Pembobotan *Vector Space Model* Korpus *Twitter* Menggunakan *Cosine Smiliarity*

TUGAS KELOMPOK

Disusun Untuk Memenuhi Tugas Mata Kuliah Temu Kembali Informasi

Dosen Pengampu : Retnani Latifah, M.Kom

Disusun Oleh :

MUHAMMAD REZA	**2019470055**
SELAMET SAPUTRA	**2019470069**
SYECHAN AHMAD ZIDAN	**2019470110**

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH JAKARTA

2022

DAFTAR ISI

BAB 1

1.1. Latar belakang masalah

Information retrieval atau pengambilan informasi adalah tugas untuk mengambil informasi yang sesuai atau relevan dari kumpulan korpus yang mewakili permintaan (kueri) (Djenouri et al., 2021).

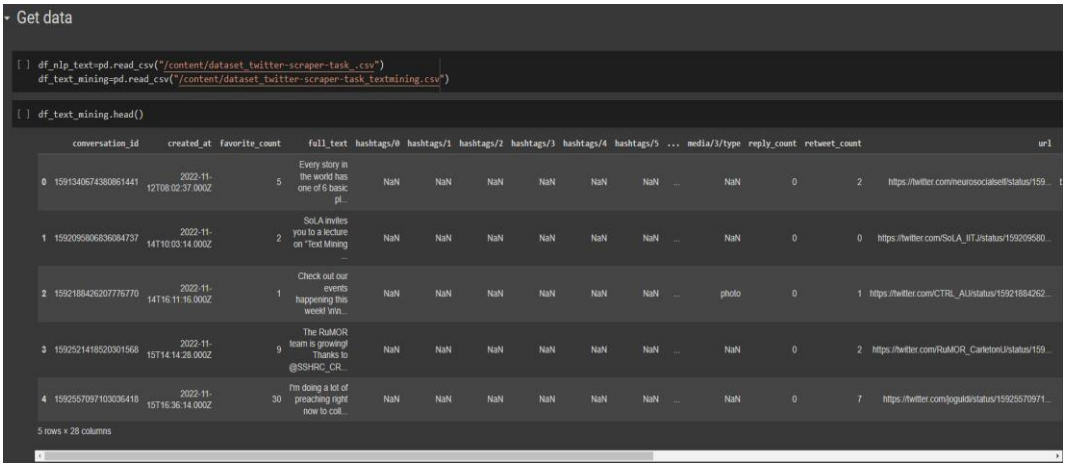
Ekstrasi fitur atau *Term Frequency Inverse Document Frequency* yaitu perkalian dari term *frequency row (tf raw)* yang dipakai untuk menghitung untuk menghitung jumlah kemunculan kata atau *term* untuk tiap kalimat pada teks. Sedangkan *inverse document frequency* adalah perhitungan untuk menentukan sebuah bobot pada suatu kata dalam suatu teks korpus. Jadi dari hasil nilai ekstrasi fitur tf-idf ini digunakan untunk perhitung similaritas, dan untuk beberapa metode dalam pendekatan statistika (Setyawan et al., 2021).

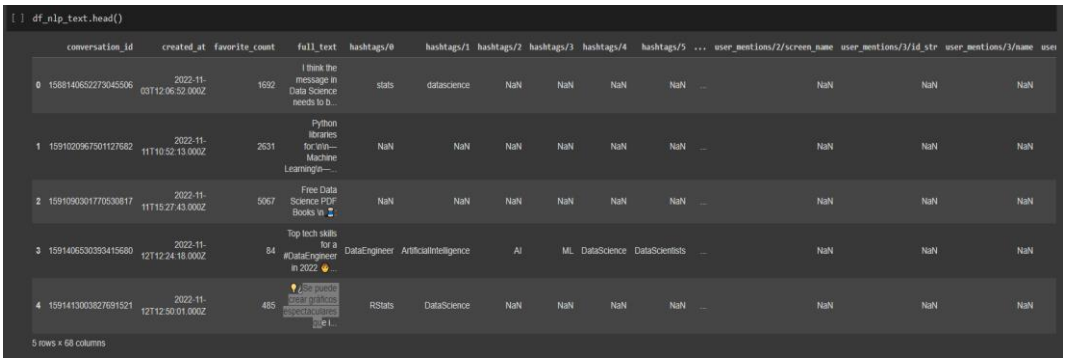
Vector Space Model (VSM). Sebuah model yang digunakan untuk mengukur sebuah kueri antara suatu dokumen dengan suatu kata kunci atau *keyword* (Susanti et al., 2020). *Vector space* adalah geometri berdimensi besar, ruang yang batas-batasnya ditentukan oleh vector. *Vector space model* yang menarik bagi penulis adalah model numerik yang menempatkan teks atau kata dalam sebuah ruang representasi dimensi tinggi. Secara lebih luas masuk akal, kita mungkin mempertimbangkan matriks jangka dokumen, yang pada dasarnya adalah tabel frekuensi kata yang disejajarkan oleh kosakata umum sehingga setiap vektor mewakili distribusi kosa kata ini dalam individu teks (Dobson, 2022). Konsep dasar *vector space model* adalah menghitung jarak vector antara dokumen dengan kata kunci yang dimasukkan kemudian mengurutkan berdasarkan tingkat kedekatannya (Susanti et al., 2020).

BAB II

2. Data Acquisition

Data yang digunakan dalam laporan kali ini adalah data atau korpus yang diambil berasal dari twitter dengan cara *scrapping*, korpus yang diambil adalah tentang text mining dan *information retrieval*.





2.2 Cleaning Data

Concat dua data atau menggabungkan kedua data yang telah didapat *df_nlp_text* atau data tentang nlp campuran dari text mining, text retrieval, bahkan data science didalamnya pada saat mengambil text. gabungkan dengan *df_text_mining* yang didalam *text* hanya tentang *text mining* pada saat pengambilan korpus.

```
df_nlp_text["full_text"]
```

```
0     I think the message in Data Science needs to b...
1     Python libraries for:\n\n— Machine Learning\n—...
2                     Free Data Science PDF Books  \n🧵:
3     Top tech skills for a #DataEngineer in 2022 👩...
4     💡¿Se puede crear gráficos espectaculares que i...
                            ...
59    Excellent retrieval skills in #BusheyHeathRead...
60    A novel adapter-based method for parameter-eff...
61    On @jhuclsp YouTube: Changes in Tweet Geolocat...
62    Yes, I am looking for a summer 2023 research i...
63    trec: TREC collection (2010). A bipartite netw...
Name: full_text, Length: 64, dtype: object
```

```
[ ] df_text_mining["full_text"]
```

```
0     Every story in the world has one of 6 basic pl...
1     SoLA invites you to a lecture on "Text Mining ...
2     Check out our events happening this week! \n\n...
3     The RuMOR team is growing! Thanks to @SSHRC_CR...
4     I'm doing a lot of preaching right now to coll...
5     Why my #Geosis package is simply the most robu...
6     Text Mining and Analytics #TextMining https://...
7     Let's speed up my booming Twitter career! Here...
8     Are you after a course that will teach you the...
9     Fundamentals of Predictive Text Mining (Texts ...
10    Awesome strategies for our humanities courses ...
11                        meaning of life is number 42
12    Brisbane Data, Power BI and AI Bootcamp speake...
13                              this is a possible tweet
14                                        It's a Tweet!
15                              this is an example tweet
16                               this is your next tweet
17                          or, maybe, a possible badger
18    Python Text Mining: Perform Text Processing, W...
19            and now for something completely different
Name: full_text, dtype: object
```

Melihat teks atau korpus dari kedua data

```
Gabungkan data terlebih dahulu
```

```
[ ] df_1=df_nlp_text["full_text"]
    df_2=df_text_mining["full_text"]
```

```
[ ] df_=pd.concat([df_1, df_2], ignore_index=True)
```

```
[ ] df_=pd.DataFrame({
        "full_text":df_
    }) # Simpan data yang sudah yang dsimpan kedalam dataframe
```

Setelah menggabungkan kedua data, data yang sudah digabung disimpan kedalam *data frame*.

2.3 Exploration Text Data

Untuk menelusuri dan mengetahui kata *stopwords* dalam Bahasa inggris yang sering muncul

Melihat berapa stopwords di data text

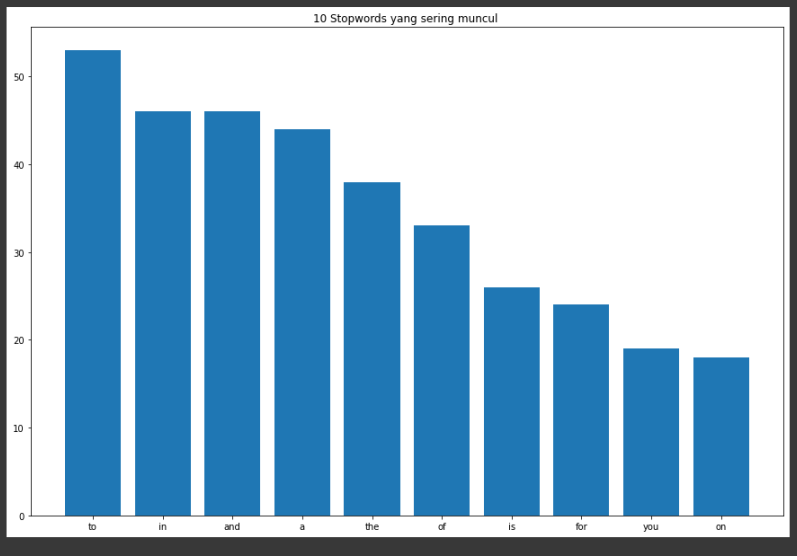
```
[ ] import nltk
    from nltk.corpus import stopwords
    nltk.download('stopwords')

    [nltk_data] Downloading package stopwords to /root/nltk_data...
    [nltk_data]   Package stopwords is already up-to-date!
    True

[ ] stop_en=stopwords.words('english')

[ ] import matplotlib.pyplot as plt
    #code from https://app.neptune.ai/neptune-ai/eda-nlp-tools/n/1-3-top-stopwords-barchart-b953763c-3fea-4331-bff0-429411793e5f/5c0fca05-ba07-4564-a02e-c44b08bfb8cb
    def plot_top_stopwords_barchart(text):
        stop=stop_en
        new= text.str.split()
        new=new.values.tolist()
        corpus=[word for i in new for word in i]
        from collections import defaultdict
        dic=defaultdict(int)
        for word in corpus:
            if word in stop:
                dic[word]+=1
        top=sorted(dic.items(), key=lambda x:x[1],reverse=True)[:10]
        x,y=zip(*top)
        plt.figure(figsize=(15,10))
        plt.bar(x,y)
        plt.title("10 Stopwords yang sering muncul ")
```

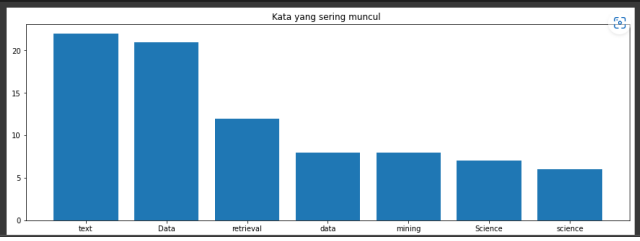
```
[ ] plot_top_stopwords_barchart(df_["full_text"])
```



Lalu didapatkan grafik berbentuk bar untuk kata-kata stopword apa saja yang kemunculannya paling sering muncul.

```
korpus_kata=["Data","text","data","mining","retrieval","Science","science"]
import matplotlib.pyplot as plt
#code from https://app.neptune.ai/neptune-ai/eda-nlp-tools/n/1-3-top-stopwords-barchart-b953763c-3fea-4331-bff0-429411793e5f/5c0fca05-ba07-4564-a02e-c44b08bfb8cb
def plot_words_barchart(text):
    stop=korpus_kata
    new= text.str.split()
    new=new.values.tolist()
    corpus=[word for i in new for word in i]
    from collections import defaultdict
    dic=defaultdict(int)
    for word in corpus:
        if word in stop:
            dic[word]+=1
    top=sorted(dic.items(), key=lambda x:x[1],reverse=True)[:10]
    x,y=zip(*top)
    plt.figure(figsize=(15,5))
    plt.bar(x,y)
    plt.title("Kata yang sering muncul ")
```

```
[ ] plot_words_barchart(df_["full_text"])
```



BAB 2

- Data ACC
- EDA
- PREPROP

BAB 3

- TFIDF
- VSM