AnalyzingSentenceStructure

November 15, 2021

```
In [ ]: import markdown
```

1 My Jupyter Notebook

1.1 Installing necessary packages

```
In [ ]: # Installing packages
        #!pip install spacy
        #!pip install markdown
        #!python -m spacy download en_core_web_sm
```

1.2 Importing spacy and pandas

```
In [1]: import spacy
        import pandas as pd
```

1.3 Reading wiki_sentences dataset

Visit source

```
In [24]: candidate_sentences = pd.read_csv("wiki_sentences_v2.csv")
```

```
In [3]: # Getting the size of pandas dataframe and looking at the first few rows for sanity ch
        candidate_sentences.shape
        candidate_sentences['sentence'].sample(5)
```

```
Out[3]: 1749    some bollywood films have been widely apprecia...
        2541    m.ănight shyamalan has been asked numerous tim...
        623     a month later, zombie officially disbanded whi...
        1854    double super 8 film  is a 16ămm wide film but ...
        843     each year the afi awards honor the ten outstan...
        Name: sentence, dtype: object
```

1.3.1 Loading en_core_web_sm, which is a small English pipeline trained on written web text (blogs, news, comments), that includes vocabulary, syntax and entities.

```
In [4]: nlp = spacy.load('en_core_web_sm')
        doc = nlp("the drawdown process is governed by astm standard d823")
        for tok in doc:
          print(tok.text, "...", tok.dep_)
```

```
the ... det
drawdown ... compound
process ... nsubjpass
is ... auxpass
governed ... ROOT
by ... agent
astm ... compound
standard ... compound
d823 ... pobj
```

1.4 Spacy.explain

```
In [8]: spacy.explain("auxpass")
```

```
Out[8]: 'auxiliary (passive)'
```

1.5 Accessing token attributes

```
In [81]: doc = nlp("Airstrikes continued into the early hours of Monday morning in Gaza.")
         # Token texts
         [token.text for token in doc]
```

```
Out[81]: ['Airstrikes',
          'continued',
          'into',
          'the',
          'early',
          'hours',
          'of',
          'Monday',
          'morning',
          'in',
          'Gaza',
          '.']
```

1.6 Accessing spans

```
In [76]: doc = nlp("I argue that states built environments of conflict are material manifestat
         span = doc[2:4]
         span.text
```

```
Out[76]: 'that states'
```

1.7 Creating a span manually

```
In [71]: from spacy.tokens import Span
         # Create a Doc object
         doc = nlp("Airstrikes continued into the early hours of Monday morning in Gaza.")
         span = Span(doc, 4, 11, label="NORP")
         span.text
```

```
Out[71]: 'early hours of'
```

1.8 Part-of-speech tags

```
In [75]: doc = nlp("Airstrikes continued into the early hours of Monday morning in Gaza.")
         # Coarse-grained part-of-speech tags
         [token.pos_ for token in doc]
         # Fine-grained part-of-speech tags
         [token.tag_ for token in doc]
```

```
Out[75]: ['NNS', 'VBD', 'IN', 'DT', 'JJ', 'NNS', 'IN', 'NNP', 'NN', 'IN', 'NNP', '.']
```

```
In [72]: spacy.explain("NNP")
```

```
Out[72]: 'noun, proper singular'
```

1.9 Syntactic dependencies

```
In [53]: doc = nlp("Airstrikes continued into the early hours of Monday morning in Gaza.")
         # Dependency labels
         [token.dep_ for token in doc]
         [token.head.text for token in doc]
```

```
Out[53]: ['continued',
          'continued',
          'continued',
          'hours',
          'hours',
          'into',
          'hours',
          'morning',
          'of',
          'continued',
          'in',
          'continued']
```

1.10 Named entities

Bias within NLP labelling

Python library recognizing Israel's legitimacy as a state without doing the same for Palestine

```
In [51]: doc = nlp("Israel's new plan is to 'shrink,' not solve, the Palestinian conflict.")
         # Text and label of named entity span
         [(ent.text, ent.label_) for ent in doc.ents]

Out[51]: [('Israel', 'GPE'), ('Palestinian', 'NORP')]
```

```
In [ ]: #spacy.explain("NORP") -->'Nationalities or religious or political groups'
        #spacy.explain("GPE") --> 'Countries, cities, states'
```

1.11 Syntax iterators – Sentences

```
In [49]: doc = nlp("This is more than just eavesdropping, its terrifying. The spyware takes co
         # doc.sents is a generator that yields sentence spans
         [sent.text for sent in doc.sents]

Out[49]: ['This is more than just eavesdropping, its terrifying.',
          'The spyware takes complete control over the phone.',
          'It can make calls to anybody, send messages and it can download content, Aboudi tol
```

1.12 Base noun phrases

```
In [48]: doc = nlp("Ghassan Halaika, a Jerusalem-based researcher with Al Haq, recently notice
         # doc.noun_chunks is a generator that yields spans
         [chunk.text for chunk in doc.noun_chunks]

Out[48]: ['Ghassan Halaika',
          'a Jerusalem-based researcher',
          'Al Haq',
          'strange things',
          'his phone']
```

1.13 Label Explanations

```
In [52]: #Exploring label explanations in spacy
         spacy.explain("NORP")
         # 'Nationalities or religious or political groups'
         #  spacy.explain("GPE")
         # 'Countries, cities, states'

Out[52]: 'Nationalities or religious or political groups'
```

2 Visualizing dependencies from various news sources

2.1 Importing displacy for visualizing sentence structure

```
In [38]: from spacy import displacy
```

2.2 Sentence from Al Jazeera article

Palestinian rights activists defiant over Israeli spyware hacks

```
In [44]: doc = nlp("The rights groups deny any links to the PFLP and Israel has failed to publi
         displacy.render(doc, style="dep")
```

```
<IPython.core.display.HTML object>
```

2.3 Sentence from NYT article

Conflict Spirals Across Israel and the Palestinian Territories

```
In [41]: doc = nlp("An American envoy landed in Israel for cease-fire talks with Palestinians a
         displacy.render(doc, style="dep")
```

```
<IPython.core.display.HTML object>
```

2.4 Visualize named entities

```
In [80]: doc = nlp("Internment, Torture and Pro-government Militia in Northern Ireland (with Sa
         displacy.render(doc, style="ent")
```

```
<IPython.core.display.HTML object>
```