

영화 평점 제공 서비스

과목 언어공학과 문화콘텐츠

교수 _____ 구영은

제출일 2023. 6. 11.

학과 인공지능융합전공

학번 2022311113

이름 _____ 이소현

목차

I. 기획 의도	3
II. 서비스 프로세스	3-4
III. 언어처리 도구	
1. 크롤링	4
2. 전처리	4-5
3. 기계학습	5
IV. 실험 결과	6
V. 기대 효과	6-7
VI. 한계	7-8
참고 문헌.....	9

I. 기획 의도

많은 사람들은 영화를 보기 전, 그 영화가 볼 만한 가치가 있는지 확인을 하기 위해 해당 영화의 평점을 확인할 수 있다. 로튼토마토, IMDb, 왓챠피디아 등의 사이트는 평론가 혹은 관람객의 평가를 기반으로 영화의 평점을 제공하고 있다. 지금은 종료되었지만 과거에는 국내 유명 포털 사이트 '네이버'에서도 각종 영화의 정보를 제공하는 웹사이트 '네이버 영화' 서비스를 통해 평론가와 관람객이 영화에 부여하는 평점과 리뷰를 찾아볼 수 있었다.

그러나 위와 같은 사이트들의 평점 서비스가 영화의 가치를 신뢰도 있게 평가해 주었다고 하기는 힘들다. 관람객들이 특정 영화를 볼 때 큰 재미 혹은 감동을 느꼈으면 최고점을, 그렇지 않았으면 최저점을 부여하여 영화 평점이 극단적으로 분포되는 사례가 존재하며, 그렇지 않다고 해도 평점 하나만으로는 어떠한 기준으로 점수가 부여되는지 잘 드러나지 않기 때문이다. 물론 영화 리뷰 글들을 하나씩 찾아 읽어보면 해결될 만한 문제이지만 이는 매우 불편하고 번거로운 방식일 것이다. 또한, 대다수의 사람들이 작품성이 현저하게 떨어진다고 느끼는 영화에 많은 네티즌들이 높은 점수를 부여함으로써 아직 해당 영화를 관람하지 않은 사람들을 속이는 경우도 존재한다. 실제로 2004년에 개봉한 한국 영화 '클레멘타인'은 개봉 당시에는 흥행에 실패했지만, 수많은 네티즌들의 단합을 통해 네이버 영화에서 9점이 넘는 높은 평점을 받은 일이 있었다.¹⁾ 이렇듯 현존하는 영화 평점 서비스의 신뢰도는 높지 않다. 단순히 네티즌들로부터 가볍게 부여된 점수들을 기반으로 평점을 계산하여 제공하는 것이 아닌, 솔직하게 작성된 장문의 후기 글들을 분석하고 영역별로 점수를 제공하는 서비스를 만든다면 사람들이 본인의 취향에 따른 영화 선정을 하는 데 더 큰 도움을 줄 수 있을 것이라고 생각하였다.

II. 서비스 프로세스

우선, 후기 글들의 감정을 분석하기 위해 프로그램이 문장의 감정을 판별할 수 있도록 기계 학습을 시킨다. 각종 영화 리뷰글들과 각 글마다 담겨 있는 감정을 사전에 학습시킴으로써 프로그램이 문장을 읽고 그 문장에 긍정적인 감정이 담겨 있는지, 부정적인 감정이 담겨 있는지 구분할 수 있도록 한다.

사용자가 프로그램에 보고 싶은 영화의 제목을 입력하면 프로그램은 해당 영화에 대한 리뷰가 작성되어 있는 수많은 블로그 글 데이터를 모은다. 그리고 각 블로그 글에 있는 수많은 문장들 중 해당 영화의 연출에 관한 내용이 담긴 문장과 스토리에 관한 내용이 담긴 문장만 구분하여 추출한다. 프로그램은 각 문장에서 '연출', 'OST', '영상미', 'CG' 등의 단어가 존재하면 연출에 관한 내용으로, '스토리', '개연성', '결말' 등의 단어가 존재하면 스토리에 관한 내용으로 인식한다. 이렇게 연출 혹은 스토리와 관련된 문장을 구분하여 리스트로 모아 놓은 다음, 각 문장마다 긍정적인 감정이 담겨 있는지 혹은 부정적인 감정이 담겨 있는지를 파악함으로써 해당 영화의 연출에 대하여 긍정적인 감정이 담

¹⁾ 최경민, "이동준 50억 날린 '클레멘타인', 평점은 9점?", <머니투데이>, 2014. 7.

17.(<https://news.mt.co.kr/mtview.php?no=2014071708364968042&outlink=1&ref=https%3A%2F%2Fsearch.naver.com>.2023. 6. 1.)

긴 문장의 개수, 부정적인 감정이 담긴 문장의 개수, 그리고 스토리에 대하여 긍정적인 감정이 담긴 문장의 개수, 부정적인 감정이 담긴 문장의 개수를 센다. 이렇게 얻은 각 문장의 개수를 통해 영화의 점수를 계산한다. 각 분야별로 $((\text{긍정적인 문장의 개수})/((\text{긍정적인 문장의 개수})+(\text{부정적인 문장의 개수}))) \times 10$ 을 계산하여 해당 영화의 연출 점수와 스토리 점수를 구한다. 하나의 블로그 글이 아닌 여러 글들을 분석하기 때문에, 각 블로그 글로부터 구한 점수의 평점을 계산함으로써 최종적이 점수를 산출한다.

사용자는 자신이 검색한 영화의 연출 점수와 스토리 점수를 각각 확인함으로써 해당 영화를 관람할지 여부를 결정할 수 있다.

III. 언어처리 도구

1. 크롤링

네이버 블로그의 글들을 추출하기 위해 네이버 API와 BeautifulSoup을 이용하였다. 네이버 API를 통해 블로그 글들의 제목, 링크, 날짜 등을 불러올 수 있지만 본문의 글을 크롤링할 수 없었다. 그래서 네이버 API로 블로그 글의 링크만 따로 추출하여 하나의 리스트에 담고, 그 리스트 안에 있는 각 링크들마다 BeautifulSoup을 이용하여 해당 링크에 담긴 블로그 본문 글을 가져와 다시 새로운 리스트 안에 담는 방식을 이용하기로 했다. 사용자로부터 영화 제목을 입력 받으면 앞선 과정을 통해 해당 영화 리뷰 글 데이터들이 하나의 리스트 안에 담기게 된다.

2. 전처리

크롤링한 상당수의 블로그 본문에는 불필요한 여백이 많이 존재하였다. 이러한 내용 분석에 도움이 되지 않는 여백을 지우기 위해 replace함수를 써서 개행 문자, 공백 문자 등을 제거하였다. 또한, 본문의 오타들을 보정해주기 위해 okt의 normalize()함수를 이용하여 본문을 정규화 하였다.

불필요한 여백을 제거하고 정규화 한 뒤 본격적인 전처리를 위하여 KoNLPy의 Kkma(꼬꼬마)와 Okt(Open Korea Text)를 이용하였다. 우선 본문의 감정을 문장 단위로 분석할 것이기 때문에 전체 본문을 문장으로 토큰화 할 필요가 있었다. 본문을 문장으로 토큰화 하기 위하여 Kkma의 sentences() 함수를 이용하였다.

Kkma의 sentences() 함수는 텍스트를 문장으로 토큰화 할 뿐만 아니라 문장의 띄어쓰기도 보정해준다. 대체로 사람들은 띄어쓰기를 하나하나 신경 쓰며 블로그 글을 작성하지 않기 때문에 sentences() 함수의 띄어쓰기 보정 기능은 유용하지만, 띄어쓰기가 잘못 적용되었을 가능성을 고려하여 띄어쓰기를 교정하는 함수를 따로 만들었다. 블로그는 사람들이 자신의 생각, 감정 등을 자유롭게 쓸 수 있는 곳이기에 대부분의 글은 구어체적 특징을 띤다. Okt는 KoNLPy의 다른 한국어 처리 도구들에 비해 구어체 분석에 유리할 것이라 생각되어 Okt를 이용하여 띄어쓰기 함수를 만들었다. 띄어쓰기 함수는 문장의 구성요소들을 Okt의 pos() 함수를 통해 tagging한 다음, 태깅된 요소의 품사가 조사, 어미, 선어말어미, 접미사, 그리고 구두점이면 앞 요소와 띄어 쓰고 그 외 나머더라면 앞 요소와 붙여진 문장을 새로 만드는 방식이다.

띄어쓰기를 교정한 문장에서 더욱 중요한 요소만 남기도록 정제하기 위하여 불용어를 제거하는 함수도 만들었다. 이는 실습 때 다루었던 함수로, 문장의 구성요소들을 다시 Okt의 pos()함수를 통해 tagging한 다음, 문장의 감정 분석에 중요한 명사, 동사, 그리고 형용사만 남겨진 새로운 텍스트

를 만든 다음, 실습 자료로 쓰였던 'stopword_ko.txt'를 통해 불용어 목록을 확인하여 새로 만든 텍스트의 불용어를 제거하는 방식이다.

위의 과정을 거쳐 정제된 텍스트 목록 중 유의미한 텍스트만 추출하는 과정을 거치도록 한다. 영화의 연출 부문 점수와 스토리 부문 점수를 계산하기 위해서 텍스트 목록 중 연출과 스토리에 관한 언급이 있는 텍스트를 찾아내어 따로 추출해야 한다. 이를 위하여 연출 관련 키워드들과 스토리 관련 키워드들을 리스트로 새로 만들었다. 연출 관련 키워드는 ["연출", "OST", "ost", "음악", "영상미", "효과음", "CG", "cg", "Cg", "편집", "트랙", "작화", "씬", "액션"]으로 리스트를 만들었고, 스토리 관련 키워드는 ["스토리", "이야기", "내용", "줄거리", "개연성", "결말", "서사", "대사"]로 리스트를 만들었다. 정제된 텍스트 목록 안의 글이 연출 관련 키워드 리스트에 담긴 단어가 존재하면 연출 관련 문장으로, 스토리 관련 키워드 리스트에 담긴 단어가 존재하면 스토리 관련 문장으로 판단하여 연출 리스트와 스토리 리스트에 분리하여 담도록 하였다. 이를 위하여 정제된 글들을 Okt의 morphs() 함수를 통해 단어로 토큰화 한 다음, 토큰화 된 단어 리스트 안에 위의 키워드들이 존재하는지 확인하는 방식으로 분리하였다. 다만, 하나의 글 안에 연출 관련 키워드와 스토리 관련 키워드 모두가 담겨 있을 경우 감정 분석에 혼동이 생길 수 있으므로 이러한 류의 글은 어떠한 리스트에도 담지 않도록 하였다.

3. 기계학습

영화의 점수를 계산하기 위해 영화 리뷰 글들의 감성을 분석할 것이므로 깃허브에 있는 Naver sentiment movie corpus(nsmc)²⁾의 "ratings_train.txt" 데이터를 이용하여 기계학습을 시키는 것이 적합하다고 생각했다. 리스트에 담긴 각 글에 긍정적인 감정이 담겨있는지 부정적인 감정이 담겨있는 지만 판단할 것이기 때문에 이진 분류 예측에 사용되는 기계학습 모델인 Logistic Regression 이 감성 분석에 적합한 모델이라고 생각하여 이를 이용하기로 결정하였다. 이때, Logistic Regression 에 문자 데이터는 들어갈 수 없기 때문에 네이버 영화 리뷰 데이터와 블로그에서 추출한 문장들을 벡터화 하였다. 문장을 토큰화 하였을 때 나오는 토큰의 등장 횟수로 벡터화 하는 것보다 빈도수로 벡터화 하는 것이 감성 분석에 더 적합하다고 판단하여 TfidfVectorizer()를 이용하여 텍스트 데이터들을 벡터화 하였다.

"ratings_train.txt" 데이터에 담긴 글들을 정규화하고 불용어를 제거하는 과정을 거친 다음, 이 정제된 글들과 해당 데이터에 담긴 label로 모델을 학습시켰다. 긍정적 감정이 담긴 글의 label은 1 이고, 부정적 감정이 담긴 글의 label은 0이다. 연출 관련 리스트와 스토리 관련 리스트에 각각 담긴 글들의 감성을 해당 모델을 통해 파악함으로써 각 블로그 글마다 담긴 긍정적 문장의 개수와 부정적 문장의 개수를 구한 다음, 앞서 서술한 방식으로 영화의 평점을 계산하였다.

IV. 실험 결과

신뢰도 있는 점수가 나오는지 확인하기 위해 앞서 언급한 영화 '클레멘타인'을 입력하였을

²⁾ Lucy Park, "Naver sentiment movie corpus v1.0", GitHub, 2016. 6.

28.(<https://github.com/e9t/nsmc>. 2023. 5. 26.)

때 나타나는 점수를 확인해 보았다.

☞ 클레멘타인의 연출 점수는 4.25입니다
클레멘타인의 스토리 점수는 3.79입니다

<클레멘타인 평점>

연출 점수는 10점 만점에 4.25점, 스토리 점수는 10점 만점에 3.79점으로 과거 네이버 영화에서 매겨졌던 9.38점보다 현저히 낮은 점수가 나타났다.

개인적으로 연출은 훌륭했지만 서사가 다소 아쉬웠다고 느꼈던 영화 '스즈메의 문단속'도 입력해보았다.

☞ 스즈메의 문단속의 연출 점수는 7.92입니다
스즈메의 문단속의 스토리 점수는 5.19입니다

<스즈메의 문단속 평점>

연출 점수는 10점 만점에 7.92점, 스토리 점수는 10점 만점에 5.19점이 나타났다. 8점에 가까운 연출 점수에 비해 스토리 점수가 상당히 낮게 나온 것을 알 수 있다.

2022년에 개봉한, 개인적으로 연출과 스토리 둘 다 나쁘지 않았다고 느꼈던 영화 '탐건: 매버릭'을 입력했을 때 나오는 결과는 다음과 같았다.

☞ 탐건: 매버릭의 연출 점수는 7.57입니다
탐건: 매버릭의 스토리 점수는 6.14입니다

<탐건: 매버릭 평점>

연출 점수는 7.57점, 스토리 점수는 6.14점으로 둘 다 점수가 낮은 편은 아니지만 스토리 점수가 연출 점수에 비해 낮게 나온 것을 알 수 있다.

V. 기대 효과

기존 대부분의 영화 평점 서비스는 관람객들이 영화에게 매긴 점수들의 평균을 통해 평점을 제공하는 방식을 이용하고 있다. 그러나 이러한 방식으로 제공되는 평점은 정확히 어떤 부분에서 이러한 점수가 나오는지 알 수 없으며 가벼운 마음으로, 혹은 장난식으로 부여되는 점수까지 평점에 반영하기 때문에 신뢰성 높은 점수를 제공하지 못하고 있다는 문제점이 있다.

그러나 이 프로그램은 장문으로 구성된 블로그의 후기 글들을 분석한다. 블로그에는 사람들이 자신이 느낀 바를 보다 솔직하게 작성한 글들이 많기 때문에 기존 서비스와 달리 장난성으로 점수가 부여된다는 문제가 발생할 가능성이 훨씬 적다는 장점이 있다. 이는 영화 '클레멘타인'을 입력할 때 나오는 점수가 네이버 영화에서의 평점보다 낮다는 점을 통해 확인이 가능하다.

또한, 이 프로그램은 블로그 글을 문장 단위로 분석하기 때문에 같은 글 내에서 부정적인 문장과 긍정적인 문장 모두 찾아냄으로써 기존 서비스의 극단적인 점수 분포 문제의 발생 가능성을 줄일 수 있을 것이다. 더 나아가 위 프로그램은 기존 서비스처럼 하나의 점수만 제공하는 것이 아닌 연출

부분에 대한 점수와 스토리 부분에 대한 점수를 구분하여 제공해준다. 이는 사용자에게 자신의 성향에 따라 영화를 선택할 수 있게 해준다는 장점이 있다. 예를 들어, 영화의 스토리보다는 영상미와 같은 연출을 중시하는 사람의 경우 스토리 부분 점수보다 연출 부분 점수를 중점적으로 확인함으로써 관람할 영화를 선택할 수 있고, 스토리를 더 중시하는 사람의 경우 스토리 부분 점수를 확인함으로써 관람할 영화를 선택할 수 있다. 즉, 사용자 본인이 영화를 볼 때 무엇을 중시하는가에 따라 본인의 성향에 맞는 영화를 알아볼 수 있다.

VI. 한계

그러나 위와 같은 방식으로 매긴 점수 역시 신뢰도가 높다고 확신하기는 힘들다. 먼저, 사용자가 인지도가 낮은 영화를 검색할 경우 충분한 양의 데이터를 얻기 힘들다. 소수의 후기 글들을 분석해서 도출한 점수는 신뢰할 만한 점수가 되지 못할 것이다.

일부 블로그 글 데이터 중에는 '블로그 포스팅 알바'를 통해 작성된 글이 존재할 가능성이 있다. 블로그 포스팅 알바는 블로거가 회사에서 주는 이미지와 원고를 받고 본인 블로그에 작성하는 것으로 진행된다.³⁾ 이러한 부류의 글은 기업이 원하는 방향으로 작성되기 때문에 작성자의 솔직한 감상이 담겨 있을 가능성이 적고, 좋은 데이터가 될 수 없다. 따라서 이 프로그램은 좋지 않은 데이터들도 분석함으로써 신뢰도가 다소 떨어지는 결과를 얻어낼 수 있다는 한계도 존재한다. 따라서 광고성 글과 그렇지 않은 글을 구분하여 분석할 수 있는 기능을 넣는 보완을 할 필요가 있다.

본문의 전처리가 완벽하지 않았다. 본문을 문장으로 토큰화 하기 위하여 Kkma의 sentences() 함수를 이용하였지만 이는 높은 정확도를 보이지 못했다. sentences() 함수가 한 문장이라고 인식한 글 안에 둘 이상이 문장이 담기기도 했었다. 본문을 더 정확하게 문장 단위로 분해할 수 있는 도구를 찾거나 만들 필요가 있다. 띄어쓰기도 제대로 교정이 되었다고 보기 어려웠다. 띄어쓰기를 교정하는 위하여 문장을 pos로 tagging하는 과정을 거쳐야 하는데, 이 tagging이 정확도가 높지 않았다. Okt의 pos() 함수가 문장의 구성요소들을 잘못 판단하여 작성자가 의도한 의미와 다른 의미로 해석하는 문제가 발생하였다. 띄어쓰기 교정 함수를 개선할 필요가 있다. 또한, 영화 제목에 고유명사가 담겨 있을 경우 전처리 도구가 고유명사를 잘 인식하지 못하여서 전처리 후 영화 제목이 깨지는 문제도 보완할 필요가 있다.

감성 분석 과정에도 한계가 존재한다. 이 프로그램은 한 문장당 하나의 감정이 담겨있음을 전제로 만들어졌기 때문에 하나의 문장 안에 연출과 스토리에 대한 언급이 있는데 두 요소에게 각기 다른 감정이 담겨 있을 경우, 그리고 한 문장에 연출이나 스토리 중 하나의 요소에 대해서 호평과 혹평이 모두 존재할 경우에는 제대로 된 감성분석을 할 수 없다. 때문에 각 특성마다 다른 감정이 담긴 문장들이 영화 평가에 기여할 수 있는 중요한 데이터로서 잘 사용되지 못한다는 문제가 발생한다. 따라서 한 문장 안에 둘 이상의 요소가 언급되어 있으며 여러 요소들마다 다른 평가가 내려졌을 경우, 각 요소에 대한 감정을 하나씩 분석할 수 있도록 보완을 할 필요가 있다.

같은 감정일지라도 그 감정의 정도가 다를 수 있다. 어떠한 영화의 개연성에 대해서 똑같이

³⁾ 신희범, "블로그 포스팅 알바, 비대면 재택부업으로 수요 증가", <JOBPOST News>, 2020. 10.

30. (<https://www.job-post.co.kr/news/articleView.html?idxno=9617>, 2023. 6. 11.)

부정적인 감정을 가질 수 있지만 누군가는 혐오와 같은 강한 부정의 감정을, 또 다른 누군가는 절고
러움과 같은 약한 부정의 감정을 느끼며 이를 글에 표현할 수 있다. 그러나 이 프로그램은 문장의 긍
부정만 구별해주지, 세부적인 감정을 인식하지 못한다. 단순히 긍정과 부정만을 판별하는 것이 아닌
강한 긍정, 약한 긍정, 약한 부정, 강한 부정 등으로 감정을 세세하게 구별할 수 있도록 보완한다면 더
욱 정확한 점수를 얻어낼 수 있을 것이다.

참고 문헌

신희범, "블로그 포스팅 알바, 비대면 재택부업으로 수요 증가", <JOBPOST News>, 2020. 10. 30.(<https://www.job-post.co.kr/news/articleView.html?idxno=9617>. 2023. 6. 11.)

최경민, "이동준 50억 날린 '클레멘타인', 평점은 9점?", <머니투데이>, 2014. 7. 17.(<https://news.mt.co.kr/mtview.php?no=2014071708364968042&outlink=1&ref=https%3A%2F%2Fsearch.naver.com>.2023. 6. 1.)

Lucy Park, "Naver sentiment movie corpus v1.0", GitHub, 2016. 6. 28.(<https://github.com/e9t/nsmc>. 2023. 5. 26.)