# Implementación de una tarea de procesamiento del lenguaje natural

**Manuel Pasieka, Joaquín Delgado Fernández, Inés Heras**

En este trabajo, se va a utilizar la herramienta Natural Language Toolkit (NLTK) para extraer alguna información linguística sobre unas palabras propuestas

In [1]:
```
import nltk

nltk.download('wordnet')
from nltk.corpus import wordnet as wn
from nltk.wsd import lesk
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/joaquindelgado/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

## 1 Se obtienen los synsets de las palabras

In [2]:
```
synset_mosquito=wn.synsets('mosquito')
synset_tsetse=wn.synsets('tsetse')
synset_housefly=wn.synsets('housefly')
synset_spider=wn.synsets('spider')
synset_cockroach=wn.synsets('cockroach')
```

In [3]:
```
print("Synset de mosquito: "+str(synset_mosquito))
print("Synset de tsetse: "+str(synset_tsetse))
print("Synset de housefly: "+str(synset_housefly))
print("Synset de spider: "+str(synset_spider))
print("Synset de cockroach: "+str(synset_cockroach))
```

```
Synset de mosquito: [Synset('mosquito.n.01')]
Synset de tsetse: [Synset('tsetse_fly.n.01')]
Synset de housefly: [Synset('housefly.n.01')]
Synset de spider: [Synset('spider.n.01'), Synset('spider.n.02'), S
ynset('spider.n.03')]
Synset de cockroach: [Synset('cockroach.n.01')]
```

Se observa que Spider tiene varios synsets

## 2 Se obtienen las definiciones

In [4]:
```
words = ['mosquito', 'tsetse', 'housefly', 'spider', 'cockroach']
for word in words:
    synsets = wn.synsets(word)
    print('\nWord [%s] : Synsets (%d) [%s]' % (word, len(synsets),
synsets))
    for syn in synsets:
        print('>[%s] : Definition [%s]' % (syn, syn.definition()))
```

```
Word [mosquito] : Synsets (1) [[Synset('mosquito.n.01')]]
>[Synset('mosquito.n.01')] : Definition [two-winged insect whose f
emale has a long proboscis to pierce the skin and suck the blood o
f humans and animals]

Word [tsetse] : Synsets (1) [[Synset('tsetse_fly.n.01')]]
>[Synset('tsetse_fly.n.01')] : Definition [bloodsucking African fl
y; transmits sleeping sickness etc.]

Word [housefly] : Synsets (1) [[Synset('housefly.n.01')]]
>[Synset('housefly.n.01')] : Definition [common fly that frequents
human habitations and spreads many diseases]

Word [spider] : Synsets (3) [[Synset('spider.n.01'), Synset('spide
r.n.02'), Synset('spider.n.03')]]
>[Synset('spider.n.01')] : Definition [predatory arachnid with eig
ht legs, two poison fangs, two feelers, and usually two silk-spinn
ing organs at the back end of the body; they spin silk to make coc
oons for eggs or traps for prey]
>[Synset('spider.n.02')] : Definition [a computer program that pro
wls the internet looking for publicly accessible resources that ca
n be added to a database; the database can then be searched with a
search engine]
>[Synset('spider.n.03')] : Definition [a skillet made of cast iron
]

Word [cockroach] : Synsets (1) [[Synset('cockroach.n.01')]]
>[Synset('cockroach.n.01')] : Definition [any of numerous chiefly
nocturnal insects; some are domestic pests]
```

## 3 Desambiguación del sentido utilizando las definiciones

Se observa que la palabra Spider tiene varios Synsets. Por ello, se va a tratar de desambiguar utilizando el algoritmo de Lesk, y tratando el resto de palabras como si fuesen el contexto

In [5]:
```
context = ['mosquito', 'tsetse', 'housefly', 'cockroach']
synset_spider_def = lesk(context, 'spider', pos='n')
```

In [6]:
```
print("El synset seleccionado es "+str(synset_spider_def))
def_spider=wn.synset(synset_spider_def.name()).definition()
print("Su definición es: "+str(def_spider))
```

```
El synset seleccionado es Synset('spider.n.03')
Su definición es: a skillet made of cast iron
```

La definición no parece la correcta para el contexto dado

## 4 Desambiguación usando similaridad por longitud de camino

Ya que el método anterior no ha dado buen resultado, se va a utilizar la similaridad por el camino. La función path_similarity, da un scoring en función de la longitud del camino entre las dos palabras a comparar dentro de la jerarquía de synsets, siendo más similares cuanto más corto sea el camino.

In [7]:
```
synset_spider_path=''
max_similar=0
for x in synset_spider:
    similar_mosq=x.path_similarity(synset_mosquito[0])
    similar_tsetse=x.path_similarity(synset_tsetse[0])
    similar_fly=x.path_similarity(synset_housefly[0])
    similar_cock=x.path_similarity(synset_cockroach[0])
    total_similar=similar_mosq+similar_tsetse+similar_fly+similar_c
ock
    if(total_similar>max_similar):
        max_similar=total_similar
        synset_spider_path=x
```

In [8]:
```
print("El synset seleccionado es "+str(synset_spider_path)+" con un
a similaridad total de "+str(max_similar))
def_spider=wn.synset('spider.n.01').definition()
print("Su definición es: "+str(def_spider))
```

```
El synset seleccionado es Synset('spider.n.01') con una similarida
d total de 0.6190476190476191
Su definición es: predatory arachnid with eight legs, two poison f
angs, two feelers, and usually two silk-spinning organs at the bac
k end of the body; they spin silk to make cocoons for eggs or trap
s for prey
```

En este caso, el resultado si que es el correcto

## 5 Extracción de sinónimos

Vamos a extraer los sinónimos o lemas pertenecientes a cada synset.

In [9]:
```
#Nos quedamos con los synsets definitivos después de desambiguar
synset_mosquito=synset_mosquito[0]
synset_tsetse=synset_tsetse[0]
synset_housefly=synset_housefly[0]
synset_spider=synset_spider_path
synset_cockroach=synset_cockroach[0]
synsets_list=[synset_mosquito,synset_tsetse,synset_housefly,synset_
spider,synset_cockroach]
```

In [10]:
```
#Para cada uno de los synsets, mostramos los lemmas que lo componen
, que son sinónimos entre si
for syn in synsets_list:
    print ('El synset [%s] lo forman los lemas:\n%s\n' % (syn, syn.
lemma_names()))
```

```
El synset [Synset('mosquito.n.01')] lo forman los lemas:
['mosquito']

El synset [Synset('tsetse_fly.n.01')] lo forman los lemas:
['tsetse_fly', 'tsetse', 'tzetze_fly', 'tzetze', 'glossina']

El synset [Synset('housefly.n.01')] lo forman los lemas:
['housefly', 'house_fly', 'Musca_domestica']

El synset [Synset('spider.n.01')] lo forman los lemas:
['spider']

El synset [Synset('cockroach.n.01')] lo forman los lemas:
['cockroach', 'roach']
```

## 6 Hiperónimos

Se muestra toda la jerarquía de hiperónimos para todos los synsets

In [11]:
```
synsets_list=[synset_mosquito,synset_tsetse,synset_housefly,synset_
spider,synset_cockroach]
hps = [syn.hypernym_paths()[0] for syn in synsets_list]
for syn, hp in zip(synsets_list, hps):
    print('Hiperónimos de [%s]:\n%s\n\n' % (syn, hp))
```

```
Hiperónimos de [Synset('mosquito.n.01')]:
[Synset('entity.n.01'), Synset('physical_entity.n.01'), Synset('ob
ject.n.01'), Synset('whole.n.02'), Synset('living_thing.n.01'), Sy
nset('organism.n.01'), Synset('animal.n.01'), Synset('invertebrate
.n.01'), Synset('arthropod.n.01'), Synset('insect.n.01'), Synset('
dipterous_insect.n.01'), Synset('mosquito.n.01')]


Hiperónimos de [Synset('tsetse_fly.n.01')]:
[Synset('entity.n.01'), Synset('physical_entity.n.01'), Synset('ob
ject.n.01'), Synset('whole.n.02'), Synset('living_thing.n.01'), Sy
nset('organism.n.01'), Synset('animal.n.01'), Synset('invertebrate
.n.01'), Synset('arthropod.n.01'), Synset('insect.n.01'), Synset('
dipterous_insect.n.01'), Synset('fly.n.01'), Synset('tsetse_fly.n.
01')]


Hiperónimos de [Synset('housefly.n.01')]:
[Synset('entity.n.01'), Synset('physical_entity.n.01'), Synset('ob
ject.n.01'), Synset('whole.n.02'), Synset('living_thing.n.01'), Sy
nset('organism.n.01'), Synset('animal.n.01'), Synset('invertebrate
.n.01'), Synset('arthropod.n.01'), Synset('insect.n.01'), Synset('
dipterous_insect.n.01'), Synset('fly.n.01'), Synset('housefly.n.01
')]


Hiperónimos de [Synset('spider.n.01')]:
[Synset('entity.n.01'), Synset('physical_entity.n.01'), Synset('ob
ject.n.01'), Synset('whole.n.02'), Synset('living_thing.n.01'), Sy
nset('organism.n.01'), Synset('animal.n.01'), Synset('invertebrate
.n.01'), Synset('arthropod.n.01'), Synset('arachnid.n.01'), Synset
('spider.n.01')]


Hiperónimos de [Synset('cockroach.n.01')]:
[Synset('entity.n.01'), Synset('physical_entity.n.01'), Synset('ob
ject.n.01'), Synset('whole.n.02'), Synset('living_thing.n.01'), Sy
nset('organism.n.01'), Synset('animal.n.01'), Synset('invertebrate
.n.01'), Synset('arthropod.n.01'), Synset('insect.n.01'), Synset('
dictyopterous_insect.n.01'), Synset('cockroach.n.01')]
```

## 7 Tesauro

Se dibuja un tesauro donde se muestre la jerarquía de los cinco synsets y sus hiperónimos desde el nivel superior de la jerarquía hasta llegar al nivel donde aparezcan los sentidos de las cinco palabras analizadas.

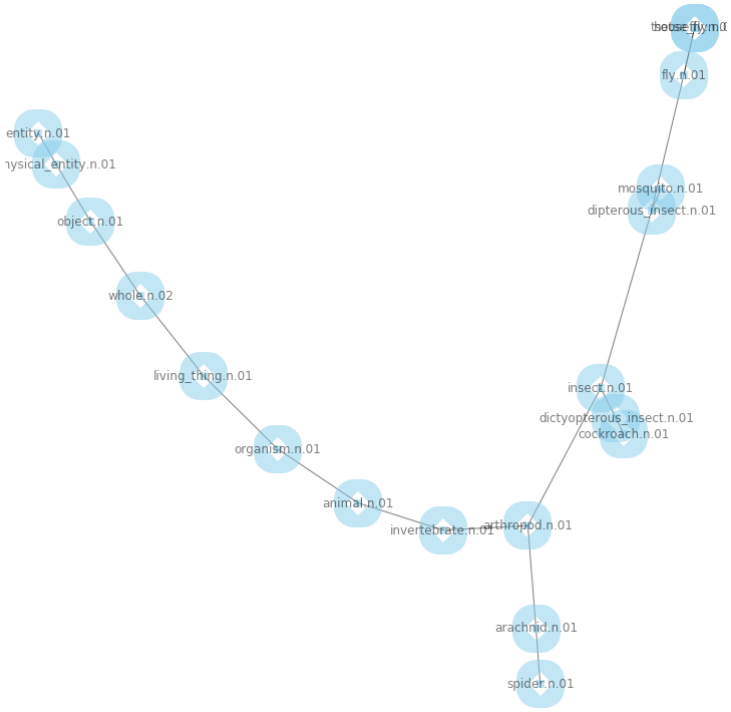
In [12]:
```
# Build from and to arrays
from_list = []
to_list = []
for hp in hps:
    for src, dest in zip(hp, hp[1:]):
        from_list.append(src.name())
        to_list.append(dest.name())
```

In [13]:
```
import pandas as pd
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt

# Build a dataframe with your connections
df = pd.DataFrame({ 'from':from_list, 'to':to_list})
df

# Build your graph
G=nx.from_pandas_edgelist(df, 'from', 'to')
plt.figure(figsize=(10,10))
nx.draw(G, pos=nx.spectral_layout(G, scale=100), with_labels=True,
node_size=50, node_color="skyblue", node_shape="s", alpha=0.5, line
widths=40, figsize=(40, 40))
plt.show()
```

```
/anaconda3/lib/python3.6/site-packages/networkx/drawing/nx_pylab.p
y:611: MatplotlibDeprecationWarning: isinstance(..., numbers.Numbe
r)
  if cb.is_numlike(alpha):
```




## 8 Búsqueda del Ancestro común más bajo

El ancestro común más bajo, es el pared entre dos synsets más bajo en la jerarquía, que se representa como un grafo.

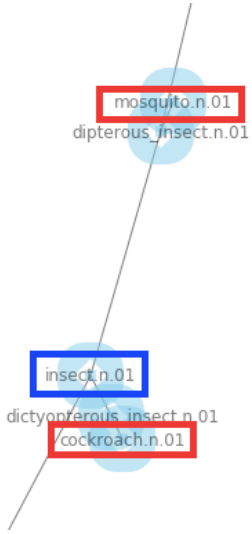
Es útil si queremos calcular la distancia entre dos nodos o synsets, ya que esta puede ser calculada como la suma de las distancias desde el ancestro a cada uno de los synsets

In [14]:
```
for x in synsets_list:
    for y in synsets_list:
        if(x!=y):

            print(str(x.lowest_common_hypernyms(y))+" es el ancestr
o común mas bajo de "+str(x)+" y "+str(y))
```

```
[Synset('dipterous_insect.n.01')] es el ancestro común mas bajo de
Synset('mosquito.n.01') y Synset('tsetse_fly.n.01')
[Synset('dipterous_insect.n.01')] es el ancestro común mas bajo de
Synset('mosquito.n.01') y Synset('housefly.n.01')
[Synset('arthropod.n.01')] es el ancestro común mas bajo de Synset
('mosquito.n.01') y Synset('spider.n.01')
[Synset('insect.n.01')] es el ancestro común mas bajo de Synset('m
osquito.n.01') y Synset('cockroach.n.01')
[Synset('dipterous_insect.n.01')] es el ancestro común mas bajo de
Synset('tsetse_fly.n.01') y Synset('mosquito.n.01')
[Synset('fly.n.01')] es el ancestro común mas bajo de Synset('tset
se_fly.n.01') y Synset('housefly.n.01')
[Synset('arthropod.n.01')] es el ancestro común mas bajo de Synset
('tsetse_fly.n.01') y Synset('spider.n.01')
[Synset('insect.n.01')] es el ancestro común mas bajo de Synset('t
setse_fly.n.01') y Synset('cockroach.n.01')
[Synset('dipterous_insect.n.01')] es el ancestro común mas bajo de
Synset('housefly.n.01') y Synset('mosquito.n.01')
[Synset('fly.n.01')] es el ancestro común mas bajo de Synset('hous
efly.n.01') y Synset('tsetse_fly.n.01')
[Synset('arthropod.n.01')] es el ancestro común mas bajo de Synset
('housefly.n.01') y Synset('spider.n.01')
[Synset('insect.n.01')] es el ancestro común mas bajo de Synset('h
ousefly.n.01') y Synset('cockroach.n.01')
[Synset('arthropod.n.01')] es el ancestro común mas bajo de Synset
('spider.n.01') y Synset('mosquito.n.01')
[Synset('arthropod.n.01')] es el ancestro común mas bajo de Synset
('spider.n.01') y Synset('tsetse_fly.n.01')
[Synset('arthropod.n.01')] es el ancestro común mas bajo de Synset
('spider.n.01') y Synset('housefly.n.01')
[Synset('arthropod.n.01')] es el ancestro común mas bajo de Synset
('spider.n.01') y Synset('cockroach.n.01')
[Synset('insect.n.01')] es el ancestro común mas bajo de Synset('c
ockroach.n.01') y Synset('mosquito.n.01')
[Synset('insect.n.01')] es el ancestro común mas bajo de Synset('c
ockroach.n.01') y Synset('tsetse_fly.n.01')
[Synset('insect.n.01')] es el ancestro común mas bajo de Synset('c
ockroach.n.01') y Synset('housefly.n.01')
[Synset('arthropod.n.01')] es el ancestro común mas bajo de Synset
('cockroach.n.01') y Synset('spider.n.01')
```

Estos resultados, se pueden comprobar en el grafo representado en el apartado anterior. Por ejemplo, entre mosquito y cockroach, el ancestro común es insect, tal y como se ve en el grafo, ya que es en insect donde el camino entre ambos synsets se separa.




## 9 Profundidad de los synsets

La profundidad de los synsets puede calcularse a partir de la jerarquía de hiperónimos, contando cada uno de ellos

In [15]:
```
for syn, hp in zip(synsets_list, hps):
    print('[%s] Path Depth: %d' % (syn, len(hp)))
```

```
[Synset('mosquito.n.01')] Path Depth: 12
[Synset('tsetse_fly.n.01')] Path Depth: 13
[Synset('housefly.n.01')] Path Depth: 13
[Synset('spider.n.01')] Path Depth: 11
[Synset('cockroach.n.01')] Path Depth: 12
```

## 10 Similitud entre Synsets

Se va a calcular el par de synsets más similares utilizando para ello el path_similarity

In [16]:
```
max_similar=0
max_pair=['x','y']
for x in synsets_list:
    for y in synsets_list:
        if(x!=y):
            path=x.path_similarity(y)
            if(path>max_similar):
                max_similar=path
                max_pair=[x,y]
```

In [17]:
```
print("Los synsets con mayor similaridad son: "+str(max_pair))
```

```
Los synsets con mayor similaridad son: [Synset('tsetse_fly.n.01'),
Synset('housefly.n.01')]
```

Tal y como se ha mencionado anteriormente, el ancestro común más bajo puede servir para calcular la similaridad entre dos palabras. Este es el método utilizado por la función path_similarity, que calcula un score basado en el camino más corto dentro de la jerarquía de hiperónimos, pasando por el ancestro común más bajo.

El camino entre tsetse_fly.n.01 y housefly.n.01, pasa por su ancestro fly.n.01, y como se puede ver en el grafo, están prácticamente superpuestas, por lo que su distancia es mínima