**Generating metadata using SpaCy and CoreNLP**

# Using spaCy

**Different types of Named Entities**

**PERSON** People, including fictional.

**NORP** Nationalities or religious or political groups.

**FAC** Buildings, airports, highways, bridges, etc.

**ORG** Companies, agencies, institutions, etc.

**GPE** Countries, cities, states.

**LOC** Non-GPE locations, mountain ranges, bodies of water.

**PRODUCT** Objects, vehicles, foods, etc. (Not services.)

**EVENT** Named hurricanes, battles, wars, sports events, etc.

**WORK\_OF\_ART** Titles of books, songs, etc.

**LAW** Named documents made into laws.

**LANGUAGE** Any named language.

**DATE** Absolute or relative dates or periods.

**TIME** Times smaller than a day.

**PERCENT** Percentage, including "%".

**MONEY** Monetary values, including unit.

**QUANTITY** Measurements, as of weight or distance.

**ORDINAL** "first", "second", etc.

**CARDINAL** Numerals that do not fall under another type.

import spacy

df3 = pd.DataFrame()

nlp = spacy.load("en\_core\_web\_sm")

for i in range(0,10):

text=df.loc[i,"event\_summary"].title()

doc = nlp(text)

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

for ent in doc.ents:

# print(ent.text, ent.label\_)

for i,token in enumerate(doc):

# token=token.capitalize()

df3.loc[i,'text'] = token.text

df3.loc[i,'lemma\_'] = token.lemma\_

df3.loc[i,'pos\_'] = token.pos\_

df3.loc[i,'tag\_'] = token.tag\_

df3.loc[i,'dep\_'] = token.dep\_

df3.loc[i,'shape\_'] = token.shape\_

df3.loc[i,'is\_alpha'] = token.is\_alpha

df3.loc[i,'is\_stop'] = token.is\_stop

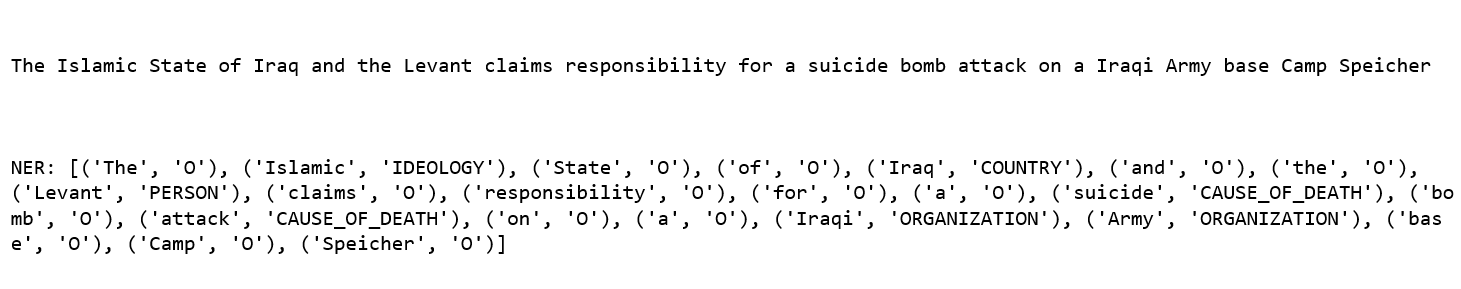
**Results**

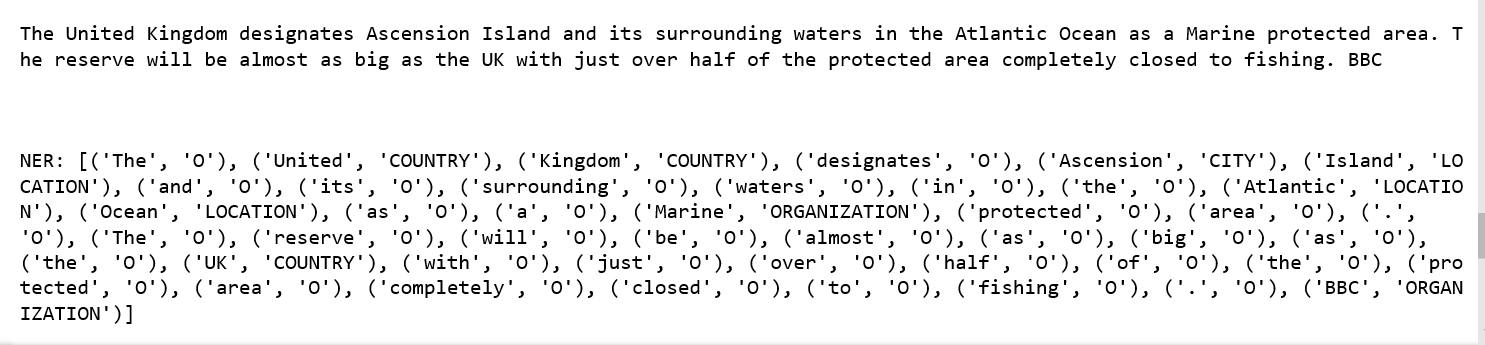


**If country first word is not capitalize it won’t identify it but corenlp does**

# Using CoreNLP

**Results:**





# Generating more NER using spaCy

**`?**

# Word Embedding

Two [Python](https://www.python.org/) natural language processing (NLP) libraries are mentioned here: (Pre-trained models)

1. [Spacy](https://spacy.io/) is a [natural language processing (NLP)](https://en.wikipedia.org/wiki/Natural-language_processing) library for Python designed to have fast performance, and with word embedding models built in, it’s perfect for a quick and easy start.
2. [Gensim](https://radimrehurek.com/gensim/) is a [topic modelling library](https://en.wikipedia.org/wiki/Topic_model) for Python that provides access to Word2Vec and other word embedding algorithms for training, and it also allows pre-trained word embeddings that you can download from the internet to be loaded.

<https://www.shanelynn.ie/word-embeddings-in-python-with-spacy-and-gensim/>