

**자연어처리 및 정보검색 0분반 – Homework 3**

Student ID: 20133096

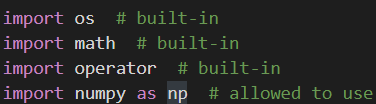
Student Name: 이현재

**프로젝트 목표:**

|  |
| --- |
| 2번 과제에서 사용자로부터 입력받은 Query를 바탕으로 Tf-Idf weight값을 인자로 가지는 Vector Space Model을 통해 Cosine Similarity를 구했다. 여기에 SVD를 통해 3개로 분리된 각각의 Matrix를 구하고, 이를 통해 K값을 입력하여 각 문서와 K만큼의 Query vector와의 Cosine similarity를 구한다. |

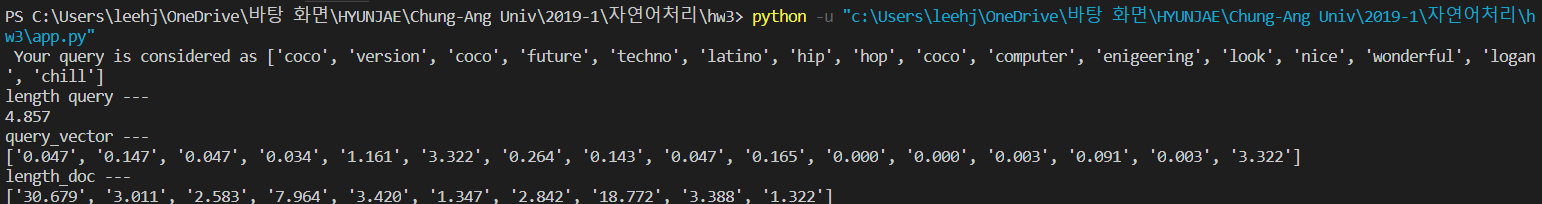
**프로젝트 설명:**

1. 입력 텍스트파일 (Query)와 주어진 텍스트 파일(Document 1 ~ 10) 까지의 유사도(similarity)를 측정하여 ranking 한다.
2. TF (Term frequency)는 특정한 단어가 문서 내에 얼마나 자주 등장하는지를 나타내고, DF(Document frequency)는 단어가 문서내에서 등장했는지를 나타낸다. 따라서 TF-IDF 가중치를 사용하기위해 DF의 역수인 IDF(Inverse Document Frequency)를 사용한다.
3. **외부라이브러리 사용의 제한으로 Python의 built-in 라이브러리인 파일 입출력을 위한 os와 log 및 계산을 위한 math, operator와 사용이 허용된 numpy만을 사용하였다.**

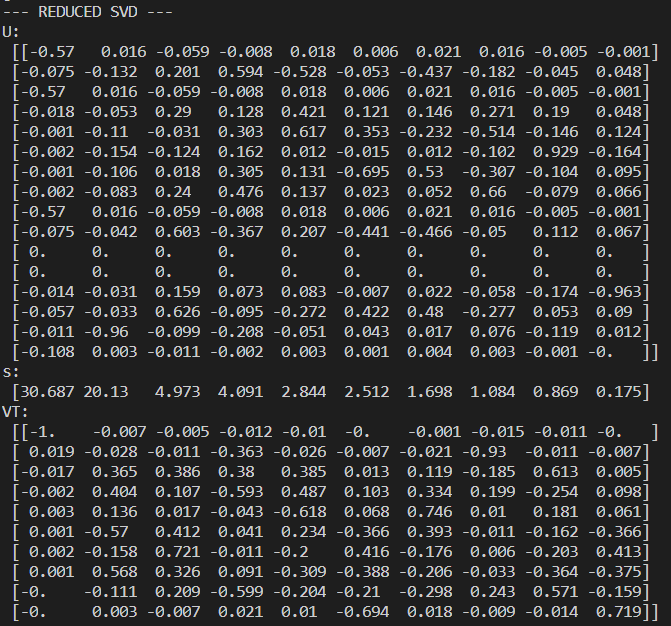


**프로젝트 결과:**

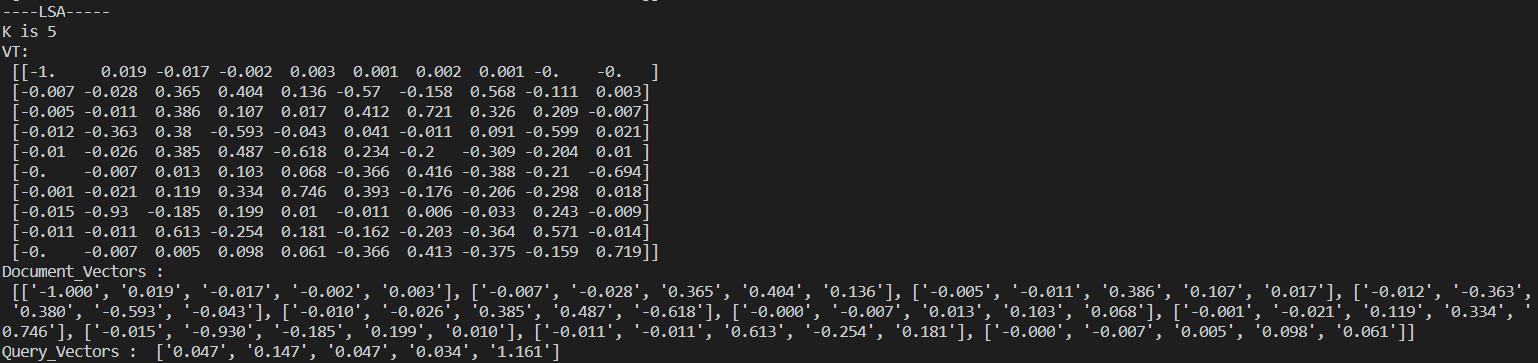
1. VSM을 통해 각 Query vector와 Document의 길이를 구한다.



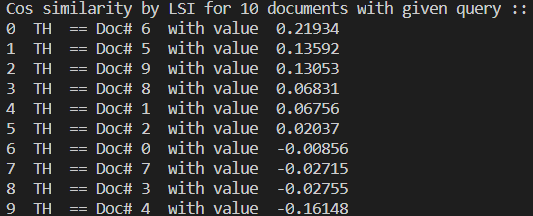
1. 주어진 VSM을 통해 각 Query 단어에 해당하는 SVD (3개의 Matrix)를 구한다.



1. N개의 Query단어 중에서 K개만을 선택해 이에 해당하는 Document vector와 Query vector를 구한다

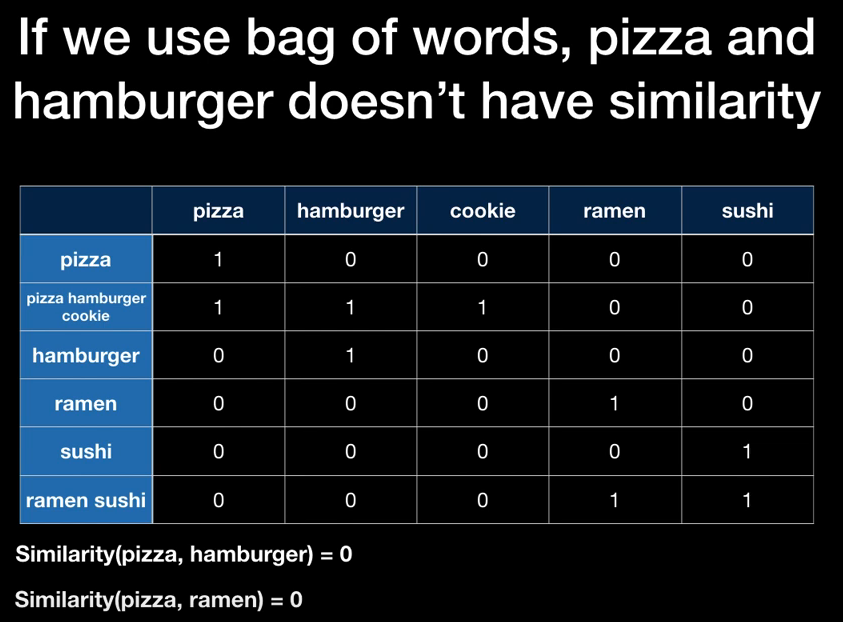


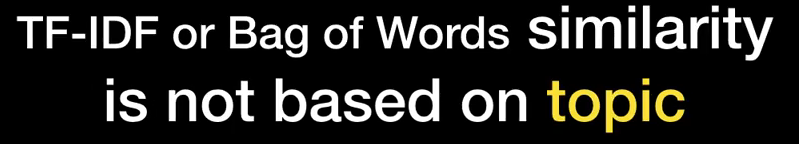
1. 이를 통해 각 문서별로 주어진 Query와의 Similarity를 구한다.

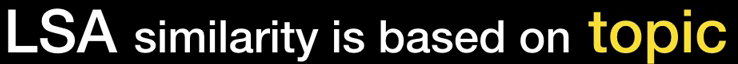


**프로젝트 과정:**

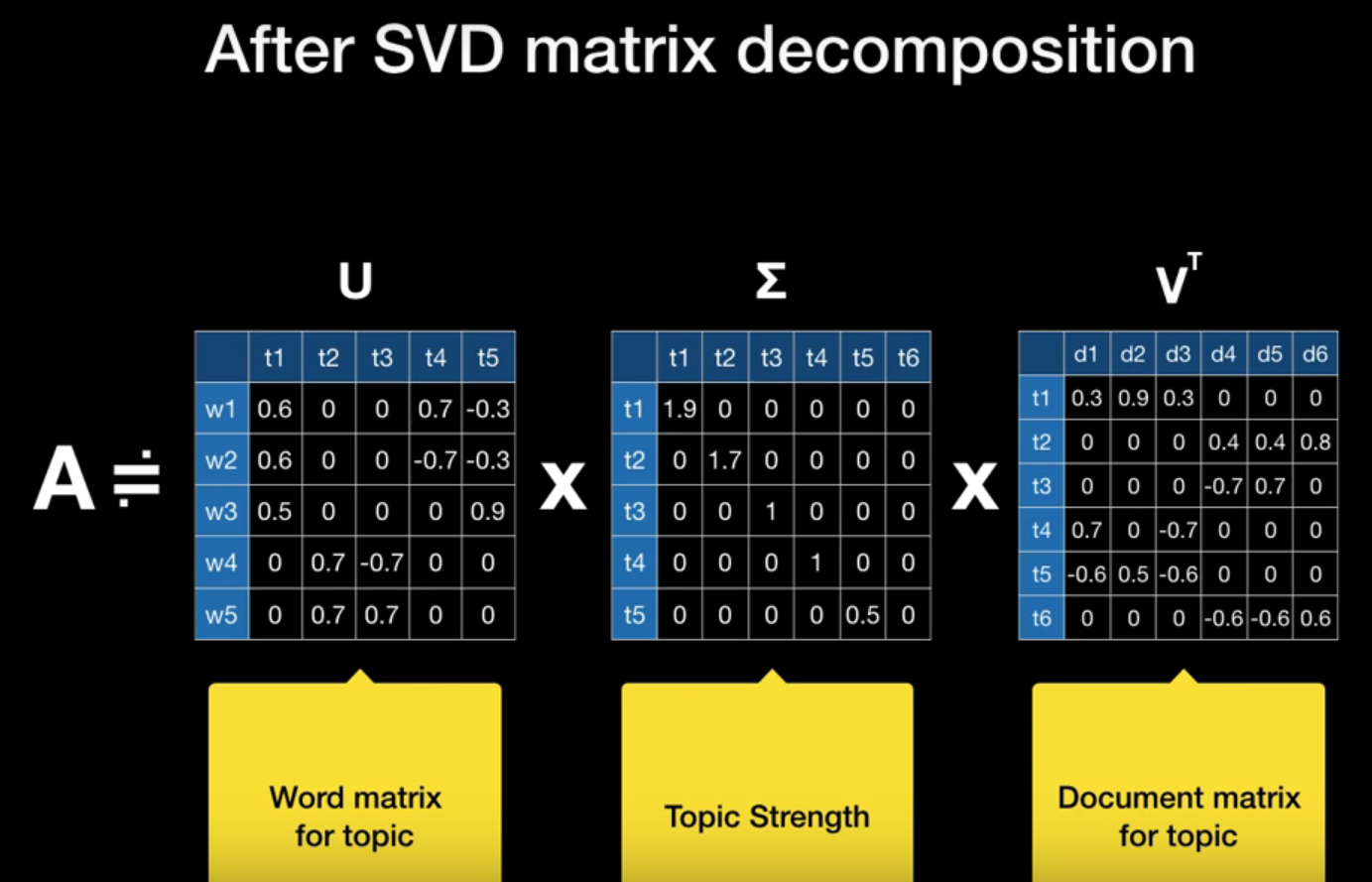
1. 먼저 참고한 자료는 해당 [URL](../../../../../../../Desktop/LSI-Eg.pdf)이다.
2. 아래의 자료를 통해 LSI의 필요성을 먼저 이해했다. 단순히 VSM과 Query만을 비교하면, Word만을 기준으로 하여 유사도가 측정되지만, Word간의 유사도는 전혀 고려되지 않는다.



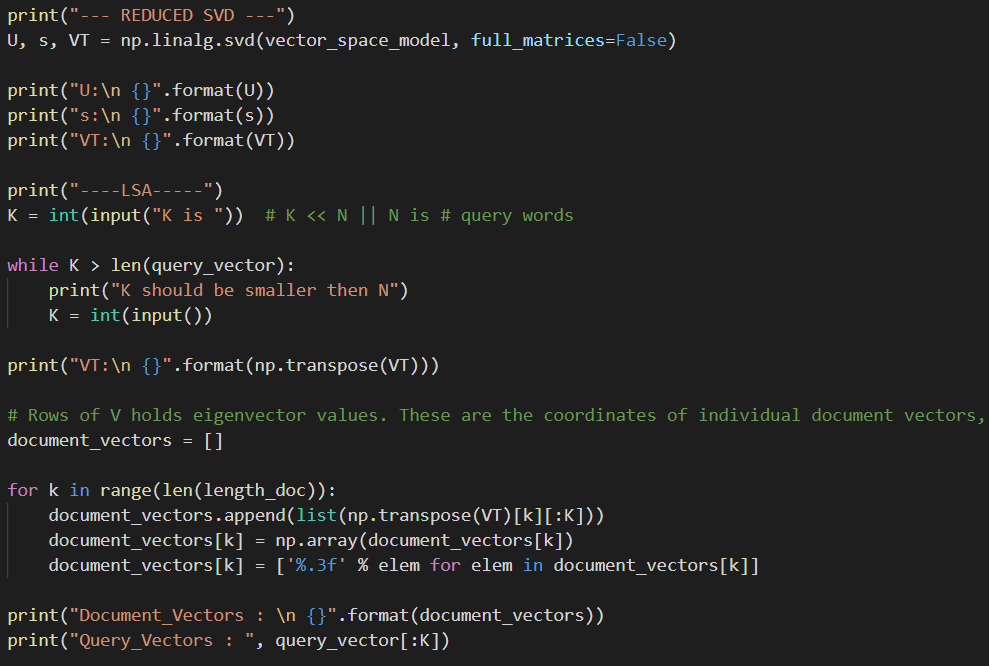




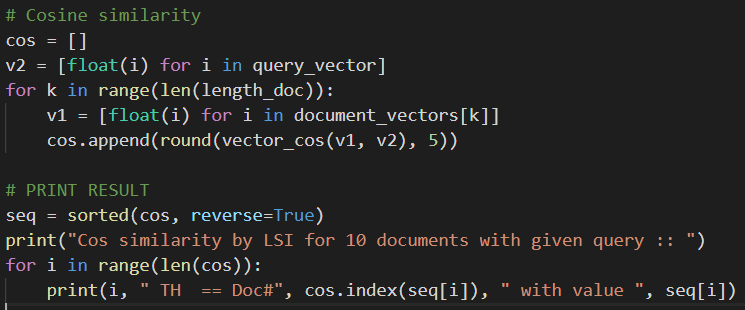
1. 따라서 LSA를 이용하여 ‘주제’를 고려하여 단어별로 유사도를 측정하고, 이를 통해 문서와 단어간 유사도를 유추할 수 있다.



1. 따라서, Numpy를 활용하여 기존의 VSM에 SVD를 다음과 같이 적용하였다.



1. 마지막으로 측정된 Document\_Vecotr와 Query\_Vector를 기준으로 문서별로 Cosine Similarity를 측정한다.



**결론:**

|  |
| --- |
| 이전과제에서 VSM을 설계하고, Query마다 각 문서별로 단어기준의 Cosine Similarity를 구했다. 하지만, 이는 단어간 유사도를 전혀 고려하지 않고, 오로지 단어의 빈도수만을 기준으로 측정된 것이다. 이점에서 단어간 유사도라고 정의 될 수 있는 ‘주제’ 를 고려하기 위해 새로운 방법인 LSI를 직접 적용해보면서 같은 데이터로부터 더 흥미로운 결과를 얻을 수 있었다. |