* + - NER-tagging (Named Entity Recognition)
      * The process of data preparation for NER tagging is sub divided into several stages:-
        + Raw data
        + Pre-processing raw data
        + Passing into Prodigy (NER-tagging tool)
        + Extracting tags from JSON-L file
        + Passing into model
        + Train and test
      * Raw data sample set:

A close up of text on a black background

Description automatically generated

* + - * Pre-processed data sample set:

A close up of text on a black background

Description automatically generated

* + - * Prodigy sample of web application:

A screenshot of a cell phone

Description automatically generated

* + - * In order to access prodigy’s web application:
        + One is needed to procure the license to Prodigy
        + Installation of prodigy



* + - * + Running of custom tagging UI

A screenshot of a cell phone

Description automatically generated

* + - * + Exporting the tags as json-L

A screenshot of a cell phone

Description automatically generated

* + - * + Tags are ready to be passed into the model.
    - Now all the data is ready in the format needed to be passed in to the script for training and pre-processing.
  + Data pre-processing
    - It involves few basic steps:
      * Removal of stop words
      * Removal of single alphabet words
      * Removal of special characters
      * Converting all into lower case
      * Under some cases numbers are to be kept
      * Lemmatization of the raw data
      * Tokenizing the vocab
    - Before passing the data into model for training the input strings needs to be converted into arrays of numbers from the tokens generated.
    - The output labels in a classification type model needs to be encoded:
      * Hot encoded
      * Integer encoded
    - Now the data is ready for passing into the model as:

X -----🡪 Input array of tokens of words

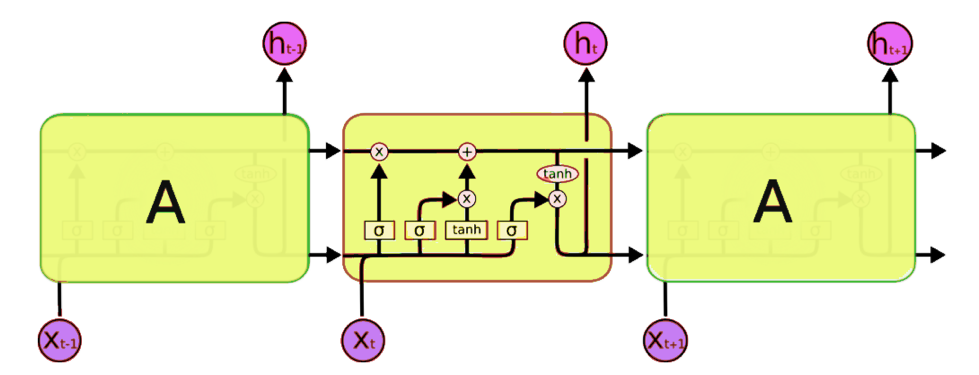
Y -----🡪 Output array of lables of entities

* + Neural architecture

A close up of a device

Description automatically generated

* + - Model architecture
      * Uni-directional LSTM
      * Bi-directional LSTM
      * LSTM model (Hidden layers for the neural model)



* + - Training
      * Passing the data as X and Y for training and validation.
    - Post-processing
      * It’s the same as pre-processing, the array of strings needs to be passed as array of tokens to the model for prediction.