**BAB 8 – TEXT CLASSIFICATION**

|  |
| --- |
| **TUJUAN BELAJAR** |

1. Memahami dan menerapkan *Text Classification* menggunakan Python

|  |
| --- |
| **DASAR TEORI** |

* Text classification merupakan istilah untuk pengkategorian data teks berlabel berdasarkan ketentuan yang telah diberikan
* Data teks dapat memiliki struktur semi maupun teks bebas
* Contoh data yang memiliki struktur semi yaitu html, json, iklan koran, daftar Pustaka, dsb
* contoh data teks bebas yaitu esai, berita, buku, dsb.
* Dalam python, toolkit yang dapat digunakan untuk melakukan text mining salah satunya adalah NLTK
* Untuk menginstall NLTK, cukup lakukan perintah pip install nltk dan lakukan import
* Tokenisasi merupakan langkah awal dalam pemrosesan teks, langkah ini mengubah sebuah teks menjadi list yang berisi elemen berupa setiap kata dari teks tersebut.
* Untuk melakukan tokenisasi, perintah string.split() dapat dilakukan, atau menggunakan NLTK
* Text

  Description automatically generated
* Text

  Description automatically generated
* Setelah tokenisasi, kita dapat mengetahui berapa kali sebuah kata muncul dalam teks tersebut dengan mencari frekuensinya
* Frekuensi kata yang muncul dalam sebuah teks dapat menjadi tanda awal tingkat kepentingan kata tersebut dalam korpus
* Untuk mencari frekuensi kata, dapat menggunakan fungsi FreqDist dari NLTK
* Text

  Description automatically generated
* Atau, dengan menggunakan method most\_common untuk mencari beberapa kata dengan frekuensi tertinggi dalam korpus
* Text

  Description automatically generated
* Untuk Bahasa Indonesia, terdapat library stemming dan lemma dengan nama PySastrawi
* Graphical user interface, text, application, email

  Description automatically generated
* Stopwords merupakan kata yang paling banyak muncul dalam sebuah teks. Biasanya tidak memiliki makna tertentu, contoh “the” “an” “a” “on” “is”.
* Text

  Description automatically generated
* Text

  Description automatically generated
* Setelah proses lemmatisasi dan penghilangan stopwords dilakukan, hal yang dapat dilakukan selanjutnya yaitu proses vektorisasi
* Vektorisasi kata ada berbagai metode, yakni TF-IDF untuk pembobotan kata berdasarkan relevansi kata pada kumpulan dokumen, dan Word Embedding untuk vektorisasi kata berdasarkan makna relative sebuah kata dimana kata tersebut digunakan dalam sebuah kalimat.
* Untuk mengetahui performa klasifikasi, jangan lupa lakukan train test split agar kita mempunyai data untuk di test
* Text

  Description automatically generated
* Menggunakan TF IDF, jalankan kode berikut
* Text

  Description automatically generated
* Untuk menggunakan random forest, jalankan kode berikut
* Text

  Description automatically generated
* Untuk MultinomialNB, jalankan kode berikut

Text

Description automatically generated

* Untuk menggunakan neural network, gunakan tensorflow / pytorch untuk mendefinisikan konfigurasi tiap layernya. Seluruh metode diatas dengan asumsi bahwa data teks sudah direpresentasikan dengan vector
* Text

  Description automatically generated
* Text

  Description automatically generated

|  |
| --- |
| **LATIHAN DAN TUGAS PRAKTIKUM** |

1. Download data teks dari halaman berikut: <https://raw.githubusercontent.com/ruzcmc/ClickbaitIndo-textclassifier/master/all_agree.csv>
2. Lakukan classification menggunakan metode MultinomialNB, RandomForest, dan metode klasifikasi pilihan kalian (terserah) dengan menggunakan fitur TF-IDF
3. Buat perbandingan performa ketiga metode supervised learning tersebut berdasarkan performanya (poin plus: lakukan cross validation)
4. Dokumentasikan setiap langkah diatas dan tulis laporannya
5. Kumpulkan dalam format pdf