Complex Networks Project : structure of co-occurence graphs

1 Co-occurrence graphs what are they ?

The co-occurence graph if a text is defined as follow:

- The nodes of the graph are the words that appear in the text.
- We say that two words co-occure in a text when they appear next to each other in the same sentence, in this case there is a link between the two nodes representing them in the graph.

2 Construction and basic observations on co-occurence graphs

First we import the librairies

In [330]:
```
import networkx as nwx
import matplotlib.pyplot as plt
```

Construction of the co-occurence graph:

Here is a fonction that constructs the co-occurence graph of an input text that has string format.

Each word of a text corrisponds to a node. The input text is splitted into a list of sentences. For each consecutive two words in a same sentence, an edge is added between the two nodes representing the words.

In [331]:
```
def text_to_graph(s):
    s= s.lower()
    g = dict()
    t = s.replace("!",".")
    t = t.replace(";",".")
    t = t.replace(",","")
    phrases = t.split(".")
    for phrase in phrases:
        mots = phrase.split()
        for i in range(len(mots) - 1):
            if (mots[i] in g) and (mots[i+1] in g[mots[i]]) :
                g[mots[i]][mots[i+1]]['weight'] =1
            elif mots[i] in g:
                g[mots[i]][mots[i+1]] = {'weight':1}
            else:
                g[mots[i]] = {mots[i+1] : {'weight': 1}}
    return(nwx.Graph(g))
```

In [332]:
```
texte = " Un long silence se fit dans la voiture. Le chauffeur rega
g = text_to_graph(texte)
```

Firsts observations using Gephi:

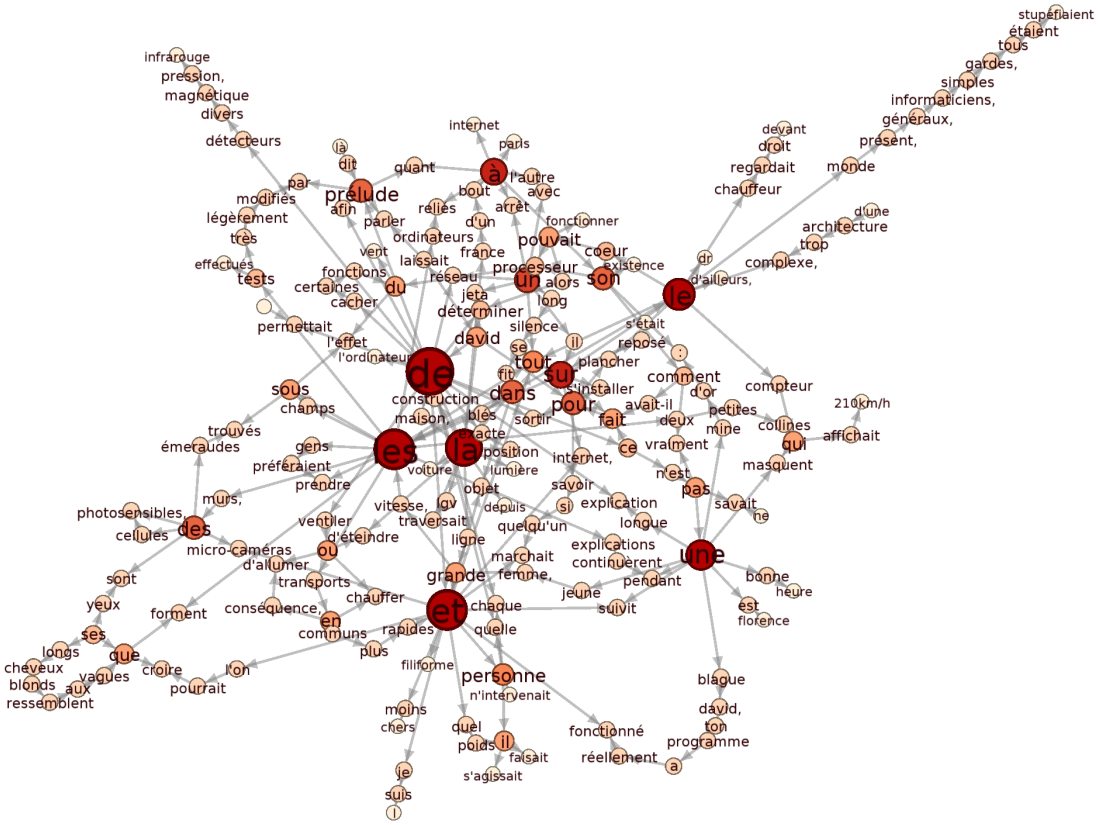
The data is exported at format graphml in order to visualize the graph using gephi:

In [333]:
```
nwx.write_gml(g,"./cooc.gml")
```

After some settings:

- color and size of the nodes are proportional to their degree
- we use the layouts "ForceAtlas 2" and "Label Adjust"

we get the following picture:



As we could expect, the networks exhibits large hubs that correspond to the most used words. Those are majoritarly "conjonction de coordination" ("et") or "articles définis et indéfinis" ("une" "la","de" ...), they characterize the french language. But there are also smallest hubs like "prélude" or "David" that seem to tell more about the topic of the text.

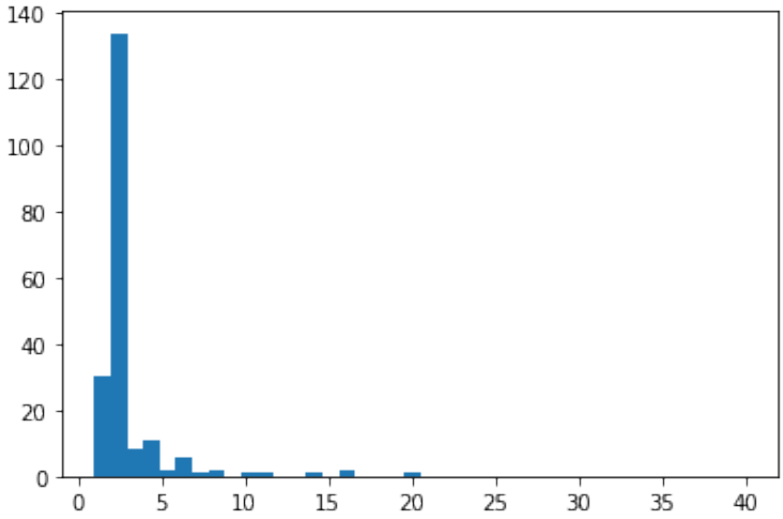
As many complex networks, this one seems to be scale-free and to exhibit small world structure. We can check those features using python.

Further observations using Python

First, lets observe the degree distribution of the network:

In [334]:
```
def plot_degree_dist(G):
    degrees = [G.degree(n) for n in G.nodes()]
    plt.hist(degrees,range=(1,40),bins=40)
    plt.show()

plot_degree_dist(g)
```



The distribution effectively does look like a power law.

Now, we check if the network has a small world structure:

In [335]:
```
largest = max(nwx.connected_components(g),key=len)
print(len(list(g.subgraph(largest).nodes())))
print(nwx.average_shortest_path_length(g.subgraph(largest)))
```

```
197
5.321972443799855
```

In [336]:
```
print(nwx.average_clustering(g))
print(nwx.density(g))
```

```
0.05039809532572691
0.013115577889447236
```

What we observe is that one the one hand, the average distance between two vertices is short as in a small world network ($\log 199 \simeq 5.41$). but on the other hand, it does not have the other characteristic of small world networks : the clustering coefficient is not significantly bigger than the density of the network. This can be explained by the fact that a lot of pairs of different words are linked with common hubs without being linked to each other.

Is it possible to distinguish between two authors writing styles thanks to co-occurence networks ?

This article (https://link.springer.com/content/pdf/10.1007/s11434-013-5711-8.pdf) showed that it was possible to distinguish between different languages by looking at basic characteristics of the co-occurence networks obtained from different texts written in these languages.

This made me wonder if it was possible to distinguish between several people writting with the same language by using simple characteristics of the co-occurence network. Unfortunately I did not get any satisfying result.

However, I present what I tried in this section.

I considered the texts of two french authors for children : Pierre Bottero and Erik L'Homme. I choose them because they wrote a serie of books together that I red some years ago, and I remembered that I had found difficult to know which of them had written which book.

For each of them, I selected 8 extracts of approximately 450 words from their books:

In [337]:
```
B1 = "Camille, abasourdie, s'apprêtait à proférer une question, mai
B2 = "Salim, lui, ne fut pas réprimandé pour son retard. À vrai dir
B3 = "Un coup de klaxon excédé le tira de ses pensées alors qu'il t
B4 = "Natan ne prit conscience de la vitesse à laquelle il roulait
B5 = "La tour ressemblait à un sablier de jade effilé coiffé d'une
B6 = "Les jours qui suivirent, Ellana évita de se retrouver seule a
B7 = "Elle avait cinq ans la première fois qu'elle vit une Armure.
B8 = "Lorsque Nawel atteignit la dernière marche de l'escalier des
L1 = "Harry Goodfellow jeta un rapide coup d’œil par la fenêtre. L’
L2 = "Il jeta par terre son pistolet-mitrailleur, vide, et brandit
L3 = "Assis dans un fauteuil de plastique bleu au centre du termina
L4 = "La réverbération du soleil sur la piste aveugla Vrânken qui p
L5 = "Mörgane ferma la porte blindée derrière elle et prit quelques
L6 = "Aujourd’hui, les dix-neuf planètes gravitant autour de Drasil
L7 = "Ambor et Bertolen éclatèrent de rire et lâchèrent les rênes d
L8 = "Il avait explosé de joie quand le Commandeur de la Confrérie
```

The parameters I looked at were (among others):

- The 30 nodes with maximum degree (I checked also for other centralities, but there were no difference)
- The average degree
- The average shortest path length
- The diameter
- The average clustering coefficient

None of these parameters permitted to show a difference between the two authors

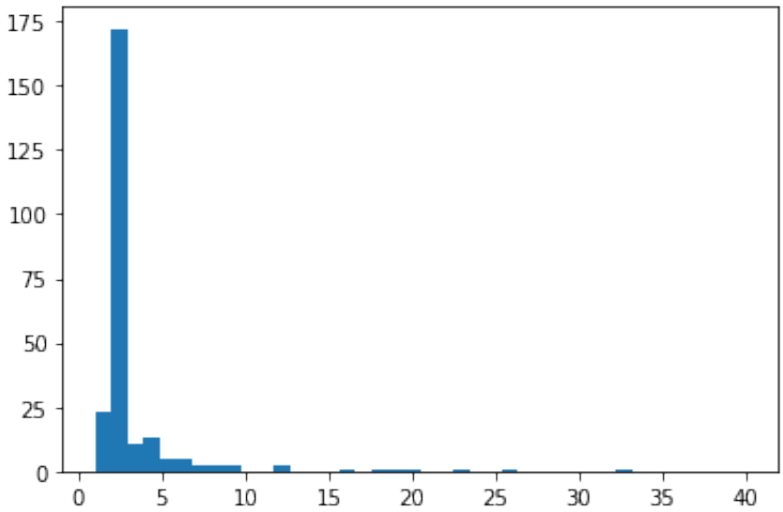
In [338]:
```
def maxdegree(g):
    lst = list(g.degree())
    l = sorted(lst, key = lambda t : t[1])
    for i in range (len(l)-30,len(l)):
        print(l[i][0])
```

In [339]:
```
def avg_deg(G):
    degrees = [G.degree(n) for n in G.nodes()]
    return(sum(degrees)/len(degrees))

def stats(t):
    gphe = text_to_graph(t)
    maxdegree(gphe)
    largest = max(nwx.connected_components(gphe),key=len)
    print("average_degree : "+str(avg_deg(gphe))+"\naverage shortes
    plot_degree_dist(gphe)
    return(None)
```

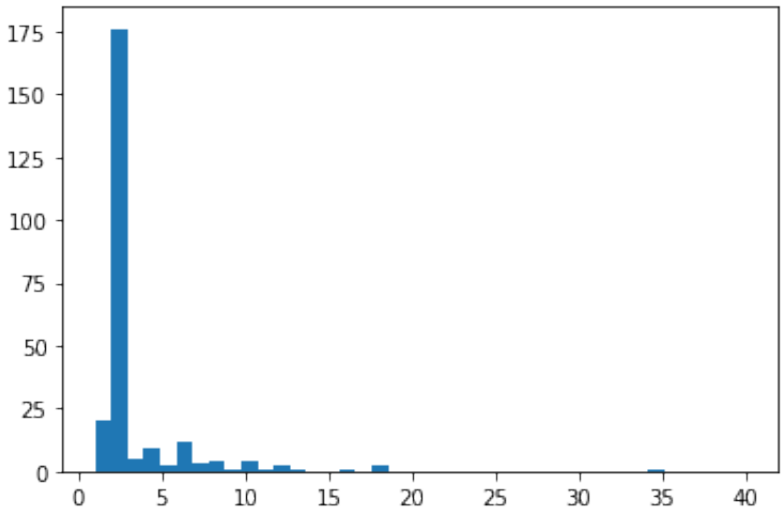
In [340]:
```
stats(B1)
```

```
pierre
chaîne
elle
du
aux
sa
hache
que
mètres
dans
se
d'une
coup
avait
au
camille
chevalier
ses
avec
d'un
en
qui
son
un
la
et
une
à
le
de
average_degree : 3.0081967213114753
average shortest path length : 4.185623692909667
average clustering coefficient : 0.052510660865038576
diameter : 12
```



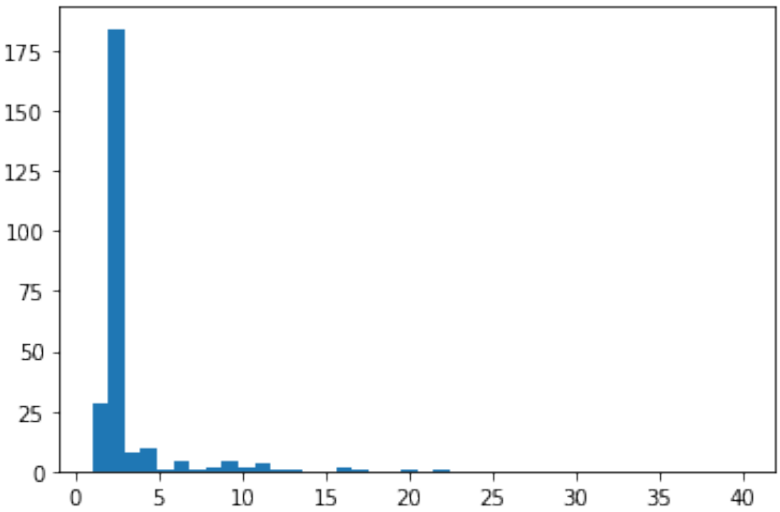
In [341]:
```
stats(B2)
```

```
son
même
se
s'il
cité
d'une
un
qui
des
collège
une
camille
avait
du
par
les
le
lui
à
dans
en
sa
il
ne
était
pas
que
et
la
de
average_degree : 3.040983606557377
average shortest path length : 4.43864264993591
average clustering coefficient : 0.03950558709478671
diameter : 11
```



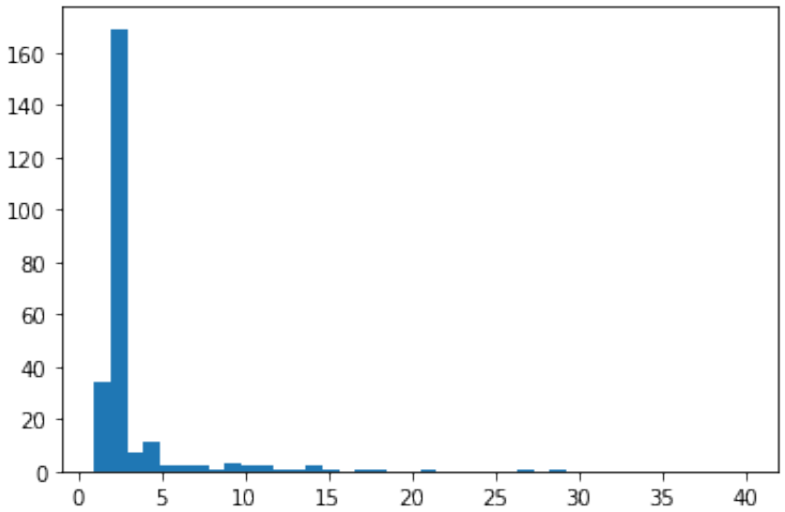
In [342]:
```
stats(B3)
```

```
dont
vers
été
très
bien
une
par
aurait
que
mais
natan
il
lui
ses
sa
pas
des
ne
à
la
qui
plus
avait
qu'il
et
les
un
le
en
de
average_degree : 2.9725490196078432
average shortest path length : 4.241994750656168
average clustering coefficient : 0.041354397474234375
diameter : 13
```



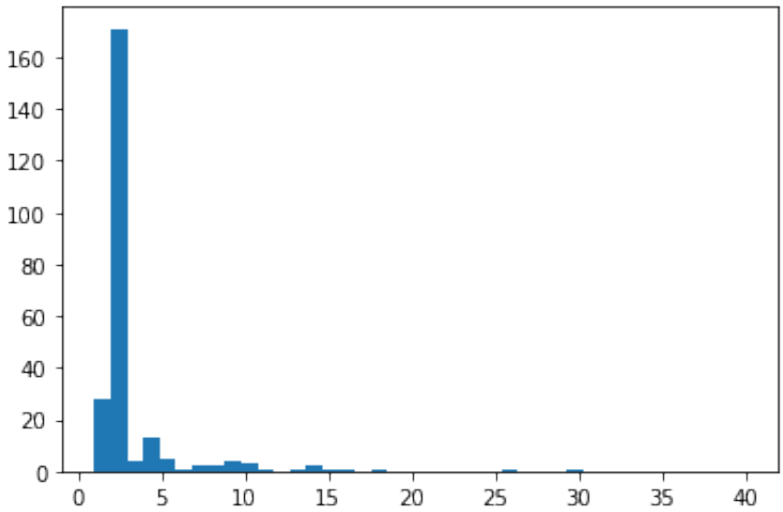
In [343]:
```
stats(B4)
```

```
lui
neige
ce
ses
voiture
avec
dix
comme
avait
était
lorsqu'il
des
du
il
sur
son
ne
un
natan
qui
que
une
et
les
se
pas
à
la
de
le
average_degree : 2.979591836734694
average shortest path length : 4.237470725995316
average clustering coefficient : 0.06143656638958207
diameter : 9
```



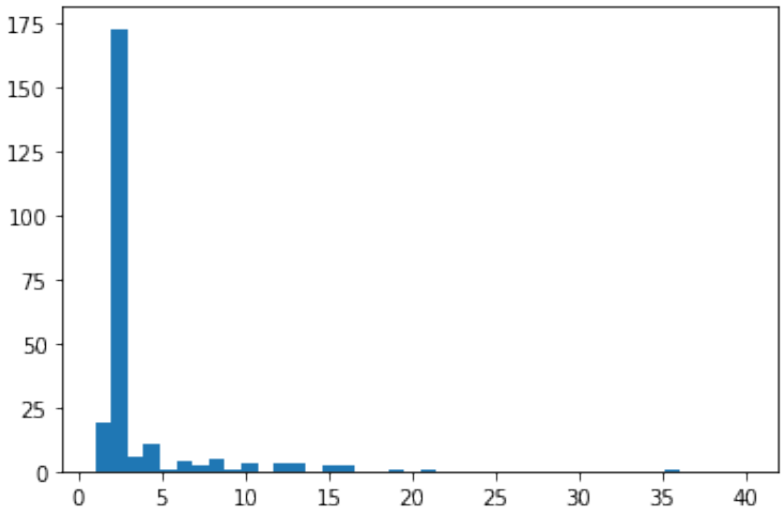
In [344]:
```
stats(B5)
```

```
bras
–
s'approcha
ce
ellana
pas
dans
d'elle
qu'elle
que
un
ne
des
en
sur
jilano
le
une
lui
son
ses
avait
pour
les
elle
qui
et
à
de
la
average_degree : 2.9669421487603307
average shortest path length : 4.378313500908748
average clustering coefficient : 0.03837249271477223
diameter : 13
```



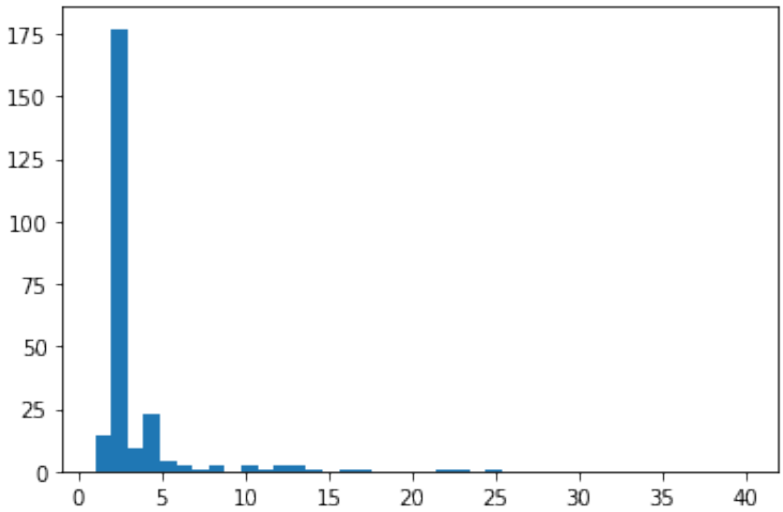
In [345]:
```
stats(B6)
```

```
une
ce
avec
mais
par
c'était
il
nillem
ne
pas
était
en
plus
sur
se
pour
son
les
lui
un
qui
que
ses
et
sa
le
à
la
elle
de
average_degree : 3.2016806722689077
average shortest path length : 3.9102932312165373
average clustering coefficient : 0.061185525614981605
diameter : 10
```



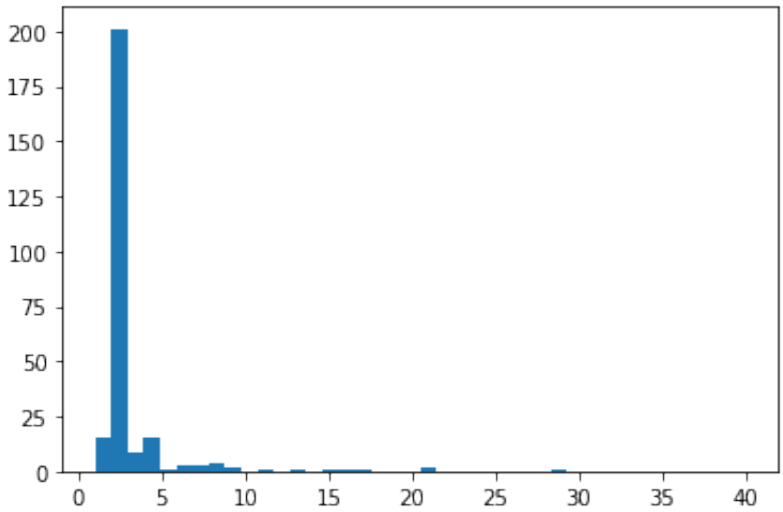
In [346]:
```
stats(B7)
```

```
par
s'y
il
soldats
guerrier
ou
poussière
devenait
le
ce
s'était
sans
son
dans
qui
en
était
du
sur
que
elle
un
avait
sa
des
les
et
à
la
de
average_degree : 3.020408163265306
average shortest path length : 4.319471395115423
average clustering coefficient : 0.024534555689104464
diameter : 12
```



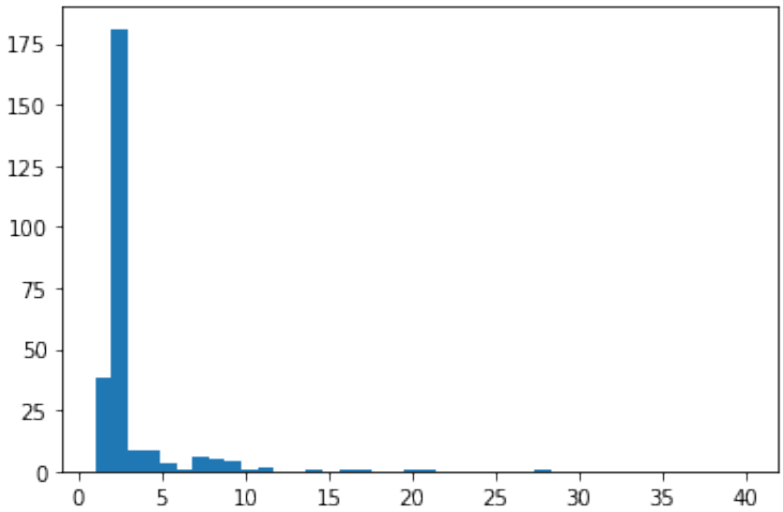
In [347]:
```
stats(B8)
```

```
aux
avaient
puissance
ou
cités
que
selon
façon
d'anknor
en
encore
plus
lorsque
elle
palais
se
du
un
avait
le
était
qui
une
les
à
ses
des
et
la
de
average_degree : 3.0076335877862594
average shortest path length : 4.182533415226229
average clustering coefficient : 0.05315729840061616
diameter : 12
```



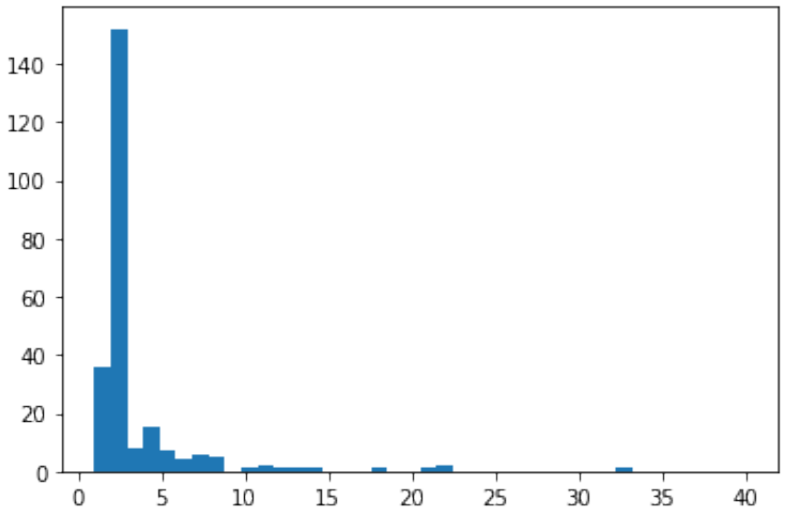
In [348]:
```
stats(L1)
```

```
l’avait
d’une
les
dans
sur
qui
un
lui
cette
pour
du
ce
par
des
ne
qu’il
une
ses
son
avait
pas
se
la
sa
il
en
et
le
à
de
average_degree : 2.830188679245283
average shortest path length : 4.813893653516295
average clustering coefficient : 0.028832716018007215
diameter : 12
```



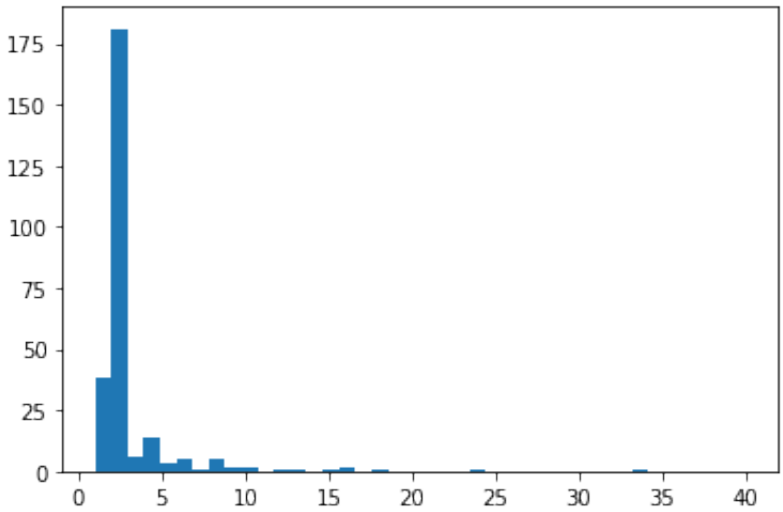
In [349]:
```
stats(L2)
```

```
yeux
j’ai
couteau
doigt
leur
nicolas
tout
sur
plus
vampire
avait
mais
pas
je
par
son
les
sa
qui
pour
en
était
elle
à
il
de
un
et
la
le
average_degree : 3.081967213114754
average shortest path length : 4.195776833299602
average clustering coefficient : 0.048263001797837865
diameter : 10
```



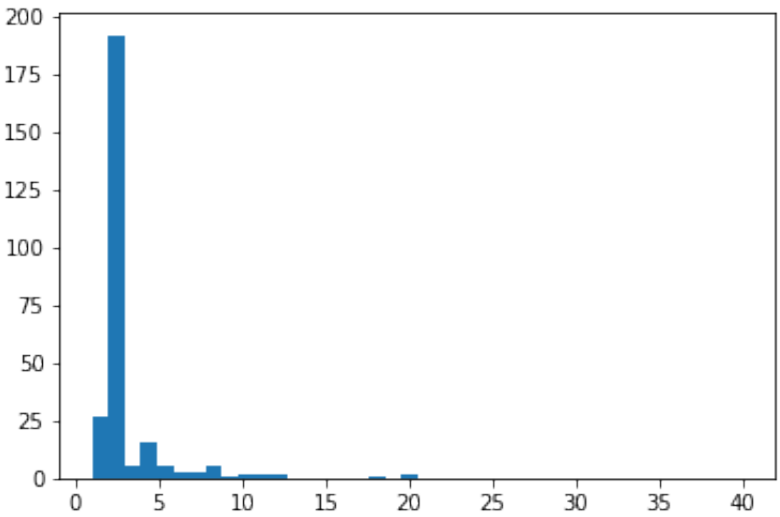
In [350]:
```
stats(L3)
```

```
l’avait
ce
leur
classes
bande
lui
pour
clarence
par
son
mais
aux
une
un
des
même
pas
était
au
sans
dans
du
qui
et
les
il
avait
la
le
de
average_degree : 2.8377358490566036
average shortest path length : 4.797169811320755
average clustering coefficient : 0.04273973545314568
diameter : 13
```



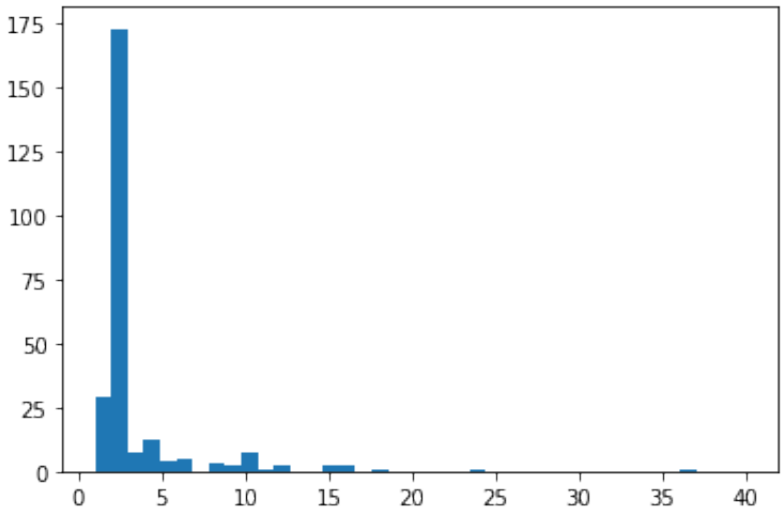
In [351]:
```
stats(L4)
```

```
se
tente
que
ses
khan
ils
fit
sous
dans
leur
pour
sur
il
lui
des
avec
et
une
au
on
du
qui
vrânken
son
les
un
à
la
le
de
average_degree : 2.9288389513108615
average shortest path length : 4.262735490411422
average clustering coefficient : 0.05522787470990762
diameter : 12
```



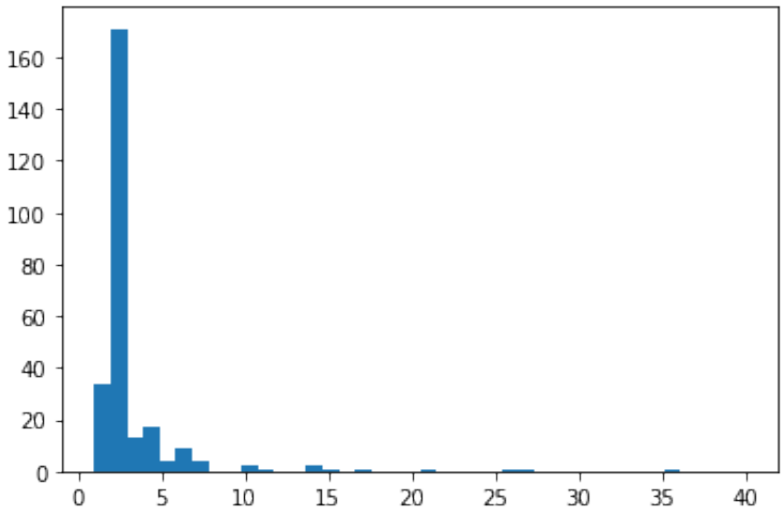
In [352]:
```
stats(L5)
```

```
ne
peu
l’ordre
avaient
où
ses
source
l’espace
sa
qui
un
le
une
que
qu’elle
avait
était
des
en
dans
au
elle
son
les
se
et
du
à
la
de
average_degree : 3.1031746031746033
average shortest path length : 4.36400430025928
average clustering coefficient : 0.03870676885535002
diameter : 15
```



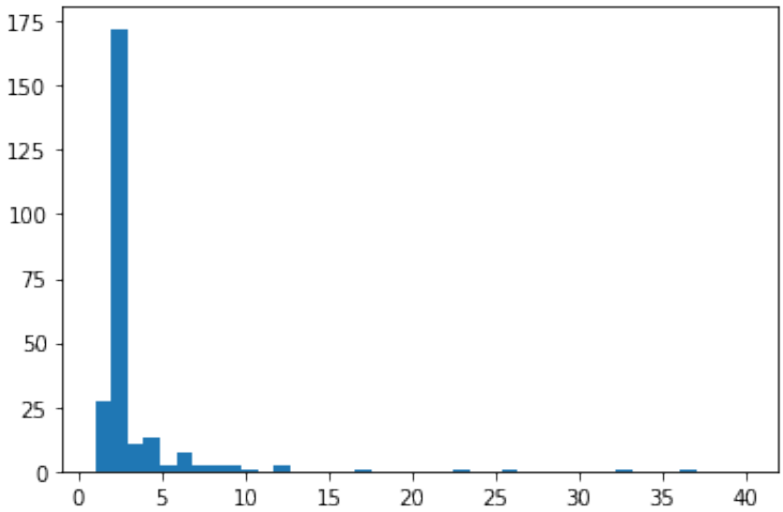
In [353]:
```
stats(L6)
```

```
frä
bien
je
otchigins
qui
elle
aux
sont
du
se
tenu
sur
un
pour
s’est
en
mörgane
lui
ce
par
dans
et
le
que
les
à
a
des
la
de
average_degree : 2.988593155893536
average shortest path length : 4.4253330624328795
average clustering coefficient : 0.046703659379794475
diameter : 10
```



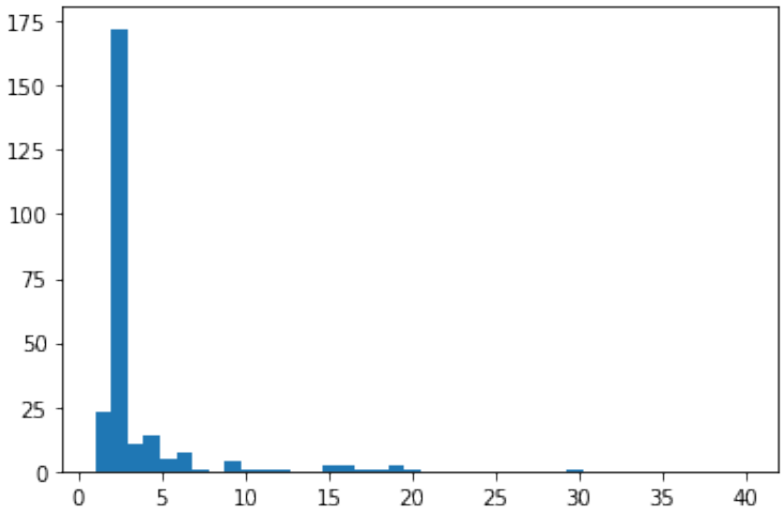
In [354]:
```
stats(L7)
```

```
leur
cour
aux
d’armes
que
soufflait
qu’il
par
d’un
au
se
du
plus
était
où
étage
ses
une
sur
étaient
des
un
en
les
qui
à
le
la
et
de
average_degree : 2.951219512195122
average shortest path length : 4.093313422930148
average clustering coefficient : 0.052511757315652885
diameter : 10
```



In [355]:
```
stats(L8)
```

```
bromotul
romaric
pas
ne
alors
était
même
un
serait
des
sa
ou
sur
ses
que
qu'il
plus
lui
se
son
avait
en
la
pour
les
le
il
à
et
de
average_degree : 3.136
average shortest path length : 4.240899598393574
average clustering coefficient : 0.039846546925430236
diameter : 12
```



Conclusion

I found it very difficult to exploit co-occurence networks to show things that we do not already know thanks to grammar.

I had no time to do so but I think that in order to find characteristics that are proper to the texts or the authors, it would be interesting to study them by comparing them to a kind of canonical language representing the french language. Maybe it would avoid to have observations polluted by what we already know about the language (for example the fact that there are always words such that "le" "une" "des"... appearing in the experiments above) (but I have no clue of how one can construct this).