word was appended to the feature list in the token2features function.
   e.g. if the word is “Rishabh9” then the corresponding encoding would be: “Xxxxd”

First Letter Capitalized:

a. For each token in the token2feature function, a feature was appended by checking if the first character of the token/word is capitalized. If the characters is capital then a feature was appended as “Capital_first”.

b. This was done since most of the Named Entities will have first letter capitalized in their names. This could act as discerning feature to identify the Named Entities and would increase the accuracy of the model.
c. Usually, the named entities have first letter capitalized therefore, this can be a useful feature.

Part of Speech Tagger:

a. For each sentence in the token2feature function, the POS tag was found for all the tokens in the sentence.
b. Later, the POS tag for the token was appended as the feature in the feature list of the token.

   e.g. My name is Rishabh

   The sentence was passed in the POS tag from NLTK library, the corresponding POS tag for each word was reflected by the pos_tag function, later for each word its tag was appended as the feature for the word.
c. This feature is useful and most of the named entities will have ‘NN’ as postag and it will help as a feature in determining named entities.

Vowel Count:

For each for word, the number of vowel were calculated along with the length of the sentence. The division of number of vowel by the length of the sentence was added as the feature to the feature list for each word in the sentence.