# Crawling Data

Crawling data merupakan proses pengumpulan data yang dilakukan pada suatu web atau sumber lain dari internet seperti media sosial, berita, artikel, dll.

Salah satu framework pada python yang digunakan untuk crawling data adalah scrapy. Scrapy dapat digunakan untuk mengekstrak, memproses dna menyimpan data dari sebuah website dengan skala besar ke format yang diinginkan.

* Instalasi Scrapy

Untuk menginstall scrapy, dapat dilakukan pada command prompt dengan menuliskan code

pip install scrapy

* Membuat file scrapy baru

Untuk memulai pembuatan file scrapy baru dapat dilakukan menggunakan kode di bawah dan kemudian diikuti dnegan namaproject yang diinginkan.

scrapy startproject namaproject

* Menjalankan Spider Baru

Sebelum membuat spider baru, maka terlebih dahulu masuk ke dalam projectscrapy yang telah dibuat terlebih dahulu

cd nama project

lalu, menuliskan command yang berisikan nama file yang akan dibuat dan juga alamat website yang akan diambil datanya sebagai berikut

scrapy genspider example example.com

* Menulis program scapper

Setelah membuat file spider, maka pada file tersebut telah tersedia file dengan nama file yang telah dibuat pada pembuatan spider.

Pada file tersebut telah tersedia code default untuk melakukan scraping yang nantinya akan diubah sesuai kebutuhan.

import scrapy

class scrapPTA(scrapy.Spider):

    name = 'PTA'

    allowed\_domains = ['pta.trunojoyo.ac.id']

    start\_urls = ['https://pta.trunojoyo.ac.id/c\_search/byprod/10/'+str(x)+" " for x in range(2,20)]

    def parse(self, response):

        for link in response.css('a.gray.button::attr(href)') :

            yield response.follow(link.get(),callback=self.parse\_categories)

    def parse\_categories(self, response):

        products = response.css('div#content\_journal ul li')

        for product in products:

            yield {

                'judul' : product.css('div a.title::text').get().strip(),

                'penulis' : product.css('div div:nth-child(2) span::text').get().strip(),

                'dosen 1' : product.css('div div:nth-child(3) span::text').get().strip(),

                'dosen 2' : product.css('div div:nth-child(4) span::text').get().strip(),

                'abstrak' : product.css('div div:nth-child(2) p::text').get().strip()

            }

Class scrapPTA() digunakan untuk melakukan perayapan (spidering) website sesuai dengan alamat yang telah ditentukan.

Class Parse digunakan untuk melakukan follow link pada button yang telah ditenukan

Calss parse\_categories merupakan callback function yang akan dipanggil pada saat setelah mengikuti link yang ditentukan pada parse().

Pada class ini, ditentukan elemen elemen apa yang akan diambil dengan melakukan copy select elemen atau class pada website yang akan diambil datanya.

* Menjalankan File Spider

Sebelum menjalankan file spider, terlebih dahulu masuk ke dalam direktori spider.

Kemudian dapat dilakukan running spider menggunakan command

# scrapy runspider namafile.py

* Menyimpan data ke file excel

Untuk menyimpan data yang telah diabil dari suatu web dapat dilakukan dengan menggunakan perintah

# scrapy crawl namafile -O namafileyangdiinginkan.xlsx

data yang telah di crawl dapat disimpan dalam format lain

# Preprocess Data

Data yang telah didapatkan merupakan data mentah yang butuh dilakukan preprosesing sebelum masuk ke dalam tahap pemrosesan data.

Text preprocessing digunakna untuk menyeleksi data text agar menjadi terstruktur melalui tahapan-tahapan preprosessing data.

Preprosesing meliputi case folding, tokenisasi, stopword removal, dan Stemming.

1. *Case Folding*

*Case folding* merupakan suatu tahap preprosessing data yang digunakan untuk menyam rataakan semua case atau ukuran huruf. Biasanya semua kata diubah menjadi lowercase (huruf kecil) pada *case folding*.

Contoh : kata “LaPtop” jika masuk pada tahap *case folding* maka akan berubah menjadi “laptop”. Semua huruf rata menggunakan *lowercase*.

1. *Tokenizing*

*Tokenizing* merupakan salah satu tahap preprocessing yang merupakan tahap untuk memotong atau membuat kalimat menjadi potongan kata.

Contoh : saya makan bakso

Maka ketika masuk pada tahap tokenizing maka akan menjadi saya, makan, bakso.

1. *Stopword Removal*

Pada tahap ini, dilakukan pembuangan terhadap kata atau karakter yang tidak dibutuhkan. Misalnya penggunaan kata hubung, kata preposisi, karakter, nomor, punctuation, selain itu juga dilakukan penghapusan terhadap kata yang tidak penting.

1. *Stemming*

*Stemming* merupakan tahap preprocessing yang digunakan untuk mengubah semua kata ke dalam bentuk baku atau dasar. Pada python *stemming* dilakukan menggunakan algoritma nadzief andriani pada library sastrawi.

Namun, pada topic modelling tidak diperlukan proses stem, dikarenakan setiap kata memiliki arti yang berbeda dan tujuan topic modelling adalah mengetahui proporsi term terhadap topik.

# Topic Modeling menggunakan LSA

* Apa itu Topic Modelling?

Topic modelling merupakan Teknik unsupervised learning atau pembelajaran tidak terawasi untuk menemukan suatu tema pada dokumen. Topic modeling akan melakukan extract data berdasarkan keyword yang menunjukkan topik.

Algoritma Topic Modelling :

* Setiap dokumen berhubungan dengan topik topik
* Setiap topik berhubungan dengan kata-kata.

Terdapat beberapa macam topic modelling, yang sering digunakan adalah Latent Dirchlet Allocation (LDA) dan Latent Semantic Analysis (LSA)

* Mengapa Menggunakan LSA (Latent Semantic Analysis?)

Latent Semantic Analysis (LSA) adalah salah satu pemodelan topik yang mengekstrak topik dari dokumen teks yang diberikan. pembelajaran LSA meliputi dekomposisi matriks pada matriks term dokumen menggunakan dekomposisi nilai Singular. LSA mengambil matriks dokumen dan istilah dan mencoba menguraikannya menjadi dua matriks yang terpisah yaitu :

* Matriks topik dokumen
* Matriks istilah topik.
* Tahapan LSA
* Menghitung TF-IDF

Langkah pertama pada topik modelling menggunakan LSA adalah menghitung TF-IDF. TF-IDF dihitung menggunakan matrix mxn. Dimana m merupakan dokumen dan n merupakan fitur tiap kata pada dokumen. TF-IDF atau Term frequency-Inverse document frequency memberikan bobot untuk term j dalam dokumen i.

Perhitungan TF-IDF dapat dilakukan dengan :

W\_{i,j}=tfi,jX\log \dfrac{N}{d\_{fj}}

Keterangan :

Wij = score TF-IDF

Tfi,j = term dari dokumen

N= Total Dokumen

Df j = dokumen

* Menampilkan SVD

Setelah menghitung TF-IDF, maka dilakukan Tindakan terhadap latent atau topik yang tersembunyi yang merepresentasikan dokumen. Sehingga kita dapat mengetahui matriks yang jarang muncul, masih terdapat noisy dan redudndan. Sehingga dilakukan pengurangan dimensi pada matrix A menjadi k.

Untuk itu dilakukan Singular Value Decomposition (SVD).

SVD merupakan proses menemukan informasi yang paling penting dengan menggunakan dimensi t untuk merepresentasikan hal yang sama. Nilai singular memfaktorkan matriks dan menjadikannya 3 matrix yang berbeda meliputi :

* Matrix kolom ortogonal (V)
* Matrix baris ortogonal, (U)
* Singular matrix (S)
* **M=U\*S\*V**

Sementara itu pada SVD dilakukan perhitungan terhadap matrix U dan juga Matrix V. dengan rumus :

A=USV^{T}

Keterangan :

Matriks U = pada matriks ini, baris mewakili vector dokumen pada topik

Matriks V = baris pada matriks ini mewakili vector istilah yang dinyatakan dalam topik.

S= matriks diagonal yang memiliki elemen-elemen diagonal sebagai nilai singular dari A.

Sehingga SVD menghasilkan representasi vektor untuk setiap dokumen dan istilah dalam data.

Panjang setiap vektor adalah k.

Salah satu kegunaan penting dari vektor ini adalah kita dapat menemukan kata dan dokumen serupa dengan bantuan metrik kesamaan kosinus.

* Mengekstrak Topic dan Term

Setelah dilakukan perhitungan matriks terhadap dokumen, dapat dilakukan ekstrak topik dokumen. Pada percobaan kali ini, dilakukan extrak sebanyak 10 topik.

Azizul, Hakim. 2020. Web Scraping menggunakan Scrapy. Diakses pada tanggal 15 Juni 2022. <https://hkalabs.com/blog/web-scraping-4-web-scraping-menggunakan-scrapy/>

Goyal, Chirag. 2021. Part 16 : Step by Step Guide to Master NLP – Topic Modelling Using LSA. Diakses pada tanggal 16 Juni 2022. <https://www.analyticsvidhya.com/blog/2021/06/part-16-step-by-step-guide-to-master-nlp-topic-modelling-using-lsa/>.

Mehrota, RAJ. 2021. Topic Modelling using LDA nad LSA in sckickit Learn. Diakses pada tanggal 15 Juni 2022. <https://www.kaggle.com/code/rajmehra03/topic-modelling-using-lda-and-lsa-in-sklearn/notebook>

Navlani, Avinash. 2021. Latent Semantic Indexing Using Scickit-Learn. Diakses pada tangga 15 Juni 2022. <https://machinelearninggeek.com/latent-semantic-indexing-using-scikit-learn/>.

Tineges, Rian. Tahapan Text Preprocessing dalam Teknik Pengolahan Data. Diakses pada tanggal 15 Juni 2022. <https://www.dqlab.id/tahapan-text-preprocessing-dalam-teknik-pengolahan-data>