Introduction to Web Science

Assignment 3

Prof. Dr. Steffen Staab
staab@uni-koblenz.de

René Pickhardt
rpickhardt@uni-koblenz.de

Korok Sengupta
koroksengupta@uni-koblenz.de

Institute of Web Science and Technologies
Department of Computer Science
University of Luxembourg

Submission until: November 16, 2016, 10:00 a.m.
Tutorial on: November 18, 2016, 12:00 p.m.

The main objective of this assignment is for you understand different concepts that are associated with the "Web". In this assignment we cover two topics: 1) DNS & 2) Internet.

These tasks are not always specific to "Introduction to Web Science". For all the assignment questions that require you to write a code, make sure to include the code in the answer sheet, along with a separate python file. Where screen shots are required, please add them in the answers directly and not as separate files.

Team Name: Abdullah Elkindy, Fiorela Ciroku, Stela Nebiaj

1 DIG Deeper (5 Points)

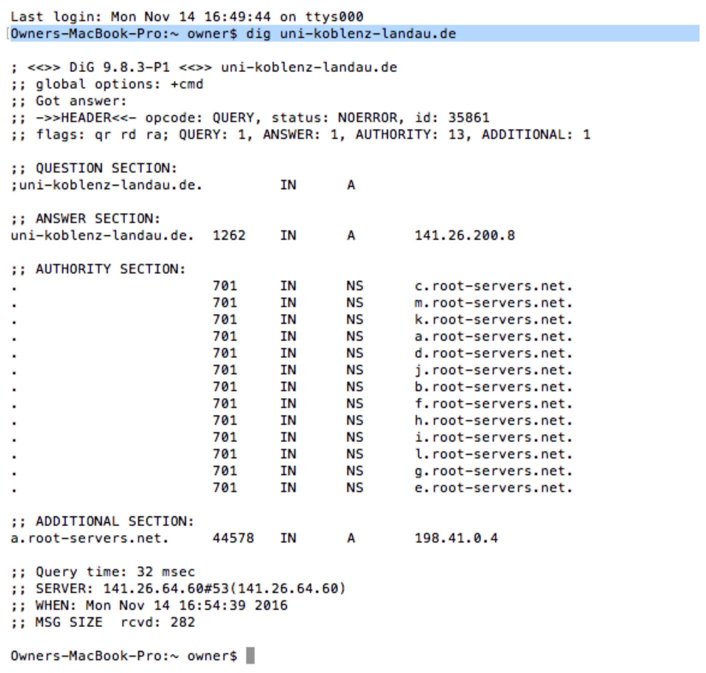
Assignment 1 started with you googling certain basic tools and one of them was "*dig*".

1. Now using that dig command, find the IP address of `www.uni-koblenz-landau.de`
2. In the result, you will find "SOA". What is SOA?
3. Copy the SOA record that you find in your answer sheet and explain each of the components of SOA with regards to your find. Merely integrating answers from the internet wont fetch you points.

Try the experiment once from University network and once from Home network and see if you can find any differences and if so, clarify why.

Answers:

1. According to Figure 1, the IP address is 141.26.200.8



```
Last login: Mon Nov 14 16:49:44 on ttys000
Owners-MacBook-Pro:~ owner$ dig uni-koblenz-landau.de

; <<>> DiG 9.8.3-P1 <<>> uni-koblenz-landau.de
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35861
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 13, ADDITIONAL: 1

;; QUESTION SECTION:
;uni-koblenz-landau.de.         IN      A

;; ANSWER SECTION:
uni-koblenz-landau.de.  1262    IN      A       141.26.200.8

;; AUTHORITY SECTION:
.                       701     IN      NS      c.root-servers.net.
.                       701     IN      NS      m.root-servers.net.
.                       701     IN      NS      k.root-servers.net.
.                       701     IN      NS      a.root-servers.net.
.                       701     IN      NS      d.root-servers.net.
.                       701     IN      NS      j.root-servers.net.
.                       701     IN      NS      b.root-servers.net.
.                       701     IN      NS      f.root-servers.net.
.                       701     IN      NS      h.root-servers.net.
.                       701     IN      NS      i.root-servers.net.
.                       701     IN      NS      l.root-servers.net.
.                       701     IN      NS      g.root-servers.net.
.                       701     IN      NS      e.root-servers.net.

;; ADDITIONAL SECTION:
a.root-servers.net.     44578   IN      A       198.41.0.4

;; Query time: 32 msec
;; SERVER: 141.26.64.60#53(141.26.64.60)
;; WHEN: Mon Nov 14 16:54:39 2016
;; MSG SIZE  rcvd: 282

Owners-MacBook-Pro:~ owner$
```

Figure 1

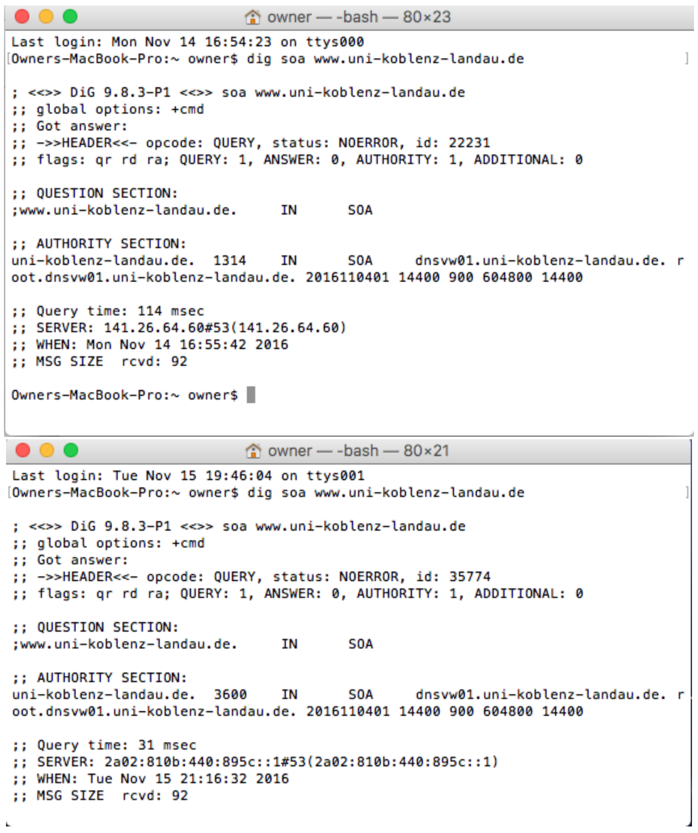
2. Start of Authority (SOA) is a record existing for each zone identifying who is the primary name server responsible for it. So that every domain when delegated to an authoritative domain is linked.

3. The SOA record is:

   uni-koblenz-landau.de. 1314 IN SOA dnsvw01.uni-koblenz-landau.de. root.dnsvw01.uni-koblenz-landau.de. 2016110401 14400 900 604800 14400

   - uni-koblenz-landau.de. is the main name for this zone.
   - 1314 is TTL
   - IN is for the class, and it implies that this is for internet.
   - dnsvw01.uni-koblenz-landau.de. root.dnsvw01.uni-koblenz-landau.de. are the authoritative name servers containing the
   - 2016110401 serial number for the DNS zone that is incremented every time a change is made to the zone file.
   - 14400 refresh in seconds to see how often a secondary server will poll a primary one to check if the zone has increased (to request new copy of the data in the zone).
   - 900 retry value is time in seconds whenever a secondary server tries to contact the primary one and fails.
   - 604800 this is the time in seconds a secondary server can use its cached version of the zone file after losing contact with the primary server. In this way we can avoid showing to the end users problems that may accrue in our primary server. Because the secondary ones, can still use the cached zone files. - 14400 This is time in seconds defining how long will a secondary server (slave server) can keep the zone file cached. Once it's passed, the zone file will be dropped and the slave server will ask the master one for a new zone file.

   The difference is in the starting value of ttl. It can be seen in Figure 2. TTL at uni starts from 1400 because we're physically closer to the zone while at home starts from 3600. The closer you are to the zone, the less ttl you will require.



owner — -bash — 80×23

```
Last login: Mon Nov 14 16:54:23 on ttys000
Owners-MacBook-Pro:~ owner$ dig soa www.uni-koblenz-landau.de

; <<>> DiG 9.8.3-P1 <<>> soa www.uni-koblenz-landau.de
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22231
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;www.uni-koblenz-landau.de.     IN      SOA

;; AUTHORITY SECTION:
uni-koblenz-landau.de.  1314    IN      SOA     dnsvw01.uni-koblenz-landau.de. r
oot.dnsvw01.uni-koblenz-landau.de. 2016110401 14400 900 604800 14400

;; Query time: 114 msec
;; SERVER: 141.26.64.60#53(141.26.64.60)
;; WHEN: Mon Nov 14 16:55:42 2016
;; MSG SIZE  rcvd: 92

Owners-MacBook-Pro:~ owner$ 
```

owner — -bash — 80×21

```
Last login: Tue Nov 15 19:46:04 on ttys001
Owners-MacBook-Pro:~ owner$ dig soa www.uni-koblenz-landau.de

; <<>> DiG 9.8.3-P1 <<>> soa www.uni-koblenz-landau.de
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35774
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;www.uni-koblenz-landau.de.     IN      SOA

;; AUTHORITY SECTION:
uni-koblenz-landau.de.  3600    IN      SOA     dnsvw01.uni-koblenz-landau.de. r
oot.dnsvw01.uni-koblenz-landau.de. 2016110401 14400 900 604800 14400

;; Query time: 31 msec
;; SERVER: 2a02:810b:440:895c::1#53(2a02:810b:440:895c::1)
;; WHEN: Tue Nov 15 21:16:32 2016
;; MSG SIZE  rcvd: 92
```

Figure 2

2 Exploring DNS (10 Points)

In the first part of this assignment you were asked to develop a simple TCP Client Server. Now, using **that** client server setup. This time a url should be send to the server and the server will split the url into the following:

`http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#InTheDocument`

1. Protocol
2. Domain
3. Sub-Domain
4. Port number
5. Path
6. Parameters
7. Fragment

The Protocol for sending the URL will be a string terminated with \r \n.

P.S.: You are **not** allowed to use libraries like `urlparse` for this question. You will also not use "Regular Expressions" for this.

Answer:

Server.py

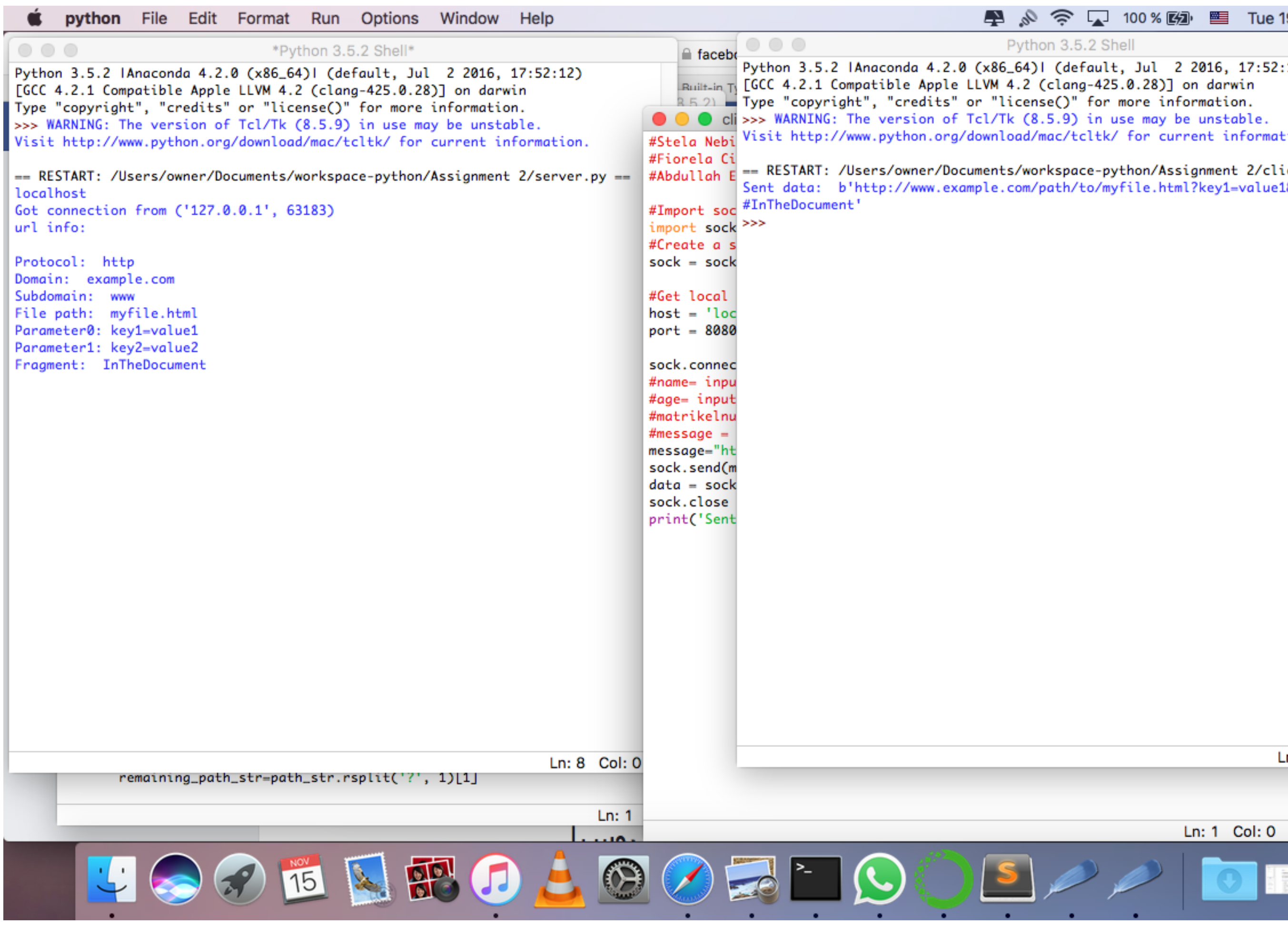
```
#Stela Nebiaj
#Fiorela Ciroku
#Abdullah Elkindy

#Import socket module
import socket
#Create a socket object
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
#Get local machine name
host = 'localhost'
port = 8080
print(host)
#Bind to the port
sock.bind((host, port))

#Now wait for client connection.
sock.listen(5)

#Establish connection with client.
connection, client_addr = sock.accept()
```

```

print ('Got connection from', client_addr)
print("url info:")
print("")

while 1:
        data= connection.recv(1024)
        if not data: break
        dns_link=str(data).split('\'')
        dns_link=str(dns_link[1])
        dns_link=dns_link.split('/')
        print("Protocol: ",dns_link[0].rsplit(':',1)[0])
        dom_str=dns_link[2]
        dom_str=str(dom_str).split('.')
        print("Domain: ",str(dom_str[1])+"."+str(dom_str[2]))
        print("Subdomain: ",dom_str[0])
        path_str=str(dns_link[5])
        print("File path: ",path_str.rsplit('?', 1)[0])
        remaining_path_str=path_str.rsplit('?', 1)[1]

        parameters_str=remaining_path_str.rsplit('#', 1)[0]
        parameters_str_array=parameters_str.split('&')
        index=0
        while index<len(parameters_str_array):
            print("Parameter"+str(index)+": "+parameters_str_array[index])
            index=index+1

        fragment=remaining_path_str.rsplit('#', 1)[1]
        print("Fragment: ",fragment)
        connection.send(data)

connection.close()
```

Client.py

```
#Stela Nebiaj
#Fiorela Ciroku
#Abdullah Elkindy

#Import socket module
import socket
#Create a socket objectki
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

#Get local machine name
host = 'localhost'
port = 8080

sock.connect((host, port))

```

```
message="http://www.example.com/path/to/myfile.html?key1=value1&key2=value2#InTheI
sock.send(message.encode('utf-8'))
data = sock.recv(1024)
sock.close
print('Sent data: ',data)
```



```
Python 3.5.2 |Anaconda 4.2.0 (x86_64)| (default, Jul  2 2016, 17:52:12)
[GCC 4.2.1 Compatible Apple LLVM 4.2 (clang-425.0.28)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.

== RESTART: /Users/owner/Documents/workspace-python/Assignment 2/server.py ==
localhost
Got connection from ('127.0.0.1', 63183)
url info:

Protocol:  http
Domain:  example.com
Subdomain:  www
File path:  myfile.html
Parameter0:  key1=value1
Parameter1:  key2=value2
Fragment:  InTheDocument
```

```
#Stela Nebi
#Fiorela Ci
#Abdullah E

#Import soc
import sock
#Create a s
sock = sock

#Get local
host = 'loc
port = 8080

sock.connec
#name= inpu
#age= input
#matrikelnu
#message =
message="ht
sock.send(m
data = sock
sock.close
print('Sent
```

```
Python 3.5.2 |Anaconda 4.2.0 (x86_64)| (default, Jul  2 2016, 17:52:
[GCC 4.2.1 Compatible Apple LLVM 4.2 (clang-425.0.28)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current informat

== RESTART: /Users/owner/Documents/workspace-python/Assignment 2/cli
Sent data:  b'http://www.example.com/path/to/myfile.html?key1=value1
#InTheDocument'
>>>
```

```
remaining_path_str=path_str.rsplit('?', 1)[1]
```

Figure 3: Output

3 DNS Recursive Query Resolving (5 Points)

You have solved the ”Routing Table” question in Assignment 2. We updated the routing tables once more. resulting in the following tables creating the following topology

Table 1: Routing Table

Router1				**Router2**				**Router3**		
Destination	Next Hop	Interface		Destination	Next Hop	Interface		Destination	Next Hop	Interface
67.0.0.0	67.68.3.1	eth 0		205.30.7.0	205.30.7.1	eth 0		205.30.7.0	205.30.7.2	eth 0
62.0.0.0	62.4.31.7	eth 1		156.3.0.0	156.3.0.6	eth 1		88.0.0.0	88.6.32.1	eth 1
88.0.0.0	88.4.32.6	eth 2		26.0.0.0	26.3.2.1	eth 2		25.0.0.0	25.03.1.2	eth 2
141.71.0.0	141.71.20.1	eth 3		141.71.0.0	141.71.26.3	eth 3		121.0.0.0	121.0.3.1	eth 3
26.0.0.0	141.71.26.3	eth3		67.0.0.0	141.71.20.1	eth 3		156.3.0.0	205.30.7.1	eth 0
156.3.0.0	88.6.32.1	eth 2		62.0.0.0	141.71.20.1	eth 3		26.0.0.0	205.30.7.1	eth 0
205.30.7.0	141.71.26.3	eth 3		88.0.0.0	141.71.20.1	eth 3		141.71.0.0	205.30.7.1	eth 0
25.0.0.0	88.6.32.1	eth 2		25.0.0.0	205.30.7.2	eth 0		67.0.0.0	88.4.32.6	eth 1
121.0.0.0	88.6.32.1	eth 2		121.0.0.0	205.30.7.2	eth 0		62.0.0.0	88.4.32.6	eth 1

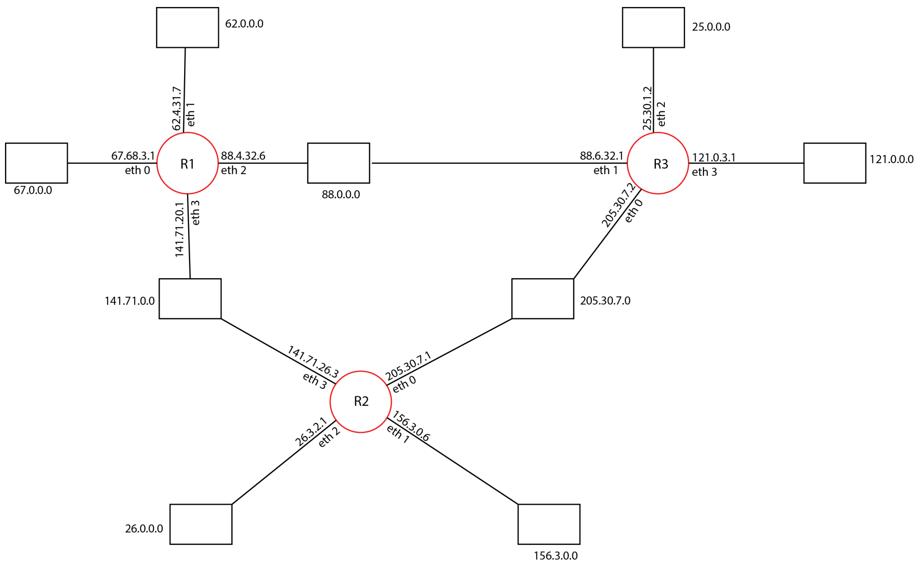


Figure 4: DNS Routing Network

Answer:

Let us assume a client with the following ip address 67.4.5.2 wants to resolve the following domain `subdomain.webscienceexampledomain.com` using the DNS.

You can further assume the root name server has the IP address of 25.8.2.1 and the name-server for `webscienceexampledomain.com` has the IP address 156.3.20.2. Finally the sub-domain is handled by a name server with the IP of 26.155.36.7.

Please explain how the traffic flows through the network in order to resolve the recursive

DNS query. You can assume ARP tables are cached so that no ARP-requests have to be made.

Client 67.4.5.2 creates an IP packet with the source address 67.4.5.2 and destination address 25.8.2.1 (IP of the root), inside there is the DNS request. This IP packet is send as an ethernet frame to Router 1 using 67.68.3.1. Router 1 receives it and sends it to 88.0.0.0 by 88.4.32.6. It leaves 88.0.0.0 through 88.6.32.1 and arrives to Router 3. Router 3 sends it to 25.0.0.0 by using 25.3.1.2. The destination 25.8.2.1 receives the frame and changes the destination to 156.3.20.2 which is the IP of the domain. The IP packet with the new destination goes back to Router 3 through 25.03.1.2 and Router 3 forwards it to 205.0.0.0 by 205.30.7.2.The encapsulated IP packet is forwarded to Router 2 with 205.30.7.1. From there is send to 156.3.0.0 by using 156.3.0.6. In this moment the IP packet has reached the IP of the domain. Here the destination is changed again, making it 26.155.36.7, which is the IP of the subdomain. To go there the IP packet is sent again to Router 2 using 156.3.0.6. Router 2 forwards the packet to the final destination 26.0.0.0 using 26.3.2.1. In the network 26.0.0.0 is also the IP 26.155.36.7, which is the final destination for our IP packet.

Important Notes

Submission

- Solutions have to be checked into the github repository. Use the directory name `groupname/assignment3/` in your group's repository.
- The name of the group and the names of all participating students must be listed on each submission.
- Solution format: all solutions as *one* PDF document. Programming code has to be submitted as Python code to the github repository. Upload *all* `.py` files of your program! Use `UTF-8` as the file encoding. *Other encodings will not be taken into account!*
- Check that your code compiles without errors.
- Make sure your code is formatted to be easy to read.
  - Make sure you code has consistent indentation.
  - Make sure you comment and document your code adequately in English.
  - Choose consistent and intuitive names for your identifiers.
- Do *not* use any accents, spaces or special characters in your filenames.

Acknowledgment

This latex template was created by Lukas Schmelzeisen for the tutorials of "Web Information Retrieval".

LaTeX

Currently the code can only be build using LuaLaTeX, so make sure you have that installed. If on Overleaf, there's an error, go to settings and change the LaTeXengine to `LuaLaTeX`.