NLP_Tugas_Vektorisasi

November 13, 2023

Muhammad Ihsan Prawira Hutomo 2211110022

1 Extract PDF

[4]:
```
pip install PyPDF2
```

```
Requirement already satisfied: PyPDF2 in /usr/local/lib/python3.10/dist-packages
(3.0.1)
```

[5]:
```
import PyPDF2
import pandas as pd

def extract_sentences(pdf_path):
    data = {'Sentence': []}

    with open(pdf_path, 'rb') as file:
        pdf_reader = PyPDF2.PdfReader(file)

        for page_num in range(len(pdf_reader.pages)):
            page = pdf_reader.pages[page_num]
            text = page.extract_text()

            # Split text into sentences
            sentences = text.split('.')

            # Add sentences to the data dictionary
            data['Sentence'].extend(sentences)

    return pd.DataFrame(data)

# Example usage
pdf_path = '/content/MALIN_KUNDANG.pdf'
df = extract_sentences(pdf_path)

# Display the DataFrame
print(df.head(10))
```

```
                                            Sentence
```

```
0  MALIN KUNDANG \nPada suatu waktu, hiduplah seb…
1   Keluarga tersebut terdiri \ndari ayah, ibu da…
2   Karena kondisi keuangan keluarga yang \nmempr…
3  \nMaka tinggallah si Malin dan ibunya di gubug…
4   Semingg u, dua minggu, sebulan, dua \nbulan b…
5   Sehingga ibunya harus menggantikan posisi aya…
6   \nMalin termasuk anak yang cerdas tetapi sedi…
7   Ia  sering mengejar ayam dan \nmemukulnya den…
8   Suatu hari ketika Malin sedang me ngejar ayam…
9   Luka terse but menjadi berbekas dilengannya \…
```

Pembersihan dengan lowercase, digit, tanda baca dan stopwords

[9]:
```
df['Sentence'] = df['Sentence'].apply(lambda x: x.lower())
```

[10]:
```
import re
```

[11]:
```
# Membersihkan /n
def clean_text1(text):
    return re.sub(r'\n([a-z])', r' \1', text)

# Apply the function to the 'Text' column
df['Sentence'] = df['Sentence'].apply(clean_text1)
```

[12]:
```
import pandas as pd
import re
import string


# digit and punctuation removal function
def remove_digits_and_punctuation(text):
    cleaned_text = re.sub(r'[\d' + re.escape(string.punctuation) + ']', '',␣
 ↪text)
    return cleaned_text

# Apply the function to the 'Sentence' column
df['Sentence'] = df['Sentence'].apply(remove_digits_and_punctuation)

# Display the result
print(df['Sentence'])
```

```
0       malin kundang  pada suatu waktu hiduplah sebua…
1        keluarga tersebut terdiri  dari ayah ibu dan …
2        karena kondisi keuangan keluarga yang  mempri…
3        maka tinggallah si malin dan ibunya di gubug …
4        semingg u dua minggu sebulan dua  bulan bahka…
                              …
3734     tubuh prabu dewata cengkar dilempar aji saka …
```

```
3735      aji saka kemudian dinobatkan menjadi raja me…
3736                        i a memboyong ayahnya ke  istana
3737     berkat pemerintahan yang adil dan bijaksana a…
3738
Name: Sentence, Length: 3739, dtype: object
```

[13]:
```
# Stopword Removal
import nltk
from nltk.corpus import stopwords

nltk.download('stopwords')

stop_words = set(stopwords.words('indonesian'))

def remove_stopwords(text):
    words = text.split()
    filtered_words = [word for word in words if word.lower() not in stop_words]
    return ' '.join(filtered_words)

df['Sentence'] = df['Sentence'].apply(remove_stopwords)
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data…
[nltk_data]   Unzipping corpora/stopwords.zip.
```

Hasil Pembersihan

[14]:
```
df['Sentence'].head(10)
```

[14]:
```
0    malin kundang hiduplah keluarga nelayan pesisi…
1       keluarga ayah anak lakilaki nama malin kundang
2    kondisi keuangan keluarga memprihatinkan sang …
3                      tinggallah si malin ibunya gubug
4    semingg u minggu sebulan ayah malin kampung ha…
5    ibunya menggantikan posisi ayah malin mencari …
6                              malin anak cerdas nakal
7                         mengejar ayam memukulnya sapu
8    malin me ngejar ayam tersandung batu lengan ka…
9           luka terse but berbekas dilengannya hilang
Name: Sentence, dtype: object
```

2 One Hot Encoding

[15]:
```
OHE = pd.get_dummies(df['Sentence'].str.split(expand=True).stack(),␣
↪drop_first=True).groupby(level=0).max()
```

[16]:
```
OHE['Sentence']= df['Sentence']
```

[17]:
```
OHE.head(10)
```

[17]:

	aaa	aaaa	aan	aat	abad	abadi	abangnya	abdi	abdinya	abu	…	\
0	0	0	0	0	0	0	0	0	0	0	…	
1	0	0	0	0	0	0	0	0	0	0	…	
2	0	0	0	0	0	0	0	0	0	0	…	
3	0	0	0	0	0	0	0	0	0	0	…	
4	0	0	0	0	0	0	0	0	0	0	…	
5	0	0	0	0	0	0	0	0	0	0	…	
6	0	0	0	0	0	0	0	0	0	0	…	
7	0	0	0	0	0	0	0	0	0	0	…	
8	0	0	0	0	0	0	0	0	0	0	…	
9	0	0	0	0	0	0	0	0	0	0	…	

	yelamatkan	yet	yik	yikan	yir	yosaku	yuk	yut	zam	\
0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	

	Sentence
0	malin kundang hiduplah keluarga nelayan pesisi…
1	keluarga ayah anak lakilaki nama malin kundang
2	kondisi keuangan keluarga memprihatinkan sang …
3	tinggallah si malin ibunya gubug
4	semingg u minggu sebulan ayah malin kampung ha…
5	ibunya menggantikan posisi ayah malin mencari …
6	malin anak cerdas nakal
7	mengejar ayam memukulnya sapu
8	malin me ngejar ayam tersandung batu lengan ka…
9	luka terse but berbekas dilengannya hilang

[10 rows x 4884 columns]

Ada modifikasi pada pd.get_dummies yang biasanya bisa langsung digunakan untuk one hot encoding karena data kata-kata berada di dalam kalimat. Modifikasinya adalah fungsi split untuk memisahkan kalimat dalam kata dan expand untuk menambah kolom baru seiring bertambahnya kata baru yang ingin dilakukan OHE. Fungsi seterusnya yaitu mengelompokkan data berdasarkan indeks utama dari MultiIndex (indeks DataFrame asli) dan menerapkan fungsi lambda untuk membuat Seri berisi nilai 1 dengan indeks yang sama seperti kata-kata. Akhirnya, unstack(fill_value=0) mengubah bentuk data, mengisi nilai NaN dengan 0, menghasilkan DataFrame yang telah di-OHE

kan. Pencarian kalimat yang terdapat kata ‘malin’ atau kolom malin bernilai == 1.

[18]:
```
OHE[OHE['malin']==1]
```

[18]:

	aaa	aaaa	aan	aat	abad	abadi	abangnya	abdi	abdinya	abu	…	\
0	0	0	0	0	0	0	0	0	0	0	…	
1	0	0	0	0	0	0	0	0	0	0	…	
3	0	0	0	0	0	0	0	0	0	0	…	
4	0	0	0	0	0	0	0	0	0	0	…	
5	0	0	0	0	0	0	0	0	0	0	…	
6	0	0	0	0	0	0	0	0	0	0	…	
8	0	0	0	0	0	0	0	0	0	0	…	
10	0	0	0	0	0	0	0	0	0	0	…	
12	0	0	0	0	0	0	0	0	0	0	…	
13	0	0	0	0	0	0	0	0	0	0	…	
14	0	0	0	0	0	0	0	0	0	0	…	
15	0	0	0	0	0	0	0	0	0	0	…	
16	0	0	0	0	0	0	0	0	0	0	…	
17	0	0	0	0	0	0	0	0	0	0	…	
18	0	0	0	0	0	0	0	0	0	0	…	
19	0	0	0	0	0	0	0	0	0	0	…	
22	0	0	0	0	0	0	0	0	0	0	…	
23	0	0	0	0	0	0	0	0	0	0	…	
24	0	0	0	0	0	0	0	0	0	0	…	
25	0	0	0	0	0	0	0	0	0	0	…	
27	0	0	0	0	0	0	0	0	0	0	…	
28	0	0	0	0	0	0	0	0	0	0	…	
30	0	0	0	0	0	0	0	0	0	0	…	
31	0	0	0	0	0	0	0	0	0	0	…	
32	0	0	0	0	0	0	0	0	0	0	…	
33	0	0	0	0	0	0	0	0	0	0	…	
34	0	0	0	0	0	0	0	0	0	0	…	
35	0	0	0	0	0	0	0	0	0	0	…	
37	0	0	0	0	0	0	0	0	0	0	…	
38	0	0	0	0	0	0	0	0	0	0	…	
40	0	0	0	0	0	0	0	0	0	0	…	
41	0	0	0	0	0	0	0	0	0	0	…	
42	0	0	0	0	0	0	0	0	0	0	…	
43	0	0	0	0	0	0	0	0	0	0	…	
44	0	0	0	0	0	0	0	0	0	0	…	
45	0	0	0	0	0	0	0	0	0	0	…	
47	0	0	0	0	0	0	0	0	0	0	…	
49	0	0	0	0	0	0	0	0	0	0	…	
50	0	0	0	0	0	0	0	0	0	0	…	
51	0	0	0	0	0	0	0	0	0	0	…	

	yelamatkan	yet	yik	yikan	yir	yosaku	yuk	yut	zam	\

0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0

	Sentence
0	malin kundang hiduplah keluarga nelayan pesisi…
1	keluarga ayah anak lakilaki nama malin kundang
3	tinggallah si malin ibunya gubug
4	semingg u minggu sebulan ayah malin kampung ha…
5	ibunya menggantikan posisi ayah malin mencari …

```
6                              malin anak cerdas nakal
8   malin me ngejar ayam tersandung batu lengan ka…
10  beranjak dewasa malin kundang kasihan ibunya b…
12  malin tertarik ajakan nakhoda kapal dagang dul…
13        malin kundang mengutarakan maksudnya ibunya
14  ibunya s emula setuju maksud malin kundang mal…
15    bekal perlengkapan malin dermaga diantar ibunya
16  anakku engkau berhasil orang berkecukupan kau …
17  kapal dinaiki malin diiringi lambaian tangan m…
18  kapal malin kundang bany ak belajar ilmu pelay…
19  perjalanan kapal dinaiki malin kundang serang …
22  malin kundang beruntung dibunuh ba jak laut pe…
23  malin kundang terkatungkatung ditengah laut ak…
24  sisa tenaga ad a malin kundang berjalan desa t…
25  sesampainya desa te rsebut malin kundang ditolong
27              desa malin terdampar desa ya ng subur
28  keuletan kegigihannya malin kelamaa n berhasil…
30  kaya raya malin kundang m empersunting gadis i…
31  berita malin kundang kaya raya te menikah mali…
32   malin kundang bersyukur gembira anaknya berhasil
33  malin kundang perg i dermaga anaknya pulang ka…
34  menikah malin istrinya pelayaran kapal indah d…
35  malin kundang menunggui anaknya kapal indah ma…
37     berdiri anaknya malin kundang beserta istrinya
38                           malin kundang turun kapal
40  ibunya belas luka dilengan kanan orang sema ki…
41  malin kundang anakku meng kau pergi mengirimka…
42  malin kundang melepaskan pelukan ibunya mendor…
43  wanita sembarangan mengaku ibuku malin kundang…
44  malin kundang purapura mengenali ibunya malu i…
45                    wanita ibumu istri malin kundang
47  mendengar pernyataan diperlakukan semenamena o…
49  kemarahannya memuncak malin menengadahkan tang…
50  angin bergemuruh kencang badai dahsyat menghan…
51  tubuh malin kundang perlahan kaku la makelamaa…

[40 rows x 4884 columns]
```

3 Hash Vectorization

[19]:
```python
import pandas as pd
import hashlib


def hash_vectoring(text, vector_size):
    # Inisialisasi vektor dengan nilai 0
```

```
    vector = [0] * vector_size

    # Konversi teks menjadi hash
    hashed_text = hashlib.sha256(text.encode()).hexdigest()

    # Ambil sebagian dari hash (sesuai dengan panjang vektor)
    hash_subset = hashed_text[:vector_size]

    # Konversi hash menjadi bilangan bulat (integer)
    hash_integer = int(hash_subset, 16)

    # Modulus hash dengan ukuran vektor untuk mendapatkan indeks
    index = hash_integer % vector_size

    # Set nilai indeks vektor menjadi 1
    vector[index] = 1

    return vector

# Define the vector size (you can adjust this based on your needs)
vector_size = 10

# Apply hash_vectoring to the 'Sentence' column
HASH = df['Sentence'].apply(lambda x: hash_vectoring(x, vector_size))
```

Model Hash vectorization adalah mengubah data teks menjadi representasi numerik yang dalam hal ini adalah biner. Output dari Hash adalah sebuah row kalimat tertentu akan lebih dekat ke salah satu fitur tertentu. Pada vector size atau jumlah fitur yang digunakan adalah 10 buah sehingga didapat hasil sebagai berikut.

[20]:
```
HASH.head(10)
```

[20]:
```
0    [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
1    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
2    [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
3    [0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
4    [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
5    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
6    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
7    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
8    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
9    [0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
Name: Sentence, dtype: object
```

4 Co-occurance Matrix

[21]:
```python
import numpy as np
import nltk
from nltk import bigrams
import itertools
import pandas as pd


# Step 4-2 Create function for co-occurrence matrix
def co_occurrence_matrix(corpus):
    vocab = set(corpus)
    vocab = list(vocab)
    vocab_to_index = {word: i for i, word in enumerate(vocab)}

    # Create bigrams from all words in corpus
    bi_grams = list(bigrams(corpus))

    # Frequency distribution of bigrams ((word1, word2), num_occurrences)
    bigram_freq = nltk.FreqDist(bi_grams).most_common(len(bi_grams))

    # Initialise co-occurrence matrix
    co_occurrence_matrix = np.zeros((len(vocab), len(vocab)))

    # Loop through the bigrams taking the current and previous word,
    # and the number of occurrences of the bigram.
    for bigram in bigram_freq:
        current = bigram[0][1]
        previous = bigram[0][0]
        count = bigram[1]
        pos_current = vocab_to_index[current]
        pos_previous = vocab_to_index[previous]
        co_occurrence_matrix[pos_current][pos_previous] = count

    co_occurrence_matrix = np.matrix(co_occurrence_matrix)

    # Return the matrix and the index
    return co_occurrence_matrix, vocab_to_index

# Merge sentences from the 'Sentence' column
merged = list(itertools.chain.from_iterable(df['Sentence'].str.split()))

# Apply the co-occurrence matrix function to the merged sentences
matrix, vocab_to_index = co_occurrence_matrix(merged)

# Create a DataFrame from the matrix and display the result
```

```
CoMatrixFinal = pd.DataFrame(matrix, index=vocab_to_index,␣
 ↪columns=vocab_to_index)
```

Menggunakan fungsi yang ada di praktikum dengan modifikasi pada variable merged yang ditambahkan fungsi split untuk memecah kata di kalimat. Hasilnya seperti berikut:

[22]:
```
CoMatrixFinal
```

[22]:

	peluru	mayang	ula	pakanya	nyenyak	dirundung	terasi	menan	\
peluru	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
mayang	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
ula	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
pakanya	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
nyenyak	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
…	…	…	…	…	…	…	…	…	
bidadari	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
hatimu	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
purapura	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
penuhi	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
kelabu	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

	erbolehkan	panas	…	pencuriga	cerdas	dahandahan	kel	terjaga	\
peluru	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
mayang	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
ula	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
pakanya	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
nyenyak	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
…	…	…	…	…	…	…	…	…	
bidadari	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
hatimu	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
purapura	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
penuhi	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
kelabu	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	

	bidadari	hatimu	purapura	penuhi	kelabu
peluru	0.0	0.0	0.0	0.0	0.0
mayang	0.0	0.0	0.0	0.0	0.0
ula	0.0	0.0	0.0	0.0	0.0
pakanya	0.0	0.0	0.0	0.0	0.0
nyenyak	0.0	0.0	0.0	0.0	0.0
…	…	…	…	…	…
bidadari	0.0	0.0	0.0	0.0	0.0
hatimu	0.0	0.0	0.0	0.0	0.0
purapura	0.0	0.0	0.0	0.0	0.0
penuhi	0.0	0.0	0.0	0.0	0.0
kelabu	0.0	0.0	0.0	0.0	0.0

```
[4884 rows x 4884 columns]
```

Co-occurrence berguna untuk mencari kata yang sering muncul bersama pada suatu konteks. Sebagai contoh kata 'malin' sering muncul berdekatan (kata sebelum dan sesudah) dengan beberapa kata berikut:

[23]:
```
CoMatrixFinal['malin'][CoMatrixFinal['malin']==1]
```

[23]:
```
istrinya         1.0
ibunya           1.0
me               1.0
menengadahkan    1.0
kun              1.0
tertarik         1.0
mencari          1.0
dermaga          1.0
mendesak         1.0
bersembunyi      1.0
anak             1.0
kampung          1.0
kelamaa          1.0
terdampar        1.0
diiringi         1.0
Name: malin, dtype: float64
```

5 Word2Vec

[24]:
```
from  gensim.models import Word2Vec
```

[55]:
```
# Split sentences into words
df['Words'] = df['Sentence'].apply(lambda x: x.split())

# Display the result
print(df['Words'].tolist()[:10])
```

```
[['malin', 'kundang', 'hiduplah', 'keluarga', 'nelayan', 'pesisir', 'pantai',
'wilayah', 'sumatra'], ['keluarga', 'ayah', 'anak', 'lakilaki', 'nama', 'malin',
'kundang'], ['kondisi', 'keuangan', 'keluarga', 'memprihatinkan', 'sang',
'ayah', 'memutuskan', 'mencari', 'nafkah', 'negeri', 'seberang', 'mengarungi',
'lautan', 'luas'], ['tinggallah', 'si', 'malin', 'ibunya', 'gubug'], ['semingg',
'u', 'minggu', 'sebulan', 'ayah', 'malin', 'kampung', 'halamannya'], ['ibunya',
'menggantikan', 'posisi', 'ayah', 'malin', 'mencari', 'nafkah'], ['malin',
'anak', 'cerdas', 'nakal'], ['mengejar', 'ayam', 'memukulnya', 'sapu'],
['malin', 'me', 'ngejar', 'ayam', 'tersandung', 'batu', 'lengan', 'kanannya',
'luka', 'terkena', 'batu'], ['luka', 'terse', 'but', 'berbekas', 'dilengannya',
'hilang']]
```

[26]:
```
model_w2v = Word2Vec(sentences = df['Words'], vector_size = 10, window = 10,␣
↪min_count=1, workers = 4)
```

[27]:
```
import numpy as np
```

[28]:
```
words = list(model_w2v.wv.index_to_key)
vector_W2V = [model_w2v.wv[word] for word in words]
vector_W2V = np.array(vector_W2V)
```

Pembuatan model Word2Vec dimulai dengan pemecahan kata-kata dalam kalimat dan membuatnya dalam array, kemudian model Word2Vec dibuat dengan beberapa parameter seperti vector_size = 10 yang berarti hasil vector akan berdimensi 10, window = 10 yang berarti model akan mempertimbangkan 10 kata sebelum dan sesudah kata target, min_count = 1 berarti sebuah kata akan tercatat jika muncul setidaknya 1 kali dalam kalimat, dan worker =4 adalah penggunaan cpu dalam training.

Hasil Word2Vec

[30]:
```
vector_W2V[:10]
```

[30]:
```
array([[ 0.26969928, -0.13966419,  0.48749143,  0.23142429, -0.05790477,
        -0.05351485,  0.63502246,  0.29603654, -0.62223953, -0.46246305],
       [ 0.30493793, -0.1286322 ,  0.28257456,  0.16946514, -0.01851519,
        -0.00075311,  0.47536942,  0.14058149, -0.53258955, -0.42636564],
       [ 0.3498392 , -0.06419192,  0.44800666,  0.12807094,  0.09419007,
        -0.02486978,  0.5178573 ,  0.20715652, -0.5920032 , -0.40948254],
       [ 0.13604753, -0.12324403,  0.44337836,  0.03345708,  0.00850896,
        -0.01179736,  0.5243239 ,  0.06836894, -0.44803712, -0.37246042],
       [ 0.10473009, -0.03340441,  0.17559138,  0.03204722,  0.02337875,
         0.01392426,  0.30181316,  0.21647091, -0.3215594 , -0.17765824],
       [ 0.1300972 , -0.04999817,  0.27088892,  0.10042419,  0.11264028,
        -0.04583785,  0.47296816,  0.0515826 , -0.4406696 , -0.21238343],
       [ 0.19846481, -0.1092798 ,  0.2915981 ,  0.05107161,  0.03990495,
         0.05417286,  0.4600036 ,  0.2011103 , -0.505834  , -0.33813724],
       [ 0.2659416 , -0.01783676,  0.28392947, -0.00279472,  0.07243159,
         0.0209108 ,  0.48865777,  0.10915539, -0.4242774 , -0.39429224],
       [ 0.20246427, -0.07420042,  0.35881296,  0.16318254, -0.03643504,
         0.03147747,  0.4807675 ,  0.07111669, -0.42959324, -0.24290286],
       [ 0.23468195, -0.09979167,  0.35754672,  0.10119189,  0.11695318,
         0.05232153,  0.33877158,  0.05384833, -0.40963244, -0.25515115]],
      dtype=float32)
```

Pencarian kata yang mirip dengan menggunakan model yang dibuat. Kata yang dicari adalah 'harta'.

[31]:
```
similar_words_w2v = model_w2v.wv.most_similar('harta',
                                              topn = 4)
print(f"Word2Vec = Kata serupa dengan 'harta':{similar_words_w2v}")
```

```
Word2Vec = Kata serupa dengan 'harta':[('kaki', 0.9693437218666077), ('bahagia',
0.9542333483695984), ('emas', 0.9537953734397888), ('sepatu',
0.9502952098846436)]
```

Hasilnya kata 'harta' mirip atau dekat dengan kata 'kaki', 'bahagia, 'emas, 'sepatu. Beberapa hasil cukup memuaskan kecuali kata 'kaki' dan 'sepatu' yang kurang cocok dengan kata harta , mungkin terdapat kesalahan bias dalam model ataupun parameter kurang optimal.

6 Fasttext

[32]:
```
from gensim.models import FastText
model_fasttext = FastText(sentences = df['Words'], vector_size = 10, window =␣
↪10, min_count=1, workers = 4)
```

[33]:
```
#Gensim fasttext
words= list(model_fasttext.wv.index_to_key)
vector_fasttext = [model_fasttext.wv[word] for word in words]
vector_fasttext = np.array(vector_fasttext)
```

[35]:
```
vector_fasttext[:10]
```

[35]:
```
array([[ 8.5749018e-01, -2.0007198e-03, -4.6173459e-01,  3.3806958e+00,
        -2.8632253e-01,  2.8312602e+00,  2.1697831e+00,  9.5786357e-01,
         5.7918262e-02,  1.0466908e+00],
       [ 6.3919568e-01, -2.1495263e-03, -3.9166978e-01,  2.5385451e+00,
        -2.6609457e-01,  2.1361856e+00,  1.6089441e+00,  7.1169800e-01,
         3.5821192e-02,  7.4107927e-01],
       [ 7.3075318e-01,  5.2232355e-02, -3.7414193e-01,  2.8263197e+00,
        -2.2936565e-01,  2.3154624e+00,  1.7668213e+00,  7.6974463e-01,
         1.3487193e-02,  7.9221272e-01],
       [ 1.8896791e+00,  2.7637986e-02, -1.0732573e+00,  7.4920888e+00,
        -6.4801693e-01,  6.2770705e+00,  4.8583241e+00,  2.0974104e+00,
         2.4018277e-01,  2.2705224e+00],
       [ 7.2141886e-01,  1.3582539e-02, -4.0914986e-01,  2.8934481e+00,
        -2.5236142e-01,  2.3892727e+00,  1.8915915e+00,  7.8911990e-01,
         6.7063324e-02,  8.8583875e-01],
       [ 8.4099865e-01,  7.8304466e-03, -5.1846319e-01,  3.3624880e+00,
        -2.5025615e-01,  2.8041079e+00,  2.2030349e+00,  9.2610019e-01,
         7.9787642e-02,  1.0675399e+00],
       [ 8.9568210e-01,  1.8560356e-02, -5.1103270e-01,  3.5676029e+00,
        -3.2196513e-01,  3.0110991e+00,  2.3116779e+00,  1.0242321e+00,
         3.9835583e-02,  1.0972910e+00],
       [ 6.8275911e-01, -1.0773021e-02, -3.8576314e-01,  2.6119812e+00,
        -2.0878464e-01,  2.1981184e+00,  1.6932329e+00,  7.0562077e-01,
         6.0844842e-02,  7.8333622e-01],
       [ 3.2438385e-01, -3.3259936e-02, -2.0692363e-01,  1.4842749e+00,
        -1.5603119e-01,  1.1783903e+00,  9.0307182e-01,  3.7340999e-01,
```

```
 -3.2703560e-02,  4.9010146e-01],
[ 1.5678836e+00, -1.9260028e-03, -8.6663568e-01,  6.0284238e+00,
 -5.2215326e-01,  5.0494628e+00,  3.9090395e+00,  1.6904857e+00,
  1.8063596e-01,  1.8716053e+00]], dtype=float32)
```

Sebagian besar syntax mirip dengan Word2Vec hanya mengganti model yang digunakan yaitu Fasttext dan tetap menggunakan parameter yang sama. Lalu, melakukan pencarian kata yang mirip, kata yang dicari adalah 'harta' seperti sebelumnya.

[36]:
```
similar_words_fasttext = model_fasttext.wv.most_similar('harta', topn = 4)
print(f"Fasttext = Kata serupa dengan 'harta':{similar_words_fasttext}")
```

```
Fasttext = Kata serupa dengan 'harta':[('jepitan', 0.9999821782112122),
('merubah', 0.9999788999557495), ('jantan', 0.9999769330024719), ('angkasa',
0.9999743103981018)]
```

Hasil didapat sangat buruk 4 kata top yang didapat tidak cocok dengan kata harta, berbeda dengan model Word2Vec sebelumnya.

[ ]:
```

```