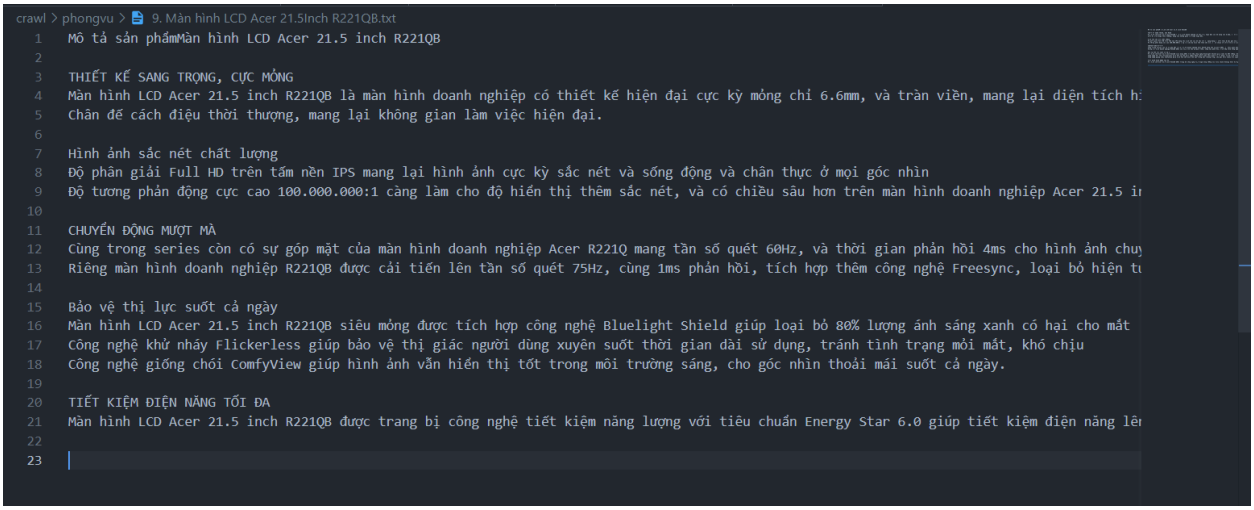
One-hot Vector (One-Hot Encoding)

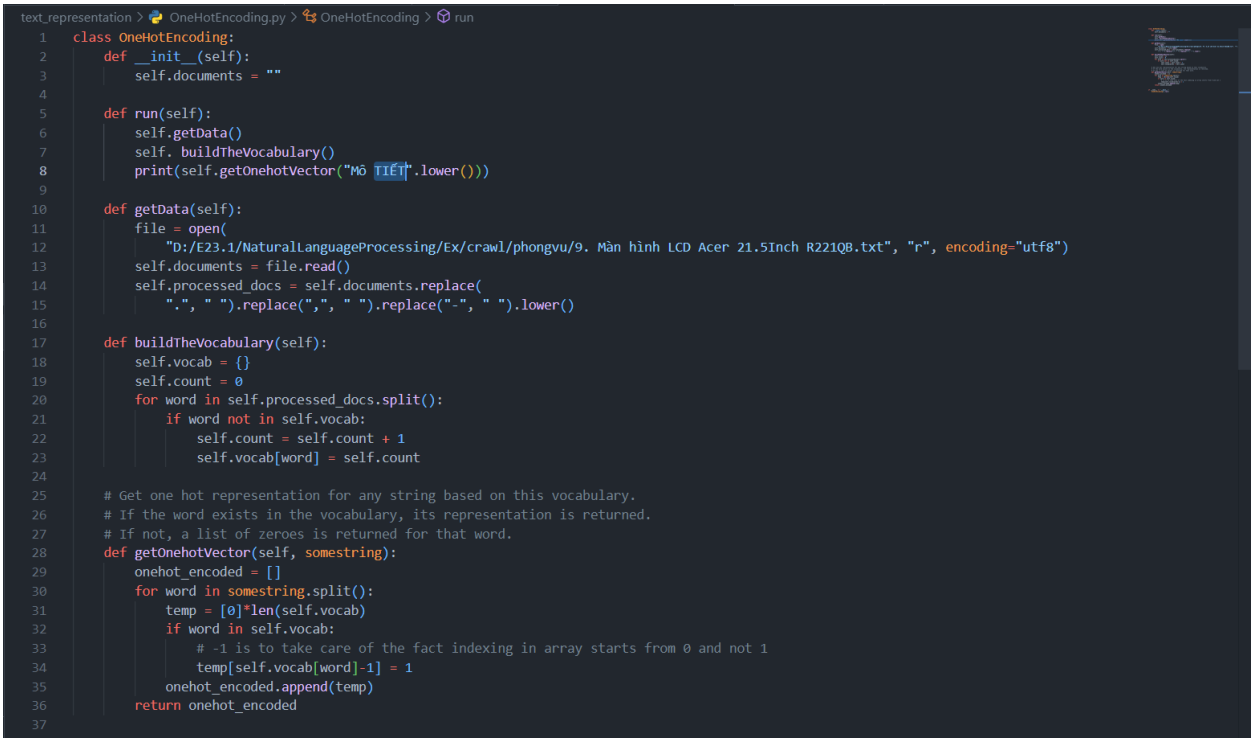
1. Dữ liệu mẫu



crawl > phongvu > 9. Màn hình LCD Acer 21.5Inch R221QB.txt

```
Mô tả sản phẩmMàn hình LCD Acer 21.5 inch R221QB

THIẾT KẾ SANG TRỌNG, CỰC MỎNG
Màn hình LCD Acer 21.5 inch R221QB là màn hình doanh nghiệp có thiết kế hiện đại cực kỳ mỏng chỉ 6.6mm, và tràn viền, mang lại diện tích h
Chân đế cách điệu thời thượng, mang lại không gian làm việc hiện đại.

Hình ảnh sắc nét chất lượng
Độ phân giải Full HD trên tấm nền IPS mang lại hình ảnh cực kỳ sắc nét và sống động và chân thực ở mọi góc nhìn
Độ tương phản động cực cao 100.000.000:1 càng làm cho độ hiển thị thêm sắc nét, và có chiều sâu hơn trên màn hình doanh nghiệp Acer 21.5 i

CHUYỂN ĐỘNG MƯỢT MÀ
Cùng trong series còn có sự góp mặt của màn hình doanh nghiệp Acer R221Q mang tần số quét 60Hz, và thời gian phản hồi 4ms cho hình ảnh chuy
Riêng màn hình doanh nghiệp R221QB được cải tiến lên tần số quét 75Hz, cùng 1ms phản hồi, tích hợp thêm công nghệ Freesync, loại bỏ hiện t

Bảo vệ thị lực suốt cả ngày
Màn hình LCD Acer 21.5 inch R221QB siêu mỏng được tích hợp công nghệ Bluelight Shield giúp loại bỏ 80% lượng ánh sáng xanh có hại cho mắt
Công nghệ khử nháy Flickerless giúp bảo vệ thị giác người dùng xuyên suốt thời gian dài sử dụng, tránh tình trạng mỏi mắt, khó chịu
Công nghệ giống chói ComfyView giúp hình ảnh vẫn hiển thị tốt trong môi trường sáng, cho góc nhìn thoải mái suốt cả ngày.

TIẾT KIỆM ĐIỆN NĂNG TỐI ĐA
Màn hình LCD Acer 21.5 inch R221QB được trang bị công nghệ tiết kiệm năng lượng với tiêu chuẩn Energy Star 6.0 giúp tiết kiệm điện năng lê
```

2. Code



text_representation > OneHotEncoding.py > OneHotEncoding > run

```python
class OneHotEncoding:
    def __init__(self):
        self.documents = ""

    def run(self):
        self.getData()
        self. buildTheVocabulary()
        print(self.getOnehotVector("Mô TIẾT".lower()))

    def getData(self):
        file = open(
            "D:/E23.1/NaturalLanguageProcessing/Ex/crawl/phongvu/9. Màn hình LCD Acer 21.5Inch R221QB.txt", "r", encoding="utf8")
        self.documents = file.read()
        self.processed_docs = self.documents.replace(
            ".", " ").replace(",", " ").replace("-", " ").lower()

    def buildTheVocabulary(self):
        self.vocab = {}
        self.count = 0
        for word in self.processed_docs.split():
            if word not in self.vocab:
                self.count = self.count + 1
                self.vocab[word] = self.count

    # Get one hot representation for any string based on this vocabulary.
    # If the word exists in the vocabulary, its representation is returned.
    # If not, a list of zeroes is returned for that word.
    def getOnehotVector(self, somestring):
        onehot_encoded = []
        for word in somestring.split():
            temp = [0]*len(self.vocab)
            if word in self.vocab:
                # -1 is to take care of the fact indexing in array starts from 0 and not 1
                temp[self.vocab[word]-1] = 1
            onehot_encoded.append(temp)
        return onehot_encoded

```

3. Kết quả

```
PS D:\E23.1\NaturalLanguageProcessing\Ex> & C:/Users/phsan/AppData/Local/Programs/Python/Python310/python.exe d:/E23.1/NaturalLanguageProcessing/Ex/te
xt_representation/OneHotEncoding.py
{'Mô': 1, 'tả': 2, 'sản': 3, 'phẩmMàn': 4, 'hình': 5, 'LCD': 6, 'Acer': 7, '21': 8, '5': 9, 'inch': 10, 'R221QB': 11, 'THIẾT': 12, 'KẾ': 13, 'SANG': 1
4, 'TRỌNG': 15, 'CỰC': 16, 'MỎNG': 17, 'Màn': 18, 'là': 19, 'màn': 20, 'doanh': 21, 'nghiệp': 22, 'có': 23, 'thiết': 24, 'kế': 25, 'hiện': 26, 'đại':
27, 'cực': 28, 'kỳ': 29, 'mỏng': 30, 'chỉ': 31, '6': 32, '6mm': 33, 'và': 34, 'tràn': 35, 'viền': 36, 'mang': 37, 'lại': 38, 'diện': 39, 'tích': 40, '
hiển': 41, 'thị': 42, 'lớn': 43, 'hơn': 44, 'Chân': 45, 'đế': 46, 'cách': 47, 'điệu': 48, 'thời': 49, 'thượng': 50, 'không': 51, 'gian': 52, 'làm': 53
, 'việc': 54, 'Hình': 55, 'ảnh': 56, 'sắc': 57, 'nét': 58, 'chất': 59, 'lượng': 60, 'Độ': 61, 'phân': 62, 'giải': 63, 'Full': 64, 'HD': 65, 'trên': 66
, 'tấm': 67, 'nền': 68, 'IPS': 69, 'sống': 70, 'động': 71, 'chân': 72, 'thực': 73, 'ở': 74, 'mọi': 75, 'góc': 76, 'nhìn': 77, 'tương': 78, 'phản': 79,
 'cao': 80, '100': 81, '000': 82, '000:1': 83, 'càng': 84, 'cho': 85, 'độ': 86, 'thêm': 87, 'chiều': 88, 'sâu': 89, 'CHUYỂN': 90, 'ĐỘNG': 91, 'MƯỢT':
92, 'MÀ': 93, 'Cùng': 94, 'trong': 95, 'series': 96, 'còn': 97, 'sự': 98, 'góp': 99, 'mặt': 100, 'của': 101, 'R221Q': 102, 'tần': 103, 'số': 104, 'qué
t': 105, '60Hz': 106, 'hồi': 107, '4ms': 108, 'chuyển': 109, 'mượt': 110, 'mà': 111, 'Riêng': 112, 'được': 113, 'cải': 114, 'tiến': 115, 'lên': 116, '
75Hz': 117, 'cùng': 118, '1ms': 119, 'hợp': 120, 'công': 121, 'nghệ': 122, 'Freesync': 123, 'loại': 124, 'bỏ': 125, 'tượng': 126, 'bóng': 127, 'ma': 1
28, 'trời': 129, 'sụt': 130, 'khung': 131, 'Bảo': 132, 'vệ': 133, 'lực': 134, 'suốt': 135, 'cả': 136, 'ngày': 137, 'siêu': 138, 'Bluelight': 139, 'Shi
eld': 140, 'giúp': 141, '80%': 142, 'ánh': 143, 'sáng': 144, 'xanh': 145, 'hại': 146, 'mắt': 147, 'Công': 148, 'khử': 149, 'nháy': 150, 'Flickerless':
 151, 'bảo': 152, 'giác': 153, 'người': 154, 'dùng': 155, 'xuyên': 156, 'dài': 157, 'sử': 158, 'dụng': 159, 'tránh': 160, 'tình': 161, 'trạng': 162, '
mỏi': 163, 'khó': 164, 'chịu': 165, 'giống': 166, 'chói': 167, 'ComfyView': 168, 'vẫn': 169, 'tốt': 170, 'môi': 171, 'trường': 172, 'thoải': 173, 'mái
': 174, 'TIẾT': 175, 'KIỆM': 176, 'ĐIỆN': 177, 'NĂNG': 178, 'TỐI': 179, 'ĐA': 180, 'trang': 181, 'bị': 182, 'tiết': 183, 'kiệm': 184, 'năng': 185, 'vớ
i': 186, 'tiêu': 187, 'chuẩn': 188, 'Energy': 189, 'Star': 190, '0': 191, 'điện': 192, 'đến': 193, '68%': 194, 'chi': 195, 'phí': 196}
```

```
PS D:\E23.1\NaturalLanguageProcessing\Ex> & C:/Users/phsan/AppData/Local/Programs/Python/Python310/python.exe d:/E23.1/NaturalLanguageProcessing/Ex/te
xt_representation/OneHotEncoding.py
[[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0]]
```

Bag of Words

1. Dữ liệu mẫu: Như bài 1
2. Code

```
text_representation > BagOfWords.py > BagOfWords > run
from sklearn.feature_extraction.text import CountVectorizer


class BagOfWords:
    def __init__(self):
        self.documents = ""

    def run(self):
        self.getData()
        self. buildTheVocabulary()
        print(self.getBagOfWords(20))

    def getData(self):
        file = open(
            "D:/E23.1/NaturalLanguageProcessing/Ex/crawl/phongvu/9. Màn hình LCD Acer 21.5Inch R221QB.txt", "r", encoding="utf8")
        self.documents = file.read()
        self.processed_docs = self.documents.replace(
            ".", " ").replace(",", " ").replace("-", " ").lower().split()

    def buildTheVocabulary(self):
        self.count_vect = CountVectorizer()
        # Build a BOW representation for the corpus
        self.bow_rep = self.count_vect.fit_transform(self.processed_docs)

        # Look at the vocabulary mapping
        print("Our vocabulary: ", self.count_vect.vocabulary_)

    def getBagOfWords(self, position):
        return self.bow_rep[position].toarray()


if __name__ == '__main__':
    BagOfWords().run()
```

3. Kết quả

```
  File "d:\E23.1\NaturalLanguageProcessing\Ex\text_representation\BagOfWords.py", line 11, in run
phản': 104, 'cao': 16, '100': 1, '000': 0, 'càng': 28, 'cho': 19, 'thêm': 127, 'chiều': 18, 'sâu': 117, 'chuyển': 20, 'mượt': 87, 'mà': 82, 'cùng': 33, 'trong': 136, '
series': 111, 'còn': 30, 'sự': 124, 'gộp': 53, 'mặt': 89, 'của': 36, 'r221q': 107, 'tần': 150, 'số': 120, 'quét': 106, '60hz': 5, 'hỗi': 60, '4ms': 4, 'riêng': 109, 'đ
ược': 165, 'cải': 35, 'tiến': 133, 'lên': 75, '75hz': 8, '1ms': 2, 'hợp': 61, 'công': 32, 'nghệ': 94, 'freesync': 45, 'loại': 72, 'bỏ': 15, 'tượng': 147, 'bóng': 12, '
ma': 80, 'trỗi': 143, 'sụt': 122, 'khung': 64, 'bảo': 13, 'vệ': 157, 'lực': 79, 'suốt': 115, 'cả': 34, 'ngày': 95, 'siêu': 113, 'bluelight': 11, 'shield': 112, 'giúp':
 49, '80': 9, 'ánh': 161, 'sáng': 116, 'xanh': 159, 'hại': 59, 'mắt': 88, 'khử': 67, 'nháy': 97, 'flickerless': 44, 'giác': 48, 'người': 96, 'dùng': 41, 'xuyên': 160,
'dài': 40, 'sử': 123, 'dụng': 42, 'tránh': 138, 'tình': 144, 'trạng': 141, 'mỏi': 91, 'khó': 65, 'chịu': 26, 'giống': 51, 'chói': 23, 'comfyview': 27, 'vẫn': 156, 'tốt
': 152, 'môi': 86, 'trường': 140, 'thoải': 126, 'mái': 84, 'tiết': 134, 'kiệm': 68, 'điện': 163, 'năng': 100, 'tối': 151, 'đa': 162, 'trang': 135, 'bị': 14, 'với': 158
, 'tiêu': 132, 'chuẩn': 21, 'energy': 43, 'star': 114, 'đến': 168, '68': 6, 'chi': 17, 'phí': 103}
[[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]]
PS D:\E23.1\NaturalLanguageProcessing\Ex>
```

Bag of n-grams

1. Dữ liệu mẫu: Như bài 1
2. Code

```
text_representation > BagOfN-Grams.py > BagOfNGrams > buildTheVocabulary
```

```python
from sklearn.feature_extraction.text import CountVectorizer


class BagOfNGrams:
    def __init__(self):
        self.documents = ""

    def run(self):
        self.getData()
        self. buildTheVocabulary()
        print(self.getBagOfWords(20))

    def getData(self):
        file = open(
            "D:/E23.1/NaturalLanguageProcessing/Ex/crawl/phongvu/9. Màn hình LCD Acer 21.5Inch R221QB.txt", "r", encoding="utf8")
        self.documents = file.read()
        self.processed_docs = self.documents.replace(
            ".", " ").replace(",", " ").replace("-", " ").lower().split()

    def buildTheVocabulary(self):
        self.count_vect = CountVectorizer(ngram_range=(1, 3))
        # Build a BOW representation for the corpus
        self.bow_rep = self.count_vect.fit_transform(self.processed_docs)

        # Look at the vocabulary mapping
        print("Our vocabulary: ", self.count_vect.vocabulary_)

    def getBagOfWords(self, position):
        return self.bow_rep[position].toarray()


if __name__ == '__main__':
    BagOfNGrams().run()
```

3. Kết quả

```
PS D:\E23.1\NaturalLanguageProcessing\Ex> & C:/Users/phsan/AppData/Local/Programs/Python/Python310/python.exe d:/E23.1/NaturalLanguageProcessing/Ex/text_representation
/BagOfN-Grams.py
Our vocabulary:  {'mô': 85, 'tả': 148, 'sản': 118, 'phẩmmàn': 105, 'hình': 57, 'lcd': 71, 'acer': 10, '21': 3, 'inch': 62, 'r221qb': 108, 'thiết': 125, 'kế': 69, 'sang
': 110, 'trọng': 142, 'cực': 37, 'mỏng': 92, 'màn': 83, 'là': 73, 'doanh': 39, 'nghiệp': 93, 'có': 31, 'hiện': 56, 'đại': 166, 'kỳ': 70, 'chỉ': 25, '6mm': 7, 'và': 155
, 'tràn': 137, 'viền': 153, 'mang': 81, 'lại': 77, 'diện': 38, 'tích': 145, 'hiển': 55, 'thị': 129, 'lớn': 78, 'hơn': 58, 'chân': 22, 'đế': 167, 'cách': 29, 'điệu': 16
4, 'thời': 130, 'thượng': 128, 'không': 66, 'gian': 47, 'làm': 74, 'việc': 154, 'ảnh': 171, 'sắc': 119, 'nét': 99, 'chất': 24, 'lượng': 76, 'độ': 169, 'phân': 102, 'gi
ải': 50, 'full': 46, 'hd': 54, 'trên': 139, 'tấm': 149, 'nền': 101, 'ips': 63, 'sống': 121, 'động': 170, 'thực': 131, 'mọi': 90, 'góc': 52, 'nhìn': 98, 'tương': 146, '
phản': 104, 'cao': 16, '100': 1, '000': 0, 'càng': 28, 'cho': 19, 'thêm': 127, 'chiều': 18, 'sâu': 117, 'chuyển': 20, 'mượt': 87, 'mà': 82, 'cùng': 33, 'trong': 136, '
series': 111, 'còn': 30, 'sự': 124, 'gộp': 53, 'mặt': 89, 'của': 36, 'r221q': 107, 'tần': 150, 'số': 120, 'quét': 106, '60hz': 5, 'hỗi': 60, '4ms': 4, 'riêng': 109, 'đ
ược': 165, 'cải': 35, 'tiến': 133, 'lên': 75, '75hz': 8, '1ms': 2, 'hợp': 61, 'công': 32, 'nghệ': 94, 'freesync': 45, 'loại': 72, 'bỏ': 15, 'tượng': 147, 'bóng': 12, '
ma': 80, 'trỗi': 143, 'sụt': 122, 'khung': 64, 'bảo': 13, 'vệ': 157, 'lực': 79, 'suốt': 115, 'cả': 34, 'ngày': 95, 'siêu': 113, 'bluelight': 11, 'shield': 112, 'giúp':
 49, '80': 9, 'ánh': 161, 'sáng': 116, 'xanh': 159, 'hại': 59, 'mắt': 88, 'khử': 67, 'nháy': 97, 'flickerless': 44, 'giác': 48, 'người': 96, 'dùng': 41, 'xuyên': 160,
'dài': 40, 'sử': 123, 'dụng': 42, 'tránh': 138, 'tình': 144, 'trạng': 141, 'mỏi': 91, 'khó': 65, 'chịu': 26, 'giống': 51, 'chói': 23, 'comfyview': 27, 'vẫn': 156, 'tốt
': 152, 'môi': 86, 'trường': 140, 'thoải': 126, 'mái': 84, 'tiết': 134, 'kiệm': 68, 'điện': 163, 'năng': 100, 'tối': 151, 'đa': 162, 'trang': 135, 'bị': 14, 'với': 158
, 'tiêu': 132, 'chuẩn': 21, 'energy': 43, 'star': 114, 'đến': 168, '68': 6, 'chi': 17, 'phí': 103}
[[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]]
```

TF-IDF

1. Dữ liệu mẫu: Như bài 1
2. Code

```python
from sklearn.feature_extraction.text import TfidfVectorizer
class TFIDF:
    def __init__(self):
        self.documents = ""

    def run(self):
        self.getData()
        self. buildTheVocabulary()
        self.getTDIDF()

    def getData(self):
        file = open(
            "D:/E23.1/NaturalLanguageProcessing/Ex/crawl/phongvu/9. Màn hình LCD Acer 21.5Inch R221QB.txt", "r", encoding="utf8")
        self.documents = file.read()
        self.processed_docs = self.documents.replace(
            ".", " ").replace(",", " ").replace("-", " ").lower().split()

    def buildTheVocabulary(self):
        self.tfidf = TfidfVectorizer()
        self.bow_rep_tfidf = self.tfidf.fit_transform(self.processed_docs)
        # IDF for all words in the vocabulary
        print("IDF for all words in the vocabulary", self.tfidf.idf_)
        print("-"*10)
        # All words in the vocabulary.
        print("All words in the vocabulary", self.tfidf.get_feature_names())
        print("-"*10)
        # TFIDF representation for all documents in our corpus
        print("TFIDF representation for all documents in our corpus\n",
              self.bow_rep_tfidf.toarray())
        print("-"*10)

    def getTDIDF(self):
        self.temp = self.tfidf.transform(["THIẾT KẾ SANG TRỌNG".lower()])
        print("Tfidf representation for 'THIẾT KẾ SANG TRỌNG':\n",
              self.temp.toarray())

if __name__ == '__main__':
```

3. Kết quả

```
All words in the vocabulary ['000', '100', '1ms', '21', '4ms', '60hz', '68', '6mm', '75hz', '80', 'acer', 'bluelight', 'bóng', 'bảo', 'bị', 'bỏ', 'cao', 'chỉ', 'chiều'
, 'cho', 'chuyển', 'chuẩn', 'chân', 'chối', 'chất', 'chỉ', 'chịu', 'comfyview', 'càng', 'cách', 'còn', 'có', 'công', 'cùng', 'cả', 'cải', 'của', 'cực', 'diện', 'doanh
, 'dài', 'dùng', 'dụng', 'energy', 'flickerless', 'freesync', 'full', 'gian', 'giác', 'giúp', 'giải', 'giống', 'góc', 'góp', 'hd', 'hiển', 'hiện', 'hình', 'hơn', 'hại
, 'hỏi', 'hợp', 'inch', 'ips', 'khung', 'khó', 'không', 'khử', 'kiểm', 'kế', 'ký', 'lcd', 'loại', 'là', 'làm', 'liên', 'lượng', 'lại', 'lớn', 'lực', 'ma', 'mang', 'mà',
 'màn', 'mái', 'mô', 'môi', 'mượt', 'mát', 'mặt', 'mọi', 'mỏi', 'mỏng', 'nghiệp', 'nghệ', 'ngày', 'người', 'nháy', 'nhìn', 'nét', 'năng', 'nền', 'phân', 'phi', 'phản',
 'phẩmmàn', 'quét', 'r221q', 'r221qb', 'riêng', 'sang', 'series', 'shield', 'siêu', 'star', 'suốt', 'sáng', 'sâu', 'sản', 'sắc', 'số', 'sống', 'sụt', 'sử', 'sự', 'thiế
t', 'thoải', 'thêm', 'thượng', 'thị', 'thời', 'thực', 'tiêu', 'tiến', 'tiết', 'trang', 'trong', 'tràn', 'tránh', 'trên', 'trường', 'trạng', 'trọng', 'trời', 'tình', 't
ích', 'tương', 'tượng', 'tả', 'tấm', 'tần', 'tối', 'tốt', 'viền', 'việc', 'và', 'vẫn', 'vệ', 'với', 'xanh', 'xuyên', 'ánh', 'đa', 'điện', 'điệu', 'được', 'đại', 'đế',
'đến', 'độ', 'động', 'ảnh']
----------
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.48265943 0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.51675901 0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.48265943
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.51675901 0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.
  0.         0.         0.         0.         ]]
PS D:\E23.1\NaturalLanguageProcessing\Ex>
```