

|  |
| --- |
| **Latent Semantic Analysis (LSA)** |

|  |  |
| --- | --- |
| 과목 | 자연어처리 및 정보검색 |
| 담당교수 | 정재은 교수님 |
| 학과 | 경제학부 |
| 학번 | 20153029 |
| 이름 | 김민지 |
| 제출일 | 2019.04.18 |

**Latent Semantic Analysis**

**1. 10 Movie scripts**

- Doc01: Beauty and the Beast

- Doc02: Coco

- Doc03: Deadpool

- Doc04: It

- Doc05: La La Land

- Doc06: The Great Gatsby

- Doc07: The Martian

- Doc08: The Secret Life of Walter Mitty

- Doc09: Thor: Ragnarok

- Doc10: War for the Planet of the Apes

**2. Analysis of previous HW**

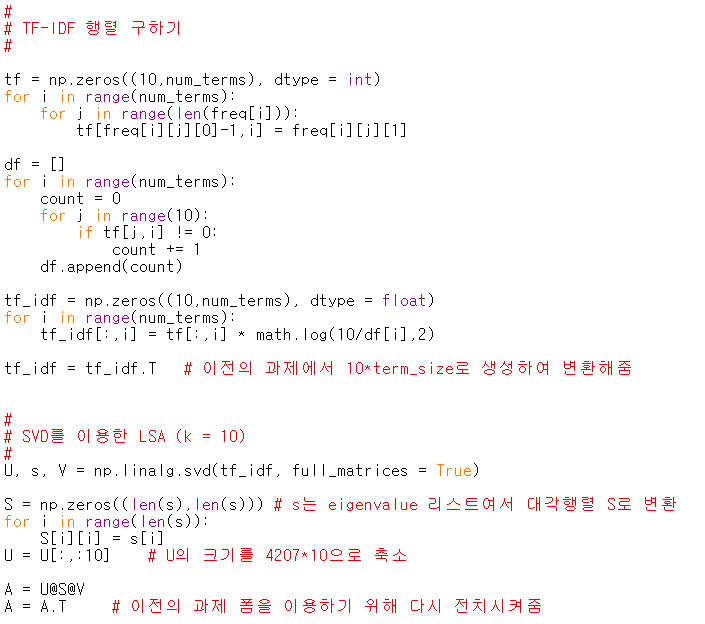
이전의 과제는 10개의 문서의 모든 단어들의 tf-idf를 구하였다. 이 행렬의 크기는 (10\*16,230) 였는데, 이 모든 단어들로 랭킹 프로그램을 생성하였다. 즉 필자는 16,230차원을 이용한 랭킹 프로그램을 생성하였다. 그 결과 행렬에 0에 가까운 값들이 매우 많은 행렬을 이용한 매우 비효율적인 랭킹 프로그램이 생성되었다. 필자는 모든 단어로 랭킹 프로그램을 생성해야 모든 단어에 대해 검색을 할 수 있으니 더 좋을 것이라 생각하였는데 이는 잘못된 생각이었다.

또한, sim을 계산하는 함수에 문제가 있었던 건지 쿼리에 “beast”를 입력하면 당연히 영화 “Beauty and the Beast”인 문서 1이 1순위로 나와야 하나, 10 - 4 - 2의 순으로 랭킹이 매겨졌다. 이는 sim을 계산하는 함수 식에 문제가 있었던 것으로 보인다.

또한, 결과에서는 정확성이 높게 나왔으나, 이는 rel의 기준을 너무 광범위하게 잡아서 정확성을 높인 것이기 때문에 옳지 않은 랭킹 프로그램이었다. 이렇게 설정해주었던 이유는 쿼리를 직접 입력하지 않고 랜덤으로 많은 단어를 뽑고 싶어 그렇게 하였더니 빈도수가 너무 낮은 단어들이 선택되어 관련된 문서가 없다고 나타나는 경우가 많아서 tf-idf값이 5를 넘는 단어는 모두 관련이 있다고 설정하였더니 정확도가 높아져서 이렇게 설정해 주었던 것이었다. 하지만 이는 정확한 결과가 아니었다.

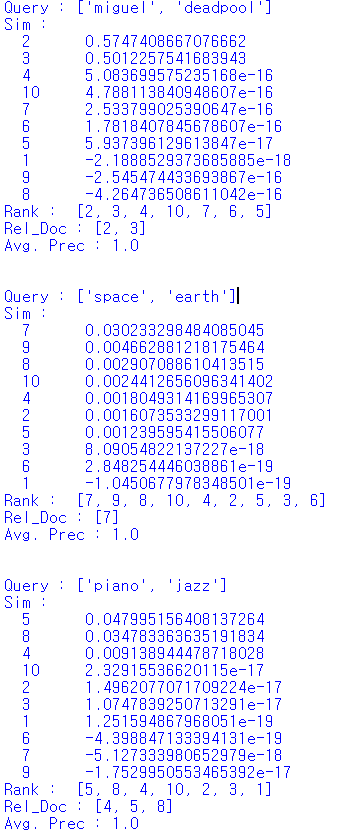
이번 과제를 하기 위해서는 이전 과제와의 비교를 통해 LSA의 특성을 살펴봐야한다고 판단하였다. 비교를 하기에는 이전의 랭킹 프로그램이 너무 정확하지 않은 프로그램이라고 판단되어 일부분을 수정해주었다. 우선, sim을 계산하는 함수를 수정해주었고, rel의 기준은 tf-idf의 값이 20이상인 것으로 높여주었다. 그리고 제거 부호에 ‘;’도 추가해주었다. 또한, 각 문서에서의 top 1000까지의 단어만 포함하여 총 4,207개의 단어로 행렬을 구성하여 주었다.

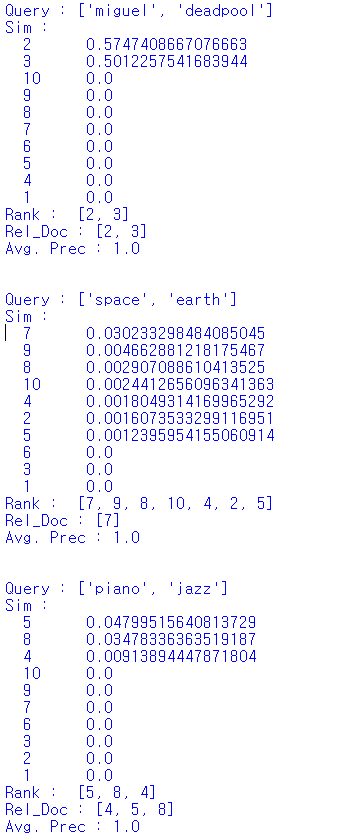
**3. Code**



코드는 과제 2의 코드와 크게 달라진 부분은 없고 TF-IDF를 이용하여 SVD를 구하여 LSA를 수행하는 부분을 추가해주었다. 이전 과제에서 TF-IDF 행렬을 10 \* terms\_num으로 설정하여 주었기 때문에 전치를 시켜주었다. 이를 이용하여 k = 10으로 SVD 처리를 하였고, 이의 값을 A 행렬에 저장해 주었다. 그리고 이를 다른 함수에 적용하기 위해 다시 전치를 시켜주었다.

**4. Coding Result**





왼쪽의 결과는 이전 과제의 결과이고, 오른쪽의 결과는 이번 과제의 결과이다. 이 둘의 비교해보면 가장 두드러지는 차이가 sim 값의 차이라고 볼 수 있다. 이전의 과제의 결과에서는 쿼리와 관련없는 문서와의 sim의 값은 0이었는데, LSA 과정을 거친 결과를 보면 0에 가까운 매우 작은 값이거나 음수의 값을 가지는 등 좀 더 정확한 유사도 분석을 수행한다. 하지만 전체적인 값의 크기를 비교해 보면 매우 비슷하다. 그리고 이 외의 다른 값들도 다 비슷했다.

**4. Analysis**

기존의 TF-IDF 행렬은 단어의 의미를 전혀 고려하지 못한다는 단점을 갖고 있었다. 그에 비해 LSA는 TF-IDF 행렬에 SVD를 사용하여 차원을 축소시키는 방식으로, 단어들의 잠재적인 의미를 끌어내는 더 좋은 랭킹 프로그램 방식이라고 알려져 있다.

실제로 구현을 해본 결과 이전의 프로그램의 결과 자체가 정확도가 높았기 때문인지 큰 차이는 보이지는 않았으나, 유사도 분석을 좀 더 정밀하게 계산해준다는 것을 알 수 있었다. 또한, 생각보다 LSA의 구현이 어렵지 않아서 차원을 쉽게 줄일 수 있는 좋은 방법이라는 것을 느낄 수 있었다.