|  |
| --- |
| **태그 데이터를 활용한**  **Word2Vec 기반 영화 추천 기법** |
| 고현웅1, 김세민1, 홍준기1, 노지우2, 김순태2  전북대학교 소프트웨어공학과  [e-mail: {gusdnd852, tpals324, [rlwns012](mailto:rlwns012), wldn306, [stkim}@jbnu.ac.kr](mailto:stkim%7d@jbnu.ac.kr)] |
| Word2Vec based movie recommendation techniqueusing tag data Hyunwoong Go1, Semin Kim1, Joongi Hong1, Jiwoo Noh2, Suntae Kim2  1Dept. of Software Engineering, Chonbuk National University |
|  |
| **요 약**  기존의 많은 영화 콘텐츠 추천 시스템이 사용자 기반의 협업 필터링 방식을 이용하여 사용자들에게 영화 콘텐츠를 추천하고 있다. 하지만 이러한 기법은 사용자가 내린 평가(평점) 정보를 이용해 유사한 사용자를 찾고, 오직 그들의 소비 패턴만을 기반해 콘텐츠를 추천하기 때문에 개별 콘텐츠가 가진 특성이나 잠재성 등을 반영할 수 없다는 문제점이 있다. 본 논문에서는 이러한 문제를 해결하기 위해 Word2Vec을 이용하여 영화에 달린 태그들 간의 유사성을 찾아내고, 이를 기반으로 개별 아이템들의 특성과 잠재성을 반영하여 사용자에게 영화를 추천하는 방식을 제안한다. | | |

**1. 서 론**

추천시스템은 정보 필터링 기술의 일종으로 특정 사용자가 관심을 가질 만한 정보(영화, 음악, 책, 이미지 등)을 추천하는 것을 말한다. 과거에 추천시스템은 전자 상거래나 뉴스 등의 특정 분야에서만 사용되던 기술이었지만, 오늘 날에는 모바일 서비스의 정착과 빅데이터 기술 등의 성장들과 함께 이제는 다양한 콘텐츠 기반 서비스에서 빼놓을 수 없는 요소가 되었다. 사용자의 취향을 더욱 적합하게 반영하여 흥미 있을 만한 것을 추천하는 것은 사용자로 하여금 해당 도메인에 더 오래 머무르게 할 수 있고 이것은 더 많은 소비로 이어질 수 있다. 추천시스템을 적극적으로 활용하여 21세기 이후 가장 큰 성공을 거둔 기업들인 Amazon과 Netflix등이 이를 반증한다.

추천 시스템, 특히 영화 추천 시스템에서 가장 선호되는 기법인 사용자 기반 협업 필터링 (User

based Collaborative Filtering)은 사용자의 평가정보를 기반으로 유사한 평가를 내린 사용자를 찾고, 그들의 정보를 바탕으로 특정 콘텐츠를 추천하는 방식이다. 그러나 이러한 방식은 오직 유사사용자의 소비패턴만을 이용하여 콘텐츠를 추천하기 때문에, 각각 개별 콘텐츠들이 가진 특성이나 잠재성들을 반영하지 못하고, 사용자가 내린 결정의 의미나, 갑작스러운 취향의 변화 등을 파악하기 매우 어렵다. 본 논문에서는 이러한 문제를 해결하기 위해 Word2Vec을 이용하여 다른 사용자들이 남겨놓은 영화 별 태그데이터를 활용해 태그들 간의 유사성을 찾아내고, 이를 기반으로 개별 콘텐츠가 가진 특성과 잠재성을 반영하여 사용자에게 영화를 추천하는 방식을 제안한다.

이후 순서로 논문을 이해하는데 필요한 배경지식과 관련 연구들을 이야기하고 그 이후에 제안 방법에 대한 자세한 설명과, 실험 결과에 대해 차례대로 이야기할 것이다.

**2. 배경지식**

본 논문을 이해하기 위해서는 협업 필터링과 Word2Vec에 대한 이해를 필요로 한다.

**2.1 협업 필터링(Collaborative Filtering)**

본 논문에서는 영화 콘텐츠 추천을 위해 협업 필터링 [1] 방식을 사용하는데, 그 방식 중 아이템 기반 협업 필터링 기법을 적용하여, 사용자가 본 영화와 유사한 영화를 추천한다.

**2.2 Word2Vec**

태그데이터들 간의 유사도를 찾기 위해 Word2Vec[2] 모델을 사용한다. 태그들을 순서대로 나열한 뒤 벡터화 하여 그들 간의 유사도를 측정한다.

**2.3 관련 연구**

본 연구는 [3]을 통해 영화 추천의 모델로 Word2Vec을 사용하는 방법에 대한 영감을 받았다. [3]에서 제안하는 방식은 유저들 간의 유사도를 Word2Vec을 통해서 구해낸다. 유저가 비슷한 시기에 점수를 달았으면 비슷한 성향의 유저라는 가정하에 유사도를 구해냈으며 유사도를 가중치로 활용하여 영화에 대한 평점을 예측하는 방법을 제안한다. 하지만 비슷한 시기에 점수를 달았다고 하여 비슷한 성향이라고 생각하는 것은 무리가 있는 가정이라고 판단하였다.

**3. 제안방법**

<그림 1>은 제안하는 방법을 한 눈에 알아볼 수 있게 나타낸 그림이다. 제안하는 방법은 크게 두가지로 나뉜다. 먼저 태그데이터를 시간의 순서로 나열한 뒤, Word2Vec 모델을 이용해 벡터화 시켜, 태그들 간의 유사도를 측정한다.

그리고, 사용자가 시청한 영화를 기반으로 추천 영화 후보를 추려내고, 그 영화들과 사용자가 본 영화의 유사도를 측정한 뒤, 유사도를 평균평점과 곱해 추천지수를 측정한다. 결과적으로 추천지수가 높은 순서대로 영화를 사용자에게 추천한다. 아래에서 이와 같은 추천기법에 대해 단계적으로 구체적인 방법을 설명한다.

**3.1.1 데이터 로드**

본 논문에서는 영화의 이름, 사용자들의 평점, 사용자들에 영화에 달아 놓은 태그 등의 데이터를 활용한다.

**3.1.2 태그 나열**

태그들 간의 유사도를 구하기 위해, Word2Vec 모델을 이용하는데, Word2Vec은 단어들 간의 순서를 기반으로 단어 간의 유사성을 파악한다. 여기에서는 유저 – 영화 별 태그들을 시간의 순서로 나열해 마치 문장과 같이 만든다.

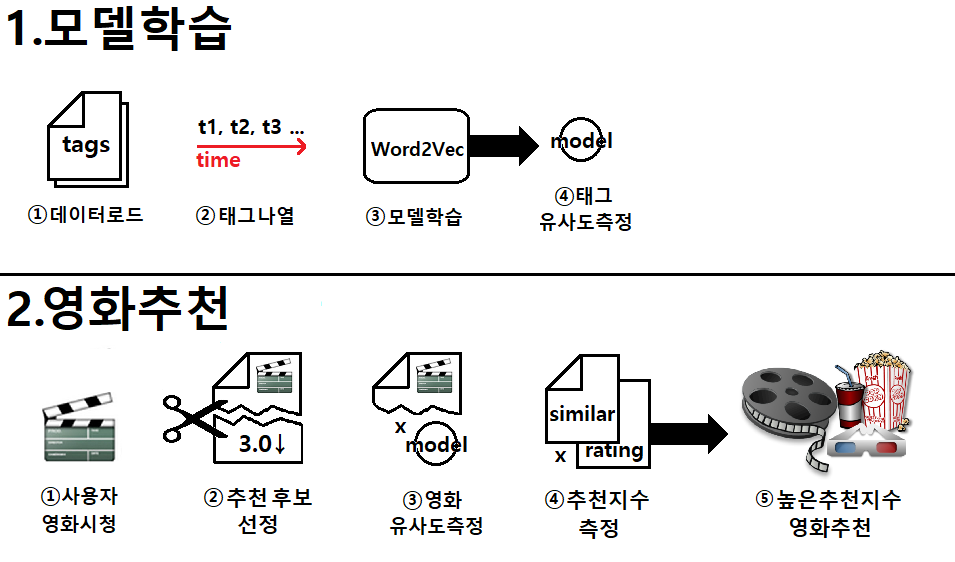
****

그림 1 제안하는 시스템 개요

표 1 MovieLens tags 원본 데이터

|  |  |  |  |
| --- | --- | --- | --- |
| **UserId** | **MovieId** | **tag** | **Timestamp** |
| **27** | 260 | classic sci-fi | 1440448113 |
| **27** | 260 | Harrison Ford | 1440448123 |
| **27** | 260 | must see | 1440448132 |
| **27** | 260 | sci-fi | 1440448094 |
| **42** | 37733 | disappointing | 1264106059 |
| **42** | 37733 | overrated | 1264106052 |
| **42** | 37733 | stupid | 1264106067 |
| **56** | 141 | cliche | 1381497460 |
| **56** | 2028 | disturbing | 1274837927 |
| **56** | 2028 | history | 1274837934 |

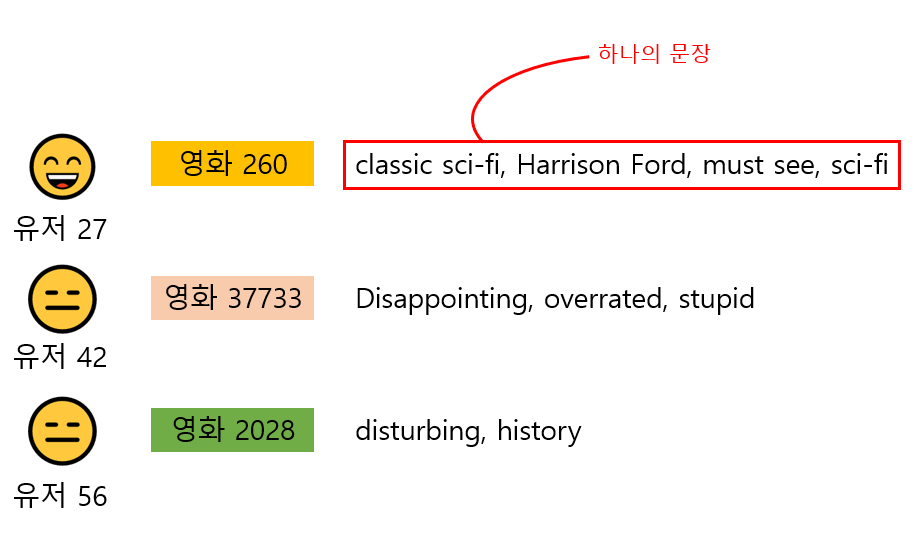


그림 1 유저-영화별 Tag 문장

원본 데이터는 <표1>과 같은 형식을 따르며 이와 같은 데이터를 <그림2>와 같이 한 유저가 한 영화에 작성한 태그들이 마치 하나의 문장과 같이 만들어지고 이는 태그들 간의 유사성을 측정하기 위한 모델을 학습시키기 위해 사용된다.

Word2Vec 모델학습을 위한 문장에서는 단어들 사이의 순서가 굉장히 중요하기 때문에 더욱 정확한 모델을 만들기 위해 태그 문장을 구성할 때 비슷한 시간에 작성한 태그는 비슷한 태그일 것이라고 가정하고 태그들을 시간의 순서로 정렬하였는데, 그 이유는 보통 사람들이 여러 개의 태그를 작성할 때, 머리속에 연상되는 순서대로 작성하는 경향이 있다고 판단했기 때문이다. 다음 예시를 보자.

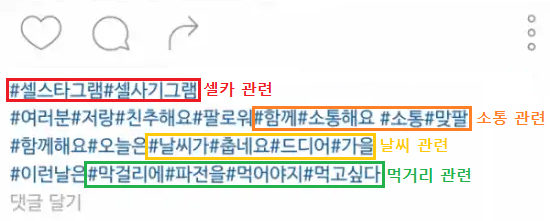


그림 2 인스타그램 해쉬태그

위 예시는 태그를 활용하는 대표적인 서비스인 인스타그램의 예시이다. 태그데이터를 분석하던 중 사람들이 태그를 작성할 때는 연상되는 단어를 시간의 순서대로 나열하는 경향이 있다는 것을 발견하였다. (셀카에 관련된 태그들이 시간 상으로 가장 먼저 작성되었고, 먹거리에 관련된 태그들은 가장 늦게 작성되었다.)

인스타그램 뿐만 아니라 태그와 관련된 다른 데이터들도 이 와 같은 경향이 있었으며, 본 논문에서 사용한 데이터셋의 태그데이터 역시 비슷한 성향이 있다는 것을 발견했다.



그림 3 한 유저의 시간 순으로 정렬된 tag

<그림4>는 한 유저가 99114번 영화에 달아 놓은 태그를 시간순으로 정렬한 것이다. 표의 왼쪽부터, 영화 아이디와 작성된 태그, 그리고 태그를 작성시간(UTC)이 나타나 있다. 위에서 말한 것처럼 영화 태그 역시 비슷한 시간대에 작성된 태그 일수록 더욱 유사한 태그라는 성질이 적용되었음을 알 수 있고, 이와 같은 성질을 활용하여 태그들을 문장으로 만들 때 시간순으로 정렬했다.

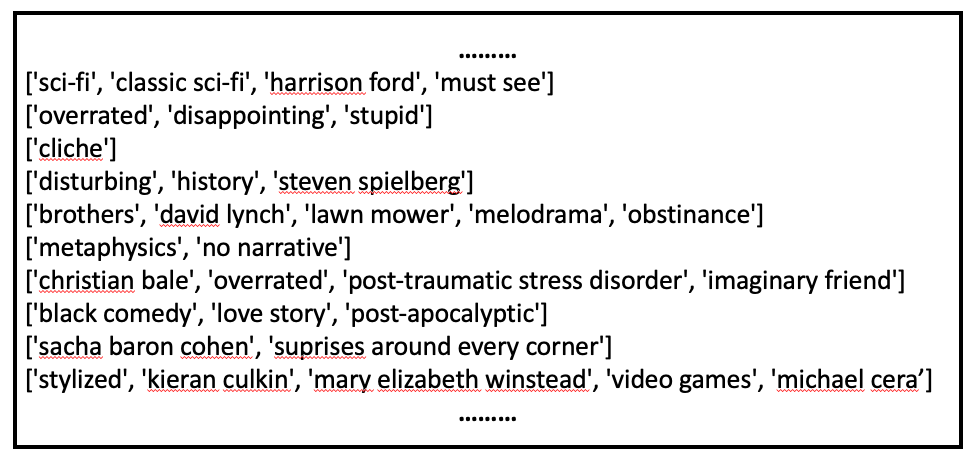


그림 4 유저-영화 별 태그 문장

위와 같은 과정을 거쳐 유저-영화 별 태그 문장을 완성한다. <그림5>는 완성된 태그문장들의 일부의 모습을 보여준다.

**3.1.3 모델 학습**

이렇게 시간의 순서로 나열한 유저 – 영화 별 태그 문장을 Word2Vec을 이용해 학습시킨다.

**3.1.4 태그 유사도 측정**

표 2 Word2Vec을 통해 구해낸 tag ‘thor’와 유사한 tag

Word2Vec 모델이 잘 학습되었는지 확인하기 위하여 태그의 유사도를 측정해본다. 여기에서는 마블 시네마틱 유니버스의 대표영화인 토르(thor) 태그를 사용해 실험하였고, <표2>는 그 결과를 나타낸다.

|  |  |  |
| --- | --- | --- |
|  | **tag** | **similarity** |
| **0** | Iron man | 0.648044 |
| **1** | mcu | 0.589653 |
| **2** | captain america | 0.570265 |
| **3** | the avangers | 0.552410 |
| **4** | marvel cinematic universe | 0.543884 |
| **5** | chris hemsworth | 0.488569 |
| **6** | marvel | 0.486233 |
| **7** | tom hiddlestone | 0.437214 |
| **8** | superhero | 0.430817 |
| **9** | superheroes | 0.404880 |

‘thor’ 태그와 유사한 태그의 목록으로 같은 마블 시네마틱 유니버스 소속의 iron man, captain america, avangers 등의 영화들과, 토르에서 주인공 토르 역을 맡은 chris hemsworth와 토르의 동생(로키)역을 맡은 tom hiddlestone 등 배우들이 유사한 태그로 선정되었다. <표2>를 통해 제시한 방법대로 만들어낸 태그문장을 이용해 모델이 적절하게 학습되었고, 모델을 통해 태그들 간의 유사성을 확인할 수 있다는 것을 알 수 있다.

**3.2.1 사용자 영화 시청**

모델 학습 섹션을 끝내고, 사용자에게 영화를 추천해주는, 영화추천 섹션의 첫번째 단계로, 사용자가 어떠한 영화를 봤다고 가정한다.

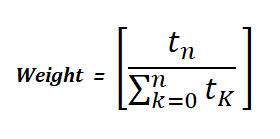
**3.2.2 추천후보 선정**

사용자 영화 선정 이후에 모든 영화리스트 중에서 평균평점이 5.0점 만점에 3.0점 아래인 영화는 사용자에게 추천해줄만 한 영화가 아니라고 판단하고 추천목록에서 제외한다.

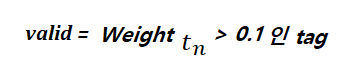
평균평점 3.0점 이상인 영화 중에 사용자에게 추천해줄만 한 영화 후보군을 선정하기 위해 다음과 같은 과정을 거친다.

*1. 사용자가 본 영화에서 태그를 빈도순으로 뽑아낸다.*

*2. 태그의 가중치를 계산한다. 여기에서 가중치란 해당 영화의 모든 태그 개수 중 해당 태그가 달린 횟수로 나눈 값을 반올림한 것을 의미한다. 의미한다.*

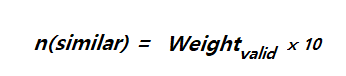
**

*3. 가중치가 0.1이상, 즉 전체에서 단일태그가 5% 이상(반올림하여 0.1로 계산됨)의 가중치를 가지고 있으면 해당 태그를 “유효 태그” 라고 판단한다.*

**

*4. 해당 영화의 모든 태그의 가중치 중에 가장 낮은 가중치를 가지는 태그를 제외한 모든 태그의 가중치 x 10한만큼 모델에서 비슷한 태그를 선출하고 이 태그들을 “유사 태그” 라고 한다.*

*예를 들어 태그 a{w=0.2}, b{w=0.1}, c, d, e{w=0}이 있다면, 태그a의 유사태그 2개와 태그b의 유사 태그 1개를 모델에서 선출한다. (w는 가중치, 아래 수식의 n은 개수를 의미한다.)*

**

*5. 추천영화 후보에는 모든 유효태그 + 유사태그 의 집합을 활용한다. 이를 “추천후보태그” 라고 한다.*

**

*6. 그러나 가장 낮은 가중치를 가지는 태그는 유사태그를 뽑지 않기 때문에, 만약 태그의 개수가 딱 1개이거나 2개등, 적은 태그를 가진 영화들은 유사태그 선별을 위한 최소가중치가 1(95%), 0.5(45%)등으로 너무 높아지기 때문에 그 만큼 유사태그를 뽑을 수 없고 활용할 태그집합의 원소 개수가 너무 적어지게 된다. 때문에, 태그 개수를 조정한다.*

*6-1. 5번까지 과정을 거친 후 추천후보 태그 집합의 원소의 개수가 1개인경우 해당 태그와 유사한 9개의 태그를 더 선별하여 총 10개의 태그를 추천 후보태그 집합에 넣는다.*

*6-2. 5번까지 과정을 거치고 추천후보 태그 집합의 원소 개수가 1개이상 10개 미만인 경우, 태그들을 순회하며 집합이 10개의 태그를 가질 때까지 유사 태그를 선별한다.*

*만약 a , b 두개의 태그가 있다면 a유사1, b유사1, a유사2, b유사2, a유사3, b유사3 … (이 경우는 태그들의 가중치가 비슷하다는 것이기 때문에 순회를 하며 추천후보태그 집합의 모든 태그들과 유사한 태그를 고르게 뽑는다.)*

*6-3. 태그의 개수가 너무 많은 경우도 있다. 태그의 개수가 너무 많으면 너무 많은 영화들이 추천 후보 영화가 되기 때문에 의미 없는 영화들까지 후보가 될 수도 있다. 6-2까지의 과정을 거친 이후 태그집합의 원소 개수가 15개 이상인 경우, 추천후보태그 집합을 모두 비운 뒤, 다시 선출한다. 유효태그와 유사태그를 뽑는데, 유사태그를 선출할 때 기준을 더욱 엄격하게 한다. 기존의 최소 가중치보다 0.1올려서 다시 유사태그를 뽑는다. (기존에 0.1이 최소 가중치였다면 이 경우에는 0.2이상의 가중치를 가진 태그만 유사 태그를 선출할 수 있다.)*

*6-4. 위 과정을 거쳤는데도 불구하고 15개가 넘는 추천 후보태그를 가지고 있다면 대부분 태그가 비슷한 가중치를 가진다는 의미이므로, 무작위로 일부 태그를 제거한다. 추천 후보태그 집합의 원소가15개가 될 때까지 태그를 제거한다.*

조금 복잡하지만, 위와 같은 과정을 거쳐 추천후보 태그 집합을 만들고, 평균 평점이 3.0이상인 영화 중에서 추천후보 태그집합의 태그를 하나라도 가진 영화를 “추천후보 영화”로 선정한다.

**3.2.3 영화 유사도 계산**

이제 사용자가 본영화와 추천후보영화들 각각 사이의 유사도를 측정한다. 두 영화는 각각의 “유효태그”의 집합을 가지고 있고, 이 유효태그들은 높은 가중치를 가진 태그들이기 때문에 해당 영화를 대표하는 단어(태그)라고 할 수 있다. 즉, 두 영화의 유효태그가 서로 유사하다면, 두 영화는 유사한 영화일 확률이 높다. 간의 각각 유사도합을 추천후보영화의 유효태그 개수로 나눈 값을 “영화 유사도”라고 한다.

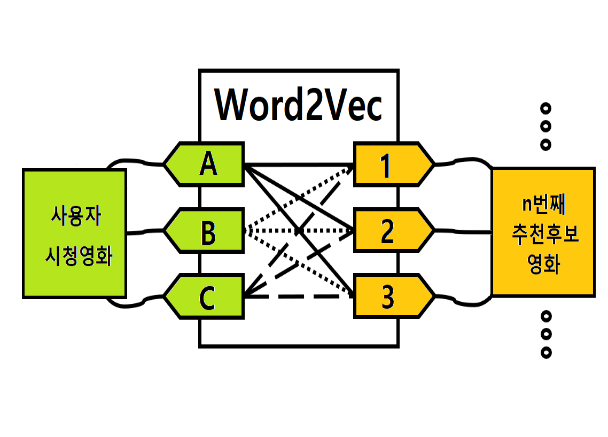


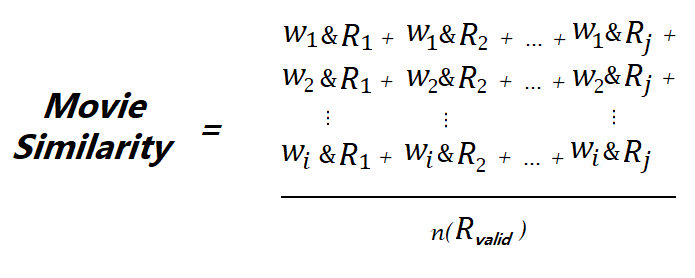
그림 5 사용자가 시청한 영화와 추천 후보 영화의 tag 유사도 도출

<그림6> 과 같이 사용자가 시청한 영화의 태그 A, B, C와 추천후보영화들의 태그들의 유사도를 구한다. 두 태그사이의 유사도 계산은 미리 만들어 둔 Word2Vec 모델을 통해 계산한다. 유사도 측정연산의 연산자를 &라고 표현한다면, A&1, A&2, A&3, B&1, B&2, B&3, C&1, C&2, C&3을 모두 구한다.

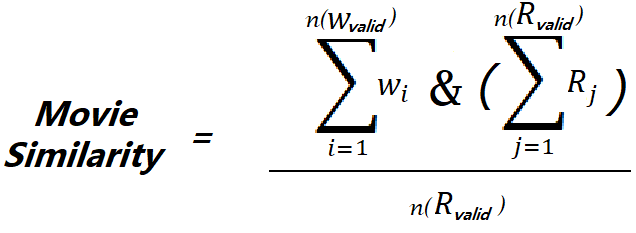
그리고 모든 유사도 연산 결과를 더한 뒤, 마지막으로, 그 것을 추천후보영화의 유효태그의 개수로 나눈다. 어떤 후보영화는 적은 수의 유효태그를 갖고 있는가 하면, 어떤 후보영화는 많은 수의 유효태그를 갖고 있을 수 있기 때문에 유효태그의 개수로 나눠주지 않으면 많은 태그를 갖는 영화가 더욱 높은 수치를 띌 확률이 높다.

이렇게 계산된 값을 “영화 유사도” 라고 한다. 사용자 시청 영화 와 n번째 추천영화 의 영화 유사도를 수식으로 나타내면 다음과 같다.

(는 영화의 번째 태그를 의미한다. 는 영화의 유효태그 개수를 의미하고, 는 유사도 연산의 연산자를 의미한다.)



이를 더 깔끔하게 표현하면 다음과 같다.



위 방법을 통해 사용자 시청영화와 각각 개별 추천후보영화가 얼마나 유사한지 측정할 수 있게 되었고, 개별 영화의 특성과 잠재성을 파악하는데 큰 지표로 활용될 것이다.

**3.2.3 추천 지수 계산**

추천후보영화의 유사도 만으로 사용자에게 영화를 추천해주면 좋은 영화를 추천해주지 못할 가능성이 있다. 가령, 사용자가 boring(지루한) 이라는 태그가 달린 영화를 시청했다면, boring태그와 유사한 태그가 달린 각종 지루한 영화들만 사용자에게 추천해주게 될 것이다.

영화들의 유사도는 영화 추천에 있어서 매우 유의미한 수치이지만, 유사한 영화를 추천해준다고 해서 사용자의 만족도를 보장할 수 없기 때문에, 이 유사도 값에 해당 영화의 평균 평점을 곱한다. 이 것을 “추천 지수” 라고 하고 추천지수를 기반으로 영화를 추천해주게 된다. 즉, 사용자가 본 영화 유사하면서, 다른 사용자들이 높은 평가를 내린 영화를 추천해주는 것이다.



**3.2.4 추천 지수 계산**

위와 같이 유사도와 평균평점을 곱해 추천 지수를 계산해냈고, 미리 만들어 놓은 추천후보 영화들의 목록에서 높은 추천지수를 가지는 영화를 순서대로 추천하게 된다.

**4. 실험 및 결과**

**4.1 실험 환경**

본 논문에서는 무비렌즈 (Movie Lens)에서 제공하는 ml-latest데이터 셋을 사용하였으며, 27,000,000개의 평점 데이터와 1,100,000개의 태그, 58,000개의 영화, 280,000명의 유저가 담겨 있다. 이 데이터가 마지막으로 최신화가 이루어진 것은 2018년 9월이다.

Word2Vec 모델 학습에는 Gensim의 라이브러리를 사용하였으며 skip-gram 방식으로 Window size는 3, 차원은 256차원으로 100번 반복시켜 학습시켰으며, 모델의 정확도와 학습 속도의 향상을 위해 100번 아래로 나온 태그는 제거하고 진행하였다.

**4.2 실험 결과**

실험 결과는 영화 “Batman Returns (1992)”를 모델에 넣었을 때 추천되는 영화들 중 10개의 영화를 표로 만든 것이다. 영화의 정보는 [표 3]과 같다.

표 3 Batman Returns (1992) 영화 정보

|  |  |  |  |
| --- | --- | --- | --- |
| **movieId** | **Title** | **Genres** | **Tags** |
| **1377** | **Batman Returns (1992)** | **Action**  **Crime** | **['batman', 'tim burton', 'superhero', 'michelle pfeiffer', 'dark hero']** |

추천 영화의 결과를 보면 영화의 제목과 장르, 영화의 대표 태그를 확인할 수 있으며, 유사도와 영화의 유저들의 평균 평점을 확인할 수 있다. [표 4]의 상위에 있을 수록 추천하는 영화이며, RecommendFactor가 가장 큰 것을 확인할 수 있다.

결과를 살펴봤을 때 태그 간의 유사도가 높을수록 높은 RecommendFactor 값을 갖는 것을 확인할 수 있으며, 평균 평점이 높은 영화일수록 추천될 확률이 높아지는 것을 확인할 수 있다.

출력된 영화 정보를 봤을 때 같은 Batman 시리즈인 *Batman vs. Robin (2015), Dark Knight, The (2008)* 등이 상위에 추천되는 것을 볼 수 있으며 장르 또한 비슷한 장르가 추천되는 것을 확인할 수 있다.

Word2Vec 모델을 기반으로 하여 태그 간의 유사도를 활용한 영화 추천은 비슷한 장르의 영화를 추천해주는 것을 확인해 볼 수 있었고, 영화의 평균 평점이 높으면서 유사도가 높은 것을 상위에 포진하게 되므로 유의미한 추천이라고 판단하고 있다.

표 4 태그 기반 영화 추천 결과

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **movieId** | **Recommend Factor** | **Title** | **Genres** | **Tags** | **Similarity** | **Average rating** |
| **131739** | **5.315181** | **Batman vs. Robin (2015)** | **Action, Adventure**  **Animation** | **[superhero, batman, robin, secret society, dc,...** | **1.586293** | **3.350694** |
| **58559** | **4.871058** | **Dark Knight, The (2008)** | **Action, Crime**  **Drama, IMAX** | **[heath ledger, batman, superhero, christian ba...** | **1.67068** | **4.173756** |
| **165639** | **4.666062** | **While You Were Fighting: A Thor Mockumentary** | **Comedy**  **Fantasy**  **Sci-Fi** | **[funny, short film, mockumentary, marvel, taik...** | **1.253077** | **3.723684** |
| **110566** | **4.554827** | **Son of Batman (2014)** | **Action, Adventure**  **Animation, Crime**  **Fantasy** | **[based on comic, superhero, super power, robin...** | **1.358728** | **3.352273** |
| **80469** | **4.391897** | **Superman/Batman: Apocalypse (2010)** | **Animation** | **[superhero, dc comics, super power, batman, su...** | **1.289705** | **3.405350** |
| **95004** | **4.355127** | **Superman/Doomsday (2007)** | **Action**  **Animation** | **[brandon vietti, bruce timm, lauren montgomery...** | **1.363170** | **3.194853** |
| **183915** | **4.214865** | **Batman: Gotham by Gaslight (2018)** | **Action, Animation**  **Sci-Fi, Thriller** | **[batman, dc comics, jack the ripper, 19th cent...** | **1.285891** | **3.277778** |
| **8636** | **4.112479** | **Spider-Man 2 (2004)** | **Action, Adventure**  **Sci-Fi, IMAX** | **[superhero, comic book, marvel, super-hero, ac...** | **1.109831** | **3.453454** |
| **167762** | **4.015859** | **Batman Beyond Darwyn Cooke's Batman 75th Anniv...** | **Action**  **Animation**  **Sci-Fi** | **[superhero, short, robot, future, based on tv ...** | **1.252257** | **3.206897** |
| **104419** | **3.912456** | **Justice League: Crisis on Two Earths (2010)** | **Action**  **Animation**  **Sci-Fi** | **[lauren montgomery, based on comic, sam liu, d...** | **1.079646** | **3.623832** |

**5. 결론**

추천 시스템은 현재 영화, 뉴스, 상품 판매 등 많은 곳에서 빼놓을 수 없는 기술로 활용되고 있다. 본 연구는 추천 시스템의 더 나은 성능을 위해서 태그 데이터와 Word2Vec를 활용하여 영화 추천 시스템을 제안하였다. 실제 추천되는 영화를 봤을 때 영화의 장르가 비슷한 장르를 추천하는 것을 확인해 볼 수 있었고 유의미한 결과를 도출해냈다.

향후 연구로는 더 나은 영화 추천을 위하여 태그 데이터의 불용어 제거 처리를 진행할 것이며, 현재는 TF를 활용하여 영화의 대표 태그로 사용하였는데 이를 TF-IDF로 바꾸어 성능을 비교할 것이다. 또한 Word2Vec의 하이퍼 파라미터의 최적값을 구하여 모델 자체의 성능을 끌어 올리는 것도 중요한 부분 중 하나이다.

**6. 참고 문헌**

[1] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th …. <https://doi.org/10.1145/371920.372071>

[2] Goldberg, Y., Levy, O., Rong, X., Perozzi, B., Al-Rfou, R., Skiena, S., … Dean, J. (2014). word2vec Parameter Learning Explained Continuous Bag-of-Word Model. ICLR. <https://doi.org/10.1162/jmlr.2003.3.4-5.951>

[3] 전근식, 공성언, 최용석 (2017). 영화 평점 예측을 위한 Word2Vec 기반 협업 필터링. 한국정보과학회 학술발표논문집 , 844-846.