1. FEATURE ENGINEERING

1.1 FEATURE DESIGN EXPLANATION

The features which contributed to improving the F1 score are explained below:

CHARACTER BASED FEATURES

1. Initial Capital
   Capitalization is an important orthographic feature in named entity recognition. However, this does not apply entirely in case of tweets as appropriate capitalization is not paid attention to when composing tweets. However, a few capitalization features are implemented in an attempt to recognize named entities.
   The first letter of a word is determined if it is in uppercase. All proper nouns begin with an uppercase letter. Hence, if the word begins with a capital letter, it is most likely to be a named entity. Hence, this binary feature helps to determine if the word is a proper noun.
   For example, consider the tweet: '@123anonymous #DT Donald Trump is the president of United States'. Here, there are 4 words in the tweet that start with an uppercase letter, and they are, 'Donald', 'Trump', 'United', 'States'. All these four words are named entities. Hence, it is a fairly reasonable to assert that words that begin with an uppercase letters are most likely to be named entities.

2. Uppercase
   The word is tested if all its letters are in uppercase. Capitalization plays a very important role in determining if a word is a named entity. However, the capitalization in Twitter is less reliable than in text, as most people do not pay attention to capitalization when tweeting. However, if a word is in full uppercase it is likely to be a named entity.
   For example, consider the tweet: '@123azx CALIFORNIA is the BEST!!!'. Here, there are 2 words in the tweet that are fully uppercase, and they are, 'CALIFORNIA' and 'BEST'. Here, 'CALIFORNIA' is a named entity of type 'location', though 'BEST' is not a named entity. However, in several cases the subject of the tweet, which is usually a named entity is in full uppercase. Hence, it is reasonable to create a feature based on this.

3. Mixed case
   The word is tested if it made up of a combination of uppercase and lowercase letters. Generally, named entities that occur in the hashtag of the tweet are multiple proper nouns concatenated with the first letters of each word being uppercase.
   For example, consider the tweet '#BarackObama is the best'. Here, the hashtag of the tweet is actually a named entity of type 'person' and it has two nouns that are strung together and both of the nouns begin with a capital letter.

4. Numeric
   The word is tested it is numeric or contains digits. Many Twitter words are numeric or contain digits. Such words generally not a named entities of type 'product'.
   For example, consider the tweets: '@jack234 I love my new iPhone 6s', 'I wish I owned an AK 47!!!'. Here, '6s' refers to a product and so does '47'.
   Thus, this binary feature is set if a word in entirely numeric or contains some digits in it.

5. Username
   Twitter usernames always begin with '@'. If the word begins with '@', it represents a Twitter username and hence this feature discriminates clearly them.

6. Hashtags
   All Twitter words starting with ‘#’ are hashtags. However, a word that starts with a ‘#’, can still refer to a named entity (after stripping out the ‘#’). Hence, the word is checked if it starts with a ‘#’. If it does, the remainder of the word is checked for its presence in any of the imported lexicons. If it does not occur, then a binary feature indicating that the word is a hashtag is set. For example, ‘#BritneySpears’ refers to a named entity of type ‘person’, ‘#iPhoneX’ refers to a ‘product’ named entity, ‘#USC’ refers to a ‘facility’ named entity, and so on.

7. Word without prefix
   The word after stripping out the first two characters is added as a feature. The form of the word without the prefix is added as a feature.

8. Word without suffix
   The word after stripping out the last two characters is added as a feature. The form of the word without the suffix is added as a feature.

9. Word
   The token (Unicode) itself is added as a feature.

10. Lowercase form of the word
    The lowercase form of the word is added as the feature.

LEXICON BASED FEATURES

1. POS Tag
   The NLTK POS tagger was used to determine the best POS tag sequence for the tokens in the sentence. And, only the POS tag of that particular word is picked up and added as a feature (however, the POS tag sequence of the entire sentence is determined for better context). Most tokens with the Noun Phrase tag and other Noun tag variations are most likely to be named entities.

2. Stopwords
   The list of English stopwords is imported from the lexicon given in the starter code of this homework. Every word is checked against the list of stopwords. If there is a match, a binary feature indicating the same is set. Stopwords are not named entities, hence it is useful to discriminate them.

3. Lexicon Match
   A set of lexicon lists that were given as a part of this homework were used in classifying named entities into the 10 categories, namely, person, facility, product, geo-location, other, movies, tv shows, company, music artists, and sports team.
   Hence, each word is tested if it occurs in one or more of these lexicons. If it does, then the appropriate feature is set.
   For instance, the word ‘Logan’ occurs in the ‘person’ lexicon and in the ‘location’, thus the features ‘IS_PERSON’ and ‘IS_LOCATION’ are set, and so on.

CONTEXT BASED FEATURES

1. First Word
   This feature checks if the current word is the first word of the sentence. In tweets, it can be observed that the first word of a tweet sometimes is a named entity.

For example, consider the tweets: ‘#LosAngeles is the best city in California @123any’, ‘Julia Roberts is a great actress #JuliaRobets’. Here, ‘LosAngeles’, and ‘Julia’ are named entities of type ‘place’ and ‘person’ respectively.

2. Last Word
   This feature checks if the current word is the last word of the sentence. In tweets, it can be observed that the last word of a tweet sometimes is a named entity.
   For example, consider the tweets: ‘@123lynn #LosAngeles is the best city in California’, ‘Julia Roberts is a great actress #JuliaRobets’. Here, the last words ‘California’ and ‘JuliaRoberts’ are named entities of type ‘place’ and ‘person’ respectively.

3. Local Context: next and previous word features
   For the immediate previous and next neighbors of the current token, all the above described features are also added to the feature set. Doing so will provide additional context to the current word.
   For example, consider the tweet ‘Beiber Justin rocks!! #JB’. Here, the word ‘Justin’ can be a ‘place’ or a ‘person’ named entity. However, the previous word ‘Bieber’ is definitely a named entity of type ‘person’. Hence, it makes most sense to tag ‘Justin’ as a person rather than a ‘place’. Hence, adding the features of the neighboring words to the feature set of the current token seems useful.

The features that were experimented with but did not contribute to the final F1 score achieved are elaborated upon below:

1. Chunk Tagging
   A grammar was written to detect noun phrases (such as, "NP: {<DT>?<JJ>*<NN>}). This was parsed using the NLTK grammar based chunk parses (i.e. nltk.RegexpParser()). This was then used to chunk tag the sentence. And, it was added as a feature.

2. Inner Capital
   If the word contains an uppercase letter in any position except the first, then this binary feature is set. However, due to the ‘mixed case’ feature overlapping with this, this feature failed to perform well.

3. Vowel Count
   The number of vowels in a word are added as a feature. As nouns tend to have a higher number of vowels in them. However, this did not help in improving the performance metrics.

4. Word Length
   The number of characters in the word are added as a feature. A noun is usually surrounded by determiners, conjunctions, prepositions, adjectives, and verbs. The number of characters in nouns would mostly be greater the determiners, prepositions, and conjunctions around it. But, its length may be less than the verbs and adjectives around it. However, due to these trade-offs, this feature failed to improve the performance metrics.

5. Lowercase
   The word is tested if it is in lowercase. If a word is in full lowercase, it is most likely not a named entity. This feature highlights that. However, since capitalization norms are not followed in tweets for the majority of the times, this feature failed to boost the performance metrics.

6. Number of iterations

The default number of iterations in tagger.py is set as 25. The number of iterations were increased to 30 and tested. As the number of iterations increased in Structured Perceptron, the training of the weights of the model on the train data set increases and hence it tends to over fit to the training data. Hence, it performs poorly on the dev data set. Decreasing the number of iterations, the weights weren't getting adjusted adequately in the specified set of iterations. Hence, the optimum number of iterations was 25 and changing it caused a degrade in performance.

7. Context of length 2
   For each token, the neighboring words and their features are added to the feature list of the current token. The same concept was extended to two neighboring words, that is, the features of two words before and after the current token were added to its feature set. This was done with the intention to establish more context. However, due to large number of features getting generated, the performance metrics on the twitter_dev.ner and twitter_dev_test.ner improved, but the performance of twitter_test.ner reduced due to overfitting.
8. Punctuation
   If the word is a punctuation symbol, then it is not a named entity, and this binary feature is set. However, I think due to the small number of punctuation tokens in the test data, this feature failed to perform well.