컨텐츠 필터링 기반 영화 추천 시스템

팀 명 최민혁 팀 원 최민혁 지도교수 이 슬 멘 토 이 슬

개발 동기 및 목적

최근에 많은 사람들이 많이 사용하는 넷플릭스는 영화 추천 시 스템을 통해 사람들이 지속적으로 자신들의 시스템을 이용하 도록 하는 대표적인 플랫폼이다. 넷플릭스는 협력 기반 필터 을 사용하여 개인의 취향을 강하게 반영하여 추천하는데, 본 연구에서는 컨텐츠 기반 필터을 사용한 추천 시스템 또한 실생활에서 사용될 수 있을 만큼 실효성이 있는지 확인하기 위 해 수행되었다. 이를 확인하기 위해서 각 영화의 줄거리를 주 요 키워드들로 Token화 시키고 각 단어들에 대한 tf-idf 점수 를 사용하여 영화 간의 Cosine Similarity를 측정하는 방법 을 사용하였다. Cosine Similarity가 높은 순으로 정렬한 뒤 상위 목록의 영화들을 선택함으로써 사용자가 선호하는 영화 와 비슷한 영화를 추천할 수 있을 것이다. 다만 특정 영화와 비 슷한 영화밖에 추천하지 못하고 개인의 취향을 반영하지 못할 것으로 예상된다. 그러므로 사용자가 매긴 평점을 모두 알 수 있는 환경이라면 협력 기반 필터을 사용한 추천 시스템이 컨텐츠 기반 필터을 사용했을 때 보다 성능이 좋을 것으로 예 상된다. 그러나 단순히 컨텐츠의 정보들만 주어진 경우 컨텐 츠 기반 필터은 비교적 간단한 방법으로 추천이 가능하다. 이러한 도메인 마다 선택된 추천 시스템은 사용자가 자신이 원 하는 상품을 보다 쉽고 빠르게 찾도록 돕는다.

주요기술

1.Content-based Filtering 컨텐츠 기반 필터을 사용하기 위해서는 줄거리의 단어 꾸러기들을 벡터화할 필요가 있었다. 모든 영화의 줄거리에서 사용된 단어들에 대한 Bag of Word를 구성하고, 그림5처럼 각 단어들에 대한 tf-idf 값을 계산하여 tf-idf 행렬을 구성하였다. 이러한 tf-idf 값들은 각 영화가 대표하는 값으로 볼 수 있다. 즉, 우리가 선호하는 영화와 다른 영화들의 유사도는 그림처럼 tf-idf끼리 Cosine 유사도를 구한 값이 된다.이 때 Cosine 유사도가 가장 높은 순으로 정렬하면 가장 유사한 영화들을 찾을 수 있다. 사용자가 선호하는 장르,감독,배우는 무조건적으로 포함해야 유사하다고 가정하였다.장르, 감독, 배우는 특성상 다를 경우 유사도가 0이라고 판단했기 때문이다. 종합적으로 Cosine 유사도에 의해 정렬된 리스트 중 상위 10개의 영화를 뽑아서 추천하는 방식으로 진행하였다.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

개발 내용

먼저 네이버 영화 사이트의 페이지 패턴을 분석한 뒤 영화의 개봉년도 변수를 사용하여 개봉년도 별 영화의 데이터를 모을 수 있었다. 이때 데이터를 읽을 각 년도 별 페이지 수를 딕셔너리로 미리 저장해 놓고 데이터를 모았다. 그 뒤 영화의 id, 제목, 평점, 등급, 장르, 감독, 배우, 줄거리를 각각의 영화 데이터 안에 할당을 하여 csv파일로 저장을 하였다. 저장된 영화의 데이터를 대표하는 값은 이제 줄거리이다.다만 영화의 줄거리를 수치화를 해주기 위해서, NLTK(자연어 처리)패키지를 사용하여 불용어를 불러오고, 줄거리에 있는 사용하지 않는 단어들을 제거하는 작업을 하였다. 또한 데이터셋에 존재했던 결측값들을 공백으로 채워주는 작업을 하였다. 불용어가 제거된 줄거리는 이제 Kkma 패키지에 의해서 명사로 토큰화 되어 존재하게 된다.그리고 이 토큰들을 공백을 사이로 두고 join시키면 정제된 줄거리가 완성된다. 그 후 이 정제된 줄거리들을 Content-based Filtering을 통해서 각 영화간의 Cosine 유사도를 구하고, 이 유사도가 높은 순으로 정렬하면 가장 유사한 영화들을 구할 수 있다.

결과 및 분석

실제로 선호하는 영화의 데이터로 장르:액션, 배우:톰크루즈를 입력했을 때, 배우와 장르를 잘 고려한 영화가 추천되는 것을 볼 수 있었다.장르나 배우가 아닌 영화 제목과 장르를 입력했을 때도 나름 합리적인 추천을 하였다. 결론적으로, 각 영화의 줄거리 내의 주요 키워드들을 tf- idf값으로 치환하여 각 영화 간의 Cosine 유사도를 측정하였고 이는 충분히 유효했다고 판단된다. 그러나 영화 데이터의 크기가 넷플릭스 정도 규모로 커진다면 BOW에 의해 생성된 tf-idf 행렬이 매우 sparse해지므로 메모리 낭비와 부족이 심해진다. 그러므로 데이터의 규모가 커지게 되면 tf-idf 점수를 사용한 유사도 비교는 힘들 것으로 판단된다. 또한 feature의 개수를 줄여야 메모리와 시간적 면에서 득이 되기 때문에 사용하지 않는 단어를 더 세세히 제거한다면 성능이 향상될 여지가 있다. 다만 컨텐츠 기반 필터의 특성상 한계가 존재하고, 개인의 취향을 배제한 특정영화와 비슷한 영화만을 추천해주기 때문에 다양성이 떨어지게 된다. 그러므로 사용자의 평점 데이터를 충분히 모을 수 있다면 협력 기반 필터을 사용하는 것이 원하는 결과를 도출하는데에 적합할 것으로 보인다.