Web Mining Digital Assignment 1

## Name: Om Ashish Mishra

## Registration Number: 16BCE0789

## Slot: F2

# The Question:

Choose an open source web crawler of your choice and measure aspects of a crawl, study the characteristics of the crawl, download web pages from the crawl and gather webpage metadata, all from pre-selected news website. Provide the steps to configure and compile the crawler. You should crawl only the news websites, and your crawler should be configured so that it does not visit pages outside of the given news website.

The maximum pages to fetch can be set and it should be set to within 1,000 pages to ensure a reasonable execution time.

Give the implementation issues of the crawler and how you rectified it.

# The Answer:

The Web Crawler I used is **Scrapy.**

## Features a crawler *should* provide

**Distributed:**

The crawler should have the ability to execute in a distributed fashion across multiple machines.

**Scalable:**

The crawler architecture should permit scaling up the crawl rate by adding extra machines and bandwidth.

**Performance and efficiency:**

The crawl system should make efficient use of various system resources including processor, storage and network bandwidth.

**Quality:**

Given that a significant fraction of all web pages are of poor utility for serving user query needs, the crawler should be biased towards fetching ``useful'' pages first.

**Freshness:**

In many applications, the crawler should operate in continuous mode: it should obtain fresh copies of previously fetched pages. A search engine crawler, for instance, can thus ensure that the search engine's index contains a fairly current representation of each indexed web page. For such continuous crawling, a crawler should be able to crawl a page with a frequency that approximates the rate of change of that page.

**Extensible:**

Crawlers should be designed to be extensible in many ways - to cope with new data formats, new fetch protocols, and so on. This demands that the crawler architecture be modular.

**The Characteristics of the crawler (Scrapy) are:**

Scrapy is a Python framework for large scale web scraping. It gives you all the tools you need to efficiently **extract** data from websites, **process** them as you want, and store them in your preferred **structure**and format.

As diverse the internet is, there is no “one size fits all” approach in extracting data from websites. Many a time ad hoc approaches are taken and if you start writing code for every little task you perform, you will eventually end up creating your own scraping framework. Scrapy is that framework.

*With Scrapy you don’t need to reinvent the wheel.*

## Why Use Scrapy?

* It is easier to build and scale large crawling projects.
* It has a built-in mechanism called Selectors, for extracting the data from websites.
* It handles the requests asynchronously and it is fast.
* It automatically adjusts crawling speed using [Auto-throttling mechanism](https://doc.scrapy.org/en/latest/topics/autothrottle.html).
* Ensures developer accessibility.

## Features of Scrapy

* Scrapy is an open source and free to use web crawling framework.
* Scrapy generates feed exports in formats such as JSON, CSV, and XML.
* Scrapy has built-in support for selecting and extracting data from sources either by XPath or CSS expressions.
* Scrapy based on crawler, allows extracting data from the web pages automatically.

## Advantages

* Scrapy is easily extensible, fast, and powerful.
* It is a cross-platform application framework (Windows, Linux, Mac OS and BSD).
* Scrapy requests are scheduled and processed asynchronously.
* Scrapy comes with built-in service called **Scrapyd** which allows to upload projects and control spiders using JSON web service.
* It is possible to scrap any website, though that website does not have API for raw data access.

## Disadvantages

* Scrapy is only for Python 2.7. +
* Installation is different for different operating systems.

**Download web pages from the crawl**

The Code: ( Taken Help From Internet = Partly Copied and Partly Self-implicated)

#Import Libraries

import time #For Delay

import urllib.request #Extracting web pages

import re

import scrapy

class firstSpider(scrapy.Spider):

name = "first"

allowed\_domains = ["wikipedia.org"]

start\_page = [

"https://en.wikipedia.org/wiki/News","https://en.wikipedia.org/wiki/Current\_affairs"

]

#Defining pages

starting\_page = ["https://en.wikipedia.org/wiki/News","https://en.wikipedia.org/wiki/Current\_affairs"]

seed\_page = "https://en.wikipedia.org" #Crawling the English Wikipedia

#Downloading entire Web Document (Raw Page Content)

def download\_page(url):

try:

headers = {}

headers['User-Agent'] = "Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.27 Safari/537.17"

req = urllib.request.Request(url, headers = headers)

resp = urllib.request.urlopen(req)

respData = str(resp.read())

return respData

except Exception as e:

print(str(e))

#Extract the title tag

def extract\_title(page):

start\_title = page.find("<span dir")

end\_start\_title = page.find(">",start\_title+1)

stop\_title = page.find("</span>", end\_start\_title + 1)

title = page[end\_start\_title + 1 : stop\_title]

return title

#####################################################################

#Extract the see also section elements

def extract\_see\_also(page):

if 'id="See\_also">' in page:

start\_see\_also = page.find('id="See\_also">')

start\_list\_items = page.find('<li>', start\_see\_also + 1)

end\_see\_also = page.find('<h2>', start\_list\_items + 1)

see\_also\_section = page[start\_list\_items: end\_see\_also]

pure\_item\_raw = (re.sub(r'<.+?>', '', see\_also\_section)).replace('\n', ',')

pure\_item\_raw2 = pure\_item\_raw.replace(',,', ',')

pure\_item = pure\_item\_raw2.replace(',,', ',')

flag = 0

else:

pure\_item = "No Related Links"

flag = 1

return pure\_item, flag

#Extract just the Introduction part of the page

def extract\_introduction(page):

start\_introduction = page.find("<p>")

stop\_introduction = page.find('<div id="toctitle">', start\_introduction + 1)

#If the page onl has introduction

if '<div id="toctitle">' not in page:

stop\_introduction = page.find('</p>', start\_introduction + 1)

else:

pass

raw\_introduction = page[start\_introduction : stop\_introduction]

return raw\_introduction

#Extract all the links

#Finding 'Next Link' on a given web page

def get\_next\_link(s):

start\_link = s.find("<a href")

if start\_link == -1: #If no links are found then give an error!

end\_quote = 0

link = "no\_links"

return link, end\_quote

else:

start\_quote = s.find('"', start\_link)

end\_quote = s.find('"',start\_quote+1)

link = str(s[start\_quote+1:end\_quote])

return link, end\_quote

#Getting all links with the help of 'get\_next\_links'

def get\_all\_links(page):

links = []

while True:

link, end\_link = get\_next\_link(page)

if link == "no\_links":

break

else:

links.append(link) #Append all the links in the list named 'Links'

#time.sleep(0.1)

page = page[end\_link:]

return links

#Remove all the HTML tags from the introduction to get the pure text

#Eliminate all the text inside '<' & '>'

def extract\_pure\_introduction(page):

pure\_introduction = (re.sub(r'<.+?>', '', page)) #From '<' to the next '>'

return pure\_introduction

#Crawl Initiation

#Check for file type in URL so crawler does not crawl images and text files

def extension\_scan(url):

a = ['.png','.jpg','.jpeg','.gif','.tif','.txt']

j = 0

while j < (len(a)):

if a[j] in url:

#print("There!")

flag2 = 1

break

else:

#print("Not There!")

flag2 = 0

j = j+1

#print(flag2)

return flag2

#URL parsing for incomplete or duplicate URLs

def url\_parse(url):

try:

from urllib.parse import urlparse

except ImportError:

from urlparse import urlparse

url = url #.lower() #Make it lower case

s = urlparse(url) #parse the given url

seed\_page\_n = seed\_page #.lower() #Make it lower case

#t = urlparse(seed\_page\_n) #parse the seed page (reference page)

i = 0

flag = 0

while i<=9:

if url == "/":

url = seed\_page\_n

flag = 0

elif not s.scheme:

url = "http://" + url

flag = 0

elif "#" in url:

url = url[:url.find("#")]

flag = 0

elif "?" in url:

url = url[:url.find("?")]

flag = 0

elif s.netloc == "":

url = seed\_page + s.path

flag = 0

#elif "www" not in url:

# url = "www."[:7] + url[7:]

# flag = 0

elif url[len(url)-1] == "/":

url = url[:-1]

flag = 0

#elif s.netloc != t.netloc:

# url = url

# flag = 1

# break

else:

url = url

flag = 0

break

i = i+1

s = urlparse(url) #Parse after every loop to update the values of url parameters

return(url, flag)

t0 = time.time()

database = {} #Create a dictionary

#Main Crawl function that calls all the above function and crawls the entire site sequentially

def web\_crawl():

to\_crawl = [starting\_page] #Define list name 'Seed Page'

#print(to\_crawl)

crawled=[] #Define list name 'Seed Page'

#database = {} #Create a dictionary

#k = 0;

for k in range(0, 3):

i=0 #Initiate Variable to count No. of Iterations

while i<3: #Continue Looping till the 'to\_crawl' list is not empty

urll = to\_crawl.pop(0) #If there are elements in to\_crawl then pop out the first element

urll,flag = url\_parse(urll)

#print(urll)

flag2 = extension\_scan(urll)

time.sleep(3)

#If flag = 1, then the URL is outside the seed domain URL

if flag == 1 or flag2 == 1:

pass #Do Nothing

else:

if urll in crawled: #Else check if the URL is already crawled

pass #Do Nothing

else: #If the URL is not already crawled, then crawl i and extract all the links from it

print("Link = " + urll)

raw\_html = download\_page(urll)

#print(raw\_html)

title\_upper = str(extract\_title(raw\_html))

title = title\_upper.lower() #Lower title to match user queries

print("Title = " + title)

see\_also,flag2 = extract\_see\_also(raw\_html)

print("Related Links = " + see\_also)

raw\_introduction = extract\_introduction(raw\_html)

#print("Raw Introduction = " + raw\_introduction)

to\_crawl = to\_crawl + get\_all\_links(raw\_introduction)

crawled.append(urll)

pure\_introduction = extract\_pure\_introduction(raw\_introduction)

print("Introduction = " + pure\_introduction.replace(' ',' '))

database [title] = pure\_introduction #Add title and its introduction to the dict

#Writing the output data into a text file

file = open('database.txt', 'a') #Open the text file called database.txt

file.write(title + ": " + "\n") #Write the title of the page

file.write(pure\_introduction + "\n\n") #write the introduction of that page

file.close() #Close the file

#Remove duplicated from to\_crawl

n = 1

j = 0

#k = 0

while j < (len(to\_crawl)-n):

if to\_crawl[j] in to\_crawl[j+1:(len(to\_crawl)-1)]:

to\_crawl.pop(j)

n = n+1

else:

pass #Do Nothing

j = j+1

i=i+1

print(i)

print(k)

#print(to\_crawl)

#print("Iteration No. = " + str(i))

#print("To Crawl = " + str(len(to\_crawl)))

#print("Crawled = " + str(len(crawled)))

return ""

print (web\_crawl())

t1 = time.time()

total\_time = t1-t0

print(total\_time)

**The Output(The first and the last):**

runfile('C:/Users/OM/(OM)/5Fifth Semester/da1\_web.py', wdir='C:/Users/OM/(OM)/5Fifth Semester')

Link = https://en.wikipedia.org/wiki/News

Title = \n<html class="client-nojs" lang="en" dir="ltr">\n<head>\n<meta charset="utf-8"/>\n<title>news - wikipedia</title>\n<script>document.documentelement.classname = document.documentelement.classname.replace( /(^|\\s)client-nojs(\\s|$)/, "$1client-js$2" );</script>\n<script>(window.rlq=window.rlq||[]).push(function(){mw.config.set({"wgcanonicalnamespace":"","wgcanonicalspecialpagename":false,"wgnamespacenumber":0,"wgpagename":"news","wgtitle":"news","wgcurrevisionid":854358990,"wgrevisionid":854358990,"wgarticleid":20781999,"wgisarticle":true,"wgisredirect":false,"wgaction":"view","wgusername":null,"wgusergroups":["\*"],"wgcategories":["pages using citations with accessdate and no url","webarchive template wayback links","cs1 maint: explicit use of et al.","wikipedia indefinitely move-protected pages","all articles lacking reliable references","articles lacking reliable references from october 2014","all articles with unsourced statements","articles with unsourced statements from october 2014","articles with curlie links","communication","news","radio terminology","sociology of knowledge","television terminology"],"wgbreakframes":false,"wgpagecontentlanguage":"en","wgpagecontentmodel":"wikitext","wgseparatortransformtable":["",""],"wgdigittransformtable":["",""],"wgdefaultdateformat":"dmy","wgmonthnames":["","january","february","march","april","may","june","july","august","september","october","november","december"],"wgmonthnamesshort":["","jan","feb","mar","apr","may","jun","jul","aug","sep","oct","nov","dec"],"wgrelevantpagename":"news","wgrelevantarticleid":20781999,"wgrequestid":"w3ll-wpamfuaahfwsncaaaaq","wgcspnonce":false,"wgisprobablyeditable":true,"wgrelevantpageisprobablyeditable":true,"wgrestrictionedit":[],"wgrestrictionmove":["sysop"],"wgflaggedrevsparams":{"tags":{}},"wgstablerevisionid":null,"wgcategorytreepagecategoryoptions":"{\\"mode\\":0,\\"hideprefix\\":20,\\"showcount\\":true,\\"namespaces\\":false}","wgwikieditorenabledmodules":[],"wgbetafeaturesfeatures":[],"wgmediavieweronclick":true,"wgmediaviewerenabledbydefault":true,"wgpopupsshouldsendmoduletouser":true,"wgpopupsconflictswithnavpopupgadget":false,"wgvisualeditor":{"pagelanguagecode":"en","pagelanguagedir":"ltr","pagevariantfallbacks":"en","usepageimages":true,"usepagedescriptions":true},"wgmfexpandallsectionsuseroption":true,"wgmfenablefontchanger":true,"wgmfdisplaywikibasedescriptions":{"search":true,"nearby":true,"watchlist":true,"tagline":false},"wgrelatedarticles":null,"wgrelatedarticlesusecirrussearch":true,"wgrelatedarticlesonlyusecirrussearch":false,"wgulscurrentautonym":"english","wgnoticeproject":"wikipedia","wgcentralnoticecookiestodelete":[],"wgcentralnoticecategoriesusinglegacy":["fundraising","fundraising"],"wgwikibaseitemid":"q38926","wgscorenotelanguages":{"arabic":"\xd8\xa7\xd9\x84\xd8\xb9\xd8\xb1\xd8\xa8\xd9\x8a\xd8\xa9","catalan":"catal\xc3\xa0","deutsch":"deutsch","english":"english","espanol":"espa\xc3\xb1ol","italiano":"italiano","nederlands":"nederlands","norsk":"norsk","portugues":"portugu\xc3\xaas","suomi":"suomi","svenska":"svenska","vlaams":"west-vlams"},"wgscoredefaultnotelanguage":"nederlands","wgcentralauthmobiledomain":false,"wgcodemirrorenabled":true,"wgvisualeditortoolbarscrolloffset":0,"wgvisualeditorunsupportededitparams":["undo","undoafter","veswitched"],"wgeditsubmitbuttonlabelpublish":true});mw.loader.state({"ext.gadget.charinsert-styles":"ready","ext.globalcssjs.user.styles":"ready","ext.globalcssjs.site.styles":"ready","site.styles":"ready","noscript":"ready","user.styles":"ready","ext.globalcssjs.user":"ready","ext.globalcssjs.site":"ready","user":"ready","user.options":"ready","user.tokens":"loading","ext.cite.styles":"ready","mediawiki.legacy.shared":"ready","mediawiki.legacy.commonprint":"ready","mediawiki.toc.styles":"ready","wikibase.client.init":"ready","ext.visualeditor.desktoparticletarget.noscript":"ready","ext.uls.interlanguage":"ready","ext.wikimediabadges":"ready","mediawiki.skinning.interface":"ready","skins.vector.styles":"ready"});mw.loader.implement("user.tokens@1dqfd7l",function($,jquery,require,module){/\*@nomin\*/mw.user.tokens.set({"edittoken":"+\\\\","patroltoken":"+\\\\","watchtoken":"+\\\\","csrftoken":"+\\\\"});\n});mw.loader.load(["ext.cite.a11y","site","mediawiki.page.startup","mediawiki.user","mediawiki.page.ready","mediawiki.toc","mediawiki.searchsuggest","ext.gadget.teahouse","ext.gadget.referencetooltips","ext.gadget.watchlist-notice","ext.gadget.drn-wizard","ext.gadget.charinsert","ext.gadget.reftoolbar","ext.gadget.extra-toolbar-buttons","ext.gadget.switcher","ext.centralauth.centralautologin","mmv.head","mmv.bootstrap.autostart","ext.popups","ext.visualeditor.desktoparticletarget.init","ext.visualeditor.targetloader","ext.eventlogging.subscriber","ext.wikimediaevents","ext.navigationtiming","ext.uls.eventlogger","ext.uls.init","ext.uls.compactlinks","ext.uls.interface","ext.3d","ext.centralnotice.geoip","ext.centralnotice.startup","skins.vector.js"]);});</script>\n<link rel="stylesheet" href="/w/load.php?debug=false&amp;lang=en&amp;modules=ext.cite.styles%7cext.uls.interlanguage%7cext.visualeditor.desktoparticletarget.noscript%7cext.wikimediabadges%7cmediawiki.legacy.commonprint%2cshared%7cmediawiki.skinning.interface%7cmediawiki.toc.styles%7cskins.vector.styles%7cwikibase.client.init&amp;only=styles&amp;skin=vector"/>\n<script async="" src="/w/load.php?debug=false&amp;lang=en&amp;modules=startup&amp;only=scripts&amp;skin=vector"></script>\n<meta name="resourceloaderdynamicstyles" content=""/>\n<link rel="stylesheet" href="/w/load.php?debug=false&amp;lang=en&amp;modules=ext.gadget.charinsert-styles&amp;only=styles&amp;skin=vector"/>\n<link rel="stylesheet" href="/w/load.php?debug=false&amp;lang=en&amp;modules=site.styles&amp;only=styles&amp;skin=vector"/>\n<meta name="generator" content="mediawiki 1.32.0-wmf.16"/>\n<meta name="referrer" content="origin"/>\n<meta name="referrer" content="origin-when-crossorigin"/>\n<meta name="referrer" content="origin-when-cross-origin"/>\n<meta property="og:image" content="https://upload.wikimedia.org/wikipedia/commons/3/33/al\_jazeera\_english\_doha\_newsroom\_1.jpg"/>\n<link rel="alternate" href="android-app://org.wikipedia/http/en.m.wikipedia.org/wiki/news"/>\n<link rel="alternate" type="application/x-wiki" title="edit this page" href="/w/index.php?title=news&amp;action=edit"/>\n<link rel="edit" title="edit this page" href="/w/index.php?title=news&amp;action=edit"/>\n<link rel="apple-touch-icon" href="/static/apple-touch/wikipedia.png"/>\n<link rel="shortcut icon" href="/static/favicon/wikipedia.ico"/>\n<link rel="search" type="application/opensearchdescription+xml" href="/w/opensearch\_desc.php" title="wikipedia (en)"/>\n<link rel="edituri" type="application/rsd+xml" href="//en.wikipedia.org/w/api.php?action=rsd"/>\n<link rel="license" href="//creativecommons.org/licenses/by-sa/3.0/"/>\n<link rel="canonical" href="https://en.wikipedia.org/wiki/news"/>\n<link rel="dns-prefetch" href="//login.wikimedia.org"/>\n<link rel="dns-prefetch" href="//meta.wikimedia.org" />\n<!--[if lt ie 9]><script src="/w/load.php?debug=false&amp;lang=en&amp;modules=html5shiv&amp;only=scripts&amp;skin=vector&amp;sync=1"></script><![endif]-->\n</head>\n<body class="mediawiki ltr sitedir-ltr mw-hide-empty-elt ns-0 ns-subject page-news rootpage-news skin-vector action-view">\t\t<div id="mw-page-base" class="noprint"></div>\n\t\t<div id="mw-head-base" class="noprint"></div>\n\t\t<div id="content" class="mw-body" role="main">\n\t\t\t<a id="top"></a>\n\t\t\t<div id="sitenotice" class="mw-body-content"><!-- centralnotice --></div><div class="mw-indicators mw-body-content">\n</div>\n<h1 id="firstheading" class="firstheading" lang="en">news</h1>\t\t\t<div id="bodycontent" class="mw-body-content">\n\t\t\t\t<div id="sitesub" class="noprint">from wikipedia, the free encyclopedia</div>\t\t\t\t<div id="contentsub"></div>\n\t\t\t\t<div id="jump-to-nav"></div>\t\t\t\t<a class="mw-jump-link" href="#mw-head">jump to navigation</a>\n\t\t\t\t<a class="mw-jump-link" href="#p-search">jump to search</a>\n\t\t\t\t<div id="mw-content-text" lang="en" dir="ltr" class="mw-content-ltr"><div class="mw-parser-output"><div role="note" class="hatnote navigation-not-searchable">for other uses, see <a href="/wiki/news\_(disambiguation)" class="mw-disambig" title="news (disambiguation)">news (disambiguation)</a>.</div>\n<div role="note" class="hatnote navigation-not-searchable">"current events" redirects here. for the album by john abercrombie, see <a href="/wiki/current\_events\_(album)" title="current events (album)"><i>current events</i> (album)</a>. for the portal, see <a href="/wiki/portal:current\_events" class="mw-redirect" title="portal:current events">portal:current events</a>.</div>\n<div class="thumb tright"><div class="thumbinner" style="width:302px;"><a href="/wiki/file:al\_jazeera\_english\_doha\_newsroom\_1.jpg" class="image"><img alt="" src="//upload.wikimedia.org/wikipedia/commons/thumb/3/33/al\_jazeera\_english\_doha\_newsroom\_1.jpg/300px-al\_jazeera\_english\_doha\_newsroom\_1.jpg" width="300" height="225" class="thumbimage" srcset="//upload.wikimedia.org/wikipedia/commons/thumb/3/33/al\_jazeera\_english\_doha\_newsroom\_1.jpg/450px-al\_jazeera\_english\_doha\_newsroom\_1.jpg 1.5x, //upload.wikimedia.org/wikipedia/commons/thumb/3/33/al\_jazeera\_english\_doha\_newsroom\_1.jpg/600px-al\_jazeera\_english\_doha\_newsroom\_1.jpg 2x" data-file-width="800" data-file-height="600" /></a> <div class="thumbcaption"><div class="magnify"><a href="/wiki/file:al\_jazeera\_english\_doha\_newsroom\_1.jpg" class="internal" title="enlarge"></a></div><a href="/wiki/al\_jazeera\_english" title="al jazeera english">al jazeera english</a> newsroom, <a href="/wiki/doha" title="doha">doha</a>, 2008</div></div></div>\n<p><b>news</b> is <a href="/wiki/information" title="information">information</a> about current events. this may be provided through many different <a href="/wiki/media\_(communication)" title="media (communication)">media</a>: <a href="/wiki/word\_of\_mouth" title="word of mouth">word of mouth</a>, <a href="/wiki/printing" title="printing">printing</a>, <a href="/wiki/postal\_system" class="mw-redirect" title="postal system">postal systems</a>, <a href="/wiki/broadcasting" title="broadcasting">broadcasting</a>, <a href="/wiki/electronic\_communication" class="mw-redirect" title="electronic communication">electronic communication</a>, and also on the <a href="/wiki/testimony" title="testimony">testimony</a> of observers and <a href="/wiki/witness" title="witness">witnesses</a> to events. it is also used as a platform to manufacture opinion for the population.\n</p><p>common topics for news reports include war, government, politics, education, health, the environment, economy, business, fashion, and entertainment, as well as athletic events, quirky or unusual events. <a href="/wiki/government" title="government">government</a> proclamations, concerning royal ceremonies, laws, taxes, public health, criminals, have been dubbed news since ancient times.\nhumans exhibit a nearly universal desire to learn and share news, which they satisfy by talking to each other and sharing information. technological and social developments, often driven by government communication and espionage networks, have increased the speed with which news can spread, as well as influenced its content. the genre of news as we know it today is closely associated with the <a href="/wiki/newspaper" title="newspaper">newspaper</a>, which originated in china as a court bulletin and spread, with <a href="/wiki/paper" title="paper">paper</a> and <a href="/wiki/printing\_press" title="printing press">printing press</a>, to europe.\n</p>\n<div id="toc" class="toc"><input type="checkbox" role="button" id="toctogglecheckbox" class="toctogglecheckbox" style="display:none" /><div class="toctitle" lang="en" dir="ltr"><h2>contents</h2><span class="toctogglespan"><label class="toctogglelabel" for="toctogglecheckbox"></label>

Related Links = Toronto School of communication theory\n

Introduction = News is information about current events. This may be provided through many different media: word of mouth, printing, postal systems, broadcasting, electronic communication, and also on the testimony of observers and witnesses to events. It is also used as a platform to manufacture opinion for the population.\n

1

0

Link = https://en.wikipedia.org/wiki/Witness

Title = \n<html class="client-nojs" lang="en" dir="ltr">\n<head>\n<meta charset="utf-8"/>\n<title>witness - wikipedia</title>\n<script>document.documentelement.classname = document.documentelement.classname.replace( /(^|\\s)client-nojs(\\s|$)/, "$1client-js$2" );</script>\n<script>(window.rlq=window.rlq||[]).push(function(){mw.config.set({"wgcanonicalnamespace":"","wgcanonicalspecialpagename":false,"wgnamespacenumber":0,"wgpagename":"witness","wgtitle":"witness","wgcurrevisionid":851192754,"wgrevisionid":851192754,"wgarticleid":50956,"wgisarticle":true,"wgisredirect":false,"wgaction":"view","wgusername":null,"wgusergroups":["\*"],"wgcategories":["cs1: julian\xe2\x80\x93gregorian uncertainty","articles lacking in-text citations from november 2009","all articles lacking in-text citations","articles needing additional references from november 2010","all articles needing additional references","articles to be expanded from november 2010","all articles to be expanded","articles using small message boxes","articles with multiple maintenance issues","interlanguage link template link number","all articles with unsourced statements","articles with unsourced statements from october 2009","wikipedia articles with gnd identifiers","wikipedia articles with nara identifiers","wikipedia articles with ndl identifiers","evidence law"],"wgbreakframes":false,"wgpagecontentlanguage":"en","wgpagecontentmodel":"wikitext","wgseparatortransformtable":["",""],"wgdigittransformtable":["",""],"wgdefaultdateformat":"dmy","wgmonthnames":["","january","february","march","april","may","june","july","august","september","october","november","december"],"wgmonthnamesshort":["","jan","feb","mar","apr","may","jun","jul","aug","sep","oct","nov","dec"],"wgrelevantpagename":"witness","wgrelevantarticleid":50956,"wgrequestid":"w3hacqpamfoaaijnh2iaaaab","wgisprobablyeditable":true,"wgrelevantpageisprobablyeditable":true,"wgrestrictionedit":[],"wgrestrictionmove":[],"wgflaggedrevsparams":{"tags":{}},"wgstablerevisionid":null,"wgcategorytreepagecategoryoptions":"{\\"mode\\":0,\\"hideprefix\\":20,\\"showcount\\":true,\\"namespaces\\":false}","wgwikieditorenabledmodules":[],"wgbetafeaturesfeatures":[],"wgmediavieweronclick":true,"wgmediaviewerenabledbydefault":true,"wgpopupsshouldsendmoduletouser":true,"wgpopupsconflictswithnavpopupgadget":false,"wgvisualeditor":{"pagelanguagecode":"en","pagelanguagedir":"ltr","pagevariantfallbacks":"en","usepageimages":true,"usepagedescriptions":true},"wgmfexpandallsectionsuseroption":true,"wgmfenablefontchanger":true,"wgmfdisplaywikibasedescriptions":{"search":true,"nearby":true,"watchlist":true,"tagline":false},"wgrelatedarticles":null,"wgrelatedarticlesusecirrussearch":true,"wgrelatedarticlesonlyusecirrussearch":false,"wgulscurrentautonym":"english","wgnoticeproject":"wikipedia","wgcentralnoticecookiestodelete":[],"wgcentralnoticecategoriesusinglegacy":["fundraising","fundraising"],"wgwikibaseitemid":"q196939","wgscorenotelanguages":{"arabic":"\xd8\xa7\xd9\x84\xd8\xb9\xd8\xb1\xd8\xa8\xd9\x8a\xd8\xa9","catalan":"catal\xc3\xa0","deutsch":"deutsch","english":"english","espanol":"espa\xc3\xb1ol","italiano":"italiano","nederlands":"nederlands","norsk":"norsk","portugues":"portugu\xc3\xaas","suomi":"suomi","svenska":"svenska","vlaams":"west-vlams"},"wgscoredefaultnotelanguage":"nederlands","wgcentralauthmobiledomain":false,"wgcodemirrorenabled":true,"wgvisualeditortoolbarscrolloffset":0,"wgvisualeditorunsupportededitparams":["undo","undoafter","veswitched"],"wgeditsubmitbuttonlabelpublish":true});mw.loader.state({"ext.gadget.charinsert-styles":"ready","ext.globalcssjs.user.styles":"ready","ext.globalcssjs.site.styles":"ready","site.styles":"ready","noscript":"ready","user.styles":"ready","ext.globalcssjs.user":"ready","ext.globalcssjs.site":"ready","user":"ready","user.options":"ready","user.tokens":"loading","ext.cite.styles":"ready","mediawiki.legacy.shared":"ready","mediawiki.legacy.commonprint":"ready","jquery.makecollapsible.styles":"ready","mediawiki.toc.styles":"ready","wikibase.client.init":"ready","ext.visualeditor.desktoparticletarget.noscript":"ready","ext.uls.interlanguage":"ready","ext.wikimediabadges":"ready","mediawiki.skinning.interface":"ready","skins.vector.styles":"ready"});mw.loader.implement("user.tokens@1dqfd7l",function($,jquery,require,module){/\*@nomin\*/mw.user.tokens.set({"edittoken":"+\\\\","patroltoken":"+\\\\","watchtoken":"+\\\\","csrftoken":"+\\\\"});\n});mw.loader.load(["ext.cite.a11y","site","mediawiki.page.startup","mediawiki.user","mediawiki.page.ready","jquery.makecollapsible","mediawiki.toc","mediawiki.searchsuggest","ext.gadget.teahouse","ext.gadget.referencetooltips","ext.gadget.watchlist-notice","ext.gadget.drn-wizard","ext.gadget.charinsert","ext.gadget.reftoolbar","ext.gadget.extra-toolbar-buttons","ext.gadget.switcher","ext.centralauth.centralautologin","mmv.head","mmv.bootstrap.autostart","ext.popups","ext.visualeditor.desktoparticletarget.init","ext.visualeditor.targetloader","ext.eventlogging.subscriber","ext.wikimediaevents","ext.navigationtiming","ext.uls.eventlogger","ext.uls.init","ext.uls.compactlinks","ext.uls.interface","ext.3d","ext.centralnotice.geoip","ext.centralnotice.startup","skins.vector.js"]);});</script>\n<link rel="stylesheet" href="/w/load.php?debug=false&amp;lang=en&amp;modules=ext.cite.styles%7cext.uls.interlanguage%7cext.visualeditor.desktoparticletarget.noscript%7cext.wikimediabadges%7cjquery.makecollapsible.styles%7cmediawiki.legacy.commonprint%2cshared%7cmediawiki.skinning.interface%7cmediawiki.toc.styles%7cskins.vector.styles%7cwikibase.client.init&amp;only=styles&amp;skin=vector"/>\n<script async="" src="/w/load.php?debug=false&amp;lang=en&amp;modules=startup&amp;only=scripts&amp;skin=vector"></script>\n<meta name="resourceloaderdynamicstyles" content=""/>\n<link rel="stylesheet" href="/w/load.php?debug=false&amp;lang=en&amp;modules=ext.gadget.charinsert-styles&amp;only=styles&amp;skin=vector"/>\n<link rel="stylesheet" href="/w/load.php?debug=false&amp;lang=en&amp;modules=site.styles&amp;only=styles&amp;skin=vector"/>\n<meta name="generator" content="mediawiki 1.32.0-wmf.16"/>\n<meta name="referrer" content="origin"/>\n<meta name="referrer" content="origin-when-crossorigin"/>\n<meta name="referrer" content="origin-when-cross-origin"/>\n<link rel="alternate" href="android-app://org.wikipedia/http/en.m.wikipedia.org/wiki/witness"/>\n<link rel="alternate" type="application/x-wiki" title="edit this page" href="/w/index.php?title=witness&amp;action=edit"/>\n<link rel="edit" title="edit this page" href="/w/index.php?title=witness&amp;action=edit"/>\n<link rel="apple-touch-icon" href="/static/apple-touch/wikipedia.png"/>\n<link rel="shortcut icon" href="/static/favicon/wikipedia.ico"/>\n<link rel="search" type="application/opensearchdescription+xml" href="/w/opensearch\_desc.php" title="wikipedia (en)"/>\n<link rel="edituri" type="application/rsd+xml" href="//en.wikipedia.org/w/api.php?action=rsd"/>\n<link rel="license" href="//creativecommons.org/licenses/by-sa/3.0/"/>\n<link rel="canonical" href="https://en.wikipedia.org/wiki/witness"/>\n<link rel="dns-prefetch" href="//login.wikimedia.org"/>\n<link rel="dns-prefetch" href="//meta.wikimedia.org" />\n<!--[if lt ie 9]><script src="/w/load.php?debug=false&amp;lang=en&amp;modules=html5shiv&amp;only=scripts&amp;skin=vector&amp;sync=1"></script><![endif]-->\n</head>\n<body class="mediawiki ltr sitedir-ltr mw-hide-empty-elt ns-0 ns-subject page-witness rootpage-witness skin-vector action-view">\t\t<div id="mw-page-base" class="noprint"></div>\n\t\t<div id="mw-head-base" class="noprint"></div>\n\t\t<div id="content" class="mw-body" role="main">\n\t\t\t<a id="top"></a>\n\t\t\t<div id="sitenotice" class="mw-body-content"><!-- centralnotice --></div><div class="mw-indicators mw-body-content">\n</div>\n<h1 id="firstheading" class="firstheading" lang="en">witness</h1>\t\t\t<div id="bodycontent" class="mw-body-content">\n\t\t\t\t<div id="sitesub" class="noprint">from wikipedia, the free encyclopedia</div>\t\t\t\t<div id="contentsub"></div>\n\t\t\t\t<div id="jump-to-nav"></div>\t\t\t\t<a class="mw-jump-link" href="#mw-head">jump to navigation</a>\n\t\t\t\t<a class="mw-jump-link" href="#p-search">jump to search</a>\n\t\t\t\t<div id="mw-content-text" lang="en" dir="ltr" class="mw-content-ltr"><div class="mw-parser-output"><div role="note" class="hatnote navigation-not-searchable">for other uses, see <a href="/wiki/witness\_(disambiguation)" class="mw-disambig" title="witness (disambiguation)">witness (disambiguation)</a>.</div>\n<table class="plainlinks metadata ambox ambox-style ambox-more\_footnotes" role="presentation"><tbody><tr><td class="mbox-image"><div style="width:52px"><img alt="" src="//upload.wikimedia.org/wikipedia/commons/thumb/a/a4/text\_document\_with\_red\_question\_mark.svg/40px-text\_document\_with\_red\_question\_mark.svg.png" width="40" height="40" srcset="//upload.wikimedia.org/wikipedia/commons/thumb/a/a4/text\_document\_with\_red\_question\_mark.svg/60px-text\_document\_with\_red\_question\_mark.svg.png 1.5x, //upload.wikimedia.org/wikipedia/commons/thumb/a/a4/text\_document\_with\_red\_question\_mark.svg/80px-text\_document\_with\_red\_question\_mark.svg.png 2x" data-file-width="48" data-file-height="48" /></div></td><td class="mbox-text"><div class="mbox-text-span">this article includes a <a href="/wiki/wikipedia:citing\_sources" title="wikipedia:citing sources">list of references</a>, but <b>its sources remain unclear</b> because it has <b>insufficient <a href="/wiki/wikipedia:citing\_sources#inline\_citations" title="wikipedia:citing sources">inline citations</a></b>.<span class="hide-when-compact"> please help to <a href="/wiki/wikipedia:wikiproject\_fact\_and\_reference\_check" title="wikipedia:wikiproject fact and reference check">improve</a> this article by <a href="/wiki/wikipedia:when\_to\_cite" title="wikipedia:when to cite">introducing</a> more precise citations.

Related Links = Bear Witness\nCourthouse facility dog or courthouse dog\nEyewitness (disambiguation)\nEyewitness identification\nEyewitness memory\nInformant\nMartyr (the word originally meant witness)\nMaterial witness\nUnited States Marshals Service\nWitness protection\n

Introduction = A witness is someone who has, who claims to have, or is thought, by someone with authority to compel testimony, to have knowledge relevant to an event or other matter of interest. In law a witness is someone who, either voluntarily or under compulsion, provides testimonial evidence, either oral or written, of what he or she knows or claims to know about the matter before some official authorized to take such testimony.\n

3

2

45.64720129966736

**Gathering the meta data**

All the meta data with the tags are represented in the output above

Some few examples are:-

<meta name="resourceloaderdynamicstyles" content=""/>

<meta name="generator" content="mediawiki 1.32.0-wmf.16"/>

etc…

**Provide the steps to configure and compile the crawler**

The Steps:

* 1. Installation of Scrapy and Newspaper3K into Anaconda(Spider 3.6)
     + In command prompt
       - conda install -c conda-forge scrapy
       - conda install Newspaper3K
  2. Opening Anaconda and taking help of the github code from Hardik Vasa on Wikipedia Crawler v3 (inserting results into MySQL)

This is Wikipedia help I took because it was mentioned the data should be news based and there should not be any other intervention.

* 1. With the help of the Wikipedia algorithm and using scrappy for news data from Wikipedia, I did the crawling.
  2. Understanding the Code:
* First the libraries are imported(time to calculate the total time at the end, urllib.request for extracting the web pages, re for regular expressions and scrapy for web crawling.
* Then I declared a class for keeping the details of the starting page and this will help to make the crawler a focused one as I initiated it to refer to Wikipedia.org only for web crawling.
* Then I declared the starting page and the relative url(seed\_page) for referral.
* Then the downloading of the page through header is done.
* This is followed by extraction of Title and metadata from the downloaded pages.
* The smaller sections are also extracted like h2,li,span etc. the smaller tags of a page and simultaneously the link tags are searched for hyperlinks.
* Various extraction has been done like get\_next\_link(searches for the url links), extract\_introduction(gets the page intro), get\_all\_links(gets all the links and stores them into the array links =[])
* Crawling of images has been to done since it was text oriented.
* In the process the repeated or duplicated url are being not considered.
* For time Checking:
  + T0 is initialized at 0 and I also maintained a database to store the scrapped pages into the file.
  + The timer starts as soon as we start the scraping.
  + After every scaning of the page and before going to next link the timer is stopped and recorded and reinitializied for the next page.
* In this way the total time is calculated and represented at the end.
  1. The numbers that we see represent the iterations for a particular page for calculations of the page scanning and storing in database.

Implementation Issues and their rectification of Scrapy:

1. Parse() was not being able to get understand what to parse, therefore I had to check separately how it works on charcters, integers, and floating points. Thus I also used ParseInt() and ParseFloat() some places in the code for better performance.
2. Removal of Duplicate elements was tough, therefore I used a code from Stack Overflow to get ride of it.
3. Since I took the main code from GitHub the indentation errors where obvious thus rectified them and since the code was based to Wikipedia, I tried to connect it to scrappy. It was a long time processing problem as the boundation of 1000 pages couldn’t be set at first and the loop was infinity. Thus an iteration check was kept to stop over search.