Named Entity Recognition on News Articles

[1]:
```python
import pandas as pd
import spacy
from nltk.tokenize import word_tokenize
from nltk.tag import pos_tag
from sklearn.feature_extraction import DictVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
```

[2]:
```python
nlp = spacy.load('en_core_web_sm')
```

[5]:
```python
import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')
nltk.download('words')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\c.sruthi\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt.zip.
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\c.sruthi\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     C:\Users\c.sruthi\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping chunkers\maxent_ne_chunker.zip.
[nltk_data] Downloading package words to
[nltk_data]     C:\Users\c.sruthi\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\words.zip.
```

[5] :
```
True
```

[6] :
```python
df = pd.read_csv('cnbc_news_datase.csv')
```

[7] :
```python
def extract_entities_spacy(text):
    doc = nlp(text)
    named_entities = [(ent.text, ent.label_) for ent in doc.ents]
    return named_entities
```

[8] :
```
def tokenize_and_pos_tag(text):
    words = word_tokenize(text)
    pos_tags = pos_tag(words)
    return pos_tags
```

[9] :
```
df['description'].fillna('', inplace=True)
```

[10] :
```
df['named_entities_spacy'] = df['description'].apply(extract_entities_spacy)
```

[11] :
```
df['tokens_pos_tags'] = df['description'].apply(tokenize_and_pos_tag)
```

[12] :
```
def visualize_named_entities_spacy(tokens, labels):
    plt.figure(figsize=(10, 5))
    plt.plot(tokens, 'b-')
    for i in range(len(tokens)):
        if i < len(labels) and labels[i]:
            plt.text(i, 0, labels[i], color='r', fontsize=12, ha='center',␣
 ↪va='bottom', rotation=45)
    plt.title('Named Entities')
    plt.xticks(rotation=45)
    plt.show()
```

[13] :
```
tokens_pos_tags = df['tokens_pos_tags'][0]
named_entities_spacy = df['named_entities_spacy'][0]
```

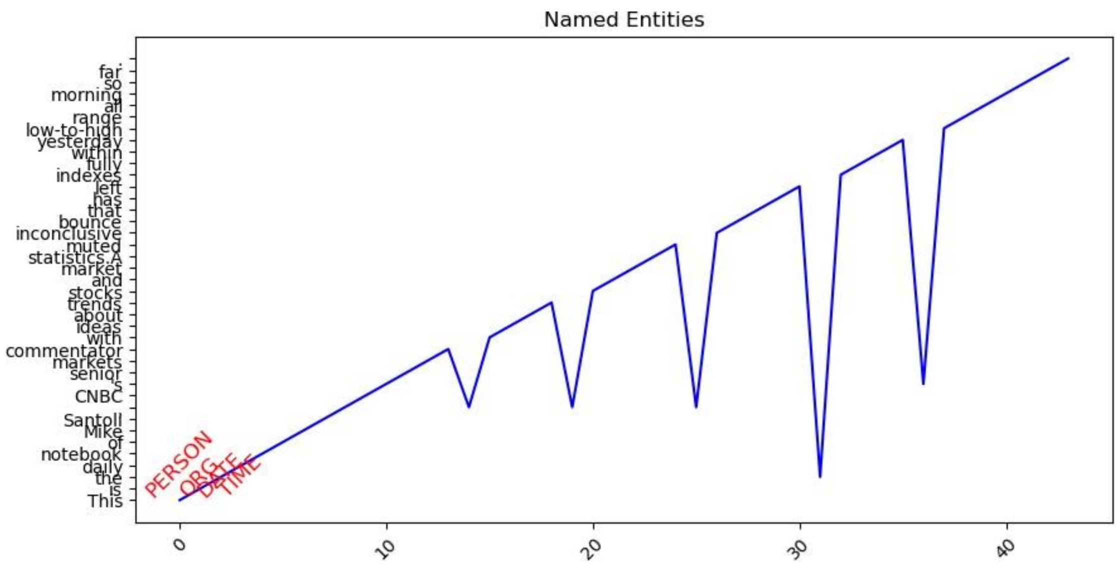
[14] :
```
tokens = [token[0] for token in tokens_pos_tags]
if named_entities_spacy:
    labels = [label[1] if len(label) > 1 else None for label in␣
 ↪named_entities_spacy]
else:
    labels = []
```

[15] :
```
print("Tokens:", tokens)
print("Labels:", labels)
```

```
Tokens: ['This', 'is', 'the', 'daily', 'notebook', 'of', 'Mike', 'Santoli', ',',
'CNBC', "'s", 'senior', 'markets', 'commentator', ',', 'with', 'ideas', 'about',
'trends', ',', 'stocks', 'and', 'market', 'statistics.A', 'muted', ',',
'inconclusive', 'bounce', 'that', 'has', 'left', 'the', 'indexes', 'fully',
'within', 'yesterday', "'s", 'low-to-high', 'range', 'all', 'morning', 'so',
'far', '.']
Labels: ['PERSON', 'ORG', 'DATE', 'TIME']
```

[16] :
```
if labels:
    visualize_named_entities_spacy(tokens, labels)
```

```
else:
    print("No named entities found in the text.")
```



[17] :
```
X = df['description']
y = df['title']
```

[18] :
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)
```

[19] :
```
def spacy_features(text):
    doc = nlp(text)
    features = {}
    for ent in doc.ents:
        features[ent.text] = 1
    return features
```

[20] :
```
vectorizer = DictVectorizer()
X_train_feats = vectorizer.fit_transform(spacy_features(text) for text in␣
↪X_train)
X_test_feats = vectorizer.transform(spacy_features(text) for text in X_test)
```

[21] :
```
classifier = LogisticRegression(max_iter=1000)
classifier.fit(X_train_feats, y_train)
```

[21] :
```
LogisticRegression(max_iter=1000)
```

[22] :
```
y_pred = classifier.predict(X_test_feats)
```

```
[23]: print(classification_report(y_test, y_pred))
```

```
                                                                              precision
recall    f1-score    support

                                     Did EA Bust the Social Gaming Bubble?        0.00
0.00        0.00        1.0
                           'Ex-sector' ETFs offer new way to bet on stocks        0.00
0.00        0.00        1.0
                                                                            'Powerful
and dangerous' Hurricane Ida is on the verge of landfall in Louisiana
0.00        0.00        0.00        1.0
                                                                 10-year Treasury yield
falls to 0.8% as investors return to safety amid pause in stock rally        0.00
0.00        0.00        1.0
                                                              22. Hexadite        0.00
0.00        0.00        0.0
                                                            22. SimpliVity        0.00
0.00        0.00        1.0
                                              5 Stocks Insiders Love Right Now        0.00
0.00        0.00        1.0
                                               A Chat With GSI Commerce's Rubin        0.00
0.00        0.00        1.0
                                                                       A.M. Best
Assigns Rating to WellPoint, Inc.'s New Senior Convertible Debentures        0.00
0.00        0.00        1.0
                        Amazon gives shoppers a glimpse at its Prime Day deals        0.00
0.00        0.00        1.0
                                              An Interview with Richard Fisher        0.00
0.00        0.00        0.0
     Apple's $1 billion data center gets Irish High Court green light        0.00
0.00        0.00        1.0
                                 Art Cashin remembers market reaction to JFK        0.00
0.00        0.00        1.0
                                                                    As Trump readies
for inauguration, environmental, climate organizations express concern
0.00        0.00        0.00        1.0
                         As it happened: China stocks stage late rally; up 4%        0.00
0.00        0.00        0.0
  BRIEF-Denison Mines offers private placement of flow-through shares        0.00
0.00        0.00        1.0
                             Banks within Wal-Mart stores collecting high fees        0.00
0.00        0.00        0.0
                            Bernanke Speech: Putting the Adults Back in Charge        0.00
0.00        0.00        0.0
                                                          Biden's national polling
lead shrinks slightly, but swing states show signs of trouble for Trump
```

```
0.00      0.00      0.0
                              Your first trade for Tuesday, January 5      0.00
0.00      0.00      1.0

                                                            accuracy
0.00    125.0
                                                           macro avg       0.00
0.00      0.00    125.0
                                                        weighted avg       0.00
0.00      0.00    125.0

C:\Users\c.sruthi\anaconda3\Lib\site-
packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\c.sruthi\anaconda3\Lib\site-
packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Recall
and F-score are ill-defined and being set to 0.0 in labels with no true samples.
Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\c.sruthi\anaconda3\Lib\site-
packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\c.sruthi\anaconda3\Lib\site-
packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Recall
and F-score are ill-defined and being set to 0.0 in labels with no true samples.
Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\c.sruthi\anaconda3\Lib\site-
packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\c.sruthi\anaconda3\Lib\site-
packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Recall
and F-score are ill-defined and being set to 0.0 in labels with no true samples.
Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

[24]:

```
person_counts = {}
organization_counts = {}
location_counts = {}
```

```
[25]: df['named_entities'] = df['description'].apply(extract_entities_spacy)
      df['named_entities_title'] = df['title'].apply(extract_entities_spacy)
```

```
[26]: for entities in df['named_entities']:
          for entity, label in entities:
              if label == 'PERSON':
                  person_counts[entity] = person_counts.get(entity, 0) + 1
              elif label == 'ORG':
                  organization_counts[entity] = organization_counts.get(entity, 0) + 1
              elif label == 'GPE' or label == 'LOC':
                  location_counts[entity] = location_counts.get(entity, 0) + 1
```

```
[27]: print("Most common persons:")
      print(sorted(person_counts.items(), key=lambda x: x[1], reverse=True)[:10])
      print("\nMost common organizations:")
      print(sorted(organization_counts.items(), key=lambda x: x[1], reverse=True)[:
      ↪10])
      print("\nMost common locations:")
      print(sorted(location_counts.items(), key=lambda x: x[1], reverse=True)[:10])
```

```
Most common persons:
[('Cramer', 108), ('Donald Trump', 70), ('Biden', 64), ('Pete Najarian', 55),
('Terranova', 52), ('Adami', 50), ('Romney', 47), ('Twitter', 46), ('Finerman',
40), ('WELCH', 37)]

Most common organizations:
[('CNBC', 423), ('Fed', 176), ('Trump', 163), ('Amazon', 130), ('Apple', 112),
('Reuters', 103), ('Google', 90), ('EU', 68), ('BAC', 50), ('GM', 45)]

Most common locations:
[('U.S.', 555), ('China', 260), ('Europe', 137), ('the United States', 85),
('U.K.', 73), ('Germany', 68), ('New York', 62), ('Washington', 61), ('Beijing',
59), ('US', 58)]
```

```
[28]: for index, row in df.iterrows():
          print(f"Article {index + 1}:")
          print("Named Entities in Title:")
          for entity, label in row['named_entities_title']:
              print(f"Entity: {entity}, Label: {label}")
          print("\nNamed Entities in Description:")
          for entity, label in row['named_entities']:
              print(f"Entity: {entity}, Label: {label}")
          print("\n")
```

```
Article 1:
Named Entities in Title:
Entity: Santoli, Label: PERSON
```

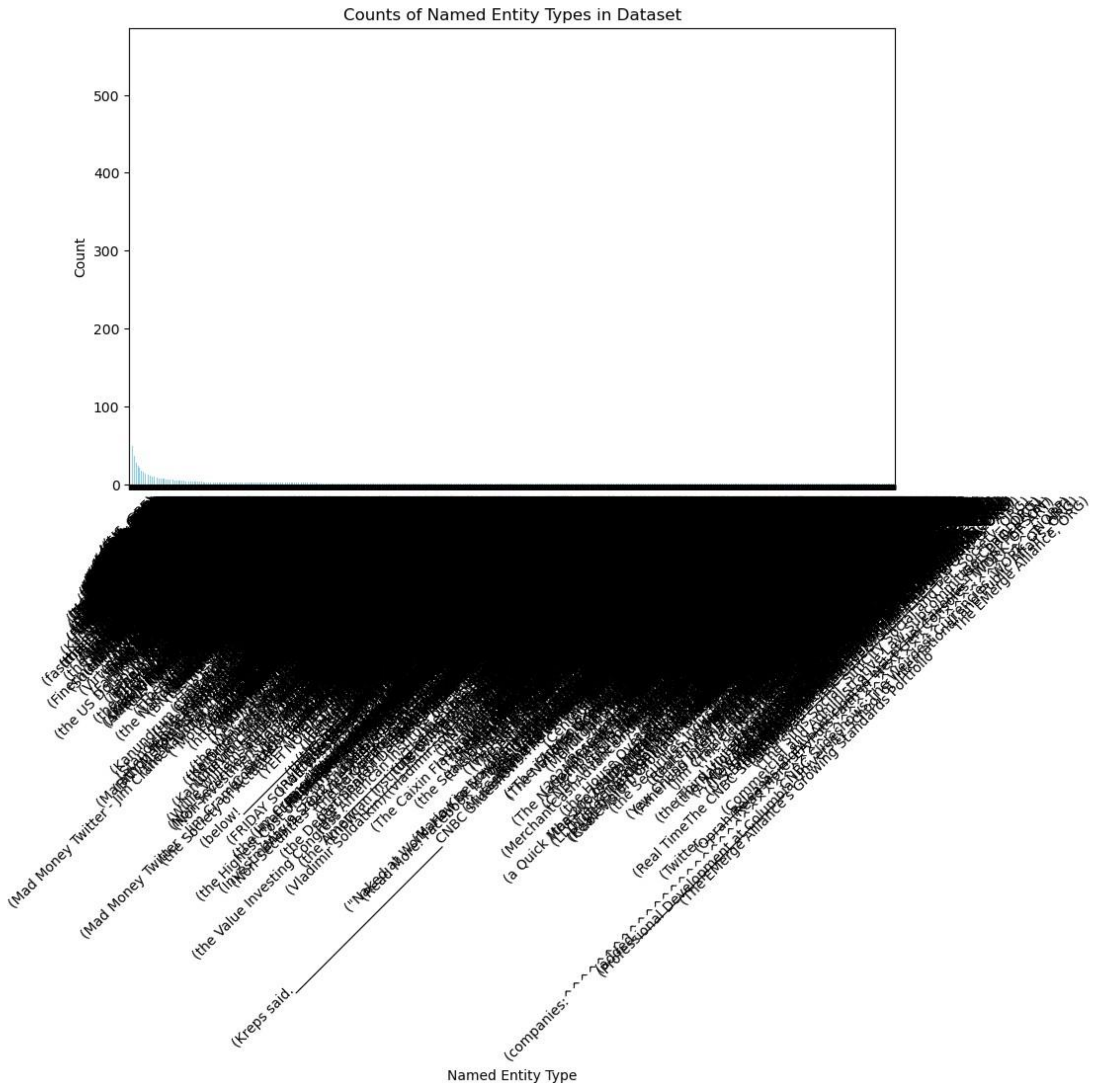
```
Entity: Wednesday, Label: DATE
Entity: September, Label: DATE
Entity: the fourth quarter, Label: DATE

Named Entities in Description:
Entity: Mike Santoli, Label: PERSON
Entity: CNBC, Label: ORG
Entity: yesterday, Label: DATE
Entity: all morning, Label: TIME


Article 2:
Named Entities in Title:
Entity: Brexit, Label: PERSON

Named Entities in Description:


Article 3:
Named Entities in Title:
Entity: Italy, Label: GPE

Named Entities in Description:


Article 4:
Named Entities in Title:
Entity: US, Label: GPE
Entity: GM, Label: ORG

Named Entities in Description:
Entity: US, Label: GPE
Entity: $13.4 billion, Label: MONEY
Entity: General Motors, Label: ORG
Entity: CNBC.The, Label: ORG
Entity: GM, Label: ORG
Entity: the United Auto Workers, Label: ORG
Entity: UAW, Label: ORG
Entity: GM, Label: ORG
Entity: Obama, Label: PERSON
Entity: UAW, Label: ORG
Entity: US, Label: GPE
Entity: GM, Label: ORG
Entity: up to $13.4 billion, Label: MONEY
Entity: December, Label: DATE
Entity: UAW, Label: ORG
Entity: New York Auto, Label: EVENT
Entity: the White House, Label: ORG
```

```
Entity: 1.7793, Label: MONEY
Entity: January 9, Label: DATE
Entity: some 11.71 cents, Label: MONEY
Entity: Fast Money's, Label: PERSON
Entity: Joe Terranova, Label: PERSON
Entity: minus 2 dollars, Label: MONEY
Entity: 7 bucks, Label: MONEY
Entity: Terranova, Label: PERSON
Entity: Own Valero, Label: PRODUCT
Entity: Sunoco, Label: ORG
Entity: less than half, Label: CARDINAL
Entity: July, Label: DATE
Entity: 4.11, Label: MONEY
Entity: fastmoney-web@cnbc.com, Label: PERSON
Entity: the Rapid Recap, Label: ORG
```

[29] :
```
all_entities = df['named_entities_title'].sum() + df['named_entities'].sum()
```

[30] :
```
entity_counts = pd.Series(all_entities).value_counts()
```

[31] :
```
plt.figure(figsize=(10, 6))
entity_counts.plot(kind='bar', color='skyblue')
plt.title('Counts of Named Entity Types in Dataset')
plt.xlabel('Named Entity Type')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```


Counts of Named Entity Types in Dataset
Count
Named Entity Type
0
100
200
300
400
500
(Mad Money Twitter - Jim Cramer
(Kreps said.
(the Value Investing Congress
(companies:
(The Emerge Alliance's Growing Standards Portfolio
(The Emerge Alliance, ORG)