Ref:
https://www.kaggle.com/code/nuhashafnan/eda-preprocessing-on-bangla-text-dataset#N-gram-Analysis

Note when reading data:

Reading the dataset is not tricky but need to be aware that we are given sequences of sentences and with labels for each of the entities. I have deliberately left `nan` values to signify the end of each sequence for when utilizing models that can handle sequences

We need to

- find out different entities found in the training set
- find out what kind of tags exist in the training set

Note that we need to handle tags that appear in training but might not in valid/test sets

FIndings

- 30798 unique entities and 14 types of tags found
- Almost 14000 sequences found
- The entities contain words as well as some kinds of punctuations

Since lots of punctuations and other chars were found we cleaned the entites to get rid of them. Doing so would increase our chosen method's learning abilities without distracting the model on useless items

- The distribution of named entities were equal overall and only unknown entities were found to be overrepresented. This is not unusual.
- We might need to use some kind of underrepresenting techniques to so that the O tags don't bias the model

We will experiment first by adding POS tags from an external dataset
https://www.kaggle.com/datasets/towhidahmedfoysal/bangla-parts-of-speechpos-tag
To get further insights

Doing so we see that the number of nouns, verbs, adverbs, adjective and pronouns are significantly higher. Ignoring the number of filler entities.

Outputting the cleaned dataset

Feature Engineering

- Model randomforest classifier
- We'll try with length, word frequency and pos as the features
- Using pos_features on its own gives very poor macro_f1 0.06
- Using a combination of length and frequency gives somewhat better results avg 0.21

Model choice

- Avg f1 macro score using GaussianNB and MultinomialNB is even poorer around 6%

Seems we need to tweak features a bit more
Let's add another feature where we indicate if a particular word is a stopword or not
We'll use external dataset
https://www.kaggle.com/datasets/shohanursobuj/bangla-stopwords
Using stop words gives around 17% macro f1 score for naive bayes and around 6% score for Randomforest

- Using so improves results to around 40% F1 score in cross validation and trees 20
- And to 25% for Naive Bayes classifier

The final inference score on the validation dataset

- Randomforest 7%
- Naive Bayes 6%

We will use pre-trained word embeddings to embed the individual entities and the tags and feed to a randomforest classifier as a final measure to improve performance

Additional Data cleaning

- We will create sequences of the cleaned sentences along with sequence of labels to feed into sequence model

The sequence model is an lstm unit unfolded till the length sequences

- We will embed the sentences using glove vectors trained on bangla text
- Each embedded token corresponds to a tag token
- And would try to predict the tag in a supervised fashion
- Since O tags are overwhelmingly large we try to reduce the weights of the tags of the O labels