2025 한양대학교

SW/ICT/AI 종합학술대회

[분야 : 404 Session Not Found]

제목: 자연어 기반 음악 유사도 분석 프로젝트

최민호(2021095441)

2025. 11. 26

한양대학교 ERICA 소프트웨어융합대학 / 한양대학교 ERICA SW중심대학사업단

연구 배경

❖현재 음악 검색 (ex. Shazam)

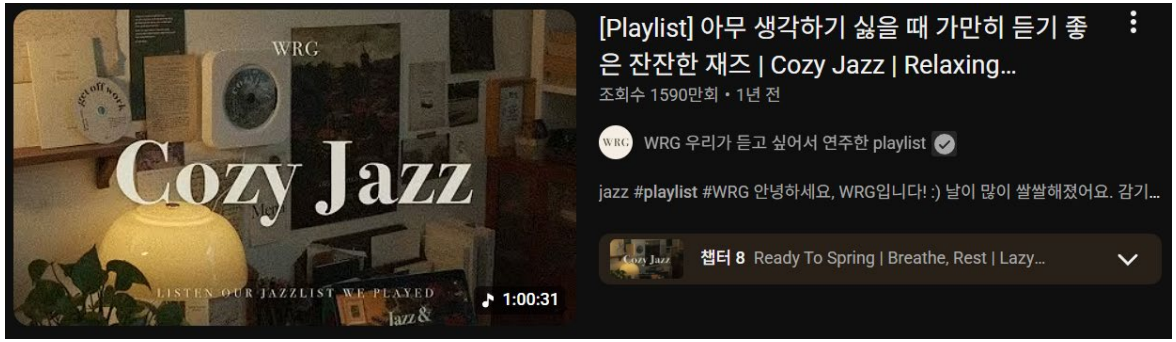
➢주변의 음악이 들리고 “이 노래 제목이 뭐지?” -> identification (식별)

➢이미 존재하는 오디오의 fingerprint와 일치하는 음악을 찾는 검색 방식

❖우리가 원하는 검색

➢Context(맥락) & Vibe(분위기) 중심의 검색

➢"이런 분위기의 곡 더 없나?", "공부할 때 듣기 좋은 차분한 곡 없나?"



❖지금의 한계

➢“새벽감성”등 이러한 감성 제목의 플레이리스트가 인기지만 사람이 수동으로 만든 플레이리스트이다

➢나만의 구체적이고 즉각적인 니즈를 반영하기 어렵다

문제 제기

❖사용자의 언어(Language)와 오디오 신호(Signal) 사이의 장벽

❖1. 기존 검색 기술의 딜레마

- ➢메타데이터(Tag) 검색: "차분한", "공부" 태그가 안 달린 음원은 검색되지 않음 (데이터의 사각지대).
- ➢오디오 신호 분석(VGGish): 음악적 특징(주파수, 리듬)은 잘 알지만, "공부할 때 좋다"는 추상적 개념을 이해하지 못함.

❖2. 새로운 대안

- ➢텍스트– 오디오 모델의 등장
- ➢CLAP 은 일반 사용자의 추상적 느낌을 잘 이해하고 검색할 수 있을까?
- ➢LLM을 사용하듯 같이 음악도 검색할 수 있으면 좋겠다

실험 및 비교 분석 (VGGish vs CLAP)

❖1. 실험환경 및 데이터셋

➢데이터셋 FMA-small (Free Music Archive) datasets

❖2. 비교 실험

➢Step 1 : 데이터 전처리 및 vector DB 구축

- FMA 데이터셋을 VGG 와 CLAP 모델을 통해 벡터 공간으로 변환하여 chroma DB 에 저장
- CLAP Embedding: 오디오와 텍스트를 같은 잠재 공간의 벡터로 임베딩해 서로 의미적으로 정렬되도록 학습된 표현.
- VGGish Embedding: 오디오를 로그-멜 스펙트로그램 기반 128차원 고수준 특징 벡터로 임베딩한 표현

➢Step 2: 동일한 타겟 음악(Query)을 두 가지 방식으로 검색(Retrieval)하여 ChromaDB의 결과를 비교.

- Case A: audio-based Retrieval
  - √ 입력 오디오를 VGGish로 임베딩 -> DB 내 벡터들과 유사도 계산
  - √ 의도 : 이 오디오 파일의 특징 벡터와 물리적으로 비슷한 특징을 찾자
- Case B:Text-based Retrieval
  - √ 입력 오디오의 특징을 텍스트로 묘사 -> CLAP 로 텍스트 임베딩 -> DB 내 CLAP 벡터들과 유사도 계산
  - √ 의도 : 이 텍스트 설명에 부합하는 특징을 가진 오디오를 찾자

실험 결과 비교 및 분석

❖검색 결과 비교

➢1. 검색 양상에 따른 차이

- VGGish(audio-to-audio): 거시적 분위기를 찾는다
  - √ 쿼리 곡과 전반적인 음색, 악기 밀도, 에너지가 유사한 곡들이 상위권에 랭크
  - √ 구체적인 악기 소리가 다르더라도 전체적인 분위기가 비슷하면 높은 유사도
  - √ 한계 : 특정 소리 객체의 유무보다는 파형의 모양에 집중하여 의미적 검색에 한계가 존재한다
- CLAP (Text-to-audio): 의미론적 의도를 찾는다
  - √ "Bell-like sounds with fast beat“ 와 같은 구체적 텍스트 입력시 해당 음향 객체와 (bell-like sound) 와 템포를 잘 포착함
  - √ Text-audio 간 공간이 효과적으로 정렬되어 있어서 묘사된 특징을 잘 찾음
  - √ 한계: 사용자의 의도(Intent)를 반영하기에 적합하나, 정교한 쿼리가 필수적임.

제안 시스템

❖**제안 : 사용자의 모호한 자연어 '감성'을 LLM을 통해 구체적 묘사를 통해 쿼리 최적화를 통한 음향 검색**

❖**시스템 파이프라인 :**

- Step 1: 자연어 입력(사용자의 의도):
  - User: "비오는 날 카페에서 듣기 좋은 곡"
- Step 2: LLM을 통한 자연어 재구성 (의도 -> 묘사 변환)
  - LLM이 사용자의 입력을 CLAP 모델이 학습된 데이터셋(캡션)의 특성에 맞춰 프롬프트 엔지니어링 진행
- Step 3: CLAP 을 통한 검색
  - 생성된 '묘사 문장'을 임베딩 하여 ChromaDB 검색
  - 묘사된 내용과 일치하는 후보군 추출
- Step 4: VGGish 를 통한 재검색
  - 사용자가 평소 좋아하는 샘플 곡 혹은 input으로 같이 넣은 샘플 이 있다면 그 음원을 기준으로 CLAP 을 통해 검색된 후보군들을 파형을 분석해 유사한 순서로 정렬
- Step 5: 최종 제공
  - 사용자 의도를 정확히 묘사하며 듣기 좋은 음악 리스트 제공

실제 검색 결과 샘플

- **Query: “비오는 날 어울리는 잔잔한 피아노 음악”**
- **LLM 재구성 : “Soft solo piano playing a slow melancholic melody, accompanied by the gentle sound of rain falling and light ambient reverb.”**
- **검색 top-3 결과 청취시 쿼리의 분위기(Vibe)와 실제 음원의 청각적 느낌이 직관적으로 일치함을 확인.**





References:

[1] Seonghyeon Go, "Real-World Music Plagiarism Detection With Music Segment Transcription System," arXiv:2509.08282, 2025.

[2] Primus, P. & Widmer, G., "Fusing Audio and Metadata Embeddings Improves Language-based Audio Retrieval," arXiv:2406.15897, 2024.

[3] Elizalde, B., Alwan, A., & Raj, B., "CLAP: Learning Audio Concepts From Natural Language Supervision," arXiv:2206.04769, 2022.

[4] Hershey, S., et al., "CNN Architectures for Large-Scale Audio Classification," in Proc. IEEE ICASSP, 2017. (VGGish)

[5] Doh, S., Lee, M., Jeong, D., & Nam, J., "Enriching Music Descriptions with a Finetuned-LLM and Metadata for Text-to-Music Retrieval," arXiv:2410.03264, 2024.

결론 및 기대 효과

❖하이브리드 아키텍쳐 구상

- ➢VGGish 의 신호 분석 능력과 CLAP의 의미 이해 능력을 결합 하여 단일 모델의 한계를 상호 보완

❖LLM 을 통한 의미론적 간극해소

- ➢기존 오디오 검색의 문제였던 사용자의 추상적 언어와 검색 모델의 구체적 요구 사이의 불일치를 LLM 기반의 쿼리 변환, 재구성 능력으로 해결

❖기대 효과 : 사용자 경험의 발전

- ➢오디오 검색 경험의 발전
  - 이제 더 이상 이 곡과 비슷한 음악을 찾는다는 글을 작성하지 않아도 됨
  - 누구나 일상적인 대화체로 정교한 음원 탐색이 가능해짐
- ➢정적 플레이리스트에서 동적 큐레이션으로
  - 유튜브에서 타인이 만든 플레이리스트를 헤매지 않아도 된다
  - AI 가 내가 서술한 감정, 느낌에 맞춰 실시간으로 생성하는 플레이리스트 구현

❖최종 결론

- ➢기존의 shazam 은 이미 세상에 존재하는 음악의 이름을 찾는 확인이었다면 본 시스템은 타인의 제공에 의존하는 수동적 추천이 아닌 지금 당장의 구체적 요구를 반영하는 능동적 검색 방법 구현