Named Entity Recognition (NER)

Named Entity Recognition is the most important or I would say the starting step in Information Retrieval. Information Retrieval is the technique to extract important and useful information from unstructured raw text documents. Named Entity Recognition NER works by locating and identifying the named entities present in unstructured text into the standard categories such as person names, locations, organizations, time expressions, quantities, monetary values, percentage, codes etc. Spacy comes with an extremely fast statistical entity recognition system that assigns labels to contiguous spans of tokens.

Spacy provides option to add arbitrary classes to entity recognition system and update the model to even include the new examples apart from already defined entities within model.

Spacy has the ‘ner’ pipeline component that identifies token spans fitting a predetermined set of named entities. These are available as the ‘ents’ property of a Doc object.

In [ ]:
```
# !pip install spacy
```

In [ ]:
```
# Perform standard imports
import spacy
nlp = spacy.load('en_core_web_sm') #en_core_web_sm is a small English pipeli
```

In [ ]:
```
#Write a function to display basic entity info:
def show_ents(doc):
    if doc.ents:
        for ent in doc.ents:
            print(ent.text+' - ' +str(ent.start_char) +' - '+ str(ent.end_ch
                  ' - '+ent.label_+ ' - '+str(spacy.explain(ent.label_)))
    else:
        print('No named entities found.')
```

In [ ]:
```
doc1 = nlp("Apple is looking at buying U.K. startup for $1 billion")

show_ents(doc1)
```

```
Apple - 0 - 5 - ORG - Companies, agencies, institutions, etc.
U.K. - 27 - 31 - GPE - Countries, cities, states
$1 billion - 44 - 54 - MONEY - Monetary values, including unit
```

Here we see tokens combine to form the entities `$1 billion`.

Text	Start	End	Label	Description
Apple	0	5	`ORG`	Companies, agencies, institutions.

Text	Start	End	Label	Description
U.K.	27	31	GPE	Geopolitical entity, i.e. countries, cities, states.
$1 billion	44	54	MONEY	Monetary values, including unit.

In [ ]:
```
doc2 = nlp(u'May I go to Washington, DC next May to see the Washington Monum

show_ents(doc2)
```

```
Washington - 12 - 22 - GPE - Countries, cities, states
next May - 27 - 35 - DATE - Absolute or relative dates or periods
the Washington Monument - 43 - 66 - ORG - Companies, agencies, institutions,
etc.
```

Here we see tokens combine to form the entities `next May` and `the Washington Monument`

Entity Annotations

`Doc.ents` are token spans with their own set of annotations.

`ent.text`	The original entity text
`ent.label`	The entity type's hash value
`ent.label_`	The entity type's string description
`ent.start`	The token span's *start* index position in the Doc
`ent.end`	The token span's *stop* index position in the Doc
`ent.start_char`	The entity text's *start* index position in the Doc
`ent.end_char`	The entity text's *stop* index position in the Doc

In [ ]:
```
doc3 = nlp(u'Can I please borrow 500 dollars from you to buy some Microsoft

for ent in doc3.ents:
    print(ent.text, ent.label_)
```

```
500 dollars MONEY
Microsoft ORG
```

Accessing Entity Annotations

The standard way to access entity annotations is the doc.ents property, which produces a sequence of Span objects. The entity type is accessible either as a hash value using **ent.label** or as a string using **ent.label_**.

The Span object acts as a sequence of tokens, so you can iterate over the entity or index into it. You can also get the text form of the whole entity, as though it were a single token.

You can also access token entity annotations using the token.ent_iob and token.ent_type attributes. token.ent_iob indicates whether an entity starts, continues or ends on the tag. If no entity type is set on a token, it will return an empty string.

```
In [ ]: doc = nlp("San Francisco considers banning sidewalk delivery robots")

        # document level
        for e in doc.ents:
            print(e.text, e.start_char, e.end_char, e.label_)
        # OR
        ents = [(e.text, e.start_char, e.end_char, e.label_) for e in doc.ents] #in
        print(ents)

        # token level
        # doc[0], doc[1] ...will have tokens stored.

        ent_san = [doc[0].text, doc[0].ent_iob_, doc[0].ent_type_]
        ent_francisco = [doc[1].text, doc[1].ent_iob_, doc[1].ent_type_]
        print(ent_san)
        print(ent_francisco)
```

```
San Francisco 0 13 GPE
[('San Francisco', 0, 13, 'GPE')]
['San', 'B', 'GPE']
['Francisco', 'I', 'GPE']
```

IOB SCHEME

I – Token is inside an entity.

O – Token is outside an entity.

B – Token is the beginning of an entity.

Text	ent_iob	ent_iob_	ent_type_	Description
San	`3`	`B`	`"GPE"`	beginning of an entity
Francisco	`1`	`I`	`"GPE"`	inside an entity
considers	`2`	`O`	`""`	outside an entity
banning	`2`	`O`	`""`	outside an entity
sidewalk	`2`	`O`	`""`	outside an entity
delivery	`2`	`O`	`""`	outside an entity
robots	`2`	`O`	`""`	outside an entity

Note: In the above example only `San Francisco` is recognized as named entity. hence rest of the tokens are described as outside the entity. And in `San Francisco` `San` is the starting of the entity and `Francisco` is inside the entity.

GPE==> Geopolitical Entity

NER Tags

Tags are accessible through the `.label_` property of an entity.

TYPE	DESCRIPTION	EXAMPLE
`PERSON`	People, including fictional.	*Fred Flintstone*
`NORP`	Nationalities or religious or political groups.	*The Republican Party*
`FAC`	Buildings, airports, highways, bridges, etc.	*Logan International Airport, The Golden Gate*
`ORG`	Companies, agencies, institutions, etc.	*Microsoft, FBI, MIT*
`GPE`	Countries, cities, states.	*France, UAR, Chicago, Idaho*
`LOC`	Non-GPE locations, mountain ranges, bodies of water.	*Europe, Nile River, Midwest*
`PRODUCT`	Objects, vehicles, foods, etc. (Not services.)	*Formula 1*
`EVENT`	Named hurricanes, battles, wars, sports events, etc.	*Olympic Games*
`WORK_OF_ART`	Titles of books, songs, etc.	*The Mona Lisa*
`LAW`	Named documents made into laws.	*Roe v. Wade*
`LANGUAGE`	Any named language.	*English*
`DATE`	Absolute or relative dates or periods.	*20 July 1969*
`TIME`	Times smaller than a day.	*Four hours*
`PERCENT`	Percentage, including "%".	*Eighty percent*
`MONEY`	Monetary values, including unit.	*Twenty Cents*
`QUANTITY`	Measurements, as of weight or distance.	*Several kilometers, 55kg*
`ORDINAL`	"first", "second", etc.	*9th, Ninth*
`CARDINAL`	Numerals that do not fall under another type.	*2, Two, Fifty-two*

User Defined Named Entity and Adding it to a Span

Normally we would have spaCy build a library of named entities by training it on several samples of text.
Sometimes, we want to assign specific token a named entity whic is not recognized by the trained spacy model. We can do this as shown in below code.

Example1

In [ ]:
```
doc = nlp(u'Tesla to build a U.K. factory for $6 million')

show_ents(doc)
```

```
U.K. - 17 - 21 - GPE - Countries, cities, states
$6 million - 34 - 44 - MONEY - Monetary values, including unit
```

Right now, spaCy does not recognize "Tesla" as a company.

In [ ]:
```
from spacy.tokens import Span
```

In [ ]:
```
# Get the hash value of the ORG entity label
ORG = doc.vocab.strings[u'ORG']

# Create a Span for the new entity
new_ent = Span(doc, 0, 1, label=ORG)

# Add the entity to the existing Doc object
doc.ents = list(doc.ents) + [new_ent]
```

In the code above, the arguments passed to `Span()` are:

- `doc` - the name of the Doc object
- `0` - the *start* index position of the token in the doc
- `1` - the *stop* index position (exclusive) in the doc
- `label=ORG` - the label assigned to our entity

In [ ]:
```
show_ents(doc)
```

```
Tesla - 0 - 5 - ORG - Companies, agencies, institutions, etc.
U.K. - 17 - 21 - GPE - Countries, cities, states
$6 million - 34 - 44 - MONEY - Monetary values, including unit
```

Example2

In [ ]:
```
doc = nlp("fb is hiring a new vice president of global policy")
ents = [(e.text, e.start_char, e.end_char, e.label_) for e in doc.ents]
print('Before', ents)
#the model didn't recognise "fb" as an entity :(

fb_ent = Span(doc, 0, 1, label="ORG") # create a Span for the new entity
doc.ents = list(doc.ents) + [fb_ent]

ents = [(e.text, e.start_char, e.end_char, e.label_) for e in doc.ents]
```

```
print('After', ents)
# [('fb', 0, 2, 'ORG')]
```

```
Before []
After [('fb', 0, 2, 'ORG')]
```

Visualizing NER

In [ ]:
```
# Import the displaCy library
from spacy import displacy
```

In [ ]:
```
text = "When S. Thrun started working on self driving cars at Google in 2007
few people outside of the company took him serious"
doc = nlp(text)
displacy.render(doc, style="ent", jupyter=True)
```

When S. Thrun **PERSON** started working on self driving cars at Google **ORG** in 2007 **DATE** few people outside of the company took him serious

In [ ]:
```
text = """Clearview AI, a New York-headquartered facial recognition company,

Over the last few years, the firm has collected images from the web and soci

The Information Commission’s Office said Monday that the company has breache

The ICO has ordered Clearview to delete data it has on U.K. residents and ba

Clearview writes on its website that it has collected more than 20 billion f

Clearview’s platform allows law enforcement agencies to upload a photo of ar

John Edwards, the U.K.’s information commissioner, said in a statement: “The

He added that people expect their personal information to be respected, rega

doc = nlp(text)

displacy.render(doc, style='ent', jupyter=True)
```

Clearview AI ORG , a New York GPE -headquartered facial recognition company, has been fined £7.5 million MONEY ($9.4 million MONEY) by a U.K. GPE privacy regulator.Over the last few years DATE , the firm has collected images from the web and social media of people in Britain GPE and elsewhere to create a global online database that can be used by law enforcement for facial recognition. The Information Commission's ORG Office said Monday DATE that the company has breached U.K. GPE data protection laws.The ICO ORG has ordered Clearview ORG to delete data it has on U.K. GPE residents and banned it from collecting any more. Clearview PERSON writes on its website that it has collected more than 20 billion CARDINAL facial images of people around the world. It collects publicly posted images from social media platforms like Facebook ORG and Instagram NORP , as well as news media, mugshot websites and other open sources. It does so without informing the individuals or asking for their consent. Clearview PERSON 's platform allows law enforcement agencies to upload a photo of an individual and try to match it to photos that are stored in Clearview PERSON 's database. John Edwards PERSON , the U.K. GPE 's information commissioner, said in a statement: "The company not only enables identification of those people, but effectively monitors their behavior and offers it as a commercial service. That is unacceptable." He added that people expect their personal information to be respected, regardless of where in the world their data is being used.

Visualizing Sentences Line by Line

In [ ]:
```
for sent in doc.sents:
    displacy.render(nlp(sent.text), style='ent', jupyter=True)
```

Clearview AI ORG , a New York GPE -headquartered facial recognition company, has been fined £7.5 million MONEY ($9.4 million MONEY) by a U.K. GPE privacy regulator.

Over the last few years DATE , the firm has collected images from the web and social media of people in Britain GPE and elsewhere to create a global online database that can be used by law enforcement for facial recognition.

The Information Commission ORG 's

Office said Monday DATE that the company has breached U.K. GPE data protection laws.

The ICO ORG has ordered Clearview ORG to delete data it has on U.K. GPE residents and banned it from collecting any more.

Clearview writes on its website that it has collected more than 20 billion CARDINAL facial images of people around the world.

It collects publicly posted images from social media platforms like Facebook ORG and Instagram NORP , as well as news media, mugshot websites and other open sources.

```
/usr/lib/python3.7/runpy.py:193: UserWarning: [W006] No entities to visualiz
e found in Doc object. If this is surprising to you, make sure the Doc was p
rocessed using a model that supports named entity recognition, and check the
`doc.ents` property manually if necessary.
  "__main__", mod_spec)
```

It does so without informing the individuals or asking for their consent.

Clearview's platform allows law enforcement agencies to upload a photo of an individual and try to match it to photos that are stored in Clearview ORG 's database.

John Edwards PERSON , the U.K. GPE 's information commissioner, said in a statement: "The company not only enables identification of those people, but effectively monitors their behavior and offers it as a commercial service.

```
/usr/lib/python3.7/runpy.py:193: UserWarning: [W006] No entities to visualiz
e found in Doc object. If this is surprising to you, make sure the Doc was p
rocessed using a model that supports named entity recognition, and check the
`doc.ents` property manually if necessary.
  "__main__", mod_spec)
```

That is unacceptable."

```
/usr/lib/python3.7/runpy.py:193: UserWarning: [W006] No entities to visualiz
e found in Doc object. If this is surprising to you, make sure the Doc was p
rocessed using a model that supports named entity recognition, and check the
`doc.ents` property manually if necessary.
  "__main__", mod_spec)
```

He added that people expect their personal information to be respected, regardless

of where in the world their data is being used.

Styling: customize color and effects

You can also pass background color and gradient options:

In [ ]:

```
options = {'ents': ['ORG', 'PRODUCT']}

displacy.render(doc, style='ent', jupyter=True, options=options)
```

Clearview AI ORG , a New York-headquartered facial recognition company, has been fined £7.5 million ($9.4 million) by a U.K. privacy regulator.Over the last few years, the firm has collected images from the web and social media of people in Britain and elsewhere to create a global online database that can be used by law enforcement for facial recognition. The Information Commission’s ORG Office said Monday that the company has breached U.K. data protection laws.The ICO ORG has ordered Clearview ORG to delete data it has on U.K. residents and banned it from collecting any more.Clearview writes on its website that it has collected more than 20 billion facial images of people around the world. It collects publicly posted images from social media platforms like Facebook ORG and Instagram, as well as news media, mugshot websites and other open sources. It does so without informing the individuals or asking for their consent.Clearview’s platform allows law enforcement agencies to upload a photo of an individual and try to match it to photos that are stored in Clearview’s database.John Edwards, the U.K.’s information commissioner, said in a statement: “The company not only enables identification of those people, but effectively monitors their behavior and offers it as a commercial service. That is unacceptable.” He added that people expect their personal information to be respected, regardless of where in the world their data is being used.

In [ ]:
```
colors = {'ORG': 'linear-gradient(90deg, #f2c707, #dc9ce7)', 'PRODUCT': 'rad
options = {'ents': ['ORG', 'PRODUCT'], 'colors':colors}
displacy.render(doc, style='ent', jupyter=True, options=options)
```

Clearview AI ORG , a New York-headquartered facial recognition company, has been fined £7.5 million ($9.4 million) by a U.K. privacy regulator.Over the last few years, the firm has collected images from the web and social media of people in Britain and elsewhere to create a global online database that can be used by law enforcement for facial recognition. The Information Commission’s ORG Office said Monday that the company has breached U.K. data protection laws.The ICO ORG has ordered Clearview ORG to delete data it has on U.K. residents and banned it from collecting any more.Clearview writes on its website that it has collected more than 20 billion facial images of people around the world. It collects publicly posted images from social media platforms like Facebook ORG and Instagram, as well as news media, mugshot websites and other open sources. It does so without informing the individuals or asking for their consent.Clearview’s platform allows law enforcement agencies to upload a photo of an individual and try to match it to photos that are stored in Clearview’s database.John Edwards, the U.K.’s information commissioner, said in a statement: “The company not only enables identification of those people, but effectively monitors their behavior and offers it as a commercial service. That is unacceptable.” He added that people expect their personal information to be respected, regardless of where in the world their data is being used.

In [ ]:
```
colors = {'ORG':'linear-gradient(90deg,#aa9cde,#dc9ce7)','PRODUCT':'radial-g
options = {'ent':['ORG','PRODUCT'],'colors':colors}
displacy.render(doc,style='ent',jupyter=True,options=options)
```

Clearview AI ORG , a New York GPE -headquartered facial recognition company, has been fined £7.5 million MONEY ($9.4 million MONEY) by a U.K. GPE privacy regulator.Over the last few years DATE , the firm has collected images from the web and social media of people in Britain GPE and elsewhere to create a global online database that can be used by law enforcement for facial recognition. The Information Commission's ORG Office said Monday DATE that the company has breached U.K. GPE data protection laws.The ICO ORG has ordered Clearview ORG to delete data it has on U.K. GPE residents and banned it from collecting any more. Clearview PERSON writes on its website that it has collected more than 20 billion CARDINAL facial images of people around the world. It collects publicly posted images from social media platforms like Facebook ORG and Instagram NORP , as well as news media, mugshot websites and other open sources. It does so without informing the individuals or asking for their consent. Clearview PERSON 's platform allows law enforcement agencies to upload a photo of an individual and try to match it to photos that are stored in Clearview PERSON 's database. John Edwards PERSON , the U.K. GPE 's information commissioner, said in a statement: "The company not only enables identification of those people, but effectively monitors their behavior and offers it as a commercial service. That is unacceptable." He added that people expect their personal information to be respected, regardless of where in the world their data is being used.

Stude Assignment

Adding Named Entities to All Matching Spans

What if we want to tag *all* occurrences of a token? In this section we show how to use the PhraseMatcher to identify a series of spans in the Doc:

In [ ]:
```
doc = nlp(u'Our company plans to introduce a new vacuum cleaner. '
          u'If successful, the vacuum cleaner will be our first product.')
```

```
show_ents(doc)
```

In [ ]:
```
# Import PhraseMatcher and create a matcher object:
from spacy.matcher import PhraseMatcher
matcher = PhraseMatcher(nlp.vocab)
```

In [ ]:
```
# Create the desired phrase patterns:
phrase_list = ['vacuum cleaner', 'vacuum-cleaner']
phrase_patterns = [nlp(text) for text in phrase_list]
```

In [ ]:
```
# Apply the patterns to our matcher object:
matcher.add('newproduct', None, *phrase_patterns)

# Apply the matcher to our Doc object:
matches = matcher(doc)

# See what matches occur:
matches
```

In [ ]:
```
# Here we create Spans from each match, and create named entities from them:
from spacy.tokens import Span

PROD = doc.vocab.strings[u'PRODUCT']

new_ents = [Span(doc, match[1],match[2],label=PROD) for match in matches]
# match[1] contains the start index of the the token and match[2] the stop i

doc.ents = list(doc.ents) + new_ents
```

In [ ]:
```
show_ents(doc)
```

In [ ]:
```
doc = nlp(u'Originally priced at $29.50, the sweater was marked down to five

show_ents(doc)
```

In [ ]:
```
len([ent for ent in doc.ents if ent.label_=='MONEY'])
```

In [ ]:
```

```

In [ ]:
```

```

This notebook was converted with convert.ploomber.io