TF-IDF 직접구현

```
[ ]  1 docs = ['오늘 동물원에서 원숭이와 코끼리를 봤어',
     2         '동물원에서 원숭이에게 바나나를 줬어 바나나를']
```

1. 토큰화 : 공백으로 토큰화 진행

```
 1 doc_ls=[]
 2 for doc in docs:
 3     doc_ls.append(doc.split())
 4 doc_ls
```

```
[['오늘', '동물원에서', '원숭이와', '코끼리를', '봤어'],
 ['동물원에서', '원숭이에게', '바나나를', '줬어', '바나나를']]
```

2. 유니크한 토큰 사전 구하기

2-1. 빈 딕셔너리 생성

```
[ ]  1 from collections import defaultdict
     2
     3 word2id = defaultdict(lambda : len(word2id))
     4 word2id
```

```
defaultdict(<function __main__.<lambda>()>, {})
```

2-2. 위에서 생성한 빈 딕셔너리에 유니크한 토큰 넣기

```
[ ]  1 for doc in doc_ls:
     2   for token in doc:
     3     word2id[token]
     4     print(token)
     5     print('\t',word2id)
     6 word2id
```

[ ]
```
오늘
        defaultdict(<function <lambda> at 0x78c876928e50>, {'오늘': 0})
동물원에서
        defaultdict(<function <lambda> at 0x78c876928e50>, {'오늘': 0, '동물원에서': 1})
원숭이와
        defaultdict(<function <lambda> at 0x78c876928e50>, {'오늘': 0, '동물원에서': 1, '원숭이와': 2})
코끼리를
        defaultdict(<function <lambda> at 0x78c876928e50>, {'오늘': 0, '동물원에서': 1, '원숭이와': 2, '코끼리를': 3})
봤어
        defaultdict(<function <lambda> at 0x78c876928e50>, {'오늘': 0, '동물원에서': 1, '원숭이와': 2, '코끼리를': 3, '봤어': 4})
동물원에서
        defaultdict(<function <lambda> at 0x78c876928e50>, {'오늘': 0, '동물원에서': 1, '원숭이와': 2, '코끼리를': 3, '봤어': 4})
원숭이에게
        defaultdict(<function <lambda> at 0x78c876928e50>, {'오늘': 0, '동물원에서': 1, '원숭이와': 2, '코끼리를': 3, '봤어': 4, '원숭이에게': 5})
바나나를
        defaultdict(<function <lambda> at 0x78c876928e50>, {'오늘': 0, '동물원에서': 1, '원숭이와': 2, '코끼리를': 3, '봤어': 4, '원숭이에게': 5, '바나나를': 6
줬어
        defaultdict(<function <lambda> at 0x78c876928e50>, {'오늘': 0, '동물원에서': 1, '원숭이와': 2, '코끼리를': 3, '봤어': 4, '원숭이에게': 5, '바나나를': 6,
바나나를
        defaultdict(<function <lambda> at 0x78c876928e50>, {'오늘': 0, '동물원에서': 1, '원숭이와': 2, '코끼리를': 3, '봤어': 4, '원숭이에게': 5, '바나나를': 6,
defaultdict(<function __main__.<lambda>()>,
            {'오늘': 0,
             '동물원에서': 1,
             '원숭이와': 2,
             '코끼리를': 3,
             '봤어': 4,
             '원숭이에게': 5,
             '바나나를': 6,
             '줬어': 7})
```

3. TF 계산하기

TF = 계산 특정단어등장빈도/문서내 전체등장단어빈도

```python
import numpy as np

TDM = np.zeros((len(word2id), len(doc_ls)), dtype=int)

for i, doc in enumerate(doc_ls):
  print(doc)
  for token in doc:
    TDM[word2id[token], i] += 1 # 해당 토큰의 위치(column)
    print(token)
    print(TDM)
    print('\t')
```

```
['오늘', '동물원에서', '원숭이와', '코끼리를', '봤어']
오늘
[[1 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]]

동물원에서
[[1 0]
 [1 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]]

원숭이와
[[1 0]
 [1 0]
 [1 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]]
```

```python
# TF 계산 특정단어등장빈도/문서내 전체등장단어빈도
def computeTF(TDM):
  doc_len = len(docs) # 문서갯수 2개
  word_len = len(word2id)
  print('문서수 : ',doc_len)
  print('전체 단어수 : ',word_len)

  tf = np.zeros((word_len,doc_len))
  print(tf)

  for doc_i in range(len(doc_ls)):
    print(docs[doc_i])
    for word_i in range(len(word2id)) :
      tf[word_i,doc_i] = TDM[word_i,doc_i]/TDM[:,doc_i].sum()
    print(tf)
  return tf
```

```python
tf = computeTF(TDM)
```

```
문서수 :  2
전체 단어수 :  8
[[0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]]
오늘 동물원에서 원숭이와 코끼리를 봤어
[[0.2 0. ]
 [0.2 0. ]
 [0.2 0. ]
 [0.2 0. ]
 [0.2 0. ]
 [0.  0. ]
 [0.  0. ]
 [0.  0. ]]
동물원에서 원숭이에게 바나나를 줬어 바나나를
[[0.2 0. ]
 [0.2 0.2]
 [0.2 0. ]
 [0.2 0. ]
 [0.2 0. ]
```

```
[ ]  1 tf
```

```
array([[0.2, 0. ],
       [0.2, 0.2],
       [0.2, 0. ],
       [0.2, 0. ],
       [0.2, 0. ],
       [0. , 0.2],
       [0. , 0.4],
       [0. , 0.2]])
```

4. IDF 계산하기

   IDF = -log(단어가등장한문서수/총문서수)

```
 1 import math
 2 # IDF계산 : -log(단어가등장한문서수/총문서수)
 3 def computeIDF(TDM):
 4   doc_len = len(docs) # 문서갯수 2개
 5   word_len = len(word2id)
 6   print('문서수 : ',doc_len)
 7   print('전체 단어수 : ',word_len)
 8
 9   idf = np.zeros(word_len)
10   print(idf)
11
12   for doc_i in range(len(doc_ls)):
13     for word_i in range(len(word2id)) :
14       idf[word_i] =  -math.log10(np.count_nonzero(TDM[word_i,:]) / doc_len)
15   print(idf)
16   return idf
```

```
[ ]  1 idf = computeIDF(TDM)
     2 idf
```

```
문서수 :  2
전체 단어수 :  8
[0. 0. 0. 0. 0. 0. 0. 0.]
[ 0.30103 -0.       0.30103  0.30103  0.30103  0.30103  0.30103  0.30103]
array([ 0.30103, -0.     ,  0.30103,  0.30103,  0.30103,  0.30103,
        0.30103,  0.30103])
```

5. TF- IDF 계산하기

```
[ ]  1 # TF-IDF 곱
     2 def computeTFIDF(TDM):
     3   tf = computeTF(TDM)
     4   idf = computeIDF(TDM).T
     5   tfidf = np.zeros(tf.shape)
     6
     7   for doc_i in range(len(doc_ls)):
     8     print(docs[doc_i])
     9     for word_i in range(len(word2id)) :
    10       tfidf[word_i,doc_i] = tf[word_i,doc_i] * idf[word_i]
    11
    12   print('TF-IDF 최종')
    13   print(tfidf)
    14   return tfidf
```

```
 1 tfidf = computeTFIDF(TDM)
```

```
문서수 :  2
전체 단어수 :  8
[[0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]]
오늘 동물원에서 원숭이와 코끼리를 봤어
[[0.2 0. ]
```

```
 [0.2 0. ]
 [0.2 0. ]
 [0.2 0. ]
 [0.  0. ]
 [0.  0. ]
 [0.  0. ]]
동물원에서 원숭이에게 바나나를 줬어 바나나를
[[0.2 0. ]
 [0.2 0.2]
 [0.2 0. ]
 [0.2 0. ]
 [0.2 0. ]
 [0.  0.2]
 [0.  0.4]
 [0.  0.2]]
문서수 :  2
전체 단어수 :  8
[0. 0. 0. 0. 0. 0. 0. 0.]
[ 0.30103 -0.       0.30103  0.30103  0.30103  0.30103  0.30103  0.30103]
오늘 동물원에서 원숭이와 코끼리를 봤어
동물원에서 원숭이에게 바나나를 줬어 바나나를
TF-IDF 최종
[[ 0.060206  0.      ]
 [-0.       -0.      ]
 [ 0.060206  0.      ]
 [ 0.060206  0.      ]
 [ 0.060206  0.      ]
 [ 0.        0.060206]
 [ 0.        0.120412]
 [ 0.        0.060206]]
```

```
[ ]   1 tfidf
```

```
array([[ 0.060206,  0.      ],
       [-0.      , -0.      ],
       [ 0.060206,  0.      ],
       [ 0.060206,  0.      ],
       [ 0.060206,  0.      ],
       [ 0.      ,  0.060206],
       [ 0.      ,  0.120412],
       [ 0.      ,  0.060206]])
```

6. 시각화로 확인

6. 시각화로 확인

```
import pandas as pd

sorted_vocab = sorted((value, key) for key, value in word2id.items())
vocab = [v[1] for v in sorted_vocab]
print('vocab : ',vocab)
tfidf= computeTFIDF(TDM)
pd.DataFrame(tfidf.T, columns=vocab)
```

```
vocab :  ['오늘', '동물원에서', '원숭이와', '코끼리를', '봤어', '원숭이에게', '바나나를', '줬어']
문서수 :  2
전체 단어수 :  8
[[0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]]
오늘 동물원에서 원숭이와 코끼리를 봤어
[[0.2 0. ]
 [0.2 0. ]
 [0.2 0. ]
 [0.2 0. ]
 [0.2 0. ]
 [0.  0. ]
 [0.  0. ]
 [0.  0. ]]
동물원에서 원숭이에게 바나나를 줬어 바나나를
[[0.2 0. ]
 [0.2 0.2]
 [0.2 0. ]
 [0.2 0. ]
 [0.2 0. ]
 [0.  0.2]
 [0.  0.4]
 [0.  0.2]]
문서수 :  2
전체 단어수 :  8
```

```
[ 0.30103 -0.        0.30103  0.30103  0.30103  0.30103  0.30103  0.30103]
오늘 동물원에서 원숭이와 코끼리를 봤어
동물원에서 원숭이에게 바나나를 줬어 바나나를
TF-IDF 최종
[[ 0.060206  0.      ]
 [-0.       -0.      ]
 [ 0.060206  0.      ]
 [ 0.060206  0.      ]
 [ 0.060206  0.      ]
 [ 0.        0.060206]
 [ 0.        0.120412]
 [ 0.        0.060206]]
```

	오늘	동물원에서	원숭이와	코끼리를	봤어	원숭이에게	바나나를	줬어
0	0.060206	-0.0	0.060206	0.060206	0.060206	0.000000	0.000000	0.000000
1	0.000000	-0.0	0.000000	0.000000	0.000000	0.060206	0.120412	0.060206

sklearn

```
[ ]
docs = ['오늘 동물원에서 원숭이와 코끼리를 봤어',
        '동물원에서 원숭이에게 바나나를 줬어 바나나를']

from sklearn.feature_extraction.text import TfidfVectorizer
tfidv = TfidfVectorizer()
tfidv = tfidv.fit(docs)
tfidv.transform(docs).toarray()
vocab = tfidv.get_feature_names_out()
```

```
import pandas as pd
df = pd.DataFrame(tfidv.transform(docs).toarray(),columns = vocab)
df
```

	동물원에서	바나나를	봤어	오늘	원숭이에게	원숭이와	줬어	코끼리를
0	0.335176	0.000000	0.471078	0.471078	0.000000	0.471078	0.000000	0.471078
1	0.278943	0.784088	0.000000	0.000000	0.392044	0.000000	0.392044	0.000000

+ 코드 + 텍스트