DATA.STAT.840 Statistical Methods for Text Data Analysis
Exercises for Lecture 2: Basic text processing

Student Name: Ahmad Sharif
Student ID: K436765
ahmad.sharif@tuni.fi

Exercise 2.1: Data acquisition - Building a better web crawler.

The web crawler code shown on Lecture 2 is poor in at least two respects:
1. It can crawl the same page multiple times, if a link on a later crawled page points to the already-crawled page.

2. It inserts all links from each page in order as pages to be crawled. If some page contains thousands of links, the crawling will crawl those first and may never get to the links from the next page, especially if the total number of pages are limited.

To fix this duplicate issue, at first before insert into the list, I check if the url already exist list data-structure.

```
import requests
import bs4

webpage_url = "https://www.sis.uta.fi/~tojape/"
webpage_html = requests.get(webpage_url)
webpage_parsed_html = bs4.BeautifulSoup(webpage_html.content, 'html.parser')


def getpageurls(webpage_parsed):
        pagelinkelements=webpage_parsed.find_all('a')
        pageurls = [];
        for pagelink in pagelinkelements:
                pageurl_isok=1
                try:
                        pageurl = pagelink['href']
                except:
                        pageurl_isok=0
                if pageurl_isok == 1:
                        if (pageurl.find('.pdf') !=-1)|(pageurl.find('.ps')!=-1):
                                pageurl_isok = 0
                        if (pageurl.find('http') ==-1 )|(pageurl.find('.fi')==-1):
                                pageurl_isok = 0
                if pageurl_isok == 1 and pageurl not in pageurls:  # Before Append we need to check
                        pageurls.append(pageurl)
        return(pageurls)
mywebpage_urls = getpageurls(webpage_parsed_html)
print(mywebpage_urls)
```

In [11]:
```
import requests
import bs4
import urllib.request
```

In [26]:
```
webpage_url = "https://www.sis.uta.fi/~tojape/"
webpage_html = requests.get(webpage_url)
webpage_parsed_html = bs4.BeautifulSoup(webpage_html.content, 'html.parser')
```

1. It can crawl the same page multiple times, if a link on a later crawled page points to the already-crawled page.
2. It inserts all links from each page in order as pages to be crawled. If some page contains thousands of links, the crawling will crawl those first and may never get to the links from the next page, especially if the total number of pages are limited. To fix this duplicate issue, at first before insert into the list, I check if the url already exist list data- structure.

In [27]:
```
def getpageurls(webpage_parsed):
    pagelinkelements=webpage_parsed.find_all('a')
    pageurls = [];
    for pagelink in pagelinkelements:
        pageurl_isok=1
        try:
            pageurl = pagelink['href']
        except:
            pageurl_isok=0
        if (pageurl.find('.pdf') !=-1)|(pageurl.find('.ps')!=-1):
            pageurl_isok = 0
        if (pageurl.find('http') ==-1 )|(pageurl.find('.fi')==-1):
            pageurl_isok = 0
        if pageurl_isok == 1 and pageurl not in pageurls: # Before Append we need to
            pageurls.append(pageurl)

    return(pageurls)
mywebpage_urls = getpageurls(webpage_parsed_html)
print(mywebpage_urls)
```

```
['https://www.tuni.fi/en', 'https://www.tuni.fi/en/about-us/faculty-information-technology-and-communication-sciences', 'https://www.tuni.fi/en/about-us/computing-sciences', 'http://cs.aalto.fi/en/', 'http://www.cis.hut.fi/projects/mi', 'http://users.ics.aalto.fi/jtpelto/', 'http://research.ics.aalto.fi/coin/', 'https://www.tuni.fi/en/study-with-us/computing-sciences-data-science?navref=curated--list', 'https://www.tuni.fi/en/study-with-us/computing-sciences-statistical-data-analytics?navref=curated--list', 'https://www.tuni.fi/studentsguide/curriculum/degree-programmes/uta-tohjelma-1717?year=2019', 'https://www.tuni.fi/studentsguide/curriculum/course-units/otm-d42bf3fb-ecd7-43ee-919e-3a18e0b7d885?year=2020&q=null', 'https://www.tuni.fi/studentsguide/curriculum/course-units/otm-386280c0-c76b-4837-b4e3-61a9d53130b4?year=2020', 'https://www.tuni.fi/studentsguide/curriculum/course-units/uta-ykoodi-48003?year=2019', 'https://www.tuni.fi/studentsguide/curriculum/course-units/uta-ykoodi-48010?year=2019', 'https://www.tuni.fi/studentsguide/curriculum/course-units/uta-ykoodi-38903?year=2019', 'https://www10.uta.fi/opas/teaching/course.htm?id=32034', 'https://www10.uta.fi/opas/teaching/course.htm?id=32030', 'https://www10.uta.fi/opas/opetusohjelma/marjapuuro.htm?id=34169', 'https://www10.uta.fi/opas/opetusohjelma/marjapuuro.htm?id=32033', 'https://www10.uta.fi/opas/opetusohjelma/marjapuuro.htm?id=32030', 'https://www10.uta.fi/opas/opetusohjelma/marjapuuro.htm?id=29901', 'https://www10.uta.fi/opas/opetusohjelma/marjapuuro.htm?id=29910', 'https://www10.uta.fi/opas/opetusohjelma/marjapuuro.htm?id=29909', 'https://www10.uta.fi/opas/opetusohjelma/marjapuuro.htm?id=25061', 'https://www10.uta.fi/opas/opetusohjelma/marjapuuro.htm?id=24888', 'https://www10.uta.fi/opas/opetusohjelma/marjapuuro.htm?id=27922', 'https://www10.uta.fi/opas/opetusohjelma/marjapuuro.htm?id=23239', 'https://www10.uta.fi/opas/opetusohjelma/marjapuuro.htm?id=20713', 'https://noppa.aalto.fi/noppa/kurssi/t-61.2020/etusivu', 'https://noppa.aalto.fi/noppa/kurssi/t-61.5010/etusivu', 'https://noppa.aalto.fi/noppa/kurssi/t-61.3050/etusivu', 'https://noppa.aalto.fi/noppa/kurssi/t-61.6040/etusivu', 'https://noppa.tkk.fi/noppa/kurssi/t-61.6040/etusivu', 'https://noppa.tkk.fi/noppa/kurssi/t-61.3040/etusivu', 'https://www.tuni.fi/en/mykola-andrushchenko', 'http://people.uta.fi/~kauppinen.joonas.t/', 'https://people.uta.fi/~olli.kuparinen/', 'http://users.ics.aalto.fi/ziyuang/', 'http://users.ics.aalto.fi/hani/', 'http://users.ics.aalto.fi/jstrahl/', 'https://www.tuni.fi/en/elizaveta-zimina', 'http://users.ics.aalto.fi/zhexie/', 'http://users.ics.tkk.fi/faisal/', 'https://www.tuni.fi/en/joni-pajarinen', 'http://users.ics.tkk.fi/lgillber/', 'http://users.ics.aalto.fi/msandhol/', 'https://www.tuni.fi/en/essi-syrjala', 'http://users.ics.tkk.fi/kongeo/', 'http://users.ics.tkk.fi/jviinika/', 'http://users.ics.aalto.fi/mlosoi/', 'https://journal.fi/sananjalka/article/view/80056', 'http://www.cis.hut.fi/projects/mi/abstracts/jmlr10.html', 'http://www.cis.hut.fi/projects/mi/abstracts/ida09b.html', 'http://www.cis.hut.fi/projects/mi/abstracts/csda09.html', 'http://www.cis.hut.fi/projects/mi/abstracts/tnn04.html', 'http://www.cis.hut.fi/projects/mi/abstracts/nn04.html', 'http://www.cis.hut.fi/projects/mi/abstracts/trnn00.html', 'http://research.ics.aalto.fi/mi/software/ne/index.shtml', 'http://www.cis.hut.fi/projects/mi/abstracts/icassp10.html', 'http://www.cis.hut.fi/projects/mi/abstracts/wsom09.html', 'http://www.cis.hut.fi/projects/mi/abstracts/icassp09_snerv.html', 'http://www.cis.hut.fi/projects/mi/abstracts/ecml07.html', 'http://www.cis.hut.fi/projects/mi/abstracts/mlsp07.html', 'http://www.cis.hut.fi/projects/mi/abstracts/pmsb2006.html', 'http://www.cis.hut.fi/projects/mi/abstracts/icml04.html', 'http://www.cis.hut.fi/projects/mi/abstracts/wsom03b.html', 'http://www.cis.hut.fi/projects/mi/abstracts/icml03.html', 'http://www.cis.hut.fi/projects/mi/abstracts/ecml03.html', 'http://www.cis.hut.fi/projects/mi/abstracts/iconip02.html', 'http://www.cis.hut.fi/projects/mi/abstracts/icann02.html', 'http://www.cis.hut.fi/projects/mi/abstracts/ijcnn01.html', 'https://politiikasta.fi/parlamentaarisen-politiikan-ajat/', 'http://www.cis.hut.fi/projects/mi/abstracts/nips08_lms_rsl.html', 'http://www.nbl.fi/~nbl924/renkula/', 'http://www.cis.hut.fi/projects/mi/abstracts/nips06_did.html', 'http://www.cis.hut.fi/projects/mi/abstracts/nips06_lce.html', 'http://www.cis.hut.fi/projects/mi/abstracts/eccb06poster.html', 'http://lib.hut.fi/Diss/2004/isbn9512273454/']
```

In [5]:
```
webpage_url = "https://www.gutenberg.org/browse/scores/top#books-last30"
webpage_html = requests.get(webpage_url)
webpage_parsed_html = bs4.BeautifulSoup(webpage_html.content, 'html.parser')
```

## 2 ) a)

In [9]:
```
pageList = webpage_parsed_html.find("h2", {"id": "books-last30"})
ol = pageList.find_next_sibling("ol")
book_list = {}

def collect_all_download_link(count):
    for x in ol.findAll('li'):
```

```
        # print(x.a.text, '19')
        url = x.a['href']
        link = 'https://www.gutenberg.org/files/' + url.split('/')[2]
        if(len(book_list) < count):
            book_list[x.a.text] = {
                "book_name": x.a.text,
                "download_link": link,
            }
collect_all_download_link(20)
print(len(book_list))
# 20
```

```
20
```

In [10]:

```
## 2 > b Name And Link

for key in book_list:
    print("Book Name", key)
```

```
Book Name Romeo and Juliet by William Shakespeare (68116)
Book Name Frankenstein; Or, The Modern Prometheus by Mary Wollstonecraft Shelley (610
98)
Book Name Pride and Prejudice by Jane Austen (58864)
Book Name Moby Dick; Or, The Whale by Herman Melville (46191)
Book Name The Scarlet Letter by Nathaniel Hawthorne (30378)
Book Name Alice's Adventures in Wonderland by Lewis Carroll (30009)
Book Name Dracula by Bram Stoker (26871)
Book Name Middlemarch by George Eliot (26013)
Book Name The Picture of Dorian Gray by Oscar Wilde (24701)
Book Name The Complete Works of William Shakespeare by William Shakespeare (24569)
Book Name A Room with a View by E. M.  Forster (23999)
Book Name Little Women; Or, Meg, Jo, Beth, and Amy by Louisa May Alcott (23366)
Book Name The Yellow Wallpaper by Charlotte Perkins Gilman (23343)
Book Name A Modest Proposal by Jonathan Swift (21248)
Book Name Metamorphosis by Franz Kafka (21075)
Book Name The Blue Castle: a novel by L. M.  Montgomery (21030)
Book Name The Enchanted April by Elizabeth Von Arnim (20223)
Book Name A Doll's House : a play by Henrik Ibsen (19179)
Book Name Cranford by Elizabeth Cleghorn Gaskell (18718)
Book Name The Adventures of Ferdinand Count Fathom — Complete by T.  Smollett (18585)
```