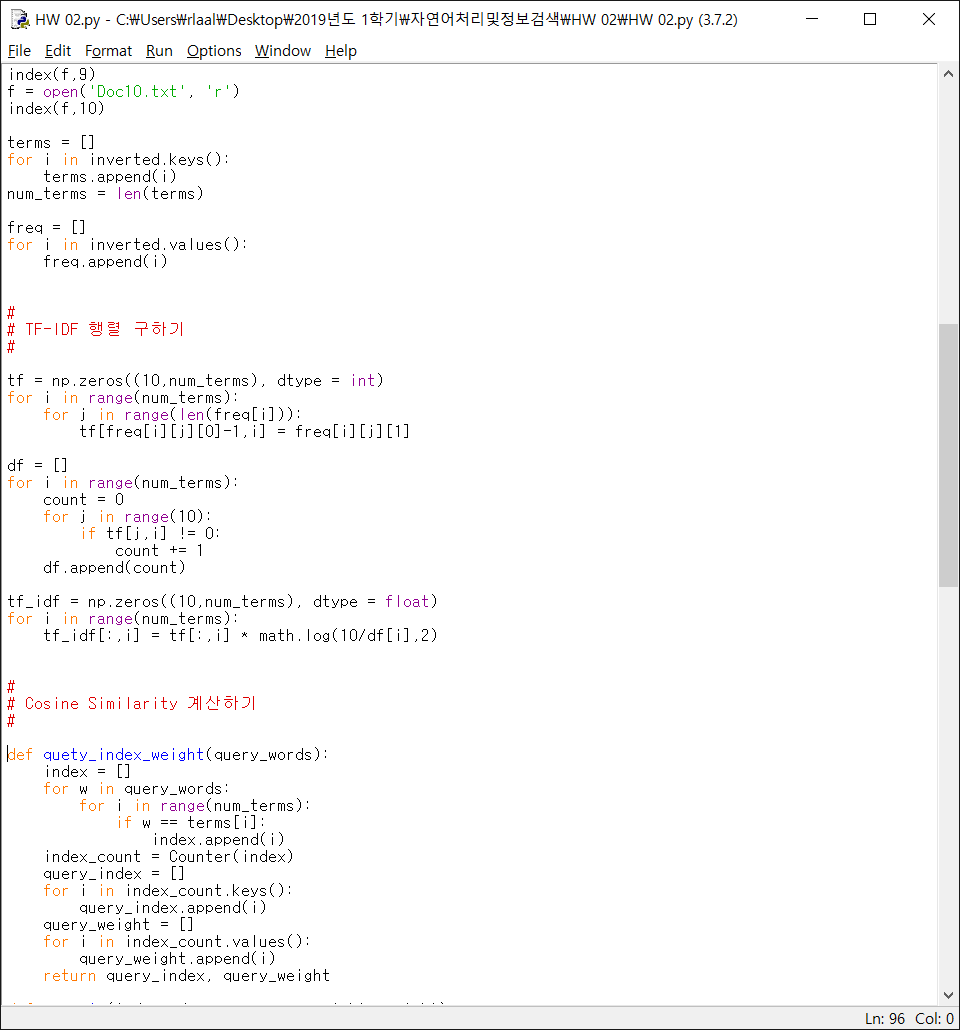
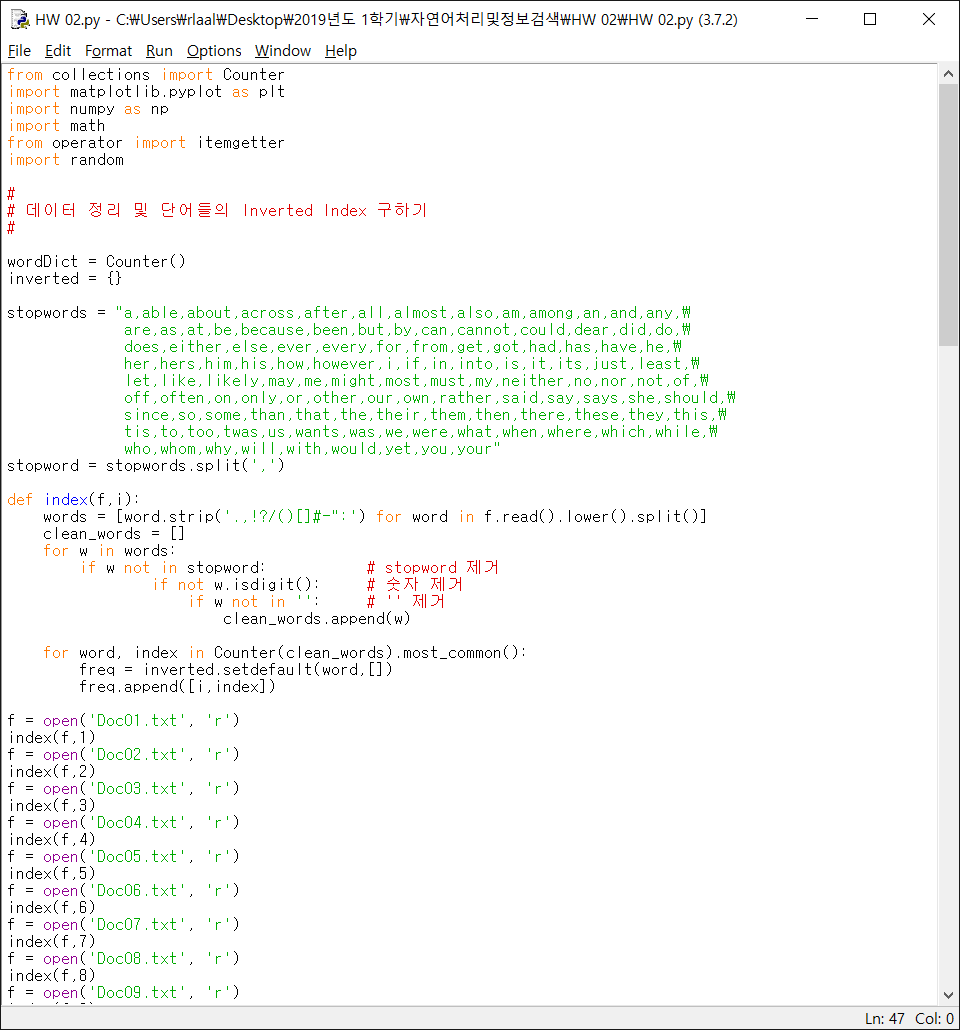


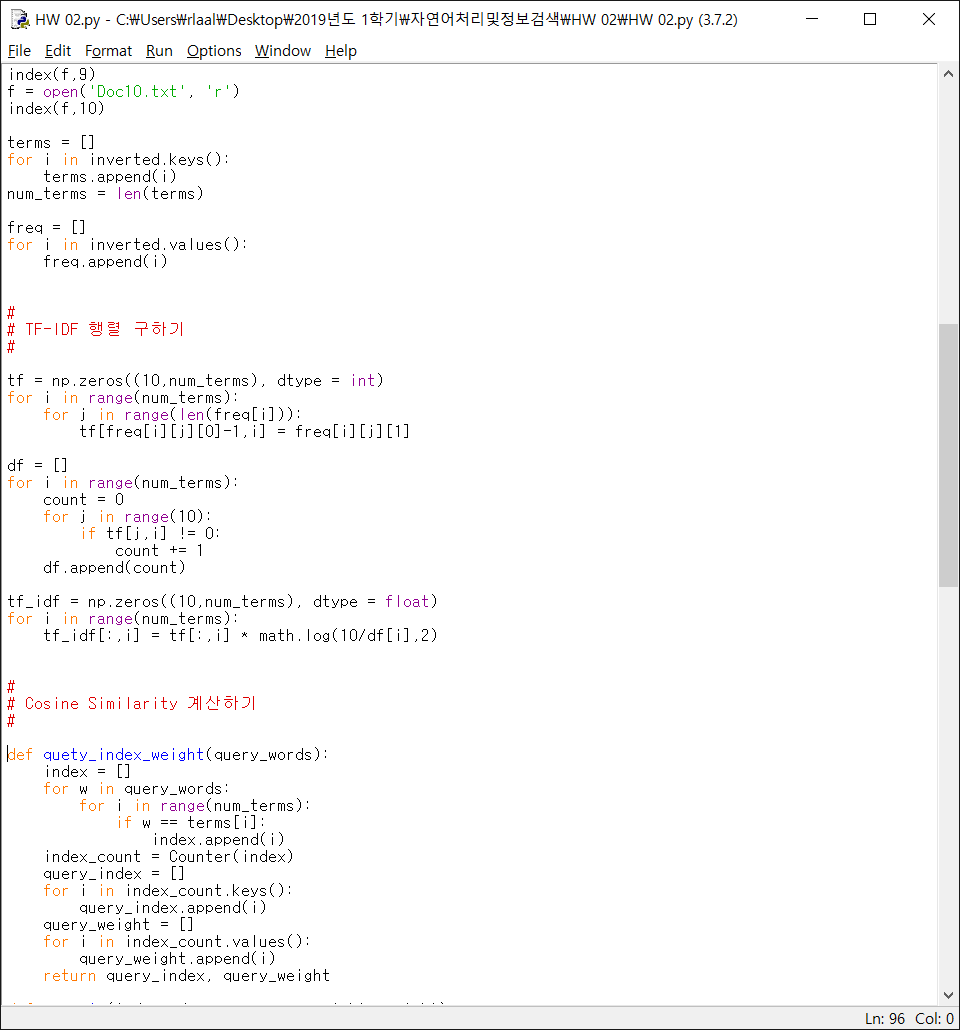
|  |
| --- |
| **Text Ranking Report**  **- Vector space model -** |

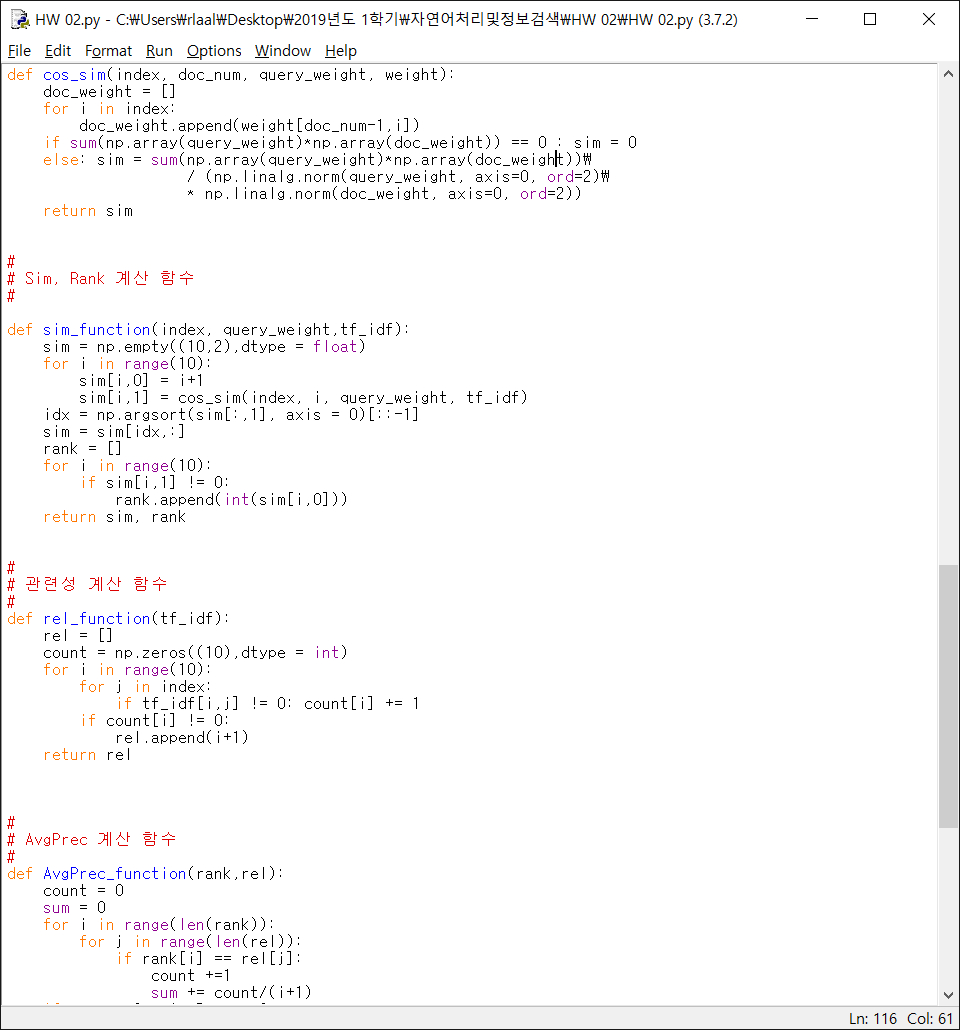
|  |  |
| --- | --- |
| 과목 | 자연어처리 및 정보검색 |
| 담당교수 | 정재은 교수님 |
| 학과 | 경제학부 |
| 학번 | 20153029 |
| 이름 | 김민지 |
| 제출일 | 2019.04.04 |

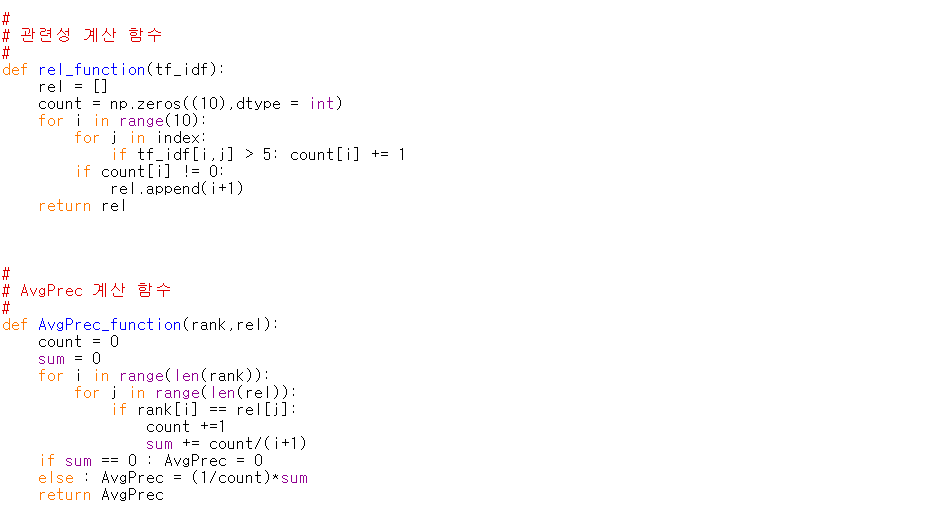
**Text Ranking**

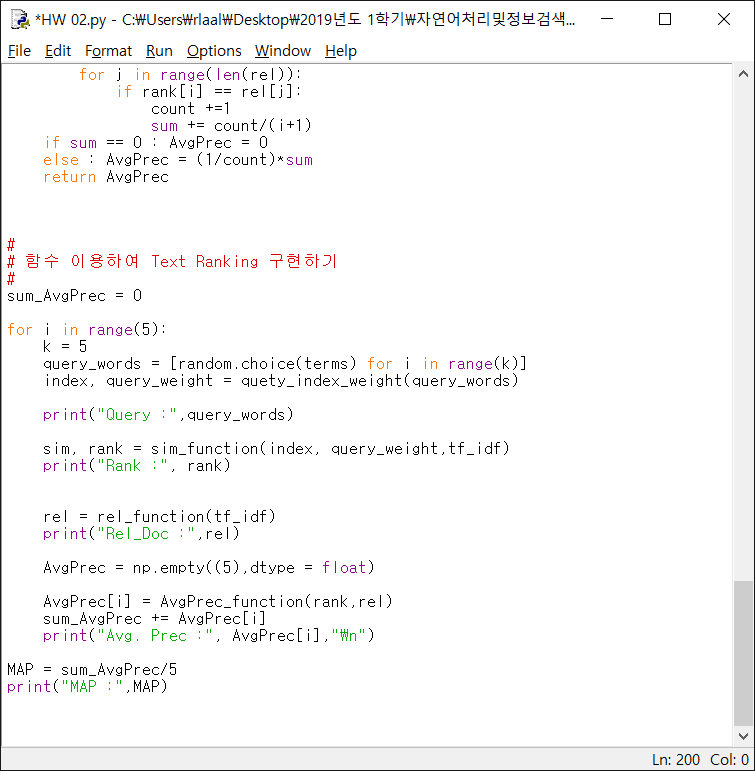
**1. Code**

****

****

****

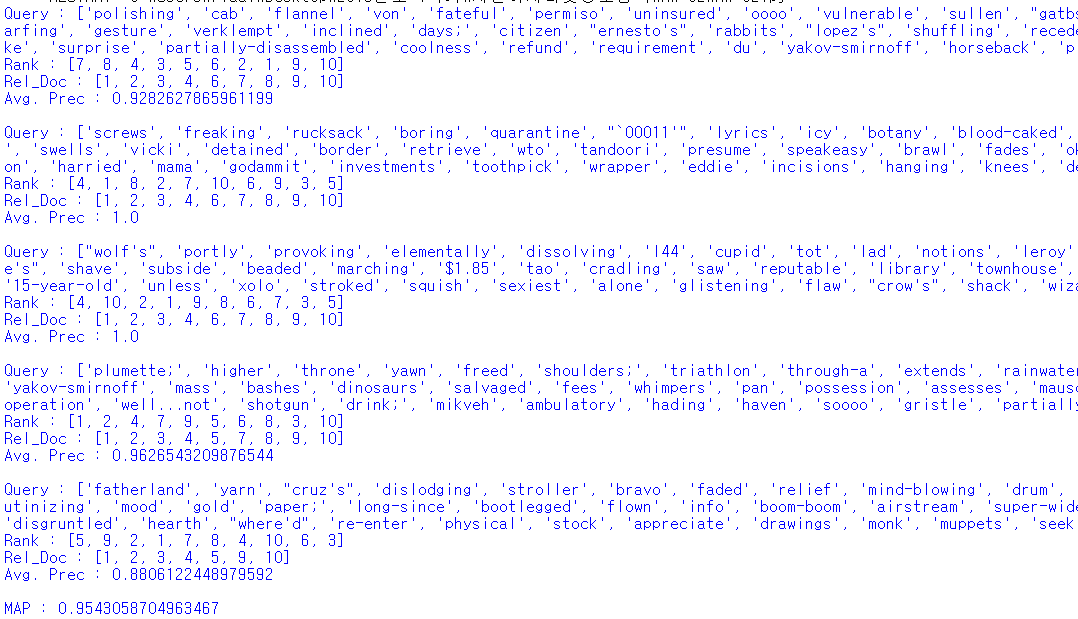
****

****

Text Ranking을 하기 위해 저번 과제에서 구한 inverted index를 이용하여 10개 문서의 모든 단어들의 TF-IDF 행렬를 구하였다. 그 후, Cosine Similarity를 이용하여 쿼리와 각문서들의 유사도를 계산하였다. 이때 vector space를 2차원으로 두고 계산하였는데, 그 이유는 2차원이 가장 이해가 쉽고, 한 문서와 쿼리 사이의 유사도를 구할 것이기 때문에 2차원이 편하다고 생각하였기 때문이다. 유사도를 구한 후, 문서 번호들을 유사도 순으로 랭킹을 만들었다. 랭킹의 정확도를 계산하기 위해 쿼리의 단어들과 문서들 간의 관련도를 설정해 주어야 한다. 그래서 쿼리의 단어들의 TF-IDF의 값이 5보다 높으면 모두 관련이 있다고 설정하였고, 이를 이용하여 쿼리와 관련이 있는 문서들의 번호를 리스트에 저장하였다. 랭킹과 관련도를 이용하여 평균정확도(AvgPrec)를 구하였고, 5개의 쿼리를 생성하여 실행한 후 이들의 MAP도 구해주었다.

**2. Coding Result**



****

쿼리를 terms 리스트에서 랜덤으로 10, 30개의 단어를 뽑아서 생성하도록 프로그래밍을 해주었다. 위의 사진들은 프로그램을 시행한 결과를 보여주고 있다. 첫번째와 두번째의 결과를 비교해보면 첫번째의 결과는 쿼리가 10개의 단어로 이루어져 있으며 MAP는 약 0.425이고, 두번째 결과의 쿼리는 30개의 단어로 이루어졌으며 이의 MAP는 약 0.95이다. 이 두 결과를 비교하여 보았을 때, 쿼리의 단어가 늘어날수록 정보검색의 정확도가 올라간다는 것을 알 수 있다.

**3. Analysis**

위의 결과로 알 수 있는 이 랭킹 프로그램의 장점은 쿼리의 단어 수가 많을 때, 정확도가 매우 높아진다는 점이다. 따라서 긴 질문을 할 때 매우 유용할 것이다. 또한, 데이터를 처리하는 과정에서 모든 문자를 소문자로 변환해주고, stopword를 제거해주고, 숫자를 제거해주는 등 불필요한 중복과 유용하지 않은 단어들을 제거해줌으로써 정확도를 더 높이는데 기여한 것도 장점이라 볼 수 있다.

하지만 단점은 단어 수가 적어지면 정확도가 떨어지며, 정확도를 높이기 위해 단어의 TF-IDF가 5 보다 높으면 무조건 관련이 있다고 설정하였기 때문에 실제적인 관련성은 결과에 비해 낮을 수 있다는 것이다. 또한, 이 프로그램에서는 쿼리를 저장된 terms리스트에서 뽑아온 단어들로 구성하였기 때문에 정확도가 좀 더 높게 나온 것으로 예상된다. Terms 리스트 안의 단어가 아닌 임의의 단어로 검색을 하였으면 정확도는 매우 낮게 나왔을 것이기 때문에 이 프로그램의 MAP 값과 실제 정확도와는 차이가 있을 것으로 보인다.

이 랭킹 프로그램의 성능을 향상시키기 위한 방안을 생각해보았다. 우선, 쿼리와 문서 간의 관련성을 정확히 하기 위해 관련성의 기준을 좀 더 바람직한 기준으로 변경해주어야 할 것이다. TF-IDF가 5보다 큰 경우 관련이 있다는 조건은 너무 현실성이 저하되기 때문이다. 또한, 정확도를 높이기 위해서는 유사성과 관련도와의 관계에서 현실을 반영한 처리를 해주어야 한다고 생각한다. 그리고 유사성, 정확성 등을 구하는 다른 방법들도 존재하기 때문에 어떤 방법을 이용하여 계산했을 때가 가장 좋은 결과를 보여주는지 확인해보는 것이 성능을 향상시키는데 도움이 될 것이라고 생각한다.