Question 1. Create a python program to find it out the Named entity recognition(NER) from t sentence using spacy and visualizing those parts in sentence.

In [5]:

```
import spacy
from spacy import displacy
```

In [2]:

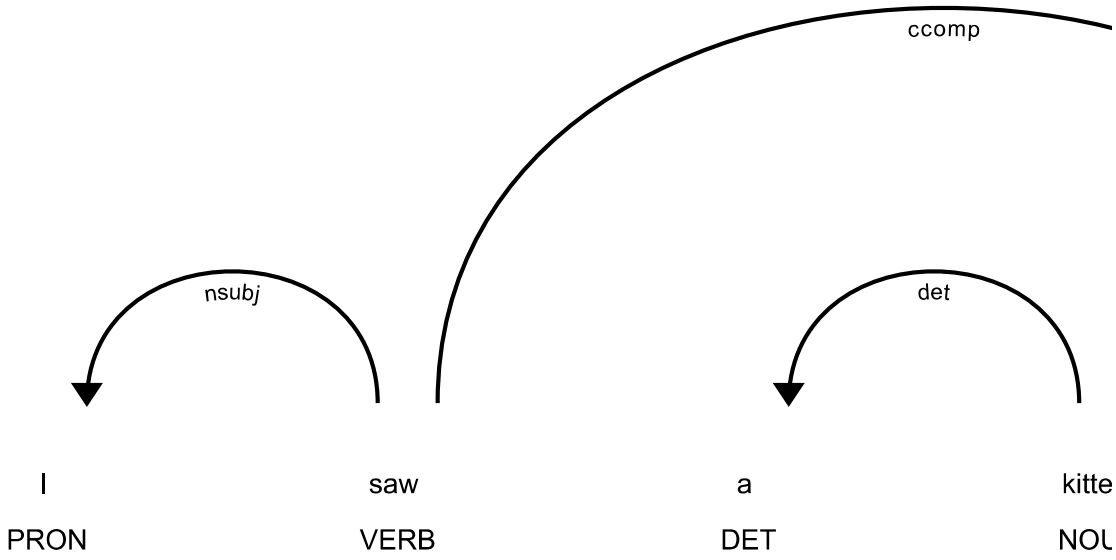
```
nlp = spacy.load("en_core_web_sm")
doc = nlp("I saw a kitten eating chicken in the kitchen")
#print(doc)
for token in doc:
    print(token.text, token.pos_, token.dep_)
```

```
I PRON nsubj
saw VERB ROOT
a DET det
kitten NOUN nsubj
eating VERB ccomp
chicken NOUN dobj
in ADP prep
the DET det
kitchen NOUN pobj
```

In [3]:

```
viz = displacy.serve(doc,style='dep')
viz
```

```
C:\Users\pc\Anaconda3\lib\site-packages\spacy\displacy\__init__.py:94: Use
arning: [W011] It looks like you're calling displacy.serve from within a J
yter notebook or a similar environment. This likely means you're already r
ning a local web server, so there's no need to make displaCy start another
ne. Instead, you should be able to replace displacy.serve with displacy.re
er to show the visualization.
  warnings.warn(Warnings.W011)
```




```
Using the 'dep' visualizer
Serving on http://0.0.0.0:5000 (http://0.0.0.0:5000) ...

Shutting down server on port 5000.
```

Question 2. Find out the similar words from following list using spacy library.

["ship car truck motor-bike jeep hagskdshd"]

In [1]:

```
import spacy.cli
spacy.cli.download("en_core_web_md")
import en_core_web_md
nlp = en_core_web_md.load()
```

```
⚠ Skipping model package dependencies and setting `--no-deps`. You don't s
m to
have the spaCy package itself installed (maybe because you've built from
source?), so installing the model dependencies would cause spaCy to be
downloaded, which probably isn't what you want. If the model package has o
er
dependencies, you'll have to install them manually.
✓ Download and installation successful
You can now load the model via spacy.load('en_core_web_md')
```

In [2]:

```
#nlp = spacy.load("en_core_web_md")
tokens = nlp('ship car truck motor-bike jeep hagskdshd')

for token11 in tokens:
    for token13 in tokens:
        print(token11.text, token13.text, token11.similarity(token13))
```

```
ship ship 1.0
ship car 0.2815587
ship truck 0.35367906
ship motor 0.2741221
ship - 0.030455196
ship bike 0.19740233
ship jeep 0.21036273

C:\Users\pc\Anaconda3\lib\site-packages\ipykernel_launcher.py:6: UserWarni
g: [W008] Evaluating Token.similarity based on empty vectors.


ship hagskdshd 0.0
car ship 0.2815587
car car 1.0
car truck 0.71134394
car motor 0.5655214
car - 0.1160374
car bike 0.53577304
car jeep 0.5652786
car hagskdshd 0.0
truck ship 0.35367906
truck car 0.71134394
truck truck 1.0
truck motor 0.52424634
truck - 0.059196077
truck bike 0.48927912
truck jeep 0.6284901
truck hagskdshd 0.0
motor ship 0.2741221
motor car 0.5655214
motor truck 0.52424634
motor motor 1.0
motor - 0.010074193
motor bike 0.46072814
motor jeep 0.40830103
motor hagskdshd 0.0
- ship 0.030455196
- car 0.1160374
- truck 0.059196077
- motor 0.010074193
- - 1.0
- bike 0.0737416
- jeep -0.01678796
- hagskdshd 0.0
bike ship 0.19740233
bike car 0.53577304
bike truck 0.48927912
bike motor 0.46072814
bike - 0.0737416
bike bike 1.0
```

```
bike jeep 0.42305312
bike hagskdshd 0.0
jeep ship 0.21036273
jeep car 0.5652786
jeep truck 0.6284901
jeep motor 0.40830103
jeep - -0.01678796
jeep bike 0.42305312
jeep jeep 1.0
jeep hagskdshd 0.0
hagskdshd ship 0.0
hagskdshd car 0.0
hagskdshd truck 0.0
hagskdshd motor 0.0
hagskdshd - 0.0
hagskdshd bike 0.0
hagskdshd jeep 0.0
hagskdshd hagskdshd 1.0
```

Question 3. Visualize sentences including parts of speech(pos) tagging and linguistic annota spacy library.

In [8]:

```
doc = nlp("Hello, My name is Aamir Iqbal.")
displacy.serve(doc,style='ent')
```

```
C:\Users\pc\Anaconda3\lib\site-packages\spacy\displacy\__init__.py:94: Use
arning: [W011] It looks like you're calling displacy.serve from within a J
yter notebook or a similar environment. This likely means you're already r
ning a local web server, so there's no need to make displaCy start another
ne. Instead, you should be able to replace displacy.serve with displacy.re
er to show the visualization.
  warnings.warn(Warnings.W011)
```

Hello, My name is Aamir Iqbal **PERSON** .

```
Using the 'ent' visualizer
Serving on http://0.0.0.0:5000 (http://0.0.0.0:5000) ...

Shutting down server on port 5000.
```

In [3]:

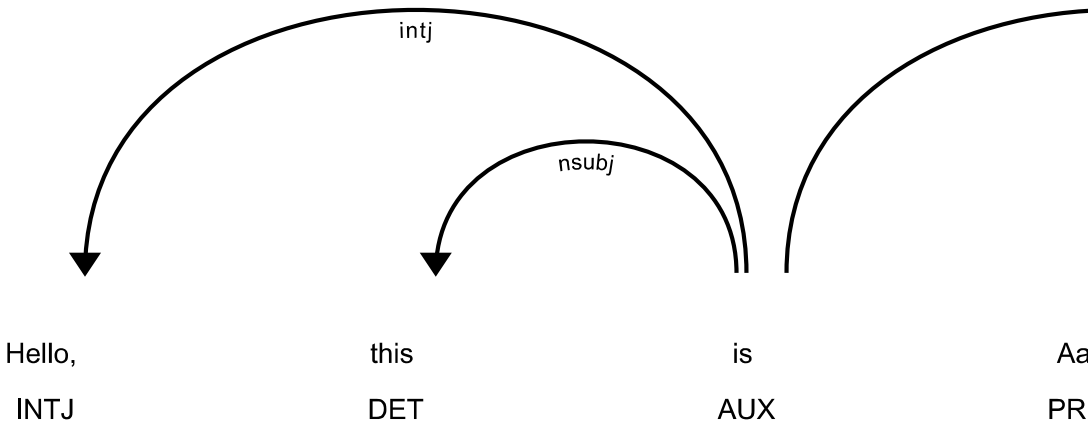
```
doc = nlp("Hello, this is Aamir Iqbal")
for token in doc:
    print(token.text, token.pos_, token.dep_)
```

```
Hello INTJ intj
, PUNCT punct
this DET nsubj
is AUX ROOT
Aamir PROPN compound
Iqbal PROPN attr
```

In [6]:

```
displacy.serve(doc,style='dep')
```

```
C:\Users\pc\Anaconda3\lib\site-packages\spacy\displacy\__init__.py:94: Use
arning: [W011] It looks like you're calling displacy.serve from within a J
yter notebook or a similar environment. This likely means you're already r
ning a local web server, so there's no need to make displaCy start another
ne. Instead, you should be able to replace displacy.serve with displacy.re
er to show the visualization.
  warnings.warn(Warnings.W011)
```




```
Using the 'dep' visualizer
Serving on http://0.0.0.0:5000 (http://0.0.0.0:5000) ...

Shutting down server on port 5000.
```

Question 4. What is Word vector? And, how could you convert words into vector?

Converting words to vectors, or word vectorization, is a natural language processing (NLP) p

process uses language models to map words into vector space. A vector space represents e
vector of real numbers. It also allows words with similar meanings have similar representatio

word vectors represent words as multidimensional numbers where similar words are mappe
points in geometric space. In simpler terms, a word vector is a row of real-valued numbers w
similar words have similar vectors. This means that words such as wheel and engine should
vectors to the word car (because of the similarity of their meanings), whereas the word bana
distant. Put differently, words that are used in a similar context will be mapped to a proximate

king - man + woman = queen

Word2vec

source : [https://www.hackdeploy.com/word2vec-explained-easily/ (https://www.hackdeploy.cc
explained-easily/)](https://www.hackdeploy.com/word2vec-explained-easily/)

Word2Vec was developed at Google by Tomas Mikolov, et al. and uses Neural Networks to l
embeddings.

The beauty with word2vec is that the vectors are learned by understanding the context in wh
The result is vectors in which words with similar meanings end up with a similar numerical re

This is a big deal and greatly improve machine learning models that utilize text as input.

For example, in a regular one-hot encoded Vector, all words end up with the same distance b
other, even though their meanings are completely different. In other words, information is los

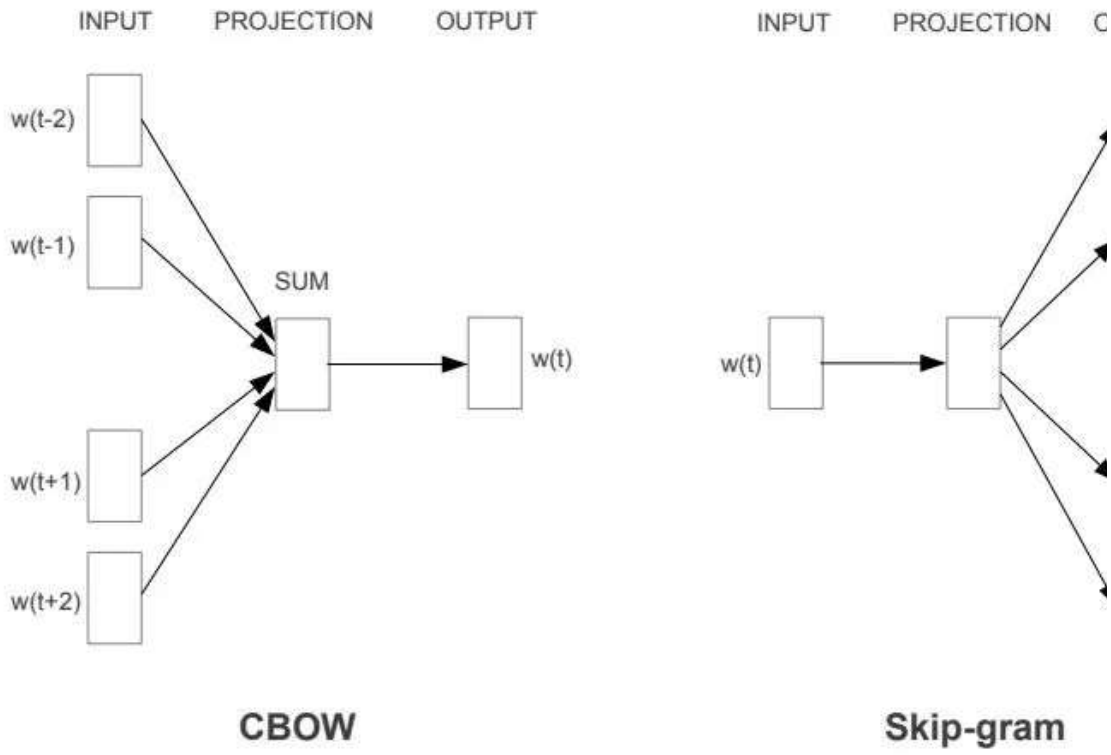
With word embeddings methods such as Word2Vec, the resulting vector does a better job of
context. For instance, cats and dogs are more similar than fish and sharks. This extra inform
difference in your machine learning algorithms.

To understand this better you need to learn about CBOW and Skip-Gram.

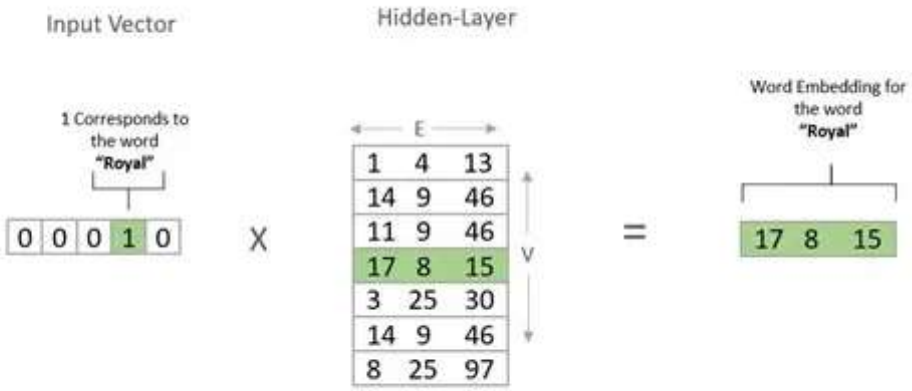
CBOW vs Skip Gram Word2Vec is composed of two different learning models, CBOW and S
stands for Continuous Bag of Words model.

Continuous Bag of Words (CBOW) model can be thought of as learning word embeddings b
to **predict a word given its context**.

Skip-Gram Model is the opposite, learning word embeddings by training a model to **predict**
word.



The word embedding is the hidden layer of the model. You can toss out the output layer, the
what you care about!



Question 5. Define Vocab, hashes and lexemes in your own word.

Vocab. It is a broad collection of corpus(vocabulary) share across the language.
whereas **lexeme** is just an entry in vocabulary it has not context, part of speech tagging or d
Hashes is typically used in memory management system. Hashes or address of any entity is

memory. It's a tough task to assign unique hashes os that there is no clashed of one entity tc help to be more efficient for example if you are using same variable over and over or assigni create any other memory it will just use the same hashes to refrence.

In [ ]: