Gensim - Creating TF-IDF Matrix

We will create Term Frequency-Inverse Document Frequency (TF-IDF) Matrix with the help of Gensim.

First, we need to import all the necessary packages as follows

In [1]:
```
import gensim
import pprint
from gensim import corpora
from gensim.utils import simple_preprocess
```

Now provide the list containing sentences. We have three sentences in our list

In [2]:
```
doc_list = [
   "Hello, how are you?", "How do you do?",
   "Hey what are you doing? yes you What are you doing?"
]
```

Next, do tokenisation of the sentences as follows

In [3]:
```
doc_tokenized = [simple_preprocess(doc) for doc in doc_list]
```

Create an object of corpora.Dictionary() as follows

In [4]:
```
dictionary = corpora.Dictionary()
```

Now pass these tokenised sentences to dictionary.doc2bow() object as follows

In [5]:
```
BoW_corpus = [dictionary.doc2bow(doc, allow_update=True) for doc in doc_tokenized]
```

Next, we will get the word ids and their frequencies in our documents.

In [6]:
```
for doc in BoW_corpus:
    print([[dictionary[id], freq] for id, freq in doc])
```

```
[['are', 1], ['hello', 1], ['how', 1], ['you', 1]]
[['how', 1], ['you', 1], ['do', 2]]
[['are', 2], ['you', 3], ['doing', 2], ['hey', 1], ['what', 2], ['yes', 1]]
```

In this way we have trained our corpus (Bag-of-Word corpus).

Next, we need to apply this trained corpus within the tfidf model models.TfidfModel().

First import the numpy package

In [7]:
```
import numpy as np
```

Now applying our trained corpus(BoW_corpus) within the square brackets of models.TfidfModel()

In [8]:
```
tfidf = gensim.models.TfidfModel(BoW_corpus, smartirs='ntc')
```

Next, we will get the word ids and their frequencies in our tfidf modeled corpus

In [9]:
```
for doc in tfidf[BoW_corpus]:
    print([[dictionary[id], np.around(freq,decimals=4)] for id, freq in doc])
```

```
[['are', 0.4025], ['hello', 0.805], ['how', 0.4025], ['you', 0.1671]]
[['how', 0.2413], ['you', 0.1002], ['do', 0.9653]]
[['are', 0.2963], ['you', 0.1845], ['doing', 0.5927], ['hey', 0.2963], ['what', 0.5927], ['yes', 0.2963]]
```

From the above outputs, we see the difference in the frequencies of the words in our documents.

In [ ]: