

비디오 벡터를 이용한 내용 기반 유튜브 영상 추천 시스템*

이현규⁰ 윤영빈 윤준현 이태현 박광훈

경희대학교 컴퓨터공학과

esot3ria@gmail.com nogadahalf12@khu.ac.kr haring157@khu.ac.kr csws79@khu.ac.kr

ghpark@khu.ac.kr

Context-based Youtube Video Recommendation System Using Video Vector

Hyungyu Lee^o Yeongbeen Yun Junhyeon Yoon Taehyun Lee Gwang Hoon Park

Department of Computer Science and Engineering, Kyung Hee University

요약

비디오 벡터를 이용한 내용 기반 유튜브 영상 추천 시스템(Context-based Youtube Video Recommendation System Using Video Vector)은 기존 유튜브의 영상 추천 알고리즘을 개선하기 위한 시스템이다. 기존 알고리즘은 동영상 투고자가 직접 남긴 태그와 이용자의 과거 시청 데이터에만 의거하여 영상을 추천한다는 한계점이 존재한다. 이에 본 논문에서는 영상의 내용을 대표하는 비디오 벡터를 통해 내용 상으로 연관성 높은 영상을 추천하는 시스템을 제안한다. 해당 시스템은 동영상 세그먼트 분석 모델을 사용하여 동영상에 태그를 할당하고, 태그를 기반으로 비디오 벡터를 생성한 후, 각 비디오 벡터 간의 유사도를 계산하여 관련 영상을 추천한다. 본 시스템을 사용하면 기존 유튜브 알고리즘보다 내용 상으로 관련이 깊은 동영상 목록을 추천할 수 있으리라 기대한다.

1. 서론

유튜브(Youtube)는 오늘날 인터넷 이용자들이 가장 애용하는 동영상 투고 및 시청 사이트이다. 이 유튜브는 동영상의 조회 수와 잔류 시간이 많을수록 더 높은 광고 수익을 받는 시스템이다.

유튜브는 이용자 접속률을 높이기 위해 효율적인 영상 추천을 해야 할 필요가 있음에도 영상 내용 기반의 추천 알고리즘이 적용되지 않은 실정이다. 이에 대하여 이 논문은 업로드 과정에서 이후에 제안할 모델을 통해 동영상 태깅을 진행할 것이다. 그리고 워드 벡터(word vector) 기반의 자연어 처리로 비디오 벡터를 만들어, 이용자가 가장 흥미를 가질만한 동영상을 추천해주는 유튜브 영상 추천 시스템을 개발하고자 한다.

2. 기존 추천 알고리즘

한국언론진흥재단의 논문 ‘유튜브 추천 알고리즘과 저널리즘’ [1]에 따르면 현재의 기술로는 영상의 내용을 파악하기 어렵기 때문에, 유튜브의 추천 알고리즘에서 영상의 내용은 반영되지 않는다고 한다. 현재의 유튜브의 추천 영상 목록을 만드는 기준을 보면, 이용자가 과거에 시청했던 영상, 좋아요 버튼을 클릭했던 영상을 기반으로 관련 영상을 추천하고 있다. 이러한 방식은 영상 자체의 내용을 분석에 활용하지 않기 때문에 이용자의 성향에 맞지 않는 영상이 추천되기도 한다.

이에 우리는 동영상 내용을 반영하여 영상을 추천하는 시스템을 만들고자 한다.

3. 내용 기반 추천 시스템

본 절에서는 내용 기반 유튜브 영상 추천 시스템의 알고리즘을 설명한다. 우선 시스템은 목표 영상 내용을 분석하여 영상과 가장 관련도 높은 태그들을 추출한다. 각 태그는 고정 차원에 임베딩된 워드 벡터 형태로 표현할 수 있으며 시스템은 이를 태그 벡터(tag vector)로 사용한다. 목표 영상에서 추출된 태그 벡터들을 적절히 합치면 영상을 대표하는 비디오 벡터(video vector)를 생성할 수 있다. 마지막으로 목표 영상과 기존 영상들의 비디오 벡터를 비교하여 가장 유사도가 높은 영상들을 선별하면 내용 기반의 영상 추천이 완료된다.

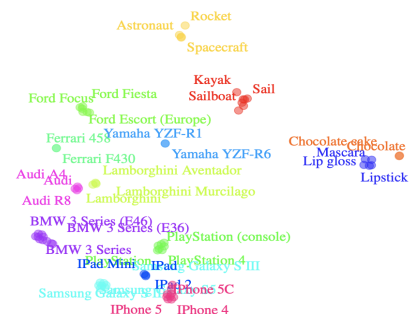


그림 1. 임베딩된 워드 벡터의 군집화[2]

본 시스템의 구현에는 영상 태그 추출, 태그의 벡터 표현, 태그 벡터의 비디오 벡터 표현이라는 세 가지 알고리즘이 필요하다. 우선 ‘Youtube-8M Video Understanding Challenge’ [3]에서 사용된 딥러닝 모델을 이

* 본 연구는 과학기술정보통신부 및 정보통신기획평가원
의 SW중심대학 사업의 연구결과로 수행되었음.

용하면 영상을 분석하여 그에 맞는 태그를 붙일 수 있다. 각 태그에 대한 벡터는 word2Vec[4] 등을 사용하여 임베딩할 수 있는데, 위키피디아 등의 단어 설명 텍스트를 모아 말뭉치(corpus)로 사용할 때 군집화가 된다는 것이 잘 알려져 있다. 그림 1에서는 의미적으로 연관이 있는 워드 벡터들이 서로 가까운 공간에 임베딩되는 현상을 보여 준다.

영상에서 태그를 추천함과 동시에 각 태그의 유사도를 추출할 수 있으므로, 유사도 기준 상위 k개의 태그를 선별하여 비디오 벡터를 만드는 데 사용할 수 있다. 이 때 각 태그의 유사도를 태그가 영상에 미치는 비중값(weight)으로 볼 수 있는데, 각각의 태그 벡터와 비중값을 곱해 전부 더한 다음 비중값의 총합으로 정규화하면 영상을 대표하는 비디오 벡터가 생성된다. 즉 비디오 벡터(v)는 다음과 같이 정의할 수 있다:

$$v = \frac{\sum_{i=1}^k t_i w_i}{\sum_{j=1}^k w_j}$$

k는 태그의 총 개수이며, t는 태그 벡터, w는 비중값이다. 각 영상의 비디오 벡터를 코사인 유사도(cosine similarity) 기법으로 연산하여 유사도 수치가 높은 영상을 선별하면 최종 추천 영상 목록을 구할 수 있다.

4. 시스템 구현 방안

해당 시스템은 웹 서비스의 형태로 구현할 것이며, 영상의 피쳐 맵(featuremap) 추출에는 Google의 Mediapipe[5], 영상 분류 및 벡터화에는 Tensorflow와 Gensim을 사용한 다. 시스템의 구성도는 다음 그림 2와 같다.

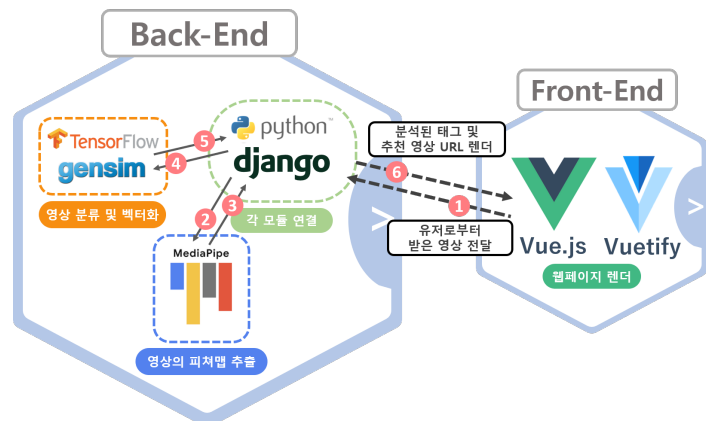


그림 2. 시스템 구성도

본 시스템은 영상의 태그를 추출하기 위해 Tensorflow로 구현된 Dbof(Deep Bag of Frames) 모델을 사용한다.

추출된 태그를 기반으로 영상마다 하나의 비디오 벡터를 생성하고, 이를 Gensim의 word2Vec 모델 형태로 저장한다. 영상이 입력되면 비디오 벡터 간의 유사도를 기반으로 입력 영상과 유사한 영상 목록을 출력한다.

사용자는 단일 영상 파일과 출력 영상의 개수 i를 입력할 수 있다. 시스템이 영상을 입력받으면 이를 태그 분류 모델의 입력으로 사용하기 위해 Mediapipe를 사용하여 피쳐 맵 형태로 변환한다. Mediapipe는 입력 영상을 초 단위로 분할해 피쳐 맵을 추출하는데, 초 단위 피쳐 맵을 다시 5초 단위의 세그먼트 단위 집합으로 분할하여 태그 분류 모델에 입력한다. 모델은 각 세그먼트에서 상위 k개의 확률을 가진 태그를 추출한다. 표 1은 영상에서 추출된 n개의 세그먼트가 각각 m개의 태그를 가짐을 보여 준다.

표 1. n개의 세그먼트와 각각에 할당된 m개의 태그

Segment 1	Segment 2	...	Segment n
Tag 1	Tag 1		Tag 1
...
Tag m	Tag m		Tag m

이후 모든 세그먼트 태그의 확률을 합산하여 최종적으로 상위 k개의 태그만 추출하고 이를 특정 영상을 대표하는 태그로 사용한다. 이 때 각 태그의 합산 확률이 전체 확률에서 차지하는 비율을 해당 태그의 비중값으로 사용한다. 표 2는 k=5인 경우의 예시이다.

표 2. k=5 일 때의 태그와 비중값

Video ID	Tag (weight)
5Uvh	Food (0.407)
	Cooking (0.327)
	Dish (0.174)
	Dessert (0.054)
	Concert (0.039)

추출된 각 태그는 위키피디아 설명 텍스트로 미리 학습시킨 워드 벡터로 변환한다. 각 태그들의 워드 벡터와 비중값을 곱한 결과 벡터들을 합산해 입력 영상의 비디오 벡터로 사용한다. 표 3은 표 2의 결과로 나온 비디오 벡터의 예시이다.

표 3. 태그 벡터와 비중값으로 계산된 비디오 벡터

Video ID	Video Vector
5Uvh	[0.02843, -0.19282, 0.14531, ..., 0.01989]

최종적으로 입력 영상의 비디오 벡터와 기존 영상들의 비디오 벡터를 코사인 유사도로 비교하면 가장 유사한

영상들을 구할 수 있다. 유사도 수치에 따라 상위 i개의 영상을 반환하면 영상 추천 프로세스가 완료된다.

5. 실험 결과

테스트는 Intel i9-9900X CPU 3.50GHz와 NVIDIA GeForce RTX 2080 Ti 환경에서 진행되었으며, 학습에는 YT8M의 1/10 scaled frame level data가 사용되었다. 태그 벡터와 워드 벡터의 차원은 100으로 설정하였으며, 출력되는 태그와 추천되는 유튜브 링크의 개수는 5개로 설정하였다.

테스트로 UNDERkg의 갤럭시 S10 5G 리뷰 영상을 이 시스템에 입력했을 때의 결과를 표 4, 표 5에 서술하였다.

표 4. 입력 영상과 분석된 태그


Input Video (Galaxy S10 5G Review)	Output Tag (weight)
	Mobile phone (0.369)
	Smartphone (0.356)
	iPhone (0.135)
	Samsung galaxy (0.097)
	Personal computer (0.043)

표 5. 추천 유튜브 링크와 유튜브 영상의 태그

Output2	
Youtube Link	Youtube Tag
https://www.youtube.com/watch?v=sEMxwhz2hCY	Mobile phone, Smartphone, iPhone, Samsung galaxy, Personal computer
https://www.youtube.com/watch?v=IFhUU5xOTzM	Mobile phone, Smartphone, iPhone, Samsung galaxy, Personal computer
https://www.youtube.com/watch?v=ao6oxQja-qA	Mobile phone, Smartphone, iPhone, Samsung galaxy, Personal computer
https://www.youtube.com/watch?v=jSMAUR78cmU	Mobile phone, Smartphone, iPhone, Samsung galaxy, Personal computer
https://www.youtube.com/watch?v=yMbWeTpyFKY	Mobile phone, Smartphone, iPhone, Samsung galaxy, Personal computer

표 4에서 먼저 입력 영상과 연관된 5개의 태그가 생성되었고, 각 태그와 입력 영상과의 유사도를 볼 수 있다.

표 5에서 입력한 영상에 대한 5개의 추천 영상 링크가 생성된 것을 알 수 있다. 해당 영상에 대한 태그가 이 시스템에서 분석한 입력 영상의 태그와 유사한 것으로 보아, 입력 영상과 유사도가 높은 유튜브 영상이 추천되었음을 추론할 수 있다.

이 결과 값을 백엔드에서 프론트엔드로 전달하였으며, 그림 3과 같이 입력 영상의 태그와 입력 영상과 관련된

유튜브 추천 URL이 웹페이지에 그려졌음을 확인할 수 있다.

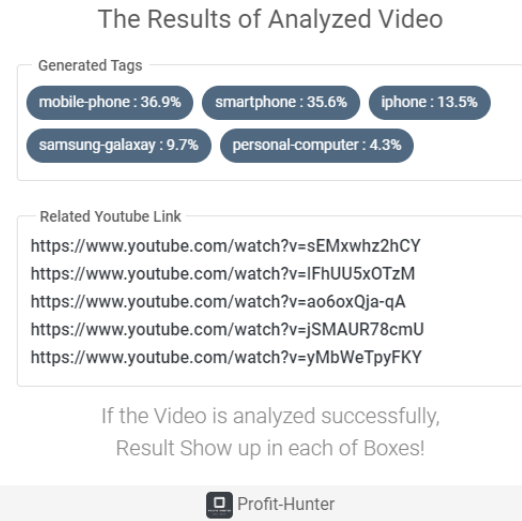


그림 3. 영상 분석 후의 웹페이지 UI

6. 결론 및 향후 연구

본 논문에서는 내용 기반 영상 추천을 위해 영상의 내용을 대표하는 비디오 벡터라는 개념을 새로 도입하였다. 동영상 업로드 시 세그먼트를 분석하여 자동으로 태그를 추출하고 비디오 벡터를 생성하였으며, 이를 기반으로 영상의 태그와 유튜브 URL을 이용자에게 추천해주는, 새로운 유튜브 영상 추천 시스템을 제안하였다.

본 시스템을 통하여 유튜브 크리에이터는 더 정확한 영상 태그와 높은 광고 효율을, 유튜브 이용자는 더 취향에 맞는 관련 영상을, 유튜브 본사는 더 높은 사이트 접속률을 기대할 수 있을 것이다.

향후 연구로는 태그 벡터를 영상 차원에서 생성하는 방법이나, 전처리 과정을 단축시킬 수 있는 방안을 모색할 계획이다.

참고 문헌

- [1] 오세욱, 송해엽, “유튜브 추천 알고리즘과 저널리즘”, 한국언론진흥재단, 2019.
- [2] Junwei Ma, Satya Krishna Gorti, Maksims Volkovs, Ilya Stanevich, “Cross-Class Relevance Learning for Temporal Concept Localization”, Layer6 AI, 3, 2019.
- [3] Youtube-8M, “Youtube-8M Large-Scale Video Understanding Challenge”, Google, 2019.
<https://research.google.com/youtube8m/workshop2019/>
- [4] Tomas Mikolov, Kai Chen, Gregory S. Corrado, Jeffrey A. Dean, “Computing numeric representations of words in a high-dimensional space”, Google, 2013.
- [5] Mediapipe, Google. <https://github.com/google/mediapipe>