CLAP With Me

Step by Step Semantic Search on Audio Sources

Content

Introduction
Who/What/Why/How

Contrastive What Now?
Brief Overview of CLAP

Building The App
Recording/Searching/Classifying/Similar Sounds

Closing
Summary/Next Steps

Introduction

About Me

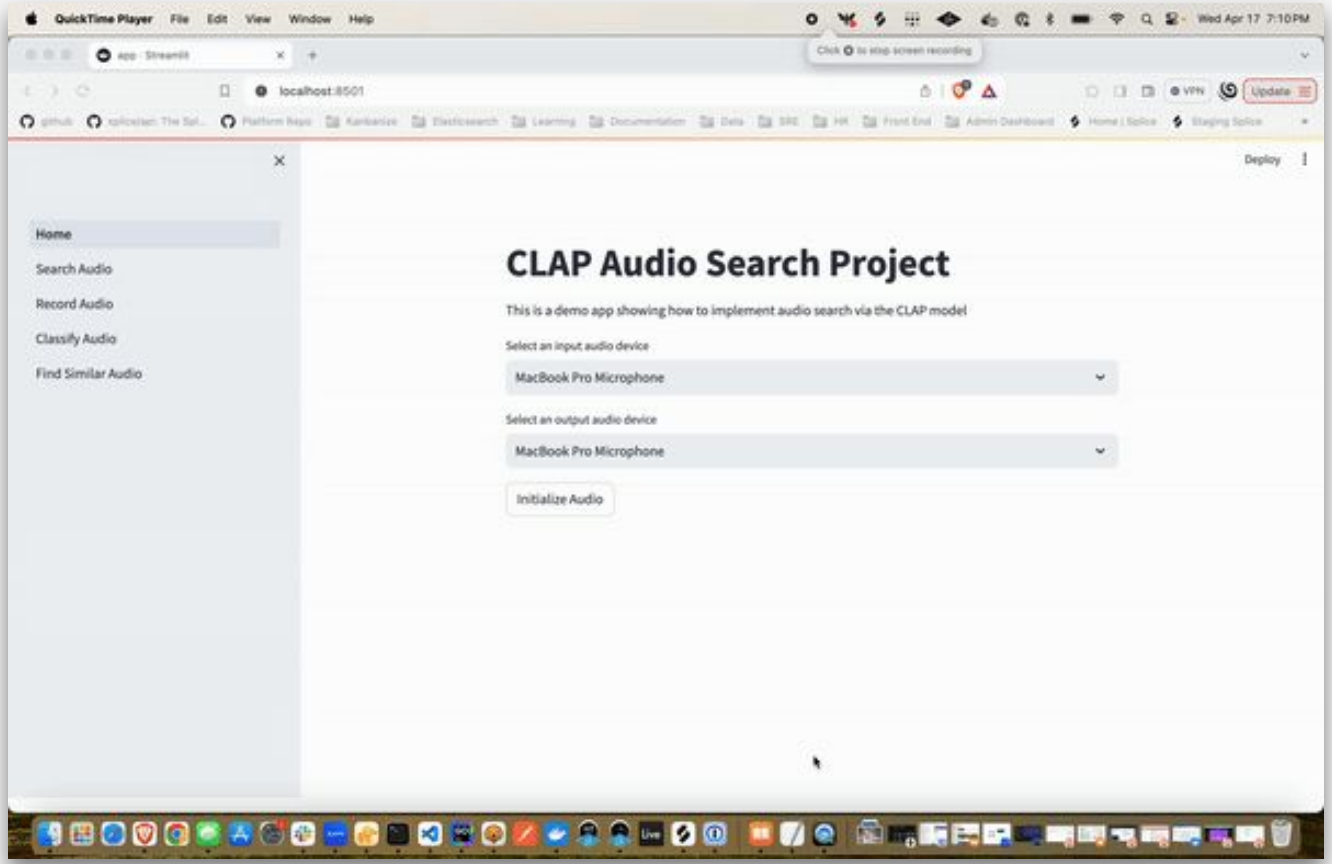
ajwallace.com

https://github.com/ajwallacemusic

Sr. Software Engineer, splice.com

Search Consultant

What We’re Gonna Build



Why?

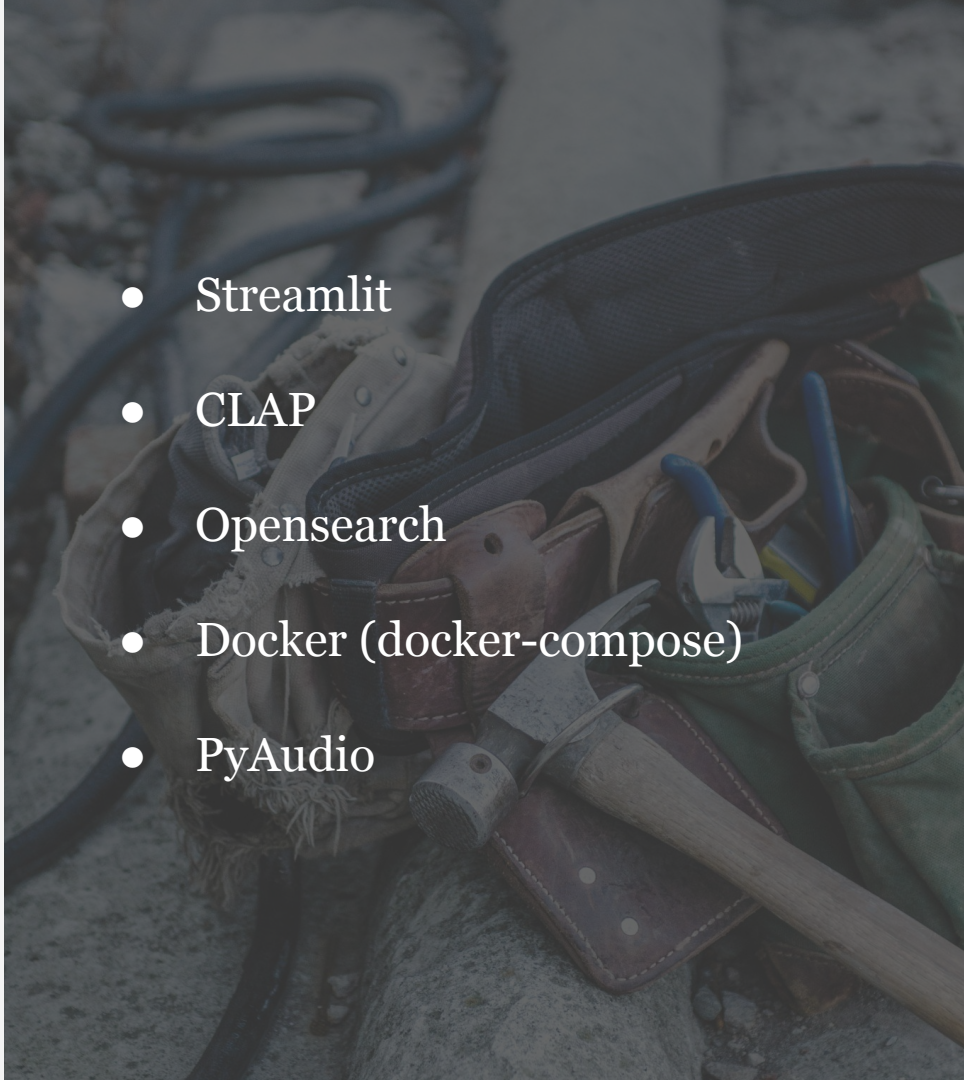
Music and Audio are Awesome.
So is Search.

- ⅘ People listen to music using a streaming service.
- Nearly 60% of musicians use AI in production.
- 50% of US uses voice search.
- Audiobooks revenue has grown double digits each year this decade.
- Multimodal Search is the future.


If I can do it…

How Are We Gonna Build It?

With code, duh.

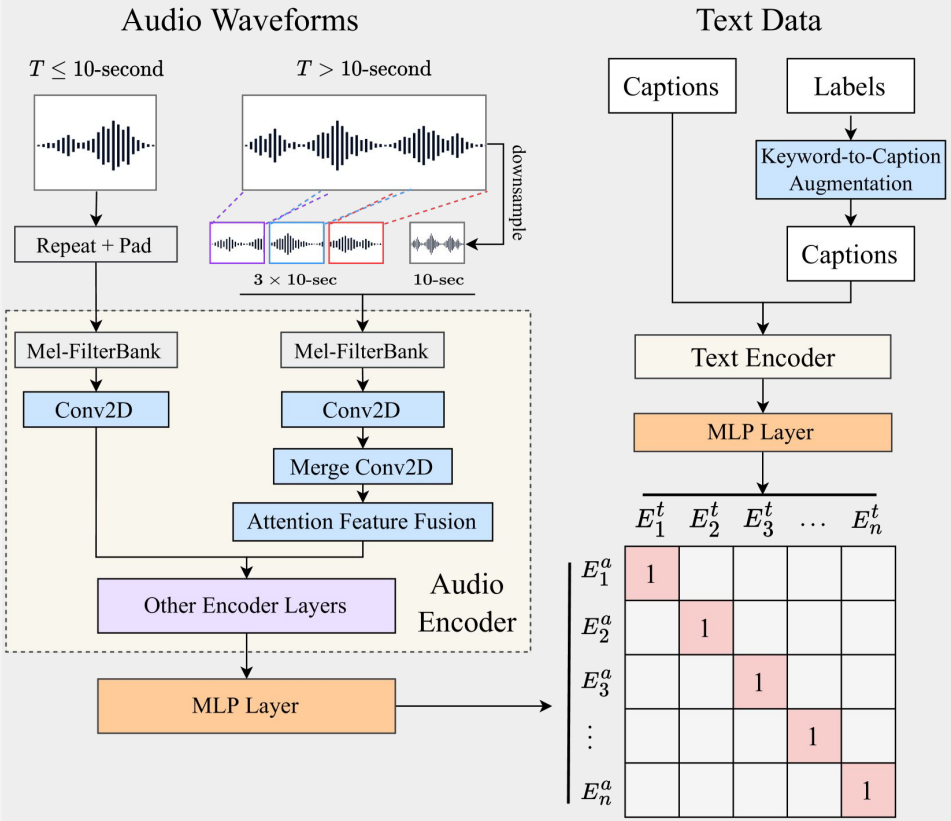


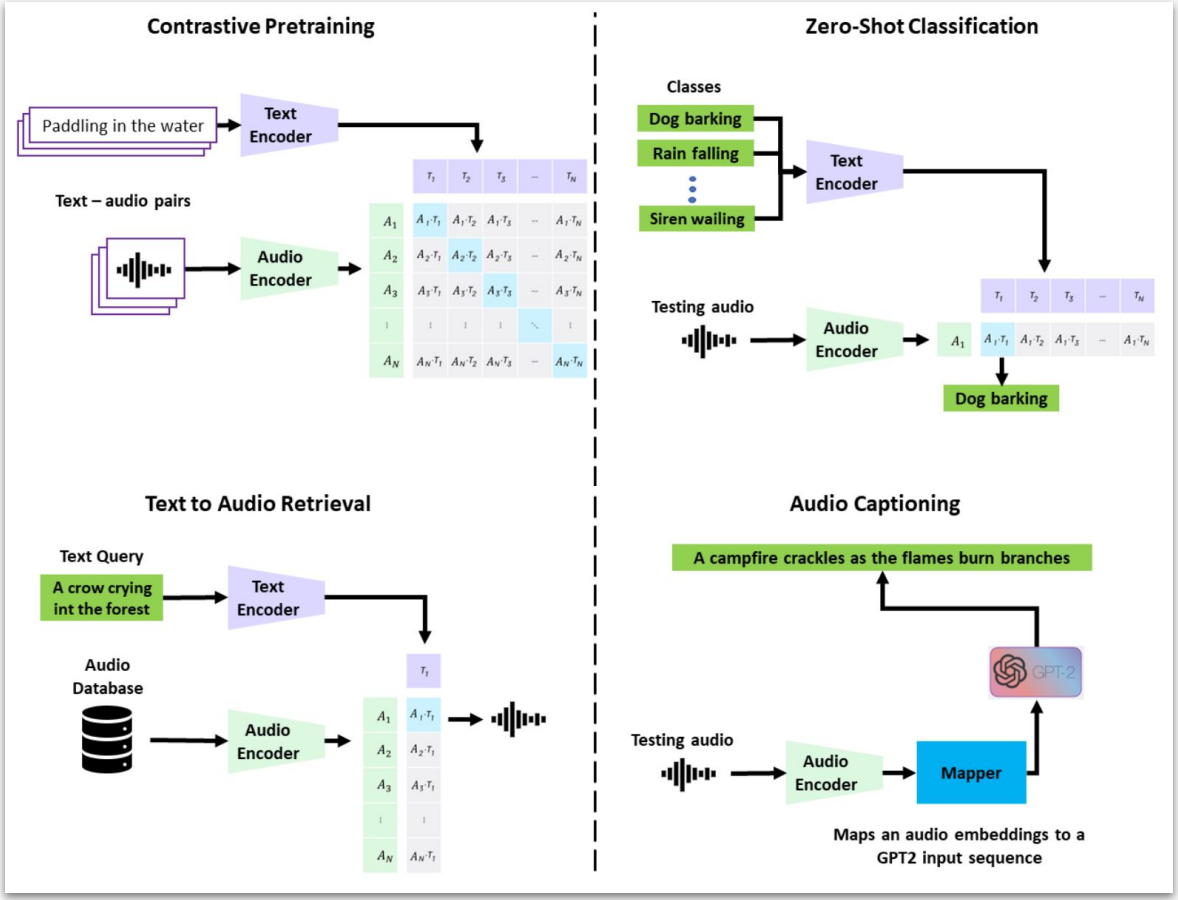
- Streamlit
- CLAP
- Opensearch
- Docker (docker-compose)
- PyAudio

Contrastive What Now?

CLAP

Contrastive Language-Audio Pretraining





Audio and text pairs are used to create a joint multimodal space.

Research Papers

[CLAP: Learning Audio Concepts From Natural Language Supervision]

[Large-scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation]

https://github.com/LAION-AI/CLAP

https://huggingface.co/docs/transformers/v4.27.2/en/model_doc/clap

LARGE-SCALE CONTRASTIVE LANGUAGE-AUDIO PRETRAINING FEATURE FUSION AND KEYWORD-TO-CAPTION AUGMENTAT

Yusong Wu [1*], *Ke Chen* [2*], *Tianyu Zhang* [1*], *Yuchen Hui* [1,3*], *Marianna Nezh*
Taylor Berg-Kirkpatrick [2], *Shlomo Dubnov* [2]

[1]Mila, Quebec Artificial Intelligence Institute, Université de Montréa
[2]University of California San Diego [3]LAION

ABSTRACT

Contrastive learning has shown remarkable success in the field of multimodal representation learning. In this paper, we propose a pipeline of contrastive language-audio pretraining to develop an audio representation by combining audio data with natural language descriptions. To accomplish this target, we first release LAION-Audio-630K, a large collection of 633,526 audio-text pairs from different data sources. Second, we construct a contrastive language-audio pretraining model by considering different audio encoders and text encoders. We incorporate the feature fusion mechanism and keyword-to-caption augmentation into the model design to further enable the model to process audio inputs of variable lengths and enhance the performance. Third, we perform comprehensive experiments to evaluate our model across three tasks: text-to-audio retrieval, zero-shot audio classification, and supervised audio classification. The results demonstrate that our model achieves superior performance in text-to-audio retrieval task. In audio classification tasks, the model achieves state-of-the-art performance in the zero-shot setting and is able to obtain performance comparable to models' results in the non-zero-shot setting. LAION-Audio-630K[1] and the proposed model[2] are both available to the public.

Index Terms— Contrastive Learning, Representation Learning, Text-to-Audio Retrieval, Audio Classification, Audio Dataset

1. INTRODUCTION

Audio is one of the most common information types in the world alongside text and image data. However, different audio tasks typically require finely-annotated data, which limits the amount of available audio data due to the labor-intensive collection procedure. Consequently, designing an effective audio representation for many audio tasks without requiring a lot of supervision remains a challenge.

The contrastive learning paradigm is a successful solution for training a model on large-scale noisy data collected from internet. The recently proposed Contrastive Language-Image Pretraining

variations of ImageNet dataset [2]. Addition
success in downstream tasks such as text-to-i
guided captioning. Similar to vision, audio
also contain overlapping information. In au
task, for instance, some text descriptions of a
to the corresponding audio. These text desc
meaning that could be learned together wi
form an audio representation of crossmodal
ally, training such a model requires simply
data, which is easy to collect.

Several recent studies [3–9] have present
contrastive language-audio pretraining mod
retrieval task. [6] utilizes Pretrained Audio N
[10] as the audio encoder, BERT [11] as the te
loss functions to evaluate the text-to-audio
[5] further ensemble HTSAT [12] and RoB
coder list to further enhance performance. Th
effectiveness of the learned representation i
of audio classification. Some other studies,
and WaveCLIP [9], focus more on the contr
image-audio-language) pretraining model.
great potential for contrastive learning in the

Nonetheless, current studies have not s
of the language-audio contrastive learning.
tioned above are trained on relatively small
large-scale data collection and augmentation
Second, prior work lacks a full investigation
parameter settings of audio/text encoders, wh
termining the basic contrastive language-aud
the model struggles to accommodate varied
larly for the transformer-based audio encoder
lution to handle audio inputs of variable-leng
of language-audio model studies focuses sol
trieval without assessing their audio represe
tasks. As a representation model, we expect
generalization ability to more downstream ta

In this paper, we make contributions t

Using It

```
# init ML processors/models/tokenizers
audio_classifier = pipeline(task="zero-shot-audio-classification",
model="laion/larger_clap_music_and_speech")
model = ClapModel.from_pretrained("laion/larger_clap_music_and_speech")
processor = ClapProcessor.from_pretrained("laion/larger_clap_music_and_speech")
tokenizer = AutoTokenizer.from_pretrained("laion/larger_clap_music_and_speech")

# function to create audio embedding
def embed_audio(filepath):
    y, sr = librosa.load(filepath)
    inputs = processor(audios=y, return_tensors="pt", sampling_rate=48000)
    audio_embed = model.get_audio_features(**inputs)
    arr = audio_embed.detach().numpy()
    return arr
```

Building The App

Opensearch

```
"mappings": {
       "properties": {
           "audio_embedding": {
               "type": "knn_vector",
               "dimension": 512,
               "method": {
               "name": "hnsw",
               "space_type": "l2",
               "engine": "lucene",
                …
```

Streamlit Homepage

```
st.title("CLAP Audio Search Project")
desc = "This is a demo app showing how to implement audio search via the CLAP model"
st.write(desc)

# initialize the things
init_session_state()
menu()
set_up_audio()
init_clap()
create_search_client()
```

Recording & Indexing

```
if start:
    st.session_state.audio_data = []
    record_audio()

if stop:
    save_audio(filename)
```

Searching

```
if st.button("Search"):
    if st.session_state.search_type == "Vector":
        response = vector_search(query)
        display_results(response)
    elif st.session_state.search_type == "BM25":
        response = bm25_search(query)
        display_results(response)
    elif st.session_state.search_type ==
"Hybrid":
        response = hybrid_search(query)
        display_results(response)
```

Classifying

```
def classify_audio(text1, text2, audiofile):
    y, _ = librosa.load(audiofile)
    audio_classifier = st.session_state.clap_audio_classifier
    output = audio_classifier(y, candidate_labels=[text1, text2])

    return get_label_with_highest_score(output)
```

Audio to Audio Search

```
if uploaded_file is not None:
    source = uploaded_file
    st.audio(source)
else:
    source = audiofile

if st.button("Search"):
    response = audio_to_audio_search(source)
    display_results(response)
```

Closing

Summary

Pretty good out of the gate

Hybrid helps

Next Steps

Opensearch hosted ML model

https://opensearch.org/docs/latest/search-plugins/multimodal-search/

Measure Relevance

https://quepid.com/

Fine Tuning

Sujit Pal: https://haystackconf.com/us2022/talk-1

Thank You

Sources

https://github.com/ajwallacemusic/clap-demo

Large-scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation

CLAP: Learning Audio Concepts From Natural Language Supervision

Images: https://unsplash.com/

Statistics: https://www.ifpi.org/, https://www.musicradar.com/, https://upcity.com/, https://www.statista.com/