Tut 3: Hướng dẫn tạo TF-IDF và sử dụng KNN

- **Tác giả: Vũ Đức Duy.**
- **Nhóm nghiên cứu về AI của Trường Đại học Bách Khoa thành phố Hồ Chí Minh**

1. Khởi chạy python

Đầu tiên các bạn chạy chương trình Python trong máy (nếu bạn chưa cài thì có thể tham khảo bài của bạn Cao Chánh Dương).

2. Import thư viện

Đoạn code sau đây sẽ import những thư viện cần thiết:

```
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier
```

3. Tạo ma trận TF-IDF

3.1. Cách 1: Sử dụng file list

Sau khi các bạn sử dụng crawler để lấy dữ liệu documents về máy thì hãy chia các document này ra từng file riêng lẻ và để trong cùng một folder. Ví dụ mình có 3 documents với file path tương ứng như sau:

/home/documents/doc1.txt

/home/documents/doc2.txt

/home/documents/doc3.txt

Sau đó ở python, khởi tạo 1 list chứa file path của tất cả các file document.

Ví dụ giả sử bạn có 100 file và nằm ở thư mục như trên:

```
fileList = ["/home/documents/doc" + str(i) + ".txt" for i in range(1,101)]
```

Các bạn sử dụng hàm TfidfVectorizer để tính ma trận TF-IDF bằng fileList trên:

```
vectorizer = TfidfVectorizer(input="filename")
X =  vectorizer.fit_transform(fileList).toarray()
```

3.2. Cách 2: Sử dụng array

Các bạn có thể đọc dataset lên và lưu trong 1 list, với mỗi phần tử trong list phải là chuỗi (nếu mỗi document chứa nhiều chuỗi thì các bạn có thể gộp chúng lại với nhau).

Ví dụ:

array = [‘Hello World’, ‘This is a paragraph.’]

Sau đó sử dụng hàm TfidfVectorizer để tính ma trận TF-IDF bằng array trên:

```
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(array).toarray()
```

*** Lưu ý**: Nếu sau khi sử dụng TfidfVectorizer trên tập data đã có nhưng muốn áp dụng nó lên 1 tập data mới thì có thể sử dụng hàm **transform**:

```
new_X = vectorizer.transform(new_array).toarray()
```

4. KNN

Trước khi đưa dữ liệu huấn luyện vào model, chúng ta cần gán label cho từng document, label ở bài lab này là category. Các bạn mã hóa category của từng document rồi gộp lại thành 1 vector có kích thước là (1,n) với n là số lượng document.

Sau khi đã có ma trận TF-IDF (X) và label vector (tạm gọi là Y), thì có thể bắt đầu đưa vào KNN.

Chạy lệnh sau:

```
n = 5                    # Số lượng neighbors của KNN

neigh = KneighborsClassifier(n_neighbors=n)
neigh.fit(X, Y)          # X và Y là training data, tương ứng với TF-IDF và label vector
```

Sau khi chạy đoạn code trên thì model đã được “fit” vào tập training , bây giờ bạn có thể dự đoán được 1 document mới sẽ được phân loại vào category nào thông qua hàm predict. Nhưng trước tiên các bạn vẫn phải dùng TfidfVectorizer để chuyển document thành ma trận tf-idf:

```
# Các bạn sử dụng vectorizer đã tạo ở mục 3.
new_X = vectorizer.transform(new_array).toarray()      # new_array là list chứa document mới
```

Sau đó sử dụng hàm predict để dự đoán label của những document mới này:

```
neigh.predict(new_X)
```

5. Tài liệu tham khảo

Chi tiết về prototype của hàm các bạn có thể tham khảo ở link dưới đây:

http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

http://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction

Để tìm hiểu kỹ hơn về KNN trong sklearn các bạn có thể tham khảo link sau:

http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

http://scikit-learn.org/stable/modules/neighbors.html#classification