

COMPUTER SCIENTISTS RETRIEVAL

WIR Project

Students:

- Luca Tomei
- Daniele Iacomini
- Andrea Aurizi

Directed by:

Andrea Vitaletti

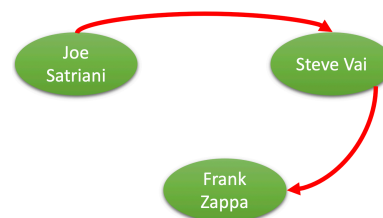
Luca Becchetti

Project Repository

Summary

The paper we chose presents a method to find the most influential rock guitarist by applying Google's PageRank algorithm to information extracted from Wikipedia articles. The influence of a guitarist is computed by considering the number of guitarists citing him/her as influence.

Basically, the experiment consists of building a directed graph where nodes are rock guitarists. There is an outgoing edge from guitarist A to another guitarist B, if guitarist A is influenced by guitarist B.



Joe Satriani is influenced by Steve Vai and the latter by Frank Zappa

We decided to replicate the same experiment with the same methodology but choosing computer scientists as the field of study.

A.Y. 2019-2020



Contents

1	Introduction	3
2	Query DBpedia with SPARQL	4
3	Manual scraping Wikipedia	5
3.1	First Phase: Collect data	5
3.2	Second Phase: Check informations in Biographic Table	5
3.3	Third Phase: Second Approach and Pagerank	6
3.3.1	My Pagerank Results	8
4	Results	9
4.1	Subsection (if any)	9
4.2	Subsection (if any)	10
5	Conclusions	12

1 Introduction

In questo progetto ci siamo soffermati sull'analisi dei dati riguardanti una categoria diversa da quella dei chitarristi: i computer scientists. A differenza di un chitarrista o di un filosofo, un ingegnere informatico non ha molti dati rilevanti su wikipedia e non vi sono nemmeno informazioni riguardanti la sua scuola di pensiero o le influenze avute durante la sua vita. Infatti la prima difficoltà riscontrata durante la fase preliminare del progetto è stata proprio quella di cercare di creare una corrispondenza biunivoca tra due ingegneri informatici, cosa che non sempre è possibile.

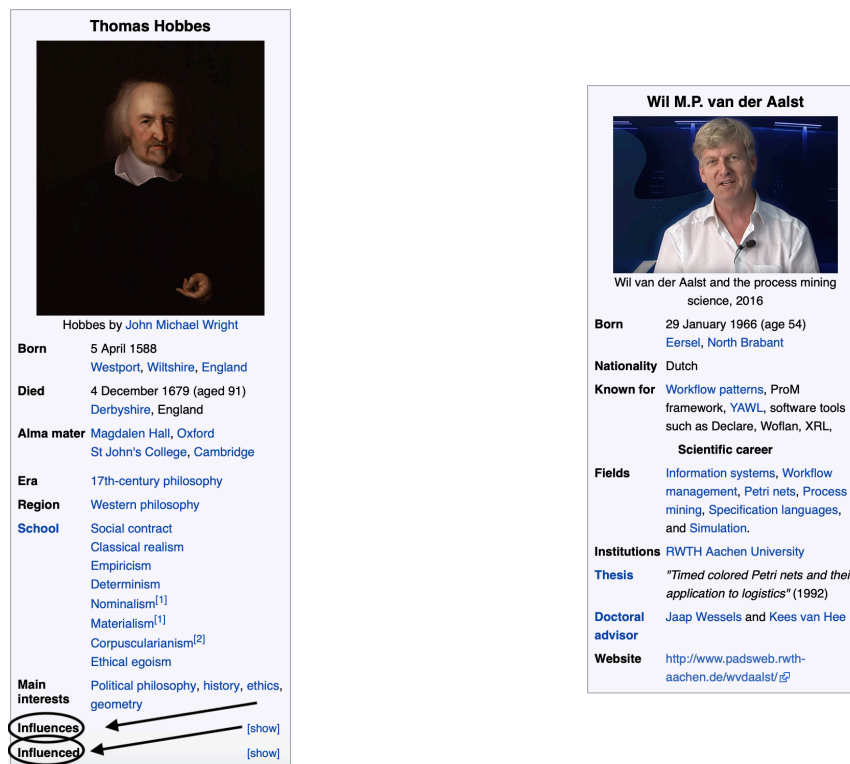


Figure 1: Poet vs Computer Scientist

Le precedenti figure mostrano i campi dati presenti sulle tabelle "infobox biography vcard" di Wikipedia. La differenza nel numero di campi tra un filosofo ed un computer scientist è netta ed inoltre quest'ultimo non presenta campi di tipo "Influences" e "Influenced" a cui possiamo attingere per creare collegamenti tra un informatico ed un altro.

Pertanto, per ovviare a tale problema, in primo luogo si è deciso di utilizzare *SPARQL* per esplorare ed estrarre le informazioni contenute in grafi RDF da una base di conoscenza distribuita su *DBpedia*.

Successivamente si è deciso di effettuare una ulteriore verifica dei risultati andando ad interrogare direttamente Wikipedia dato che è senza dubbio una delle maggiori risorse del Web per tutte le conoscenze.

2 Query DBpedia with SPARQL

Per interrogare DBpedia si è pensato di utilizzare *SPARQLWrapper*, wrapper con lo scopo di creare l'URI della query e, possibilmente, a convertire il risultato in un formato più gestibile.

```
1  """Query DBpedia with SPARQL Code"""
2  from SPARQLWrapper import SPARQLWrapper, JSON
3
4  sparql = SPARQLWrapper("http://dbpedia.org/sparql")
5  sparql.setQuery("""
6      PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
7      SELECT *
8      WHERE {
9          ?p a
10             <http://dbpedia.org/ontology/Computer_scientists> .
11             ?p <http://dbpedia.org/ontology/influenced> ?influenced.
12      }
13  """)
14  sparql.setReturnFormat(JSON)
15  results = sparql.query().convert()
```

Listing 1: Query DBpedia

Con i dati memorizzati in questo modo, la creazione del grafo è semplice: è sufficiente elaborare ogni riga (ognuna è costituita da due voci: influenzato e influenza) e inserire un arco corrispondente nel grafo.

Dopo aver incapsulato l'output della query nella variabile `result` siamo pronti ad esaminare i dati ottenuti e a creare un grafo per calcolare il pagerank.

```
35  G = nx.DiGraph()
36
37  for result in results["results"]["bindings"]:
38      try:
39          cs = result["p"]["value"].split("/resource/")[1]
40          influenced = result["influenced"]["value"].split("/resource/")[1]
41
42          G.add_edge(influenced, cs)
43      except: print("", end="")
44
45  """ Computing Pagerank """
46  pr = nx.pagerank(G, alpha=0.85)
47  sorted_pr = sorted(pr.items(), key=operator.itemgetter(1))
48  sorted_pr.reverse()
49
50  for i in sorted_pr: print(i)
```

Listing 2: Computing Pagerank

3 Manual scraping Wikipedia

DBpedia offre pochi risultati per i Computer Scientists, motivo per cui si è deciso di intraprendere una strada più impegnativa ma che ci restituisse più risultati: il Web Scarping.

Si è pensato di suddividere tale processo in 4 fasi logiche:

1. Collezionare in un file tutti i link di computer scientists presenti nella versione inglese di Wikipedia (*List Of Computer Scientists*).
2. Per ogni link di un informatico ottenuto dalla fase precedente, si è verificato se la relativa pagina web contenesse una tabella informazioni che includesse la lista di influenze e influenzati di tale informatico.
3. Dai risultati ottenuti si è calcolato il Pagerank con lo scopo di quantificare l'importanza dell'informatico relativa all'interno dell'insieme di documenti connessi.
4. Si è pensato di soffermarci sulla classificazione dei vari campi di studio di un informatico.

3.1 First Phase: Collect data

The English version of Wikipedia contains a list of all the computer scientists (*link*) with an existing article, alphabetically sorted. In the first phase, we simply copy each link and its associated name in a .json file.

The total number of computer scientists retrieved is 509.

3.2 Second Phase: Check informations in Biographic Table

Prima di iniziare ad estrapolare le informazioni di ogni informatico, per un questione di ottimizzazione del codice e di efficienza in termini prestazionali si è pensato di rendere offline le intere pagine Wikipedia di ognuno di essi, in modo tale da non effettuare $n_{cs} = 509$ richieste.

Dopo aver collezionato le pagine web di ogni informatico, per ognuno di essi si è verificato se la corrispettiva pagina contenesse una tabella informativa (*Infobox HTML*) che includesse la lista di influences and influencers del suddetto informatico.

Il numero di Computer Scientists che soddisfano tale requisito è pari a 62.

```

1  def bio_table(self, page):
2      name = page.rsplit('/')[ -1]
3      page = open(page, 'r')
4      soup = BeautifulSoup(page, "html.parser")
5      table = soup.find('table', class_='infobox biography vcard')
6      try: influencers = table.find_all('ul', class_='NavContent')[0]
7      except: influencers = []
8      try: influenced = table.find_all('ul', class_='NavContent')[1]
9      except: influenced = []
10     final_influencers, final_influenced = ([] for i in range(2))
11     if influencers != []:
12         for a in influencers.find_all('a'):
13             final_influencers.append(a.get('title'))
14     if influenced != []:
15         for a in influenced.find_all('a'):
16             final_influenced.append(a.get('title'))

```

Listing 3: Checking Bio Table Function

3.3 Third Phase: Second Approach and Pagerank

Dato che il risultato ottenuto dalla seconda fase è stato per nulla soddisfacente, si è deciso di utilizzare un secondo approccio: invece di verificare solamente la tabella biografia di un informatico si è analizzata l'intera pagina a lui associata.

```

1  def make_links(self, path):
2      # Define output dict
3      inlinks = SortedDict()
4      outlinks = SortedDict()
5
6      SetofNames = SortedSet()
7
8      #reading all the folders from the path and creating a set of CS names
9      for name in self.read_names(path):
10         if name == "Guy_L._Steele,_Jr": name = "Guy_L._Steele,_Jr."
11
12         SetofNames.add(name)
13
14         #creating an empty inlinks of names as sortedSet
15         inlinks[name] = SortedSet()
16
17     #reading their inlinks and outlinks
18     for name in SetofNames:
19         SetOfInLink = SortedSet()
20         fp = open(path + "/" + name, 'r', encoding = "utf-8")
21         soup = BeautifulSoup(fp.read(), "html.parser")
22         linksFound = []
23         linksFound = soup.findAll('a', href=re.compile("/wiki/"))
24
25         HTML = ""
26         for link in linksFound:
27             HTML = HTML + str(link)
28             HTML = HTML + " and "
29
30         #get All the outlinks by calling get_links
31         outlinks[name] = self.get_links(SetofNames, HTML)
32
33         if name in outlinks[name]: outlinks[name].remove(name)
34
35         for outlink in outlinks[name]:
36             SetOfInLink.add(name)
37             inlinks[outlink].update(SetOfInLink)
38     return (inlinks, outlinks)

```

Listing 4: Making Links

La precedente funzione, dopo aver letto in input le pagine html di ogni computer scientist, crea e restituisce due *SortedDict*:

- *inlinks*: maps from a name to a SortedSet of names that link to it.
- *outlinks*: maps from a name to a SortedSet of names that it links to.

Ad esempio:

- `inlinks['Ada_Lovelace'] = SortedSet(['Charles_Babbage', 'David_Gelernter'], key=None, load=1000)`
- `outlinks['Ada_Lovelace'] = SortedSet(['Alan_Turing', 'Charles_Babbage'], key=None, load=1000)`

To obtain all the *outlinks* we call `self.get_links(SetofNames, HTML)`, which return a *SortedSet* of computer scientist names that are linked from this html page. The return set is restricted to those people in the provided set of names. The returned list should contain no duplicates. This function take as input:

- A *SortedSet* of computer scientist names, one per filename.
- A string representing one html page.

```

1 def get_links(self, names, html):
2     listofHrefs = []
3     listofHrefTexts = []
4     FinalSortedSet = SortedSet()
5     splice_char = '/'
6
7     for i in range(0, len(listofHrefs)):
8         value = listofHrefs[i][6:]
9         listofHrefTexts.append(value)
10
11     listofHrefTexts = re.findall(r'href', html)
12
13     for i in listofHrefTexts:
14         value = i[6:]
15         listofHrefs.append(value)
16     listofHrefs = list(set(listofHrefs))
17
18     for href in listofHrefs:
19         for name in names:
20             if(name == "Guy_L._Steele,_Jr"):
21                 names.remove(name)
22                 names.add("Guy_L._Steele,_Jr.")
23             if(href == name): FinalSortedSet.add(name)
24
25     return FinalSortedSet

```

Listing 5: Get Links

Con i risultati ottenuti dalla funzione *make_links* possiamo calcolare il pagerank chiamando la funzione *compute_pagerank*.

```

1 def compute_pagerank(self, urls, inlinks, outlinks, b=.85, iters=20):
2     rw = defaultdict(lambda:0.0)
3     pageRank = defaultdict(lambda:1.0)
4
5     for outlink in outlinks: rw[outlink]=len(outlinks[outlink])
6
7     #initialize page ranks scores to 1
8     for url in urls: pageRank[url] = 1.0
9
10    for i in range(iters):
11        for url in urls:
12            summ = 0.0
13            for link in inlinks[url]: summ += 1.0 * pageRank[link]/rw[link]
14            pageRank[url] = (1/len(urls))* (1.0-b)+b*summ
15    return SortedDict(dict(pageRank))

```

Listing 6: Computing Pagerank

This function return a *SortedDict* mapping each url to its final PageRank value (float) by using this formula:

$$R(u) = \left(\frac{1}{N}\right)(1 - b) + b \cdot \sum_{w \in B_u} \frac{R(w)}{|F_w|}$$

where:

- $R(u)$ = Valore pagerank della pagina u che vogliamo calcolare;
- $B(u)$ = Insieme di pagine che contengono almeno un link verso la pagina u . w rappresenta ognuna di tali pagine;
- $R(w)$ = Valori di PageRank di ogni pagina w ;
- F_w = Numero complessivo di link contenuti nella pagina che offre il link;
- b = Damping factor. It is generally assumed that the damping factor will be set around 0.85.

3.3.1 My Pagerank Results

La seguente figura mostra l'output del terminale dopo aver effettuato la chiamata a *compute_pagerank* con 20 iterazioni e 509 computer scientists.

Top 20 Page Ranks with 10 iterations	
Donald_Knuth	0.08372
Rudy_Rucker	0.07940
Fred_Brooks	0.07389
Adi_Shamir	0.06984
Douglas_Engelbart	0.06941
Allen_Newell	0.06934
Stephen_Wolfram	0.06704
Alan_Kay	0.06534
Niklaus_Wirth	0.06421
Alan_Perlis	0.06368
E_Allen_Emerson	0.06327
Herbert_A_Simon	0.06236
Dennis_Ritchie	0.06233
Edmund_M_Clarke	0.06215
Amir_Pnueli	0.06165
Dana_Scott	0.06021
John_McCarthy_(computer_scientist)	0.05920
John_Cocke	0.05905
Barbara_Liskov	0.05878
Charles_Bachman	0.05854

Figure 2: *compute_pagerank* output

Dopo aver applicato il pagerank si è pensato di costruire un grafo diretto che evidenziasse i collegamenti tra ogni computer scientist. Dalla figura mostrata in seguito, costruita con la libreria PIL di Python, si evince che vi è un'addensamento nella parte superiore che evidenzia i numerosi collegamenti che vi sono tra i top computer scientists ottenuti in output dalla funzione *compute_pagerank*.

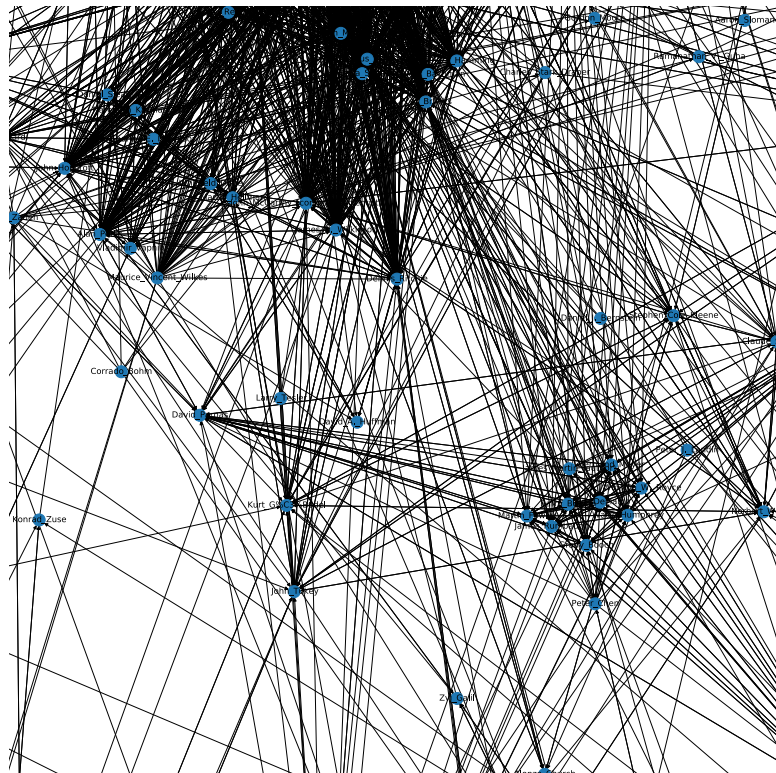


Figure 3: Pagerank Graph

4 Results

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4.1 Subsection (if any)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis.

Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

4.2 Subsection (if any)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut

massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

5 Conclusions

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

References

- [1] SPARQLWrapper: <https://pypi.org/project/SPARQLWrapper/>
- [2] LÓPEZ, F. (SF) *Espacios Topoógicos*, Depto. de Geometría y Topología, Universidad de Granada, E-18071 Granada, España
- [3] MACHO, M. (2002), *¿Qué es la topología?*, Universidad del País Vasco-Euskal Herriko Unibertsitatea.
- [4] O'CONNOR, J., ROBERTSON, E. (1996). *A history of Topology*.