pusilkom ui
quality results

Crawling Data from Web

Bayu Distiawan

Natural Language Processing & Text Mining
Short Course

Pusat Ilmu Komputer UI
22 – 26 Agustus 2016

Preparing Tools

- Install Scrapy
  - pip install scrapy
- Create file: coba1.py, and run file:
  - `scrapy runspider coba1.py`

```
import scrapy

class DemoSpider(scrapy.Spider):
    name = "coba"
    allowed_domains = ["kompas.com"]
    start_urls = [
        "http://www.kompas.com",
        "http://bola.kompas.com/"
    ]

    def parse(self, response):
        filename = response.url.split("/")[-2] + '.html'
        with open(filename, 'wb') as f:
            f.write(response.body)
```

Expanding the links

- To show all links in a web page, add the following command
  - `link = sel.xpath('a/@href').extract()`

```
import scrapy

class DemoSpider(scrapy.Spider):
        name = "coba"
        allowed_domains = ["kompas.com"]
        start_urls = [
        "http://www.kompas.com",
        "http://bola.kompas.com/"
        ]

        def parse(self, response):
                for sel in response.xpath('//ul/li'):
                        link = sel.xpath('a/@href').extract()
                        print link
```

Expanding the links

- Using RSS Feed

```
from scrapy.spiders import XMLFeedSpider

class DemoSpider(XMLFeedSpider):
        name = "coba"
        allowed_domains = ["detik.com"]
        start_urls = [
        "http://rss.detik.com/index.php/sepakbola"
        ]

        def parse_node(self, response, node):
                links = node.xpath('link').extract()
                print links
```

Using the expanding links

```
from scrapy.spiders import XMLFeedSpider
from lxml import etree
import scrapy

class DemoSpider(XMLFeedSpider):
        name = "coba"
        allowed_domains = ["detik.com"]
        start_urls = [
        "http://rss.detik.com/index.php/sepakbola"
        ]

        def parse_node(self, response, node):
                result = etree.fromstring(node.xpath('link').extract()[0])
                link = response.urljoin(result.text)
                yield scrapy.Request(link, callback=self.parse_dir_contents)

        def parse_dir_contents(self, response):
                filename = response.url.split("/")[-2] + '.html'
                with open(filename, 'wb') as f:
                        f.write(response.body)
```

pusilkom ui
quality results

Getting the article

```
from bs4 import BeautifulSoup
from lxml.html.clean import clean_html

with open('3276323.html', 'r') as myfile:
      html=myfile.read()

soup = BeautifulSoup(html, 'html.parser')

#print soup.find_all('article')[0]
#print soup.article
print soup.find_all("div",
class_="detail_text").text
```

pusilkom ui
quality results

Getting the article

- Using **python-readability**
  - `pip install readability-lxml`

```
from readability import Document

with open('3276323.html', 'r') as
myfile:
       html=myfile.read()

doc = Document(html)
print doc.summary()
```

Task

- Clean the article from HTML Tag
  - Hints:
    - Using replace method (http://www.tutorialspoint.com/python/string_replace.htm)
    - https://tutorialedge.net/remove-html-tags-python
    - Google!
- Do The preprocessing task
  - Stopword remover
  - etc

Thank you