빅데이터프로젝트: 캡스톤디자인 보고서

라디오 사연 기반 노래 추천 시스템

2018 년 6월 14일

텍 봇

빅데이터경영통계전공 20132706 전민재
빅데이터경영통계전공 20152645 민향숙
빅데이터경영통계전공 20152648 박주란

목 차

Ⅰ. 주제 선정 배경

추천 시스템(Recommendation system)은 정보 필터링을 사용하여 **사용자**에게 흥미로운 아이템을 제공하는 시스템이다. 정보 시스템은 사용자에게 **개인 신상, 관심 분야, 선호도** 등을 질의하여 **사용자의 정보 프로파일을 획득하는 기법**으로, 추천 시스템은 이러한 정보를 기반으로 고객의 심리 정보와 선호도 정보에 알맞은 **정보 및 상품을 추천하거나 제공하는 방법**이다.

수많은 컨텐츠들 중에 사용자에게 적합한 것을 효율적으로 제공하기 위한 개인 맞춤형 서비스 시대가 도래하였다. 소비자들이 직접 나서서 컨텐츠를 찾아보는 데에는 한계가 있는데, 방대한 전체 컨텐츠의 양에 비해 그들이 접하는 부분은 다소 한정적이기 때문이다. 소비자가 아직 경험 또는 인식하지 못했지만, 흥미를 느낄 만한 새로운 컨텐츠를 추천해 주기 위해 다양한 추천 시스템들이 제안, 연구되고 있다.

본 프로젝트에서는 **라디오 사연을 분석**해 **음악을 추천**함으로써 문서 분석을 통한 **사용자의 상황 분석과 그에 따른 음악 추천의 가능성을 확장**시킬 수 있다는 점에서 의의가 있다. 또한, 이를 발전시켜 최근 화제가 되고 있는 **인공지능 스피커를 통한 음악 추천에 응용이 가능**할 것으로 보여 진다.

Ⅱ. 관련 연구

1) **라디오 사연 분석을 통한 음악 추천시스템(2012)** 논문의 경우 문서 간 유사도를 통해 사연 간의 유사도를 구하고 가장 유사한 사연의 신청 곡을 추천해주는 시스템을 제시했다. LSA(Latent Semantic Analysis)를 통해 벡터화한 사연간의 문맥적 의미를 파악하고, 학습시킨 벡터를 기반으로 새로운 사연이 들어오면 가장 유사한 사연의 신청곡을 추천해주는 알고리즘을 제시한다. 하지만, 사연 간의 유사도를 바탕으로 유사한 사연의 신청 곡만을 추천해주기 때문에 사연에서 신청한 곡만 추천된다는 한계가 있다. [2]

2) **MusicSense: Contextual Music Recommendation using Emotional Allocation Modeling (2007)** 논문의 경우 웹 블로그의 게시글 등을 대상으로 사람이 직접 감정 사전 등에 기반한 '감정 라벨링'을 진행한 후, 라벨링된 감정과 가장 유사한 감정을 갖는 노래를 추천해주는 방식을 이용했다. 기존의 영어 감정 사전에 대한 다양하고 수많은 연구가 선행되었기 때문에 텍스트를 대상으로 사람이 직접 라벨링을 할 수 있었으며, 해당 라벨을 모델 성능 평가 척도로 사용했다는 점에서 의의를 갖는다. 하지만, 라벨링 자체를 사람이 했다는 점에서 데이터가 증가할 경우 라벨링에 어려움이 있다는 한계가 있다. [8]

3) **Social Tagging and Music Information Retrieval (2008)** 논문의 경우 기존의 노래를 설명할 수

있는 특징인 아티스트, 앨범, 장르 뿐만 아니라 노래를 표현하는 자유 텍스트(Tag)를 이용하여 노래 추천 시스템에 대한 개념을 제시 하였다. 본 논문에서는 태그 유사도를 기반으로 추천했을 때, 일반적인 추천 시스템인 협업 필터링(CF)에 비해 더욱 투명하고 설득력을 있다는 이점을 가진다고 설명한다. 하지만, 소셜 태그의 경우 태그를 지정하는 사람이 주로 젊고, 유행에 민감하며, 인터넷에 익숙한 계층이기 때문에 일반적인 음악 취향을 대변하는데는 미흡할 수 있다는 한계가 있다. [7]

4) **InCarMusic: Context-Aware Music Recommendations in a Car(2011)** 논문의 경우 상황의 유사성에 대한 주관적 평가에 대한 툴을 만들어 확률론적 모델을 이용하여 노래를 추천한다. 본 논문의 경우 앱을 이용하여 사용자들의 Rating을 평가 받기 때문에 추천 이후에 피드백을 통해 계속해서 유저 인터페이스를 갱신하여 노래를 추천한다. 하지만, 앱을 시작하면서 몇 가지 음악에 대한 사전 평가를 진행하고 이를 기반으로 음악을 평가하기 때문에 사용자에 대한 기본 평가 정보가 없으면 음악 추천에 어려움이 있다는 한계가 있다. [6]

Ⅲ. 라디오 사연기반 음악 추천시스템

제 1절 시스템 개요

우리는 최종적으로 2개의 모델을 제안하고자 한다.

텍스트의 감정과 상황을 분석하여 노래를 추천해주는 시스템을 제안하고자 한다. 그러나 한국어는 영어와 다르게 제대로 된 감정사전이 구축되어 있지가 않다. 따라서 우리는 텍스트 자체의 감정과 상황을 분석하는 것이 아닌, 비슷한 감정과 상황을 가진 텍스트를 찾아서 노래를 추천해주는 시스템을 구축해보고자 한다. 그리하여 우리는 텍스트와 신청곡이 하나의 쌍으로 묶여 있는 라디오 사연 데이터를 기반으로 모델링을 진행하고자 한다.

두번째는, 기존의 방식은 제대로 된 한국어 감정사전이 존재하지 않기 때문에 고안한 방법이다. 여기에 우리만의 감정사전을 구축하여 새로운 모델링을 시도해보았다. 노래 정보를 수집하면서 노래의 태그와 가사가 노래의 감정을 반영해준다는 사실에서 고안한 것이다. 그러나 감정단어라는 것은 의미론적으로 유사함을 뜻하지 단어 적인 일치성은 뜻하는 것이 아니다. 사연과 노래 태그 간에 단어 적인 일치성을 기대하기가 어렵다. 그래서 단어의 의미를 벡터로 표현해주는 Word2vec 모델을 사용하고자 한다. 태그와 가사 그리고 사연을 바탕으로 Word2vec 모델을 생성함으로써 비슷한 감정을 가지는 단어들은 비슷한 벡터 값을 가지도록 하였다. 따라서 우리는 해당 모델을 한국어 감정사전으로 대체하여 모델링을 진행하고자 한다.

결론적으로 두개의 모델은 서로 다른 장점과 단점을 가지고 있기 때문에, 우리는 두개의 모델을 모두 제안하고자 한다.

제 2절 데이터 수집 및 1차 전처리

1) 노래 데이터 수집

노래 정보는 멜론, 엠넷, 그리고 지니 총 3곳의 대표적인 음원사이트에서 수집하였다. 노래정보로는 노래태그, 가사, 인기도값 그리고 장르를 수집하였다. 노래를 나타내는 값으로는 태그와 가사를 이용하고자 한다.

(1) 먼저 3개의 사이트에서 노래 태그를 수집하는 과정이다.

Python에서 웹 페이지에 HTTP 요청을 보내는 모듈인 requests와 HTML을 파싱하는 모듈인 BeautifulSoup을 이용하여 웹크롤링을 진행하였다. 노래 데이터를 수집하는 전체 과정은 다음과 같다.

1. Python의 requests를 이용해 DJ앨범이 있는 웹사이트에 HTTP 요청을 보낸다.
2. BeautifulSoup을 이용해 해당 웹사이트의 HTML에서 DJ앨범의 URL에 해당하는 부분만 파싱하여 DJ앨범의 URL만 따로 수집한다.
3. requests를 이용해 수집된 URL에 접근한 뒤, BeautifulSoup을 이용해 노래와 태그 정보를 파싱하여 저장한다.

먼저 각 사이트에서 DJ 앨범의 URL을 수집하였다. 아래에는 사이트별로 DJ앨범의 URL 수집 방법과 DJ앨범에 대해 설명하였다.

① '멜론'에서는 2014년 1월부터 2018년 4월까지 주별로 인기있는 DJ앨범들의 URL을 먼저 수집하였다. Python의 requests와 BeautifulSoup을 이용해서 앨범DJ앨범은 개인 사용자가 비슷한 감정과 상황을 가지는 노래들을 선별하여 선곡한 앨범이다. DJ앨범에는 해당 앨범의 대표적인 감정과 상황을 표현하는 '태그'들이 붙여져 있고, 이 태그들과 어울리는 음악들이 담겨 있다. (그림 1 참고)

② '지니'에서는 '지니'가 정한 대표적인 상황과 감정을 바탕으로 각각 인기도 기준 상위 100개의 DJ앨범 URL을 수집하였다. DJ앨범에는 해당 DJ앨범의 태그와 수록곡이 포함되어 있다. 엠넷은 장르, 시대, 연령, 날씨/시간 등의 메뉴별로 DJ앨범이 분류되어 있다. 각 메뉴별로 인기도 기준 상위 100개의 DJ앨범 URL을 수집하였다. DJ앨범에는 동일하게 태그와 음악들이 포함되어 있다. (그림 2와 그림 3 참고)

③ '엠넷'에도 동일하게 DJ앨범이라는 것이 존재하는데, 장르, 느낌, 장소 등의 메뉴로 분류가 되어 있다. 각 메뉴에 접근하여 인기도 순으로 정렬한 뒤 상위 100개 앨범을 각각 크롤링하였다. 엠넷 DJ앨범에는 감정과 상황을 나타내는 테마와 기타태그가 있고, 하단에는 앨범의 감정과 상황에 어울리는 음악들이 담겨 있다. (그림 4 참고)

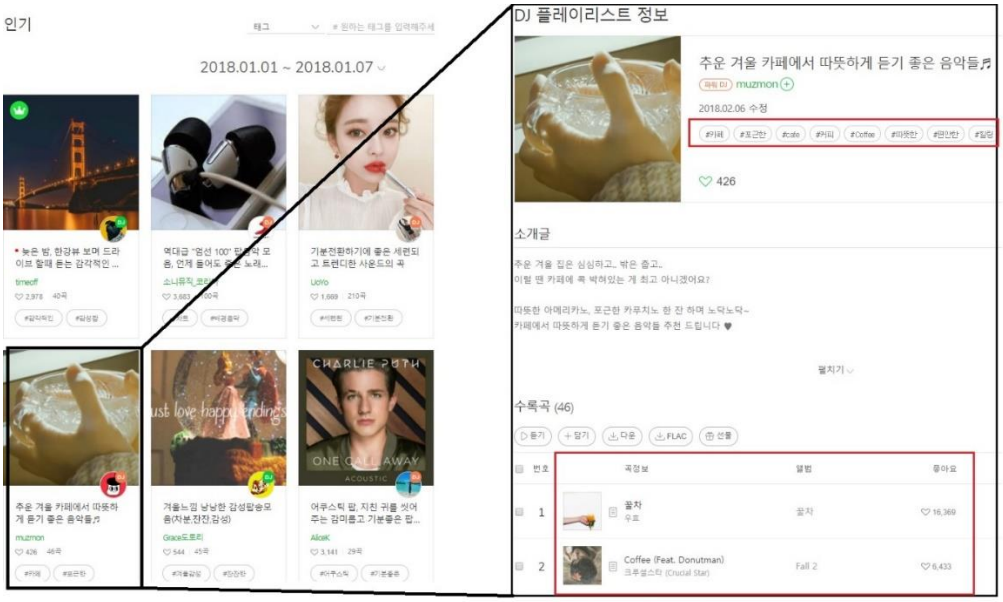


그림 1 멜론 DJ앨범과 DJ앨범 정보

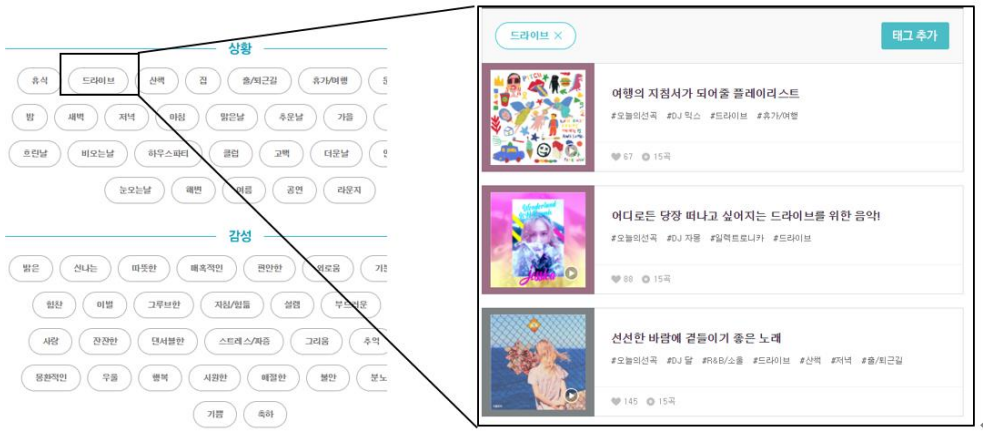


그림 2 지니에서 태그 정보와 DJ앨범

그림 3 지니 DJ앨범 정보

그림 4 엠넷 DJ앨범

그 다음에는 DJ앨범 URL에 각각 접근하여 해당 앨범의 태그정보와 수록곡을 수집하였다. 데이터 수집 결과는 멜론: 54787곡 태그 1759개, 지니: 17402곡 태그 134개, 그리고 엠넷: 46597곡 태그 77개이다. 각 음원 사이트 별로 데이터를 <그림5>와 같이 변형시켰다. 행은 노래 정보, 열은 태그로 하여 각 노래 당 태그 빈도수 행렬을 생성하였다.

songid	songebid	songebnm	title	singer	가을	감사	감성	겨울	설레는	추억	차분	외로운	맑은	사랑	즐거운	분위기	용
10015192	2760765	Gran Bufe	Dizzi Laica	Shmaltz!	0	0	0	0	0	0	0	0	0	0	0	0	
1001880	310104	Greatest H	Pop	N Sync	0	0	0	0	0	0	0	0	0	0	3	0	
1001881	310104	Greatest H	Gone	N Sync	0	0	1	0	0	0	0	0	0	0	0	0	
10024197	2761728	Some Oth	Lucky To	Bill Evans	0	0	1	0	0	0	2	0	0	0	0	0	
10024584	2761766	Moonriver	Moon Rivi	Audrey He	0	0	2	0	0	0	0	0	0	0	2	0	
1002536	310155	Next Wav	Blaze It Up	Mondo Gr	0	0	0	0	0	0	0	0	0	0	0	1	
10027254	2762033	YYY ACA	Summer F	Yoyoyo A	0	0	0	0	0	0	0	0	0	0	1	0	
1002851	310185	김현철 Be	그대니까요	김현철	0	0	3	0	0	0	0	0	0	0	0	0	
1002853	310185	김현철 Be	왜그래	김현철	0	0	0	0	0	2	0	0	0	0	0	0	
10029052	2762194	Unforgettz	The Christ	Nat King	0	0	0	0	0	0	0	0	0	0	0	0	
1002911	310191	Discovery	One More	Daft Punk	0	0	0	0	0	0	0	0	1	0	5	0	
1002914	310191	Discovery	Harder, Be	Daft Punk	0	0	2	0	0	1	0	0	3	0	15	1	
1002919	310191	Discovery	Something	Daft Punk	2	0	31	3	0	0	3	1	2	1	7	5	
1002921	310191	Discovery	Veridis Qu	Daft Punk	0	0	0	0	0	0	2	0	0	0	0	0	
1002923	310191	Discovery	Face To Fa	Daft Punk	0	0	2	0	0	0	0	0	0	0	1	0	

그림 5 노래 데이터 전처리1

(2) 두번째로 3개의 노래 데이터를 합치는 과정이다.

각 음원 사이트별로 노래 제목과 가수명을 동일하게 만들기 위해서 노래 제목과 가수 텍스트 전처리를 <그림6>과 같이 수행하였다.

그림 6 노래 데이터 전처리2

그 다음으로 멜론 데이터을 바탕으로 노래 제목과 가수를 기준으로 3개의 음원 사이트 데이터를 결합하였고, 태그 빈도 값은 동일한 태그일 경우 합을 해주었다. 그 결과 총 54787곡을 태그

약 1800개를 수집하였다.

	songid	songtitle	singer	가을	감사	감성	겨울	눈	맑은	봄	...	여유	외로운	응원	이별	즐거운	차분	추억	크리스마스	행복	흐린
0	10015192	Dizzi Laicas	Shmaltz!	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0
1	1001880	Pop	N Sync	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	5.0	0.0	0.0	0.0	9.0	0.0	0.0	0.0	1.0	0.0
2	1001881	Gone	N Sync	0.0	0.0	1.0	1.0	0.0	0.0	0.0	...	6.0	0.0	0.0	0.0	2.0	0.0	2.0	0.0	0.0	0.0
3	10024197	Lucky To Be Me	Bill Evans	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	2.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0
4	10024584	Moon River	Audrey Hepburn	0.0	0.0	2.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0
5	1002536	Blaze It Up	Mondo Grosso	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	10027254	Summer Fling	Yoyoyo Acapulco	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	3.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
7	1002851	그대니까요	김현철	0.0	0.0	5.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0

그림 7 노래 데이터 전처리3

(3) 세번째로 노래 가사, 장르, 그리고 인기도 값을 수집하고 장르에 따라 노래를 필터링 하였다.

완성된 노래 데이터를 바탕으로 멜론 사이트에서 노래에 대한 추가 정보를 수집하였다. 각 노래별로 장르와 가사 그리고 인기도 값에 대해 추가적으로 웹 크롤링을 진행하였다.

songid	genre	lyric	singer	title	year
3045598	<dd>국외	I'm runnin	Lincoln Br	Best Days	<dd>2010.09.28</dd:
3045600	<dd>국외	You're the	Lincoln Br	More Thai	<dd>2010.09.28</dd:
30456119	<dd>Balla	I always lc	거미	I I YO (Prc	<dd>2017.06.05</dd:
30456129	<dd>Balla	There is a	LambC (랜	Butterfly (	<dd>2017.06.06</dd:
30456252	<dd>Elec	Don't you	DUVV	DARE TO	<dd>2017.06.07</dd:
30456435	<dd>Drar	If I told yc	Yael Meye	No matter	<dd>2017.06.07</dd:

songid	like
10015192	11
1001880	2462
1001881	2322
10024197	18
10024584	921

그림 8 노래의 가사, 장르, 인기도값 데이터

앞서 만든 노래 데이터와 조인 한 후, Pop, Animation, New Age, Classic 등의 장르는 제거하였다. 그리하여 노래 데이터는 노래 40439곡으로 완료하였다.

(4) 네번째로 노래 태그와 가사 전처리 과정이다.

노래 데이터는 행렬 형태로 두지 않고 다시 태그를 하나의 문장으로 바꾸어 주었다. 예시는 아래와 같다. 노래 태그로 감성이 총 5번, 여유가 1번 그리고 추억이 4번 나온 '김현철의 그대니까요'는 '감성 감성 감성 감성 감성 여유 추억 추억 추억 추억'과 같이 문장으로 바꾸어 주었다.

	songid	songtitle	singer	가을	감사	감성	겨울	눈	맑은	봄	...	여유	외로운	응원	이별	즐거운	차분	추억	크리스마스	행복	흐린
7	1002851	그대니까요	김현철	0.0	0.0	5.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0

songid	songtitle	singer	tag
1002851	그대니까요	김현철	감성 감성 감성 감성 감성 여유 추억 추억 추억 추억

그 다음에는 태그와 가사를 하나의 문장으로 이어 준 뒤, Python의 한국어 형태소 분석 모듈인 konlpy의 Twitter 형태소 분석기를 사용하여 명사와 형용사 그리고 숫자만을 추출하여 불필요한 단어들을 제거해주었다. 그리하여 노래별로 태그와 가사로 이루어진 하나의 문장을 생성해주었다.

태그와 가사를 형태소 분석을 진행 한 이유는 태그 내 불필요한 단어들을 제거해 주고, ‘감성가득’, ‘감성힙합’, ‘감성재즈’, ‘재즈힙합’을 ‘감성’, ‘힙합’, ‘재즈’로 각각 분리하기 위해서이다. 또한, 가사를 추가한 이유는 개인이 주관적으로 생성한 태그만으로는 노래 정보를 모두 파악할 수 없다고 판단 하였기 때문이다.

2) 라디오 사연 수집

KBS, MBC, SBS 에서 아침7시부터 밤 12시까지 다양한 시간대의 라디오에서 사연을 수집하였다. 약 30개 이상의 라디오에서 사연을 수집하였다. Python에서 웹 페이지에 HTTP 요청을 보내는 모듈인 requests와 웹 어플리케이션 테스팅 프레임 워크인 selenium 그리고 HTML을 파싱하는 모듈인 beautifulsoup을 이용하여 웹크롤링을 진행하였다. Python을 이용해 라디오 사연 게시판에서 데이터를 수집하여 엑셀 파일로 저장하였다.

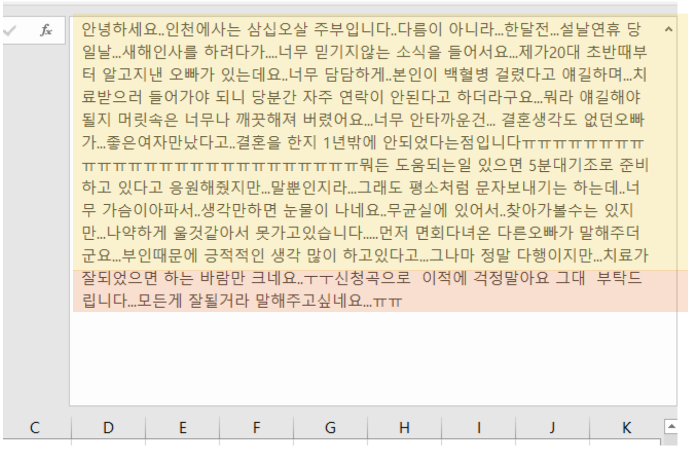


그림 9 라디오 사연의 구성

라디오 사연은 일반적으로 사연 – 신청곡으로 구성되어 있다. 노란색 부분이 자신의 이야기를 소개하는 사연 부분이고 붉은 부분은 사연에 따라 작성자가 듣고 싶은 신청곡을 기재한 부분이다. 사연은 2014년부터 2018년까지 분포하여 있고, 총 10000여 개를 수집하였다. 수집된 사연은 직접 엑셀을 이용하여 사연과 신청곡을 분리하는 전처리 과정을 진행하였다.

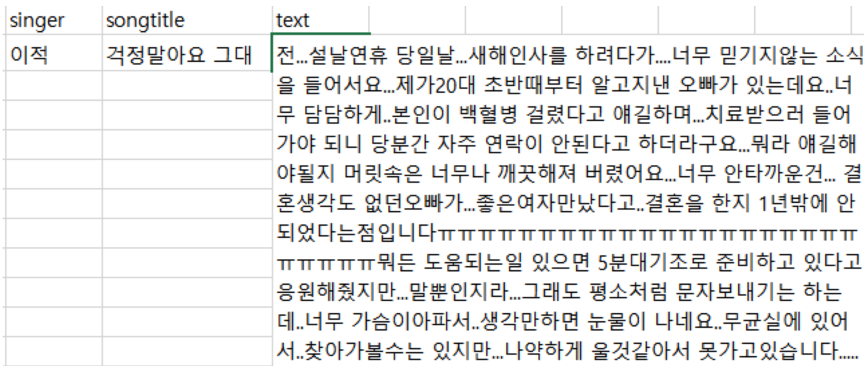


그림 10 라디오 사연 전처리 후

가수와 제목을 입력할 때에는 앞서 생성한 노래 데이터의 텍스트와 동일하게 기입하였다. 예를 들면, 사연에는 '빅뱅-판다스틱베이비'라고 적혀 있다면, 이를 노래 데이터의 텍스트와 동일하게 'BIGBANG-FANTASTIC BABY'로 바꾸어 기입하였다. 전처리 과정 중 사연의 총 길이가 350자 미만이거나, 신청곡이 존재하지 않는 사연들을 분리하였다. 사연 총 길이에 제한을 둔 이유는 사연 내용의 품질 때문이다. 모든 사연이 자신의 이야기를 담고 있는 것이 아니다. 일부 사연의 경우 라디오 DJ에게 하고싶은 말을 쓰거나, 신청곡만 적혀 있는 경우가 있기 때문에 이러한 데이터를 제거하기 위해서 사연 길이에 제한을 두게 되었다. 그 결과 사연은 총 1905개로 마무리 짓게 되었다. 사연 길이가 350자 이상이고 신청곡이 없는 사연 데이터 4552개는 따로 저장하였다.

제 3 절 시스템 구성

1) 첫번째 추천 시스템

첫번째 추천 시스템은 먼저 2개의 모델을 제안하고, 객관적인 모델평가 이후 최종적으로 한 개의 모델을 정하고자 한다.

(1) 유사한 사연 기반 노래 추천 시스템(모델1)

유사한 사연 기반 노래 추천 시스템의 전체 구성은 아래 그림과 같다.

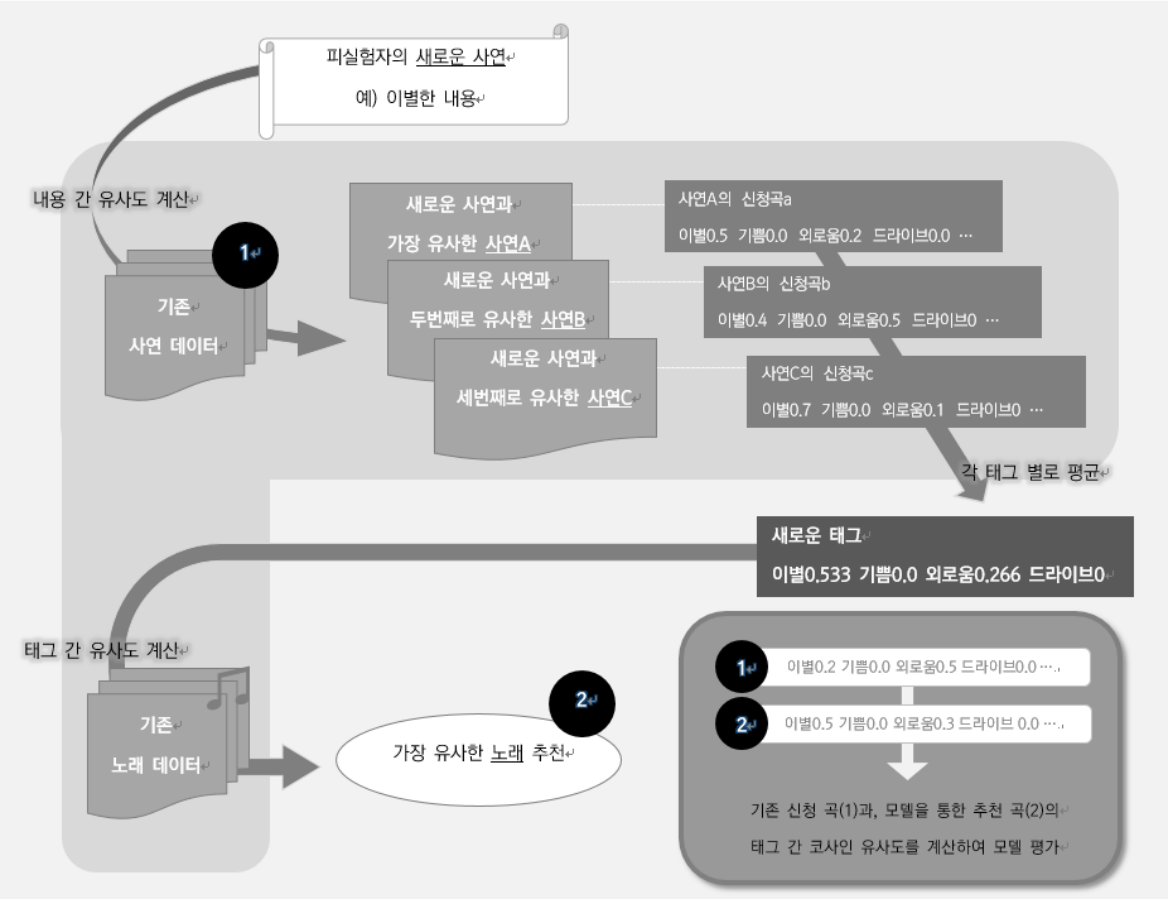


그림 11 유사한 사연 기반 노래 추천 시스템 구성도(1)

먼저, 사연 데이터를 형태소 분석을 진행한 뒤, 텍스트에서 키워드에 가중치를 주는 TF-IDF 행렬을 구한 뒤, SVD를 통해 사연의 차원을 축소시켜 준다. 두번째로, 사연간 코사인 유사도 값을 구해 유사도 값이 높은 순서대로 정렬 한 후 상위 7개의 사연을 추출한다. 세번째로 상위 7개 사연의 신청곡 들의 태그 벡터들을 모은 뒤, 각 태그별로 평균값을 구해 새로운 태그 벡터를 생성한다. 마지막으로 새로 생성된 태그 벡터와 노래 태그 벡터들 간에 코사인 유사도 값을 구한다. 코사인 유사도 값이 가장 높은 노래를 추천해 준다. 모델 평가 방법은 '모델 추천곡이 기존의 신청곡과 유사한 노래인가?'이다. 사연의 신청곡 태그 벡터와 추천시스템을 통해 나온 추천곡 태그 벡터 간의 코사인 유사도 값들을 구한다. 그리고 유사도 값의 전체 평균을 계산한다. 이 평균값이 높을수록 성능이 좋다는 것을 의미한다.

1. 라디오 사연 형태소 분석

수집한 라디오 사연 1905개를 Python의 한국어 형태소 분석 모듈인 konlpy의 Twitter 형태소 분석기를 이용하여 사연의 명사와 형용사를 추출하였다. 그 결과, 사연을 구성하는 단어는 총 14324개이다. 형태소 분석기가 띄어쓰기, 오타 등은 자동으로 고쳐주지 못한다. 또한 사연에는 개인의 이름과 같은 명사가 등장하기도 한다. 이러한 단어들은 제거해주기 위해 빈도수가 5 이하인 단어들은 제거해주었다. 그리하여 총 3084개의 단어만을 사용하였다.

2. TF-IDF를 활용한 문서 분석

수집한 라디오 사연은 1905개이고, 이를 구성하는 단어 수는 3084개이다. 각 사연들은 단어들로 구성되는데 사연 내에서 이 모든 단어가 같은 중요도를 가지는 것이 아니다. 사연에는 사연의 특징을 나타내 주는 단어들과 일반적으로 포함되는 단어들이 존재한다.

```
'불안 마음 안녕 당진 결혼 년차 세댁 입니 저희 연애 결혼 아직 아기 소식 없네 아기 하늘 선물 마음 편하 가끔 너무 불안 기분 우울 ㅠㅠ 신랑 괜찮 그게 우울 노래 불안 마음 안녕 당진 결혼 년차 세댁 입니 저희 연애 결혼 아직 아기 소식 없네 아기 하늘 선물 마음 편하 가끔 너무 불안 기분 우울 ㅠㅠ 신랑 괜찮 그게 우울 노래'
```

예를 들면, '입니','저희'와 같이 문서를 구성하기 위해 필요하지만 특별한 뜻이 없고, 모든 문서에서 동일하게 나타나는 단어가 있는 반면, '아기', '연애', '결혼'과 같이 의미가 있고, 모든 문서에서 동일하게 나타나지 않는 단어가 있다. 문서의 특징을 부각시키기 위해선 전자의 경우는 가중치를 낮게, 그리고 후자의 경우에는 가중치를 높게 줄 필요가 있다. 이것을 해결하기 위해서 문서에 TF-IDF를 사용하였다. TF-IDF는 문서 내에서 키워드를 추출하는데 많이 사용되고 있다. TF-IDF 값이 높은 단어들을 키워드로 선정하는 것이다. 국내 연구 중에는 소설에서 키워드를 추출하거나 또는, 뉴스 기사로부터 분야별 키워드를 추출하기 위해 TF-IDF 모델을 사용하였다. [1,3]

TF-IDF란 TF와 IDF값의 곱을 의미한다. TF값은 한 문서에서 사용된 모든 단어들의 출현 빈도를 나타낸 값으로, 출현 빈도가 높은 단어일수록 해당 문서에서 중요도가 높은 것으로 판단한다. 하지만 해당 문서에서 출현 빈도가 높다고 해서 그 단어가 해당 문서에서 중요하다고 볼 수 없다. 이것을 해결하기 위한 것이 IDF값이다. IDF값은 전체 문서에서 보편적으로 등장하는 단어일수록 값이 낮아진다. 결론적으로 TF값과 IDF값을 곱해줌으로써, 모든 문서에서 보편적으로 등장하는 단

어일 경우 최종적으로 가중치 값을 낮게 만들고, 하나의 문서에서만 보편적으로 등장하는 단어일 경우에는 최종적으로 높은 가중치 값을 가지게 된다.

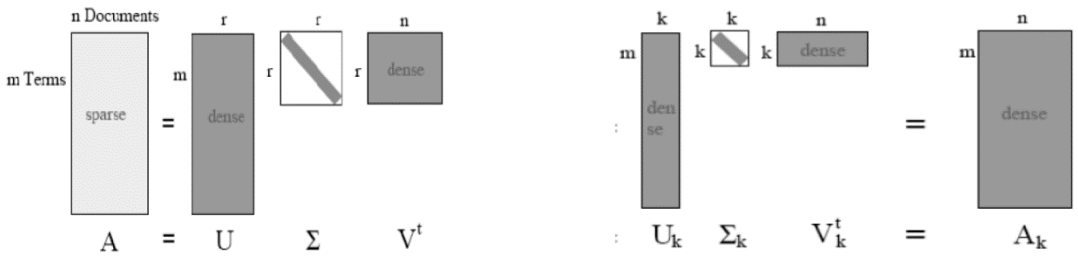
TF-IDF 행렬을 생성하기 위해 Python sklearn모듈의 TfidfVectorizer를 사용하였다. 라디오 사연 형태소 분석 다시 빈도수가 5번 이상인 단어만 사용한다고 하였기 때문에, 단어 최소 빈도수 제한 파라미터를 5로 설정하고, TF-IDF 행렬을 생성하였다. 그리하여 1905(사연개수)X3084(단어개수) 행렬을 만들었다.

3. SVD를 통한 차원 축소

TF-IDF 행렬은 1905X3084의 크기를 가진다. 그러나 한 개의 사연에는 3084개의 단어가 모두 등장하지 않고, 평균적으로 30여개의 단어만이 사용된다. 따라서, 현재 생성한 행렬은 굉장히 낮은 밀도를 가지게 된다. 따라서 각 행끼리 유사도를 구하게 된다면 겹치는 값이 많지 않아 의미있는 유사도 값을 얻기가 어렵다. 이러한 문제를 해결하기 위해 SVD를 이용해 행렬의 차원을 축소시키고 행렬의 밀도를 높이고자 한다. Boling(2015)에 의하면 SVD를 이용해 차원축소를 하게 되면 문서 내의 잡음을 줄여주고 차원은 축소하였지만 문서의 특징은 보존한다. 또한, 문서간 코사인 유사도를 구할 경우, SVD를 사용할 때 더 의미 있는 값이 나온다고 한다. [5]

SVD는 Singular Value Decomposition으로 특이값분해를 의미한다. SVD는 직사각형의 행렬을 분해하는 방법이다. 예를 들면, MxN 크기의 행렬 A를 다음과 같이 분해하는 것을 의미한다.

$$A = U\Sigma V^T.$$



그 다음 A의 0보다 큰 고유값의 개수를 r이라고 할 때, r보다 작은 k를 임의로 설정하여 Σ_k를 만든다. 그리고 U와 V행렬에서 k에 대응하는 부분만 남겨 동일하게 U_k와 V_k^T를 만든다. 마지막으로 $U_k * \Sigma_k * V_k^T$를 해줌으로써 행렬 A와 비슷하지만 k차원으로 축소된 A_k행렬을 구축한다. 따라서 SVD를 이용하면 행렬의 단어 차원을 K개로 축소시킬 수 있게 된다. 단어 차원을 축소하면 비슷한 패턴을 가지게 되는 단어는 근접하게 되어 문서간 유의미한 유사도 값을 얻을 수 있게 된다. 최적의 K값은 이후 모델 평가 방법에서 설명하고자 한다.

4. 유사한 사연 선정

유사한 사연을 선정하는 과정은 코사인 유사도 값을 기준으로 한다. 대각행렬의 코사인유사도 값은 0으로 바꾸어 준다. 사연간 코사인 유사도 값을 계산하여 유사도 값이 가장 높은 사연 N개를 선정한다. 최적의 N개의 개수는 모델 평가 방법에서 설명하고자 한다.

5. 노래 태그 행렬 생성

앞서 전처리를 끝낸 노래 데이터는 형태소 분석을 진행한 태그와 가사가 문장으로 이어져 있는 형태이다. 노래 별로 태그 벡터를 생성하기 위해 노래X태그 행렬을 생성하고자 한다. Python sklearn 모듈의 CountVectorizer 함수를 이용하여 2번이하로 등장한 단어는 제외하였다. 그리하여 노래 40439곡, 태그 1727개로 노래X태그 행렬을 생성하였다. <그림 12>를 통해 각 노래 별 전체 태그 수 빈도를 살펴보면, 노래별로 태그 수가 크게 차이나는 것을 알 수 있다.

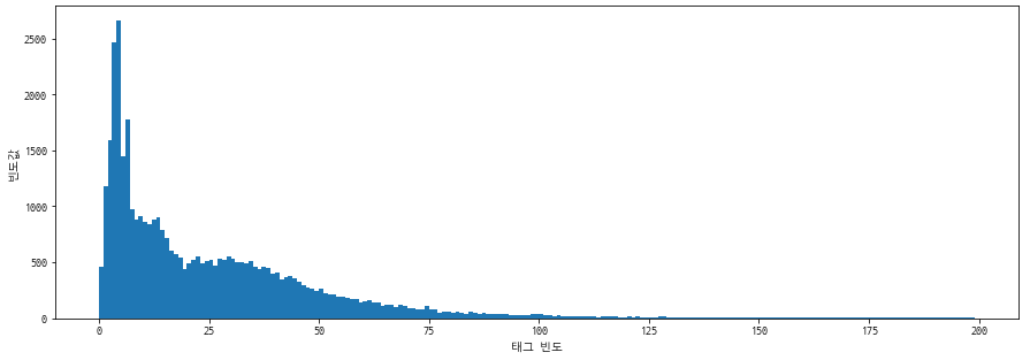


그림 12 노래별 전체 태그 수 히스토그램

노래 간 태그 수 차이를 줄이기 위해 노래의 태그를 비율로 바꾸어 주었다. 예를 들면, 어떤 노래의 태그 빈도 값이 <표1> 같을 때, 이를 비율로 바꿔주게 되면 <표2>가 된다.

노래제목	응원	위로	슬픔	기쁨	신나는	행복	즐거운
힘 내!	23	17	2	10	5	5	8

< 표 1 >

노래제목	응원	위로	슬픔	기쁨	신나는	행복	즐거운
힘 내!	0.33	0.24	0.03	0.14	0.07	0.07	0.11

< 표 2 >

이렇게 노래별로 태그 값을 비율로 바꿔 줌으로써, 노래 별 전체 태그 수의 격차를 줄이고자 한다.

6. 유사한 사연들 기반 새로운 노래 벡터 생성

4번에서 선정한 유사한 사연들의 신청 곡들을 뽑아낸다. 그 다음으로 각 노래 별로 태그 벡터를 찾은 뒤, 태그별로 평균값을 구해 새로운 노래 태그 벡터를 생성한다.

7. 노래 태그 벡터간 유사도 기반 음악 추천

새로운 노래 태그 벡터와 기존 노래 태그 행렬간 코사인 유사도 값을 구하여, 코사인 유사도 값이 가장 높은 노래를 추천해준다.

8. 모델평가

모델의 성능을 평가하는 객관적 지표로는 사연의 기존 신청곡과 모델의 추천곡 간의 코사인 유사도 값으로 한다. 전체 1905개 사연의 신청곡과 추천곡 간의 각각 코사인 유사도 값을 구한 후 유사도 값의 전체 평균이 높을수록 좋은 모델이다. 따라서 해당 지표를 바탕으로 최적의 SVD 차원 수 K 와 사연 개수 N을 정하고자 한다. 그리하여 K값은 500에서 1300까지 200간격으로, N값은 3에서 7까지 2간격으로 총 15개의 경우의 수를 진행해보았고, 전체평균은 그 결과 값이다.

번호	N	K	전체 평균	번호	N	K	전체 평균	번호	N	K	전체 평균
0	3	500	0.1949	5	5	500	0.2390	10	7	500	0.2665
1	3	700	0.1950	6	5	700	0.2414	11	7	700	0.2684
2	3	900	0.1946	7	5	900	0.2409	12	7	900	**0.2686**
3	3	1100	0.1956	8	5	1100	0.2417	13	7	1100	0.2678
4	3	1300	0.1943	9	5	1300	0.2401	14	7	1300	0.2657

표 3 경우의 수와 결과 값

해당 결과를 그래프로 그리면 <그림 13>과 같다.

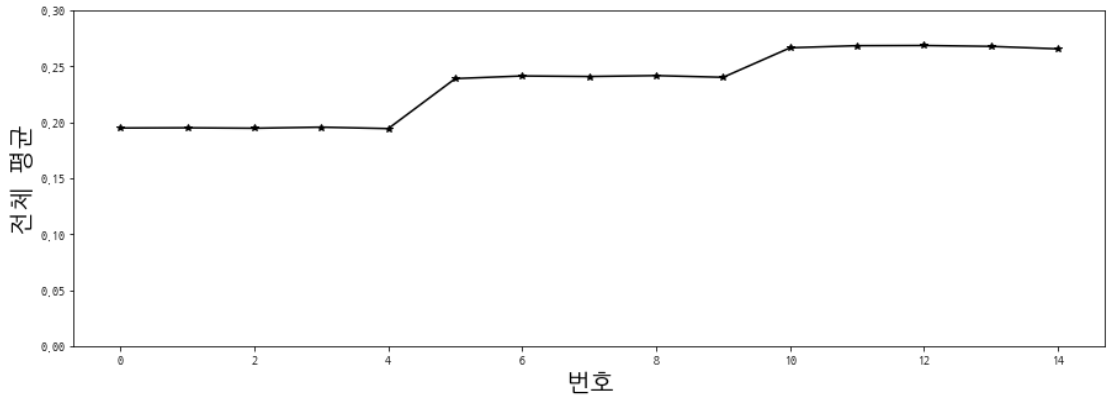


그림 13 전체 평균 그래프

따라서 표에 의하면 최적의 K값은 900이고, 사연의 개수는 7개 이다. 사실상 전체 평균에 가장 큰 영향을 미치는 것은 사연의 개수이다. 사연의 개수가 증가할수록 유사도 값의 평균이 증가는 것을 볼 수 있다.

(2) 유사한 사연 기반 노래 추천 시스템(모델2)

해당 모델은 1번 모델을 조금 변형시킨 모델이다. 1번 모델은 기존 사연 텍스트 내용은 전혀 반영이 되지 않는 모델이다. 이 점을 보완하기 위해서 2번 모델에서는 노래 추천 과정에서 기존 사연 텍스트 내용을 반영해보았다.

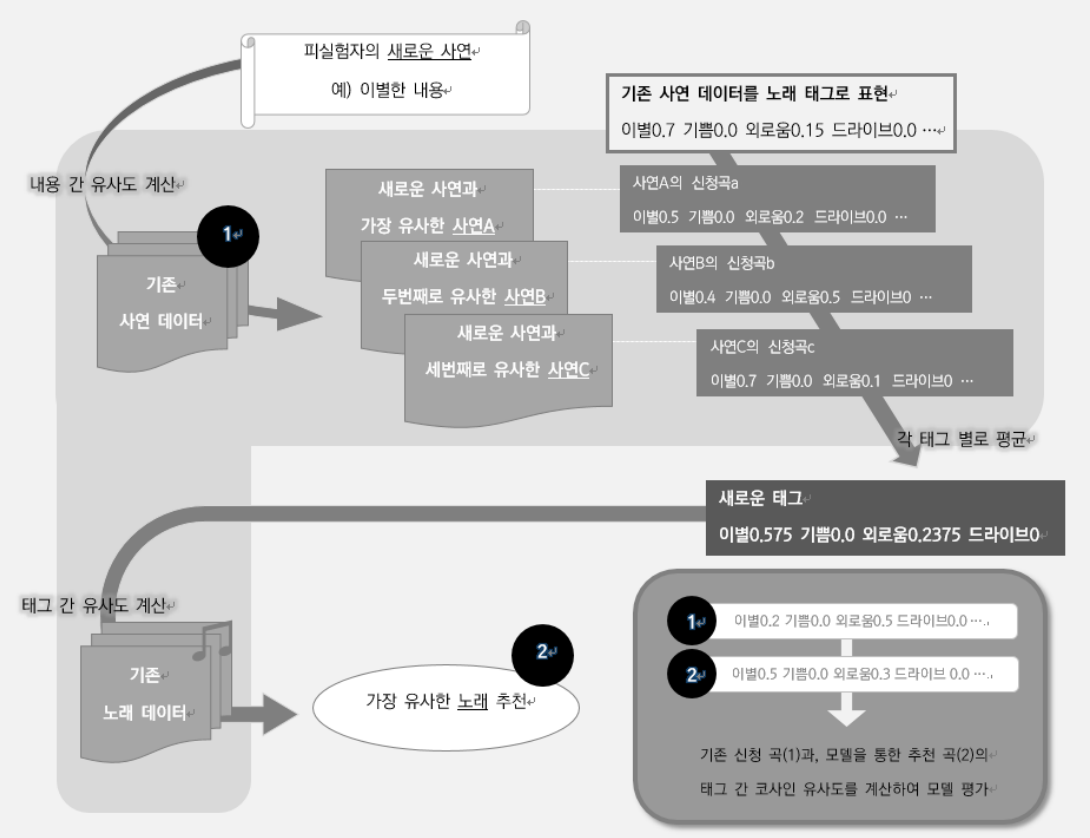


그림 14 유사한 사연 기반 노래 추천 시스템 구성도(2)

1번 모델과 1번부터 5번까지 과정과 7번 과정은 동일하다. 새로운 태그 벡터를 생성하는 6번 과정만 기존과 달라졌다.

6. 사연 텍스트 내용과 유사한 사연들 기반 새로운 노래 벡터 생성

기존에 새로운 노래 벡터는 유사한 사연들의 신청곡 태그만 사용하였다. 그러나 해당 모델에서는 기존 사연의 텍스트 내용도 반영하고자 한다. 그리하여 기존 사연 텍스트 내용을 노래 태그 단어로 표현한 새로운 사연X태그단어 행렬을 생성하였다. 즉, 사연 내용을 1727개의 태그 단어로 표현하는 것이다. 위의 행렬도 노래 행렬처럼 비율로 바꾸어 주었다. 따라서 사연 내용을 반영하기 위해서 새로운 노래 벡터를 생성할 때, 해당 정보도 포함시켜 태그별로 평균값을 구하였다.

8. 모델평가

모델의 성능을 평가하는 객관적 지표로는 사연의 기존 신청곡과 모델의 추천곡 간의 코사인 유사도 값으로 한다. 전체 1905개 사연의 신청곡과 추천곡 간의 각각 코사인 유사도 값을 구한 후 유사도 값의 전체 평균이 높을수록 좋은 모델이다. 따라서 해당 지표를 바탕으로 최적의 SVD 차원 수 K 와 사연 개수 N을 정하고자 한다. 그리하여 K값은 500에서 1300까지 200간격으로, N값은 3에서 7까지 2간격으로 총 15개의 경우의 수를 진행해보았고, 전체평균은 그 결과 값이다.

번호	N	K	전체 평균	번호	N	K	전체 평균	번호	N	K	전체 평균
0	3	500	0.1949	5	5	500	0.2390	10	7	500	0.2665

1	3	700	0.1950	6	5	700	0.2414	11	7	700	0.2684
2	3	900	0.1946	7	5	900	0.2409	12	7	900	**0.2686**
3	3	1100	0.1956	8	5	1100	0.2417	13	7	1100	0.2678
4	3	1300	0.1943	9	5	1300	0.2401	14	7	1300	0.2657

표 4 경우의 수와 결과 값

해당 결과를 그래프로 그리면 <그림 15>와 같다.

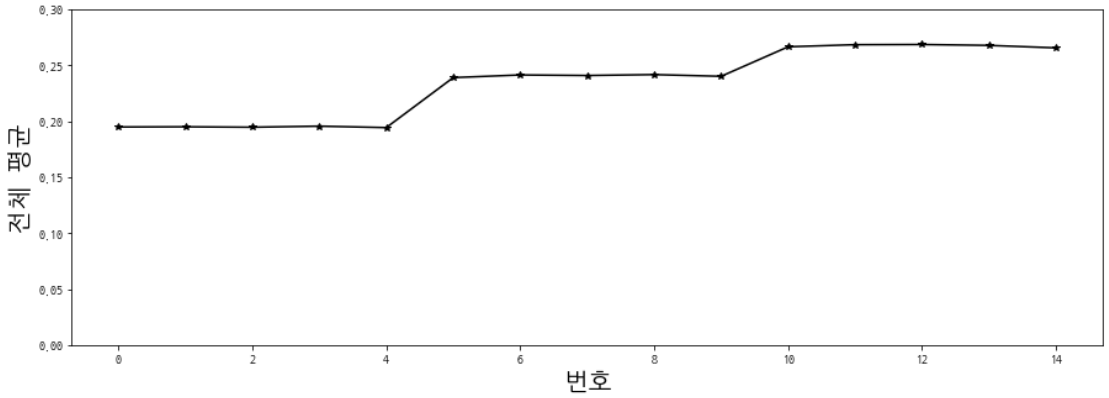


그림 15 전체 평균 그래프

모델1의 결과값인 <표 3>과 모델2의 결과값인 <표 4>를 비교해보면 전체 평균 값이 동일한 것을 알 수 있다. 즉, 사연의 내용을 반영해도 추천곡은 큰 차이가 없다는 것을 의미한다. 또는, 사연의 내용이 제대로 반영되지 못했다는 것을 뜻한다. 사연의 내용이 잘 반영되지 않은 이유는 다음과 같다고 볼 수 있다. 사연의 내용을 노래 태그 단어로 표현을 할 때, 사연에 쓰인 단어와 노래 태그가 겹치는 부분이 적기 때문이다. 따라서, 모델1과 모델2간에 큰 차이가 없기 때문에, 기존의 모델1을 사용하고자 한다.

2) 두번째 추천 시스템

(1) Word2Vec을 통한 사연과 노래 간 의미론적 유사도 기반의 음악 추천 모델

노래는 하나의 속성으로 표현하기 어렵다. 음정, 가사, 가수의 목소리, 장르, 사용된 악기 등 여러 가지 요소가 그 노래의 특징을 결정짓기 때문이다. 따라서 노래를 설명하는 방법으로 태그와 더불어 노래 데이터 내에 있는 가사, 장르, 노래 제목 등의 텍스트 정보를 추가하였다. [4]

1. 라디오 사연 형태소 분석

① 기존의 모델1과는 다르게, 사연 내용만 존재하는 데이터 중 사연 길이가 350자 이상인 4552개의 데이터를 대상으로 하였다.

② 형태소 분석 과정은 모델1에서의 형태소 분석 과정과 동일하게 수행하였다.

2. 노래 데이터 전처리

① 노래 태그 정보

- 전체 노래 데이터에 존재하는 속성으로, 고유 개수는 1,536개이다.
- 노래 당 태그 개수 분포(중복 포함)의 평균값은 15.12개, 중위 수는 9.0개로 나타났다. 가장 적은 1개의 태그를 가진 노래는 1,603곡이 있었고, 가장 많은 태그를 가진 곡은 3,661개의 태그를 가지고 있었는데, 이는 대중적인 노래일수록 태그의 길이가 길기 때문으로 보인다. 3~5개의 태그를 가지는 노래가 12,303곡으로 나타났다.
- 가장 많이 등장한 태그는 드라이브(17,421회), 밤(17,354회), 기분(16,546회), 기분전환(16,075회), 가요(15,624회), 발라드(13,852회), 힙(11,092회), 사랑(10,838회), 힙합(10,642회), 팝(10,476회) 순으로 나타났다.

② 노래 가사 정보

가사 데이터 수집 과정은 제 2절의 1) 노래 데이터 수집에 나와있으며, 전처리 전 노래 데이터의 개수는 한국어가 아니거나 가사가 미등록 된 노래를 포함하여 총 62,783곡이다. 한국어가 아닌 가사를 가진 노래를 제거한 이유는, 팝송을 제외한 국내 노래만을 추천 곡 대상으로 하기 때문이다.

(1) 가장 먼저 가사가 미등록 된 노래를 제거한다.

- 가사가 미등록 된 노래는 가사가 '[가사 준비중] 멜론 회원 여러분! 가사 등록을 기다리고 있어요' 라는 문구로 대체되어 있다.
- 문자열 포함여부를 확인하는 'contains()'를 사용해해당 문구를 포함한 행을 제외한 나머지 행 만을 선택하여 따로 저장한다.

(2) 다음으로 한국어가 아닌 가사를 제거한다.

- 모든 가사를 대상으로 Konlpy의 Twitter 형태소 분석기를 사용해 명사, 형용사, 또는 동사만을 추출하였을 때 길이가 10글자 미만인 것은 한국어 가사가 아니라고 판단하여 삭제하였다. 이 때 형태소 분석 결과를 사용하지 않았으며 한국어 가사 여부를 판단하는 데에 한글 형태소 분석기를 활용하였다.

(3) 위 과정을 거쳐 남은 가사를 대상으로 Konlpy의 Komoran 형태소 분석기를 사용해 명사, 형용사, 동사를 추출한다.

(4) 추출한 단어들을 출현 빈도 순으로 정렬 후, 출현 빈도 상위 100개 단어 중 불용어로 설정할 단어들을 선택했다. 감성이나 상황을 나타낸다고 보기 어렵지만 대부분에 가사에 등장한 단어 등을 불용어로 설정했다. 아래는 불용어로 설정한 단어들의 리스트이다.

- ['펼치', '모르', '사람', '마음', '시간', '생각', '오늘', '이렇', '그렇', '이제', '그러', '보이', '세상', '가슴', '하루', '모습', '위하', '느끼', '노래', '순간', '부르', '지나', '그때', '나오', '지금', '얘기', '처음', '미치', '남자', '여자', '바람', '친구', '변하', '조금', '만들', '소리', '거리', '머리', '기분', '시작', '내일', '이러', '어리', '어제', '대답', '나가', '동안', '영화', '다음', '이야기', '계절']

(5) Konlpy의 Komoran 형태소 분석기를 사용해, 불용어 리스트 내 단어를 제외한 동사, 명사, 형용사 중 길이가 2이상인 문자를 새로운 가사 데이터로 사용한다.

- 불용어 제거 전과 후

그림 16 가사의 불용어 제거 전과 후 단어 빈도

(6) 이렇게 처리된 최종 가사 데이터는 19,666곡을 대상으로 하며, 노래 당 가사 길이의 평균값은 101.61자, 중위 수는 87.0자의 분포를 보였다. 가사에 사용된 단어 수(하나의 가사 내 중복되는 단어는 하나만 남김)의 분포는 평균 65.58회, 중위 수 54.0회로 나타났다.

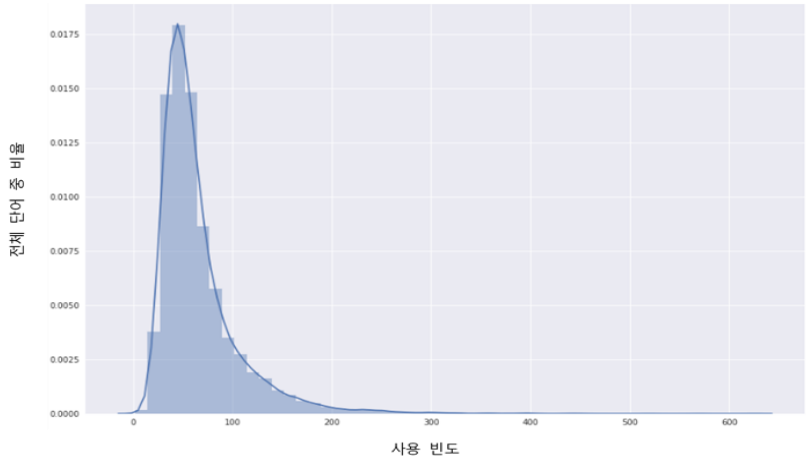


그림 17 가사 내 단어 별 사용 빈도 분포

③ 노래 장르 정보

- ['Foreign', 'Animation', 'New Age', 'Adult Contemporary', '동요', 'World', 'CCM', 'Blues', 'Instrumental', '워십', '찬송가', '기타']를 제외한 나머지 장르의 텍스트 자체를 사용하였다. 결과적으로 총 12,792개 곡에 대한 장르와, 39가지 장르 종류로 구성되었다.

④ 노래 제목 정보

- 제목은 노래의 내용을 함축적으로 표현하는 단어/어구/짧은 문장이다. 길이가 짧아 불용어 처리를 따로 하지 않았으며, 대부분의 노래를 포괄하는 속성이라는 점에서 추가하게 되었다. 총 34,612개 곡에 대해 제목 데이터가 존재하며, 총 9,669개의 단어로 구성되어 있다.

(1) 영어가 포함되어 있거나, 전체가 영어인 노래 제목들을 Google 스프레드시트 'GOOGLETRANSLATE' 함수를 사용해 처리하였다

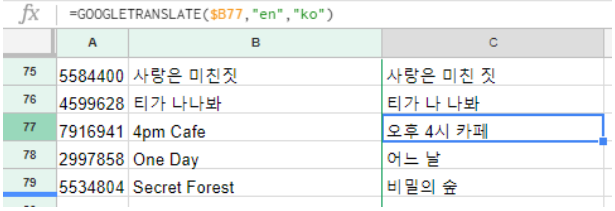


그림 18 구글 스프레드 시트를 통한 영어 노래 제목 처리

(2) 처리한 노래 제목들을 대상으로 Konlpy의 Twitter 형태소 분석기를 사용하여 2글자 이상의 동사, 명사, 형용사를 추출한 텍스트를 새로운 노래 제목 데이터로 사용하였다.

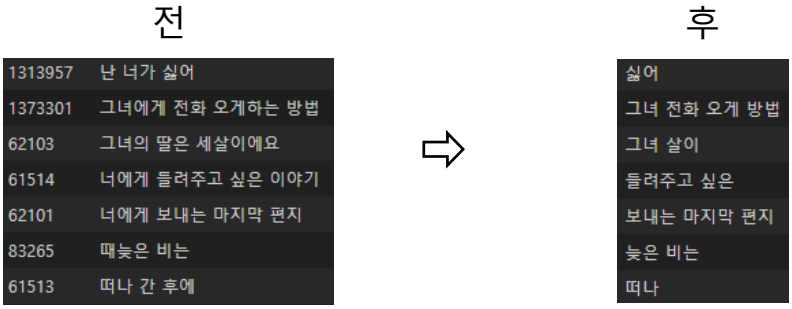


그림 19 처리 전과 후의 노래 제목

⑤ 텍스트 전처리가 끝난 사연 데이터와 노래 관련 데이터들의 Dictionary 형태 저장

- 데이터 프레임보다 용량이 작다는 점과, 사연ID와 다르게 불연속형의 Key값을 갖는 노래 데이터의 특성을 고려하여 { Key: Value } 형태에 특화 된 Dictionary 형태로 사연 데이터, 노래 태그/가사/장르/제목 데이터를 저장하였다.

- 사연 데이터는 사연ID(t1~t4451)와 사연 내용 문장이 짝을 이루는 dictionary 형태로 저장하였다.

- 노래 데이터는 노래ID(자연수 형태이며 불규칙적이다.)와 각 노래 정보가 짝을 이루는 dictionary로 태그, 가사, 장르, 노래 제목 별로 저장하였다.

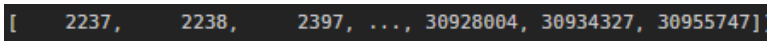


그림 20 불규칙한 형태의 노래 ID

3. 노래 정보와 사연 내용 통합

① Pandas.Dataframe의 data에 여러 Dictionary들을 리스트 형태로 함께 입력하고 NA값을 공백 처리하여 데이터프레임의 Index값이 노래ID 41,148개, 사연ID 4,552개 값을 갖고, 각 열이 각 Dictionary별 Value값을 갖는 데이터 프레임을 생성하였다.

② Index를 기준으로 열을 통합한 문자열(문장)을 생성한다

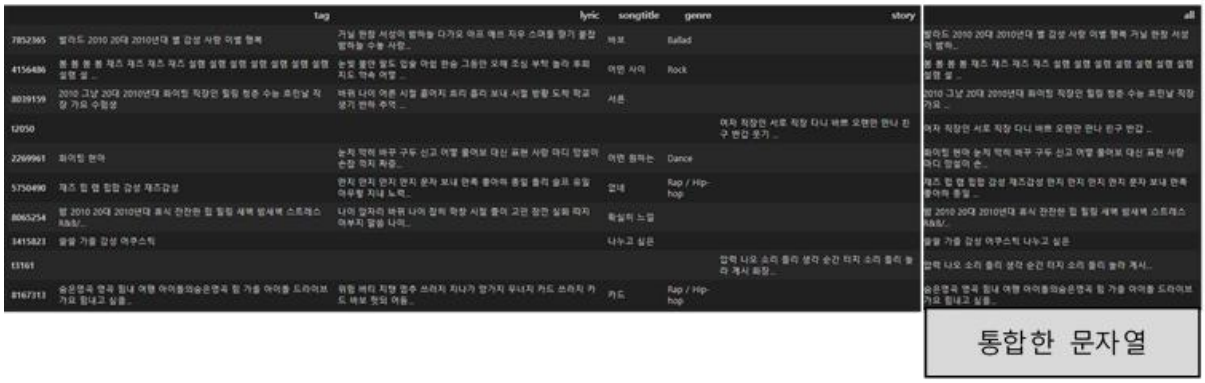


그림 21 사연, 노래 태그/가사/장르/제목을 통합한 데이터프레임

③ 통합한 문자열을 list형태로 별도로 저장한다.

통합한 문자열 내 사용 빈도(하나의 문자열 내에서 중복 단어 제거 후) 기준으로 가장 많이 쓰인 단어는 사랑(17,038), 가요(10,265), 밤(10,121), 새벽(9,550), 기분(9,087), 기분전환(8,325), 휴식(7,967), 발라드(7,197), 드라이브(7,009), 추억(6,975), 팝(6,740), 20대(6,726), 행복(6,653) 순으로 나타났다.

4. Word2Vec을 통한 워드 임베딩 모델 생성

- word2vec은 2013년 구글에서 발표된 연구로, Tomas Mikolov을 필두로 여러 연구자들이 모여서 만든 Continuous Word Embedding 학습 모형이다. 다차원의 공간에 단어를 매핑 가능하게 함으로써 각 단어들 간의 유사도를 측정할 수 있으며, 여러 단어에 대해 다룰 때에도 수치적으로 다루기 쉽다는 장점이 있다. 또한, 단어의 의미 자체가 벡터로 수치화 되어 있기 때문에, 벡터 연산을 통해 의미에 대한 추론이 가능하다.

- Word2vec 모델에서 단어 매핑 방법은 크게 CBOW(Continuous Bag-of-Words)와 Skip-gram 이 있다. Skip-gram 모델의 경우 현재 주어진 단어 하나를 가지고 주위에 등장하는 나머지 몇 가지의 단어들의 등장 여부를 유추하는 것이다. 이 때 예측하는 단어들의 경우 현재 단어 주위에서 샘플링 하는데, '가까이 위치해 있는 단어일수록 현재 단어와 관련이 더 많은 단어일 것이다' 라는 가정을 기반으로 멀리 떨어져 있는 단어일수록 낮은 확률로 유추한다.

- 빠른 속도가 Word2Vec을 워드 임베딩 방법으로 채택한 가장 큰 이유이다. 다음으로 등장하는 간단한 모델 테스트를 바탕으로 Skip-gram을 사용하였다.

① Word2Vec 모델 파라미터 설정을 위한 테스트

파라미터	설정 내용	테스트 값
Size	차원 수	50, 200, 300
Min_count	단어의 출현 빈도 최소 값	10, 100
sg	CBOW 또는 skip gram	0 1

표 5 Word2Vec 모델 테스트를 위한 파라미터 조정

- 모델을 테스트 할 수 있는 데이터 형태가 아니기 때문에, 특정 단어를 입력했을 때 모델에서 출력되는 단어를 확인하여 파라미터를 정하고자 했다.

- 총 12가지 모델을 대상으로 most_similar()을 이용해, 테스트를 위해 Word2Vec 모델 생성 시 이용한 문자열에서 가장 많이 등장한 단어 중 '밤', '추억'을 입력해 보았다.

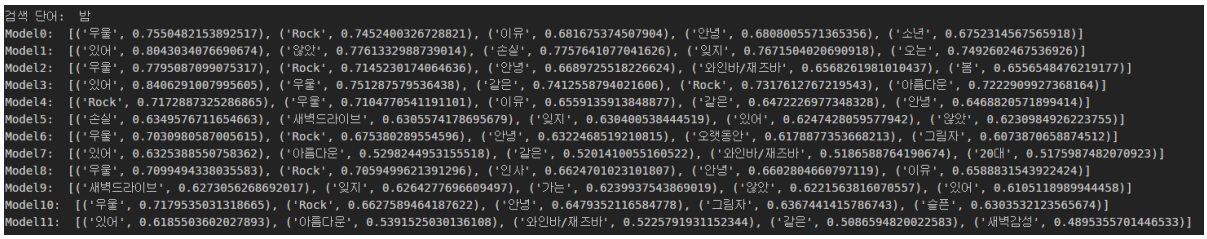


그림 22 12가지 Word2Vec 모델의 '밤'과 가장 유사한 단어 각각 5가지를 출력

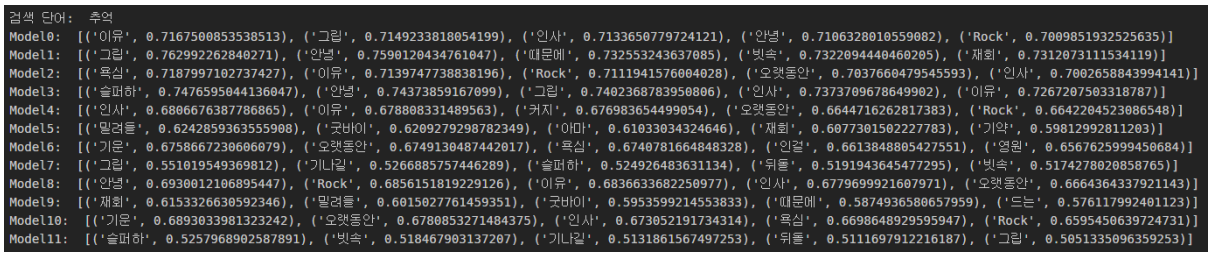


그림 23 12가지 Word2Vec 모델의 '추억'과 가장 유사한 단어 각각 5가지를 출력

- 입력한 단어와 각 모델의 출력 단어를 비교해 봤을 때, 다른 모델들에 비해 9번 모델의 성능이 우수하다고 판단하였고 이를 워드 임베딩을 위한 최종 모델로 결정하였다.

- 9번 모델: size = 300, min_count = 10, sg = 1 (300차원, 문서 간 출현 빈도 10번 이상, skip-gram 사용)

5. 워드 임베딩 기반의 사연에 어울리는 노래 추천

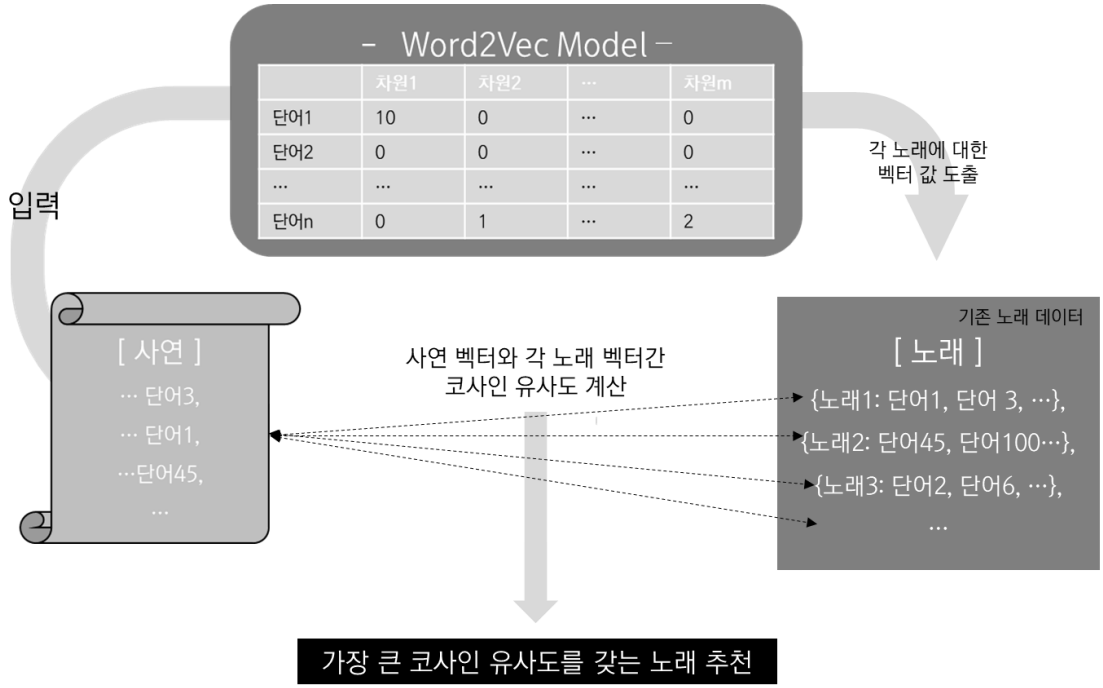


그림 24 Word2Vec을 통한 사연과 노래 간 의미론적 유사도 기반의 음악 추천 알고리즘 도식화

- 모델의 입력 값은 문장 형태의 사연으로, 직접 입력하는 형식이다. 출력 값은 추천 곡에 대한 간략한 정보(가수 이름 – 노래 제목)이다.

① 문장 형태의 사연을 입력 받는다.

그림 25 사연 입력 방법

② 입력 받은 사연에 대한 형태소 분석을 진행한다. 이 때 형태소 분석은 기존 사연에 적용했던 것과 동일하게 수행한다.

③ 형태소 분석 후의 단어들로 이루어진 사연 문장을 임베딩 모델 내의 단어들과 비교하고 겹치는 단어만 추출한 뒤, 문장 형태로 만든다.

- 이 때 문장 형태는 단어가 리스트 형태로 나열되고, 하나로 묶인

  [['단어1'], ['단어2'], ['단어3'], … ,['단어n']] 의 구조를 의미한다.

④ 새롭게 만든 사연 문장의 임베딩 벡터 평균 값과 전체 노래 데이터(40,439개)의 임베딩 벡터 평균 값의 코사인 유사도를 비교하여, 가장 높은 코사인 유사도를 갖는 노래를 추천 곡으로 선정한다.

⑤ 추천 곡의 ID를 Key값으로 사용해 노래 정보 Dictionary에서 '노래 제목 – 가수' 형태의 문자열을 출력 값으로 불러온다.

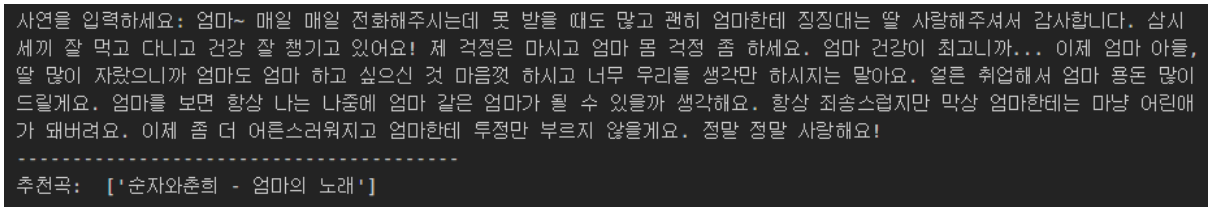


그림 26 입력된 사연과 추천 모델을 통해 출력된 추천 곡

IV. 활용 방안

1) 인공지능 스피커에 대한 소비자의 인식

한국소비자원에서 소비자 인식도를 조사한 설문조사를 바탕으로 쓰여진 보고서이다. 설문조사에 따르면, 구매자들은 일반적으로 음성인식 스피커에서 기기와의 일상화 라는 특성을 기대하는 것으로 보여진다. 이에 더해 자주 사용하는 음성인식 스피커의 기능에는 음악 재생이 있다. 이처럼 일상적인 대화를 기반으로 노래를 추천해주는 서비스를 제공한다면, 음성인식 스피커 이용자에게 높은 호응을 얻을 수 있을 것으로 기대된다. [9]

(1) 조사대상 및 방법

구분	내용
① 조사 목적	- 음성인식 스피커 이용실태 현황 및 소비자 만족도 조사
② 조사 대상	- 국내외 음성인식 스피커*를 사용하고 있는 이용자 300명 * 기가지니(KT), 누구(SKT), 에코(아마존), 홈(구글)
③ 조사 방법	- 온라인 설문조사기관 이용, 신뢰도 95%, 표본오차 ±5.65% (6% 이하의 대표성 있는 신뢰수준)
④ 조사 기간	17.6.21. ~ 17.6.30.
⑤ 조사 내용 **1) 사용자 이용실태 조사**	- 구매 동기, 사용기간 - 자주 사용하는 기능 및 전반적인 제품 만족도 수준

(2) 응답자 현황

◦ 음성인식 스피커를 사용하는 이용자 300명을 대상으로 온라인 설문조사를 실시함.

구분		빈도(명)	구성비(%)	구분		빈도(명)	구성비(%)
성별	남	114	38%	지역	서울	119	39.7%
	여	186	62%		경기	55	18.3%
연령	20대 이하	74	24.7%		인천	20	6.7%
	30대	143	47.7%		강원	3	1%
	40대	59	19.6%		충청	17	5.7%

	50대 이상	24	8%		전라	19	6.3%
					경상	65	21.7%
					제주	2	0.6%

□ (**구매 이전 기대한 제품의 특성**)

- 응답자의 46.3%(139명)이 '쉽고 간편한 음성인식 시스템 이용환경" 이라고 응답

△ **기기와의 일상 대화(23%, 69명)**, △ 일정 및 스케쥴 관리(13%, 39명), △ 가정용 사물인터넷 기능(12.4%, 37명), △ 생활편의(날씨, 교통정보 등) 순

[표] 구매 이전 기대한 음성인식 스피커의 특징

[단위 : %(명)]

구분	인공지능에 대한 호기심	지인의 소개	음성인식 기능 필요	선물용	기타
응답자(300명)	67.7%(203명)	20.0%(60명)	6.0%(18명)	5.0%(15명)	1.3%(4명)

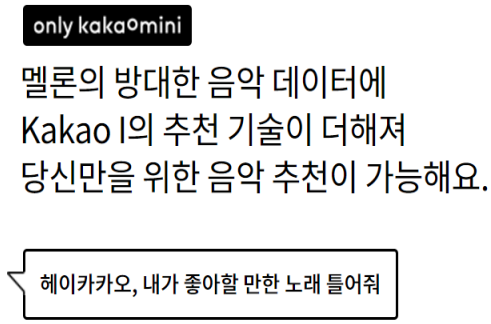
[표] 자주 이용하는 음성인식 스피커의 기능(복수응답)

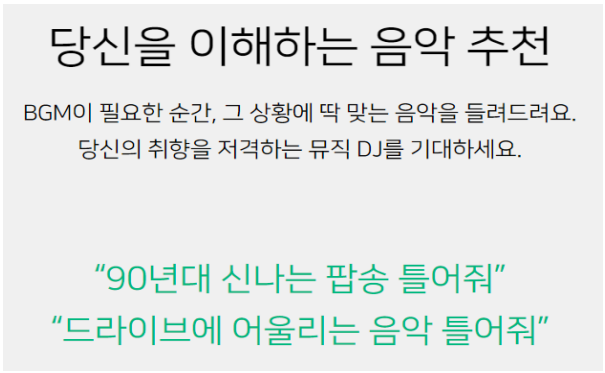
[단위 : %(명)]

항목	응답자(300명)
① 음악재생	**71.3%(214명)**
② 날씨, 교통정보	41.0%(123명)
③ 인터넷 정보검색	40.3%(121명)
④ 타이머스케쥴 관리	35.7%(107명)
⑤ 라디오,뉴스 팟 캐스트	31.0%(93명)
⑥ IPTV 연동 서비스	29.0%(87명)
⑦ 쇼핑 및 음식주문	17.7%(53명)
⑧ 가정용 사물인터넷	14.3%(43명)
⑨ 일상대화	14.0%(42명)
⑩ 기타 편의기능	5.7%(17명)

설문에 따르면, 인공지능 스피커를 구매하기 이전 가장 기대했던 기능은 '기기와의 일상 대화' 인 것으로 나타났다. 그리고 인공지능 스피커 사용자가 가장 많이 이용하는 기능은 음악재생 기능이다. 즉, 인공지능 스피커 소비자들은 일상적인 대화기능과 음악재생 기능에 관심이 있다는 것을 파악할 수 있다. 따라서 일상적인 대화를 바탕으로 노래를 추천해주는 서비스를 제공한다면, 이용자들에게 높은 호응을 얻을 수 있을 것으로 기대된다.

2) 기존 음성 인식 스피커의 음악 추천 시스템





KaKao MINI _ KAKAO

- 음악 플랫폼인 멜론과 결합하여 음악 감상 히스토리를 기반으로 노래를 추천
- 추천 시스템에서 일반적으로 사용되는 협업필터링을 이용하여 추천

Clova _ Naver

- 추천 시스템에서 일반적으로 사용되는 협업필터링을 이용하여 추천
- 콘텐츠 기반 필터링을 통해 사용자, 곡의 기타 정보 등을 입력해 특징을 뽑아내는 방식과 딥러닝을 이용하여 음악을 추천
- 딥러닝을 기반으로 네이버 뮤직에서 설정한 해시태그를 이용하여 노래 추천

GIGA Genie _ KT

- 음악 플랫폼 Genie와 결합하여 사용자별로 음악 감상 패턴 및 노래의 장르와 아티스트, 발매일 등을 분석해 협업필터링을 통해 추천

3) 유사 서비스의 음악 추천시스템 현황

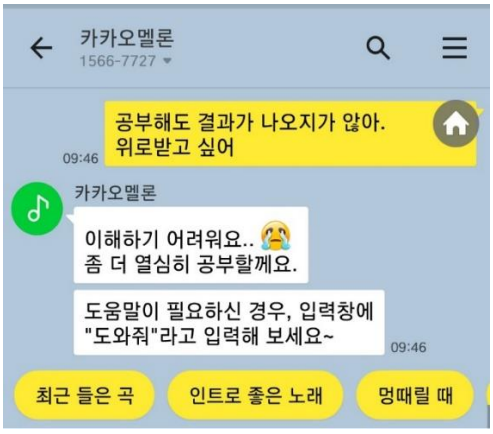


그림 27 Kakao 플러스 친구 - 멜론 로니

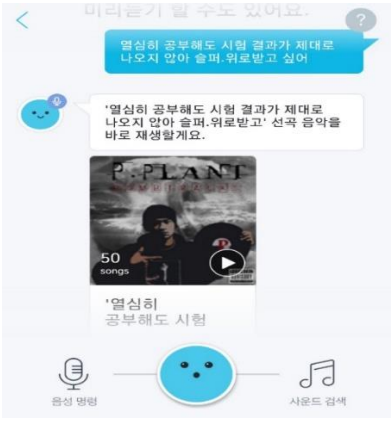


그림 28 Genie 뮤직 - Genius

멜론과 지니는 현재 카카오톡 플러스 친구와 지니 뮤직 애플리케이션을 통해 인공지능 기반 음

악 추천 서비스를 제공하고 있다. 멜론은 '로니'라는 이름으로, 지니는 'Genius'라는 이름으로 서비스를 현재 제공중이다. 각각의 플랫폼에 접근하여 원하는 상황 또는 감정을 입력하면 그에 걸맞은 노래 목록을 추천해준다. '비오는날', '우울', '90년대 신나는 노래'와 같이 문구를 전송하면 추천 시스템에 의해 노래를 추천해준다. 그러나 두 서비스는 텍스트 길이가 조금만 길어지면 이를 인식하지 못하고, 노래 추천을 제대로 해주지 못한다. 텍스트 길이가 우리가 앞서 사용했던 노래 데이터의 '태그'와 같은 내용이 아니라면 추천 시스템이 제대로 작동하지 않게 된다. <그림27>과 <그림28>은 텍스트 내용이 길어질 경우 노래 추천을 해주지 못하는 사례를 보여주고 있다.

협업 필터링의 경우 유사한 취향의 사용자는 서로 비슷한 것을 선호할 것이라고 전제하고 있다. 하지만, 유사한 사용자라고 하더라도 기분이나 날씨, 사적인 경험에 의해 갑작스럽게 취향이 달라 질 수 있다. 알고리즘이 딛고 있는 전제 자체가 무너지는 상황이 발생할 수 있다는 것이다. 또한, 사용자의 기존 스트리밍 기록 내역이 없으면 추천할 수 없는 콜드스타트 문제를 갖는다. 이에 더해 사용자나 곡이 늘어날수록 추천의 정확도가 떨어진다. 특히, 앨범은 다르지만 같은 곡들이 중복되거나, 악의적으로 해당 곡에 나쁜 평가 점수를 부여하는 경우, 신곡에 대한 로그가 없어 유사성을 분석할 방법이 없다는 한계를 가진다.

기존의 음성인식 스피커의 경우 각 회사들이 가지고 있는 음악 플랫폼을 이용하여 노래를 추천하는 방식이다. KAKAO의 경우 Melon, Naver의 경우 Naver Music, KT의 경우 Genie을 이용하여 사용자들의 스트리밍 기록을 이용하여 추천하는 협업 필터링이 일반적이다.

본 프로젝트에서는 라디오 사연을 적은 신청자의 개인적인 감정/상황 정보의 다른 사연과의 유사도를 통해 사연이 신청한 각각의 노래 정보를 이용하여 노래를 추천하는 추천 시스템을 소개하였다. 이를 위해 본 프로젝트에서는 다양한 라디오 사연 데이터가 수집을 위해 KBS, MBC, SBS에서 다양한 시간대의 라디오 사연을 수집하였다. 추천의 근거가 되는 노래 정보인 노래 태그, 가사, 인기도(좋아요) 값, 장르를 음악 플랫폼인 Mnet, Genie, Melon에서 수집하였다.

기존의 음성인식 스피커와 다르게 본 프로젝트의 경우 Mnet, Genie, Melon 세 곳의 음악 플랫폼에서 태그 및 노래 정보를 크롤링하여 진행하였다는 다양한 음악 플랫폼에서 정보를 얻어와 더욱 다양한 감정/상황을 표현할 수 있다는 점에서 의의를 가진다. 사용자의 과거 스트리밍 기록 없이도 추천할 수 있는 방법을 사용하기 때문에 기존의 음악 추천 시스템이 갖고 있는 콜드 스타트 문제를 해결 할 수 있을 것이다. 또한, 문서 유사도 기반 음악추천시스템은 실제 음원 서비스와 연동시켜 실험 및 평가가 이루어진다면 다양한 텍스트를 기반으로 음악을 추천할 수 있는 앱의 가능성을 제공할 수 있을 것으로 보여진다.

V 결 론

추천 시스템(Recommendation system)은 사용자 정보를 기반으로 고객의 심리와 선호도에 맞게 정보를 필터링하여 제공하는 시스템으로 정의할 수 있다. 방대한 정보에 비해 소비자가 접하게 되는 부분은 다소 한정적이기 때문에 추천 시스템이 갖는 의미는 더욱 커지고 있다. 본 프로젝트 라디오 사연 분석을 통해 사연자의 감정과 상황에 따른 음악을 추천해 주는 알고리즘을 구현하였

다.

텍스트 기반 음악 추천 시스템에 대한 연구는 이미 여러가지가 진행된 바가 있다. 하지만 기존의 연구들은 한계점이 존재한다. 텍스트 분석 이후 추천되는 노래가 한정적이거나, 모델에 사용되는 데이터를 구축하는데 너무 오랜 시간이 걸리고, 노래를 설명하는 태그가 보편적이지 않거나 그리고 콜드 스타트의 문제점을 가지고 있다.

본 프로젝트에서 우리의 모델은 이 한계점을 극복하고자 했다. 텍스트의 감정과 상황을 분석하여 노래를 추천해주는 시스템에 대해 최종적으로 2개의 모델을 제안했는데, 첫 번째는 유사한 감정과 상황을 가진 텍스트를 찾아 음악을 추천해주는 시스템이고, 두번째는 Word2vec 모델을 이용하여 텍스트와 노래 간의 의미적인 유사성을 측정하여 음악을 추천해주는 시스템이다. 텍스트와 노래가 쌍으로 이루어진 라디오 사연을 수집함으로써 데이터 수집에 큰 시간을 소요하지 않았다. 그리고 유사한 사연을 찾은 이후, 노래 추천 방식을 새롭게 발전 시켜 사연에서 언급되지 않는 노래도 추천될 수 있도록 하였다. 또한, 노래 정보 생성시 개인이 지정한 태그만을 사용하지 않고 노래의 가사, 제목도 포함시켜 노래 정보의 개인적인 주관성을 극복하고자 하였다. 두번째 모델을 통해서는 신청곡이 없는 일반 텍스트를 이용해서도 비슷한 내용과 감정을 가진 노래를 추천이 가능하다.

첫번째 모델은 성능 평가하기 위한 객관적인 지표로 기존 사연의 신청곡과 모델의 추천곡 간의 유사도 값을 이용하였다. 그러나 두번째 모델은 모델 성능을 평가하기 위한 객관적인 지표가 없다는 것이 한계로 존재한다. 따라서, 첫 번째 모델은 사연과 신청 곡이 쌍으로 존재하는 'case-by-case'식의 형태로 인해, 추천의 정확도나 만족도 측면은 상대적으로 높을 수 있으나, 모델 내에 없는 사연의 내용, 형태 등에 대해서는 다소 약한 모습을 보인다. 두 번째 모델은 사연의 벡터 값과 노래 데이터셋의 벡터 값들을 비교함으로써 앞의 모델보다 확장성 측면에서는 우수하지만, 정확도나 모델 평가 측면에서는 어려움이 있다. 따라서 자체 평가가 가능한 첫 번째 모델에 두 번째 모델을 사용해 신청 곡 부분을 보완할 수 있다. 결론적으로 두개의 모델은 서로 다른 장점과 단점을 가지고 있기 때문에, 우리는 두개의 모델을 모두 제안하는 바이다.

우리의 모델은 인공지능 스피커의 서비스로 활용할 가능성이 있다. 인공지능 스피커에 대한 소비자의 인식에 따르면, 구매자들은 일반적으로 음성인식 스피커에서 '기기와의 일상화' 라는 특성을 기대하는 것으로 보여지며, 자주 사용하는 음성인식 스피커의 기능에는 음악 재생이 있다. 이처럼 일상적인 대화를 기반으로 노래를 추천해주는 서비스를 제공한다면, 음성인식 스피커 이용자에게 높은 호응을 얻을 수 있을 것으로 기대가 된다. [9] 즉, 우리의 모델을 적용함으로써 일상적인 대화 내용에 걸맞은 노래를 추천해 주는 서비스를 제공한다면 높은 호응을 얻을 수 있을 것이다.

기존의 음성인식 스피커의 경우 각 회사들이 가지고 있는 음악 플랫폼을 이용하여 노래를 추

천하는 방식이다. 그러나 본 프로젝트의 경우 Mnet, Genie, Melon 세 곳의 음악 플랫폼에서 태그 및 노래 정보를 크롤링하여 진행하였다. 다양한 음악 플랫폼에서 정보를 얻어와 더욱 한 곳의 데이터를 이용하는 것보다 풍부한 감정/상황을 표현할 수 있다는 점에서 의의를 가진다. 그리고 사용자의 과거 스트리밍 기록 없이도 추천할 수 있는 방법을 사용하기 때문에 기존의 음악 추천 시스템이 갖고 있는 콜드 스타트 문제를 해결 할 수 있을 것이다. 또한, 문서 유사도 기반 음악추천 시스템은 실제 음원 서비스와 연동시켜 실험 및 평가가 이루어진다면 텍스트를 기반으로 음악을 추천할 수 있는 앱의 가능성을 제공할 수 있을 것으로 보여진다.

참고문헌

[1] 유은순, 최건희, 김승훈 (2015). TF-IDF와 소설 텍스트의 구조를 이용한 주제어 추출 연구. 한국컴퓨터 정보학회논문지, 20(2).

[2] 이명아. (2012). *라디오 사연 분석을 통한 음악 추천시스템*(공학석사). 서울대학교 대학원 융합과학기술대학원 디지털정보융합학과

[3] 이성직, 김한준 (2009). TF-IDF의 변형을 이용한 전자뉴스에서의 키워드 추출 기법. 한국전자거래학회지, 14(4).

[4] Bingjun Zhang, Jialie Shen, Qiaoliang Xiang, Ye Wang. CompositeMap: a Novel Framework for Music Similarity Measure . n.p.: School of Computing, National University of Singapore, 2009.

[5] Chelsea Boling and Kumer Das. Article: Reducing Dimensionality of Text Documents using Latent Semantic Analysis. International Journal of Computer Applications 112(5):9-12, February 2015

[6] Linas Baltrunas, Marius Kaminskas, Bernd Ludwig, Omar Moling, InCarMusic: Context-Aware Music Recommendations in a Car (2011)

[7] Paul Lamere, Social Tagging and Music Information Retrieval (2008)

[8] Rui Cai, Chao Zhang, Chong Wang, Lei Zhang, Wei-Ying Ma, MusicSense: Contextual Music Recommendation using Emotional Allocation Modeling (2007)

[9] Anon, (2018). [online] Available at: https://www.kca.go.kr/brd/m_32/view.do?seq=2305 [Accessed 8 Sep. 2017].