r-ner-and-pos-tagging-from-scratch

March 18, 2024

Named Entity Recognition(NER)

This is the process of extracting pieces of structured information such as Name, location, a company, activity sectors etc , from unstructured text.NER can be used to recognize and parse important information from resumes etc. Apart from its use in information extraction it is used in preprocessing step for many NLP applications like Machine translation ,questions answering etc.

Parts of Speech(POS) Tagging

This is fundamental task in NLP which invloves assigning grammatical categories(such as Noun,Verb adjectives etc) to each word in a sentence in the data.The aim of POS tagging is to identify the syntactic structure of the sentence and identify he grammatical roles of individual words.

Relationship of POS tagging and NER in NLP

NER is used in Identifying and classifying entities (e.g., persons, organizations, locations) in text.while the role of POS Tagging is to help in identifying proper nouns, which are often indicative of named entities.

Bidirection Long Term Short term Memory (BiLSTM)

This is a combination of two LSTM-one runs forward from the right to the left the other runs backwards from left to right,this provides an additional context to the neural network that results in faster and even fuller learning on the problem.

Implimentation of BiLSTM for NER and POS tagging from scratch

###**Import all the Neccesary Libraries**

We'll perform the following steps:

1.Tokenize the text.

2.Label the tokens for POS tagging.

3.Label the tokens for NER.

[28]:
```python
# Import The Required Library
import numpy as np
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Embedding, Dense, TimeDistributed,␣
  ↪Bidirectional
```

```
[29]: import nltk
      nltk.download('punkt')
      nltk.download('averaged_perceptron_tagger')
      nltk.download('universal_tagset')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data…
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package universal_tagset to /root/nltk_data…
[nltk_data]   Package universal_tagset is already up-to-date!
```

```
[29]: True
```

```
[30]: # Dataset
      text = """
      The recently released 2019 national census report putting Nairobi as the most␣
        ↪populous county is an indicator of rising job and investment opportunities,
      Nairobi Governor Mike Sonko has said. Addressing guests during the launch of␣
        ↪the International Conference on Population (ICPD25),
      Sonko said the new resident population will shape development planning for the␣
        ↪county.
      This reality poses endless social and economic opportunities for all Kenyans␣
        ↪and our friends from all over the world,Sonko said.
      The Kenya Population and Housing Census 2019 results indicate that Nairobi has␣
        ↪a resident population of 4.4 million people.
      Over and above this figure, Nairobi has a day population in excess of six␣
        ↪million people,he added.
      Nairobi still generates over 60% of the country’s Gross Domestic Product and␣
        ↪was recently ranked as the sixth wealthiest city in Africa.
      However, the governor admitted that there are still challenges in terms of␣
        ↪infrastructure and social development.
      Our investment in these sectors has not grown in tandem with the population␣
        ↪over the years and we must improve on this,
      he said. The theme of this year’s conference is Accelerating The Promise and␣
        ↪the Governor has asked
      participants to focus on offering practical solutions that can be infused in␣
        ↪the development planning of the city.
```

```
National Treasury and Planning CS Ukur Yatani said the national government will␣
↪continue working closely with the county in financing urban development
"""
```

[31]:
```
# Step 1: Tokenize the text
from nltk.tokenize import word_tokenize
tokens = word_tokenize(text)
tokens
```

[31]:
```
['The',
 'recently',
 'released',
 '2019',
 'national',
 'census',
 'report',
 'putting',
 'Nairobi',
 'as',
 'the',
 'most',
 'populous',
 'county',
 'is',
 'an',
 'indicator',
 'of',
 'rising',
 'job',
 'and',
 'investment',
 'opportunities',
 ',',
 'Nairobi',
 'Governor',
 'Mike',
 'Sonko',
 'has',
 'said',
 '.',
 'Addressing',
 'guests',
 'during',
 'the',
 'launch',
 'of',
 'the',
 'International',
 'Conference',
 'on',
 'Population',
 '(',
 'ICPD25',
 ')',
 ',',
 'Sonko',
 'said',
 'the',
 'new',
 'resident',
 'population',
 'will',
 'shape',
 'development',
 'planning',
 'for',
 'the',
 'county',
 '.',
 'This',
 'reality',
 'poses',
 'endless',
 'social',
 'and',
 'economic',
 'opportunities',
 'for',
 'all',
 'Kenyans',
 'and',
 'our',
 'friends',
 'from',
 'all',
 'over',
 'the',
 'world',
 ',',
 'Sonko',
 'said',
 '.',
 'The',
 'Kenya',
 'Population',
 'and',
 'Housing',
 'Census',
 '2019',
 'results',
 'indicate',
 'that',
 'Nairobi',
 'has',
 'a',
 'resident',
 'population',
 'of',
 '4.4',
 'million',
 'people',
 '.',
 'Over',
 'and',
 'above',
 'this',
 'figure',
 ',',
 'Nairobi',
 'has',
 'a',
 'day',
 'population',
 'in',
 'excess',
 'of',
 'six',
 'million',
 'people',
 ',',
 'he',
 'added',
 '.',
 'Nairobi',
 'still',
 'generates',
 'over',
 '60',
 '%',
 'of',
 'the',
 'country',
 '’',
 's',
 'Gross',
 'Domestic',
 'Product',
 'and',
 'was',
 'recently',
 'ranked',
 'as',
 'the',
 'sixth',
 'wealthiest',
 'city',
 'in',
 'Africa',
 '.',
 'However',
 ',',
 'the',
 'governor',
 'admitted',
 'that',
 'there',
 'are',
 'still',
 'challenges',
 'in',
 'terms',
 'of',
 'infrastructure',
 'and',
 'social',
 'development',
 '.',
 'Our',
 'investment',
 'in',
 'these',
 'sectors',
 'has',
 'not',
 'grown',
 'in',
 'tandem',
 'with',
 'the',
 'population',
 'over',
 'the',
 'years',
 'and',
 'we',
 'must',
 'improve',
 'on',
 'this',
 ',',
 'he',
 'said',
 '.',
 'The',
 'theme',
 'of',
 'this',
 'year',
 ''',
 's',
 'conference',
 'is',
 'Accelerating',
 'The',
 'Promise',
 'and',
 'the',
 'Governor',
 'has',
 'asked',
 'participants',
 'to',
 'focus',
 'on',
 'offering',
 'practical',
 'solutions',
 'that',
 'can',
 'be',
 'infused',
 'in',
 'the',
 'development',
 'planning',
  'of',
  'the',
  'city',
  '.',
  'National',
  'Treasury',
  'and',
  'Planning',
  'CS',
  'Ukur',
  'Yatani',
  'said',
  'the',
  'national',
  'government',
  'will',
  'continue',
  'working',
  'closely',
  'with',
  'the',
  'county',
  'in',
  'financing',
  'urban',
  'development']
```

[33]:
```
# Step 2: Label the tokens for POS tagging
# We will use NLTK's pos_tag function for this task
pos_tags = nltk.pos_tag(tokens)
pos_tags
```

[33]:
```
[('The', 'DT'),
 ('recently', 'RB'),
 ('released', 'VBN'),
 ('2019', 'CD'),
 ('national', 'JJ'),
 ('census', 'NN'),
 ('report', 'NN'),
 ('putting', 'VBG'),
 ('Nairobi', 'NNP'),
 ('as', 'IN'),
 ('the', 'DT'),
 ('most', 'RBS'),
 ('populous', 'JJ'),
 ('county', 'NN'),
 ('is', 'VBZ'),
('an', 'DT'),
('indicator', 'NN'),
('of', 'IN'),
('rising', 'VBG'),
('job', 'NN'),
('and', 'CC'),
('investment', 'NN'),
('opportunities', 'NNS'),
(',', ','),
('Nairobi', 'NNP'),
('Governor', 'NNP'),
('Mike', 'NNP'),
('Sonko', 'NNP'),
('has', 'VBZ'),
('said', 'VBD'),
('.', '.'),
('Addressing', 'VBG'),
('guests', 'NNS'),
('during', 'IN'),
('the', 'DT'),
('launch', 'NN'),
('of', 'IN'),
('the', 'DT'),
('International', 'NNP'),
('Conference', 'NNP'),
('on', 'IN'),
('Population', 'NNP'),
('(', '('),
('ICPD25', 'NNP'),
(')', ')'),
(',', ','),
('Sonko', 'NNP'),
('said', 'VBD'),
('the', 'DT'),
('new', 'JJ'),
('resident', 'JJ'),
('population', 'NN'),
('will', 'MD'),
('shape', 'VB'),
('development', 'NN'),
('planning', 'NN'),
('for', 'IN'),
('the', 'DT'),
('county', 'NN'),
('.', '.'),
('This', 'DT'),
('reality', 'NN'),
('poses', 'VBZ'),
('endless', 'JJ'),
('social', 'JJ'),
('and', 'CC'),
('economic', 'JJ'),
('opportunities', 'NNS'),
('for', 'IN'),
('all', 'DT'),
('Kenyans', 'NNP'),
('and', 'CC'),
('our', 'PRP$'),
('friends', 'NNS'),
('from', 'IN'),
('all', 'DT'),
('over', 'IN'),
('the', 'DT'),
('world', 'NN'),
(',', ','),
('Sonko', 'NNP'),
('said', 'VBD'),
('.', '.'),
('The', 'DT'),
('Kenya', 'NNP'),
('Population', 'NNP'),
('and', 'CC'),
('Housing', 'NNP'),
('Census', 'NNP'),
('2019', 'CD'),
('results', 'NNS'),
('indicate', 'VBP'),
('that', 'IN'),
('Nairobi', 'NNP'),
('has', 'VBZ'),
('a', 'DT'),
('resident', 'JJ'),
('population', 'NN'),
('of', 'IN'),
('4.4', 'CD'),
('million', 'CD'),
('people', 'NNS'),
('.', '.'),
('Over', 'IN'),
('and', 'CC'),
('above', 'IN'),
('this', 'DT'),
('figure', 'NN'),
(',', ','),
('Nairobi', 'NNP'),
('has', 'VBZ'),
('a', 'DT'),
('day', 'NN'),
('population', 'NN'),
('in', 'IN'),
('excess', 'NN'),
('of', 'IN'),
('six', 'CD'),
('million', 'CD'),
('people', 'NNS'),
(',', ','),
('he', 'PRP'),
('added', 'VBD'),
('.', '.'),
('Nairobi', 'NNP'),
('still', 'RB'),
('generates', 'VBZ'),
('over', 'IN'),
('60', 'CD'),
('%', 'NN'),
('of', 'IN'),
('the', 'DT'),
('country', 'NN'),
('’', 'NNP'),
('s', 'VBZ'),
('Gross', 'NNP'),
('Domestic', 'NNP'),
('Product', 'NNP'),
('and', 'CC'),
('was', 'VBD'),
('recently', 'RB'),
('ranked', 'VBN'),
('as', 'IN'),
('the', 'DT'),
('sixth', 'JJ'),
('wealthiest', 'JJS'),
('city', 'NN'),
('in', 'IN'),
('Africa', 'NNP'),
('.', '.'),
('However', 'RB'),
(',', ','),
('the', 'DT'),
('governor', 'NN'),
('admitted', 'VBD'),
('that', 'IN'),
('there', 'EX'),
('are', 'VBP'),
('still', 'RB'),
('challenges', 'NNS'),
('in', 'IN'),
('terms', 'NNS'),
('of', 'IN'),
('infrastructure', 'NN'),
('and', 'CC'),
('social', 'JJ'),
('development', 'NN'),
('.', '.'),
('Our', 'PRP$'),
('investment', 'NN'),
('in', 'IN'),
('these', 'DT'),
('sectors', 'NNS'),
('has', 'VBZ'),
('not', 'RB'),
('grown', 'VBN'),
('in', 'IN'),
('tandem', 'NN'),
('with', 'IN'),
('the', 'DT'),
('population', 'NN'),
('over', 'IN'),
('the', 'DT'),
('years', 'NNS'),
('and', 'CC'),
('we', 'PRP'),
('must', 'MD'),
('improve', 'VB'),
('on', 'IN'),
('this', 'DT'),
(',', ','),
('he', 'PRP'),
('said', 'VBD'),
('.', '.'),
('The', 'DT'),
('theme', 'NN'),
('of', 'IN'),
('this', 'DT'),
('year', 'NN'),
('’', 'VBZ'),
('s', 'JJ'),
('conference', 'NN'),
('is', 'VBZ'),
('Accelerating', 'VBG'),
('The', 'DT'),
('Promise', 'NNP'),
('and', 'CC'),
('the', 'DT'),
('Governor', 'NNP'),
('has', 'VBZ'),
('asked', 'VBN'),
('participants', 'NNS'),
('to', 'TO'),
('focus', 'VB'),
('on', 'IN'),
('offering', 'VBG'),
('practical', 'JJ'),
('solutions', 'NNS'),
('that', 'WDT'),
('can', 'MD'),
('be', 'VB'),
('infused', 'VBN'),
('in', 'IN'),
('the', 'DT'),
('development', 'NN'),
('planning', 'NN'),
('of', 'IN'),
('the', 'DT'),
('city', 'NN'),
('.', '.'),
('National', 'NNP'),
('Treasury', 'NNP'),
('and', 'CC'),
('Planning', 'NNP'),
('CS', 'NNP'),
('Ukur', 'NNP'),
('Yatani', 'NNP'),
('said', 'VBD'),
('the', 'DT'),
('national', 'JJ'),
('government', 'NN'),
('will', 'MD'),
('continue', 'VB'),
('working', 'VBG'),
('closely', 'RB'),
('with', 'IN'),
('the', 'DT'),
('county', 'NN'),
('in', 'IN'),
('financing', 'VBG'),
 ('urban', 'JJ'),
 ('development', 'NN')]
```

[34]:
```
# Step 3: Label the tokens for NER
# Manually labeling some entities in the text
ner_labels = [
    ('Nairobi', 'GPE'), ('2019', 'DATE'), ('Mike Sonko', 'PERSON'), ('Nairobi',␣
 ↪'GPE'), ('ICPD25', 'ORGANIZATION'),
    ('Kenya', 'GPE'), ('2019', 'DATE'), ('Nairobi', 'GPE'), ('Nairobi', 'GPE'),␣
 ↪('Nairobi', 'GPE'), ('Africa', 'LOCATION'),
    ('Nairobi', 'GPE'), ('CS Ukur Yatani', 'PERSON')
]
```

[35]:
```
# Step 4: Prepare the data for training
#Creating sequences of tokens and their corresponding POS and NER labels
pos_data = [(token, pos) for token, pos in pos_tags]
ner_data = [(token, label) for token, label in ner_labels]
```

[36]:
```
pos_data
```

[36]:
```
[('The', 'DT'),
 ('recently', 'RB'),
 ('released', 'VBN'),
 ('2019', 'CD'),
 ('national', 'JJ'),
 ('census', 'NN'),
 ('report', 'NN'),
 ('putting', 'VBG'),
 ('Nairobi', 'NNP'),
 ('as', 'IN'),
 ('the', 'DT'),
 ('most', 'RBS'),
 ('populous', 'JJ'),
 ('county', 'NN'),
 ('is', 'VBZ'),
 ('an', 'DT'),
 ('indicator', 'NN'),
 ('of', 'IN'),
 ('rising', 'VBG'),
 ('job', 'NN'),
 ('and', 'CC'),
 ('investment', 'NN'),
 ('opportunities', 'NNS'),
 (',', ','),
 ('Nairobi', 'NNP'),
 ('Governor', 'NNP'),
 ('Mike', 'NNP'),
('Sonko', 'NNP'),
('has', 'VBZ'),
('said', 'VBD'),
('.', '.'),
('Addressing', 'VBG'),
('guests', 'NNS'),
('during', 'IN'),
('the', 'DT'),
('launch', 'NN'),
('of', 'IN'),
('the', 'DT'),
('International', 'NNP'),
('Conference', 'NNP'),
('on', 'IN'),
('Population', 'NNP'),
('(', '('),
('ICPD25', 'NNP'),
(')', ')'),
(',', ','),
('Sonko', 'NNP'),
('said', 'VBD'),
('the', 'DT'),
('new', 'JJ'),
('resident', 'JJ'),
('population', 'NN'),
('will', 'MD'),
('shape', 'VB'),
('development', 'NN'),
('planning', 'NN'),
('for', 'IN'),
('the', 'DT'),
('county', 'NN'),
('.', '.'),
('This', 'DT'),
('reality', 'NN'),
('poses', 'VBZ'),
('endless', 'JJ'),
('social', 'JJ'),
('and', 'CC'),
('economic', 'JJ'),
('opportunities', 'NNS'),
('for', 'IN'),
('all', 'DT'),
('Kenyans', 'NNP'),
('and', 'CC'),
('our', 'PRP$'),
('friends', 'NNS'),
('from', 'IN'),
('all', 'DT'),
('over', 'IN'),
('the', 'DT'),
('world', 'NN'),
(',', ','),
('Sonko', 'NNP'),
('said', 'VBD'),
('.', '.'),
('The', 'DT'),
('Kenya', 'NNP'),
('Population', 'NNP'),
('and', 'CC'),
('Housing', 'NNP'),
('Census', 'NNP'),
('2019', 'CD'),
('results', 'NNS'),
('indicate', 'VBP'),
('that', 'IN'),
('Nairobi', 'NNP'),
('has', 'VBZ'),
('a', 'DT'),
('resident', 'JJ'),
('population', 'NN'),
('of', 'IN'),
('4.4', 'CD'),
('million', 'CD'),
('people', 'NNS'),
('.', '.'),
('Over', 'IN'),
('and', 'CC'),
('above', 'IN'),
('this', 'DT'),
('figure', 'NN'),
(',', ','),
('Nairobi', 'NNP'),
('has', 'VBZ'),
('a', 'DT'),
('day', 'NN'),
('population', 'NN'),
('in', 'IN'),
('excess', 'NN'),
('of', 'IN'),
('six', 'CD'),
('million', 'CD'),
('people', 'NNS'),
(',', ','),
('he', 'PRP'),
('added', 'VBD'),
('.', '.'),
('Nairobi', 'NNP'),
('still', 'RB'),
('generates', 'VBZ'),
('over', 'IN'),
('60', 'CD'),
('%', 'NN'),
('of', 'IN'),
('the', 'DT'),
('country', 'NN'),
('’', 'NNP'),
('s', 'VBZ'),
('Gross', 'NNP'),
('Domestic', 'NNP'),
('Product', 'NNP'),
('and', 'CC'),
('was', 'VBD'),
('recently', 'RB'),
('ranked', 'VBN'),
('as', 'IN'),
('the', 'DT'),
('sixth', 'JJ'),
('wealthiest', 'JJS'),
('city', 'NN'),
('in', 'IN'),
('Africa', 'NNP'),
('.', '.'),
('However', 'RB'),
(',', ','),
('the', 'DT'),
('governor', 'NN'),
('admitted', 'VBD'),
('that', 'IN'),
('there', 'EX'),
('are', 'VBP'),
('still', 'RB'),
('challenges', 'NNS'),
('in', 'IN'),
('terms', 'NNS'),
('of', 'IN'),
('infrastructure', 'NN'),
('and', 'CC'),
('social', 'JJ'),
('development', 'NN'),
('.', '.'),
('Our', 'PRP$'),
('investment', 'NN'),
('in', 'IN'),
('these', 'DT'),
('sectors', 'NNS'),
('has', 'VBZ'),
('not', 'RB'),
('grown', 'VBN'),
('in', 'IN'),
('tandem', 'NN'),
('with', 'IN'),
('the', 'DT'),
('population', 'NN'),
('over', 'IN'),
('the', 'DT'),
('years', 'NNS'),
('and', 'CC'),
('we', 'PRP'),
('must', 'MD'),
('improve', 'VB'),
('on', 'IN'),
('this', 'DT'),
(',', ','),
('he', 'PRP'),
('said', 'VBD'),
('.', '.'),
('The', 'DT'),
('theme', 'NN'),
('of', 'IN'),
('this', 'DT'),
('year', 'NN'),
('’', 'VBZ'),
('s', 'JJ'),
('conference', 'NN'),
('is', 'VBZ'),
('Accelerating', 'VBG'),
('The', 'DT'),
('Promise', 'NNP'),
('and', 'CC'),
('the', 'DT'),
('Governor', 'NNP'),
('has', 'VBZ'),
('asked', 'VBN'),
('participants', 'NNS'),
('to', 'TO'),
('focus', 'VB'),
('on', 'IN'),
('offering', 'VBG'),
('practical', 'JJ'),
('solutions', 'NNS'),
('that', 'WDT'),
('can', 'MD'),
('be', 'VB'),
('infused', 'VBN'),
('in', 'IN'),
('the', 'DT'),
('development', 'NN'),
('planning', 'NN'),
('of', 'IN'),
('the', 'DT'),
('city', 'NN'),
('.', '.'),
('National', 'NNP'),
('Treasury', 'NNP'),
('and', 'CC'),
('Planning', 'NNP'),
('CS', 'NNP'),
('Ukur', 'NNP'),
('Yatani', 'NNP'),
('said', 'VBD'),
('the', 'DT'),
('national', 'JJ'),
('government', 'NN'),
('will', 'MD'),
('continue', 'VB'),
('working', 'VBG'),
('closely', 'RB'),
('with', 'IN'),
('the', 'DT'),
('county', 'NN'),
('in', 'IN'),
('financing', 'VBG'),
('urban', 'JJ'),
('development', 'NN')]
```

```
[37]: ner_data
```

```
[37]: [('Nairobi', 'GPE'),
       ('2019', 'DATE'),
       ('Mike Sonko', 'PERSON'),
       ('Nairobi', 'GPE'),
       ('ICPD25', 'ORGANIZATION'),
       ('Kenya', 'GPE'),
       ('2019', 'DATE'),
       ('Nairobi', 'GPE'),
       ('Nairobi', 'GPE'),
       ('Nairobi', 'GPE'),
       ('Africa', 'LOCATION'),
       ('Nairobi', 'GPE'),
       ('CS Ukur Yatani', 'PERSON')]
```

```
[38]: # Define the vocabulary based on unique words in the text
vocab = set(tokens)
vocab
```

```
[38]: {'%',
 '(',
 ')',
 ',',
 '.',
 '2019',
 '4.4',
 '60',
 'Accelerating',
 'Addressing',
 'Africa',
 'CS',
 'Census',
 'Conference',
 'Domestic',
 'Governor',
 'Gross',
 'Housing',
 'However',
 'ICPD25',
 'International',
 'Kenya',
 'Kenyans',
 'Mike',
 'Nairobi',
 'National',
 'Our',
 'Over',
 'Planning',
 'Population',
 'Product',
 'Promise',
 'Sonko',
 'The',
 'This',
 'Treasury',
 'Ukur',
 'Yatani',
 'a',
 'above',
 'added',
 'admitted',
 'all',
 'an',
 'and',
 'are',
 'as',
 'asked',
 'be',
 'can',
 'census',
 'challenges',
 'city',
 'closely',
 'conference',
 'continue',
 'country',
 'county',
 'day',
 'development',
 'during',
 'economic',
 'endless',
 'excess',
 'figure',
 'financing',
 'focus',
 'for',
 'friends',
 'from',
 'generates',
 'government',
 'governor',
 'grown',
 'guests',
 'has',
 'he',
 'improve',
 'in',
 'indicate',
 'indicator',
 'infrastructure',
 'infused',
 'investment',
 'is',
 'job',
 'launch',
 'million',
 'most',
 'must',
 'national',
 'new',
 'not',
 'of',
 'offering',
 'on',
 'opportunities',
 'our',
 'over',
 'participants',
 'people',
 'planning',
 'population',
 'populous',
 'poses',
 'practical',
 'putting',
 'ranked',
 'reality',
 'recently',
 'released',
 'report',
 'resident',
 'results',
 'rising',
 's',
 'said',
 'sectors',
 'shape',
 'six',
 'sixth',
 'social',
 'solutions',
 'still',
 'tandem',
 'terms',
 'that',
 'the',
 'theme',
 'there',
  'these',
  'this',
  'to',
  'urban',
  'was',
  'we',
  'wealthiest',
  'will',
  'with',
  'working',
  'world',
  'year',
  'years',
  '’'}
```

```
[40]: # Define word-to-index dictionary
      word2idx = {word: idx + 1 for idx, word in enumerate(vocab)}
      word2idx
```

```
[40]: {'over': 1,
       'participants': 2,
       'National': 3,
       '4.4': 4,
       'poses': 5,
       'world': 6,
       'focus': 7,
       'Domestic': 8,
       'people': 9,
       'is': 10,
       'social': 11,
       'from': 12,
       'ICPD25': 13,
       'Addressing': 14,
       'as': 15,
       'excess': 16,
       'Kenya': 17,
       'be': 18,
       'on': 19,
       'must': 20,
       'Nairobi': 21,
       'populous': 22,
       'year': 23,
       'working': 24,
       'during': 25,
       'day': 26,
       's': 27,
       'million': 28,
'governor': 29,
'friends': 30,
'will': 31,
'reality': 32,
'%': 33,
'the': 34,
'new': 35,
'indicator': 36,
'development': 37,
'country': 38,
'grown': 39,
'2019': 40,
'added': 41,
'generates': 42,
'Ukur': 43,
'still': 44,
'planning': 45,
'putting': 46,
'Census': 47,
'a': 48,
'Population': 49,
'sectors': 50,
'for': 51,
'above': 52,
'shape': 53,
'financing': 54,
'International': 55,
'all': 56,
'figure': 57,
'report': 58,
'can': 59,
'CS': 60,
'to': 61,
'Kenyans': 62,
'Africa': 63,
'asked': 64,
'there': 65,
'census': 66,
'terms': 67,
'Promise': 68,
'ranked': 69,
'launch': 70,
'economic': 71,
'most': 72,
'.': 73,
'government': 74,
'solutions': 75,
'’': 76,
'tandem': 77,
'was': 78,
'Planning': 79,
'job': 80,
'city': 81,
'recently': 82,
'that': 83,
'these': 84,
'urban': 85,
'indicate': 86,
'Mike': 87,
'admitted': 88,
'conference': 89,
'released': 90,
'this': 91,
'years': 92,
'60': 93,
'and': 94,
'with': 95,
'population': 96,
'Accelerating': 97,
'Product': 98,
'Treasury': 99,
'our': 100,
'This': 101,
'Over': 102,
'Our': 103,
'infused': 104,
'investment': 105,
'six': 106,
'Conference': 107,
'national': 108,
'continue': 109,
'said': 110,
'offering': 111,
'The': 112,
'Gross': 113,
'he': 114,
'we': 115,
'infrastructure': 116,
'in': 117,
'has': 118,
'endless': 119,
'sixth': 120,
'challenges': 121,
'of': 122,
'theme': 123,
'closely': 124,
'improve': 125,
'guests': 126,
'Housing': 127,
'opportunities': 128,
'an': 129,
'wealthiest': 130,
')': 131,
'not': 132,
'resident': 133,
'results': 134,
'county': 135,
'are': 136,
',': 137,
'However': 138,
'Sonko': 139,
'Yatani': 140,
'(': 141,
'practical': 142,
'Governor': 143,
'rising': 144}
```

[41]:
```
# Define the inverse mapping from index to word
idx2word = {idx: word for word, idx in word2idx.items()}
word2idx["ENDPAD"] = 0  # Add padding token
```

[42]:
```
# Step 4: Prepare the data for training
# Convert words to indices using word2idx dictionary
pos_X = []
for token, _ in pos_data:
    if token in word2idx:
        pos_X.append(word2idx[token])
    else:
        pos_X.append(word2idx["ENDPAD"])  # Use padding token for␣
↪out-of-vocabulary words

ner_X = []
for token, _ in ner_data:
    if token in word2idx:
        ner_X.append(word2idx[token])
    else:
        ner_X.append(word2idx["ENDPAD"])  # Use padding token for␣
↪out-of-vocabulary words
```

[43]:
```
ner_labels
```

```
[43]: [('Nairobi', 'GPE'),
       ('2019', 'DATE'),
       ('Mike Sonko', 'PERSON'),
       ('Nairobi', 'GPE'),
       ('ICPD25', 'ORGANIZATION'),
       ('Kenya', 'GPE'),
       ('2019', 'DATE'),
       ('Nairobi', 'GPE'),
       ('Nairobi', 'GPE'),
       ('Nairobi', 'GPE'),
       ('Africa', 'LOCATION'),
       ('Nairobi', 'GPE'),
       ('CS Ukur Yatani', 'PERSON')]
```

```
[44]: # Convert POS and NER labels to indices
      pos_y = [nltk.tag.map_tag('en-ptb', 'universal', pos) for (_, pos) in pos_tags]
      pos_y
```

```
[44]: ['DET',
       'ADV',
       'VERB',
       'NUM',
       'ADJ',
       'NOUN',
       'NOUN',
       'VERB',
       'NOUN',
       'ADP',
       'DET',
       'ADV',
       'ADJ',
       'NOUN',
       'VERB',
       'DET',
       'NOUN',
       'ADP',
       'VERB',
       'NOUN',
       'CONJ',
       'NOUN',
       'NOUN',
       '.',
       'NOUN',
       'NOUN',
       'NOUN',
       'NOUN',
       'VERB',
       'VERB',
       '.',
       'VERB',
       'NOUN',
       'ADP',
       'DET',
       'NOUN',
       'ADP',
       'DET',
       'NOUN',
       'NOUN',
       'ADP',
       'NOUN',
       '.',
       'NOUN',
       '.',
       '.',
       'NOUN',
       'VERB',
       'DET',
       'ADJ',
       'ADJ',
       'NOUN',
       'VERB',
       'VERB',
       'NOUN',
       'NOUN',
       'ADP',
       'DET',
       'NOUN',
       '.',
       'DET',
       'NOUN',
       'VERB',
       'ADJ',
       'ADJ',
       'CONJ',
       'ADJ',
       'NOUN',
       'ADP',
       'DET',
       'NOUN',
       'CONJ',
       'PRON',
       'NOUN',
       'ADP',
       'DET',
       'ADP',
       'DET',
       'NOUN',
       '.',
       'NOUN',
       'VERB',
       '.',
       'DET',
       'NOUN',
       'NOUN',
       'CONJ',
       'NOUN',
       'NOUN',
       'NUM',
       'NOUN',
       'VERB',
       'ADP',
       'NOUN',
       'VERB',
       'DET',
       'ADJ',
       'NOUN',
       'ADP',
       'NUM',
       'NUM',
       'NOUN',
       '.',
       'ADP',
       'CONJ',
       'ADP',
       'DET',
       'NOUN',
       '.',
       'NOUN',
       'VERB',
       'DET',
       'NOUN',
       'NOUN',
       'ADP',
       'NOUN',
       'ADP',
       'NUM',
       'NUM',
       'NOUN',
       '.',
       'PRON',
       'VERB',
       '.',
       'NOUN',
       'ADV',
       'VERB',
       'ADP',
       'NUM',
       'NOUN',
       'ADP',
       'DET',
       'NOUN',
       'NOUN',
       'VERB',
       'NOUN',
       'NOUN',
       'NOUN',
       'CONJ',
       'VERB',
       'ADV',
       'VERB',
       'ADP',
       'DET',
       'ADJ',
       'ADJ',
       'NOUN',
       'ADP',
       'NOUN',
       '.',
       'ADV',
       '.',
       'DET',
       'NOUN',
       'VERB',
       'ADP',
       'DET',
       'VERB',
       'ADV',
       'NOUN',
       'ADP',
       'NOUN',
       'ADP',
       'NOUN',
       'CONJ',
       'ADJ',
       'NOUN',
       '.',
       'PRON',
       'NOUN',
       'ADP',
       'DET',
       'NOUN',
       'VERB',
       'ADV',
       'VERB',
       'ADP',
       'NOUN',
       'ADP',
       'DET',
       'NOUN',
       'ADP',
       'DET',
       'NOUN',
       'CONJ',
       'PRON',
       'VERB',
       'VERB',
       'ADP',
       'DET',
       '.',
       'PRON',
       'VERB',
       '.',
       'DET',
       'NOUN',
       'ADP',
       'DET',
       'NOUN',
       'VERB',
       'ADJ',
       'NOUN',
       'VERB',
       'VERB',
       'DET',
       'NOUN',
       'CONJ',
       'DET',
       'NOUN',
       'VERB',
       'VERB',
       'NOUN',
       'PRT',
       'VERB',
       'ADP',
       'VERB',
       'ADJ',
       'NOUN',
       'DET',
       'VERB',
       'VERB',
       'VERB',
       'ADP',
       'DET',
       'NOUN',
       'NOUN',
       'ADP',
       'DET',
       'NOUN',
       '.',
       'NOUN',
       'NOUN',
       'CONJ',
       'NOUN',
       'NOUN',
       'NOUN',
       'NOUN',
       'VERB',
       'DET',
       'ADJ',
       'NOUN',
       'VERB',
       'VERB',
       'VERB',
       'ADV',
       'ADP',
       'DET',
       'NOUN',
       'ADP',
       'VERB',
       'ADJ',
       'NOUN']
```

[107]:
```
# Update NER Labels Dictionary
ner_labels = {'GPE': 11, 'DATE': 6, 'PERSON': 12, 'ORGANIZATION': 4, 'LOCATION':
 ↪ 10}

# Update NER Labels Mapping
ner_y = [ner_labels[label] if label in ner_labels else -1 for _, label in␣
 ↪ner_data]

# Print updated ner_y
print(ner_y)
```

```
[11, 6, 12, 11, 4, 11, 6, 11, 11, 11, 10, 11, 12]
```

[108]:
```
# Split the data into training and testing sets
from sklearn.model_selection import train_test_split
pos_X_train, pos_X_test, pos_y_train, pos_y_test = train_test_split(pos_X,␣
 ↪pos_y, test_size=0.2, random_state=42)
ner_X_train,ner_X_test, ner_y_train, ner_y_test = train_test_split(ner_X,␣
 ↪ner_y, test_size=0.2, random_state=42)
```

[110]:
```
# Define the number of POS tags based on the unique tags in the training data
num_pos_tags = len(set(pos_y_train))

# Ensure output data has the same length as input sequences
max_seq_length = 100

# Reshape pos_y_train and pos_y_test to match input sequence length
pos_y_train = pad_sequences([pos_y_train], maxlen=max_seq_length,␣
 ↪padding='post', truncating='post')[0]
pos_y_test = pad_sequences([pos_y_test], maxlen=max_seq_length, padding='post',␣
 ↪truncating='post')[0]

# Convert to one-hot encoding after reshaping
pos_y_train = to_categorical(pos_y_train, num_classes=num_pos_tags)
pos_y_test = to_categorical(pos_y_test, num_classes=num_pos_tags)
```

[111]:
```
# Convert POS and NER labels to one-hot encodings
num_ner_tags = len(set(ner_y_train))
```

[89]:
```
# Convert POS tags to integer indices using NLTK's tag mapping
pos_tagset = sorted(set(pos_y_train))  # Get unique POS tags
pos_tag_indices = {tag: idx for idx, tag in enumerate(pos_tagset)}

# Map POS tags to integer indices
pos_y_train_idx = [pos_tag_indices[tag] for tag in pos_y_train]
pos_y_test_idx = [pos_tag_indices[tag] for tag in pos_y_test]

# Convert integer indices to one-hot encodings
pos_y_train = to_categorical(pos_y_train_idx, num_classes=num_pos_tags)
pos_y_test = to_categorical(pos_y_test_idx, num_classes=num_pos_tags)
```

[113]:
```
# Calculate the actual number of unique NER tags
num_ner_tags_actual = len(set(ner_y_train))

# Create a set of unique NER labels present in the training data
unique_ner_labels = set(ner_y_train)

# Filter out labels not present in the ner_labels dictionary
```

```
ner_labels_filtered = {label: idx for label, idx in ner_labels.items() if label␣
↪in unique_ner_labels}

# Convert NER labels to integer indices using the updated mapping
ner_y_train_idx = [ner_labels_filtered.get(label, -1) for label in ner_y_train]
ner_y_test_idx = [ner_labels_filtered.get(label, -1) for label in ner_y_test]

# Convert integer indices to one-hot encodings
ner_y_train = to_categorical(ner_y_train_idx, num_classes=num_ner_tags_actual)
ner_y_test = to_categorical(ner_y_test_idx, num_classes=num_ner_tags_actual)
```

[101]:
```
# Define the BiLSTM model for POS tagging
pos_model = Sequential()
pos_model.add(Embedding(input_dim=len(word2idx), output_dim=50,␣
↪input_length=MAX_SEQ_LENGTH))
pos_model.add(Bidirectional(LSTM(units=100, return_sequences=True)))
pos_model.add(TimeDistributed(Dense(num_pos_tags, activation='softmax')))
```

[102]:
```
# POS model Summarry
pos_model.summary()
```

```
Model: "sequential_8"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_7 (Embedding)     (None, 100, 50)           7250

 bidirectional_7 (Bidirecti  (None, 100, 200)          120800
 onal)

 time_distributed_7 (TimeDi  (None, 100, 2)            402
 stributed)

=================================================================
Total params: 128452 (501.77 KB)
Trainable params: 128452 (501.77 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[103]:
```
# compiling the model
pos_model.compile(optimizer='adam', loss='categorical_crossentropy',␣
↪metrics=['accuracy'])
```

[114]:
```
# Set the batch size to match the size of pos_y_train
batch_size = len(pos_y_train)

# Train the POS tagging model with adjusted batch size
```

```
pos_model.fit(pos_X_train, pos_y_train, validation_data=(pos_X_test,␣
↪pos_y_test), epochs=10, batch_size=batch_size, shuffle=True)
```

[115]:
```
# Define maximum sequence length for padding
MAX_SEQ_LENGTH = 100
# Define the BiLSTM model for NER
ner_model = Sequential()
ner_model.add(Embedding(input_dim=len(word2idx), output_dim=50,␣
↪input_length=MAX_SEQ_LENGTH))
ner_model.add(Bidirectional(LSTM(units=100, return_sequences=True)))
ner_model.add(TimeDistributed(Dense(num_ner_tags, activation='softmax')))

ner_model.compile(optimizer='adam', loss='categorical_crossentropy',␣
↪metrics=['accuracy'])

# Train the NER model
ner_model.fit(ner_X_train, ner_y_train, validation_data=(ner_X_test,␣
↪ner_y_test), epochs=10, batch_size=32)
```

References

1.https://zhoubeiqi.medium.com/named-entity-recognition-ner-using-keras-lstm-spacy-da3ea63d24c5.

2.https://medium.com/@sujathamudadla1213/what-is-parts-of-speech-pos-tagging-natural-language-processing-in-2b8f4b07b186

3.https://medium.com/@ikeokaforcasper/bidirectional-lstms-how-do-they-work-bc7247fda05c