

Create Your Music

The music you hear is beautiful, but the music you do not hear is even more beautiful.

딥러닝을 활용한 작곡 작사 표절 프로그램

TEAM: CRP
조 현국, 권 혜민, 서 성민

목차

1. 프로젝트 개요
2. 팀 구성 및 역할
3. 프로젝트 수행 절차
4. 개발 언어 및 툴
5. 화면 - 프론트엔드
6. 표절 프로그램
7. 작곡 프로그램
8. 작사 프로그램

| 프로젝트 개요



음원 표절 논란이 많은 현대시대에

파이썬 딥러닝 기술로

“음원표절을 잡아내는 **표절프로그램**이 있으면 어떨까?”

“그렇다면 **작곡과 작사** 또한 할 수 있지 않을까?”

“또 그렇다면! **하나의 웹사이트** 내에서 표절, 작곡, 작사
음악에 모든 것들을 담아 낼 수는 없을까?”

이 세가지의 궁금점으로 시작한 프로젝트 입니다.

| 팀 구성 및 역할

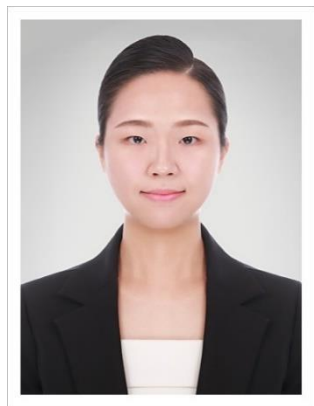


PM/PO

조 현국

Lyric Programmer

Python / NLP / Pytorch



PL

권 혜민

Composition Programmer

Python / GAN / Pytorch



PA

서 성민

Front Programmer

JavaScript / TypeScript / REACT

| 프로젝트 수행 절차

구분	기간	활동	비고
사전 기획	▶ 4/25(월)~5/2(월)	▶ 프로젝트 기획 및 주제 선정 ▶ 기획안 작성	▶ 아이디어 선정
데이터 수집	▶ 5/3(화)~5/10(화)	▶ 필요 악보, mp3파일 수집 ▶ 외부 데이터 수집	▶ 필요한 DB 수집
기본 화면 구성	▶ 5/10(수)~5/26(목)	▶ 화면 기본 구성들을 REACT로 구현	
파이썬 모델 찾기 및 학습	▶ 05/27(월)~7/24(일)	▶ 파이썬 모델 구현을 위한 학습	▶ 팀별 중간보고 실시 화면 TYPE으로 전환
프로젝트 서비스 구축	▶ 7/25(월)~8/8(월)	▶ AWS 및 학습된 모델 정리 ▶ 프로젝트 발표영상 준비 및 서류 준비	▶ 최적화, 오류 수정
총 개발기간	▶ 4/25(월)~8/8(월)(총 15주)	-	-

| 개발언어 및 라이브러리



JavaScript



python



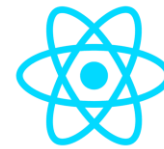
TypeScript



PyTorch



TensorFlow



React



NumPy



pandas

matplotlib



OpenCV



Redux

| 화면 (FrontEnd) – Next.js + TypeScript

- TypeScript와 Next.js를 베이스로 화면을 구현함.
- React 기반으로 음악 콘텐츠를 중심으로 필요한 기능과 화면을 구현함.

NEXT.js + TypeScript

| 화면 (FrontEnd) – TypeScript + Next (component)

게시판

- 게시판에 대한 Component와 Pages 부분

```
import { Article } from '@modules/types'
import React from 'react'

type Props = {

  onChange : (e: React.FormEvent<HTMLInputElement> | any ) => void
  onSubmit : (e: React.FormEvent<HTMLFormElement> ) => void
}

const AddBoard: React.FC<Props> = ({onChange, onSubmit}) => {
  const date = new Date();
  const parseDate = date.toDateString()
```

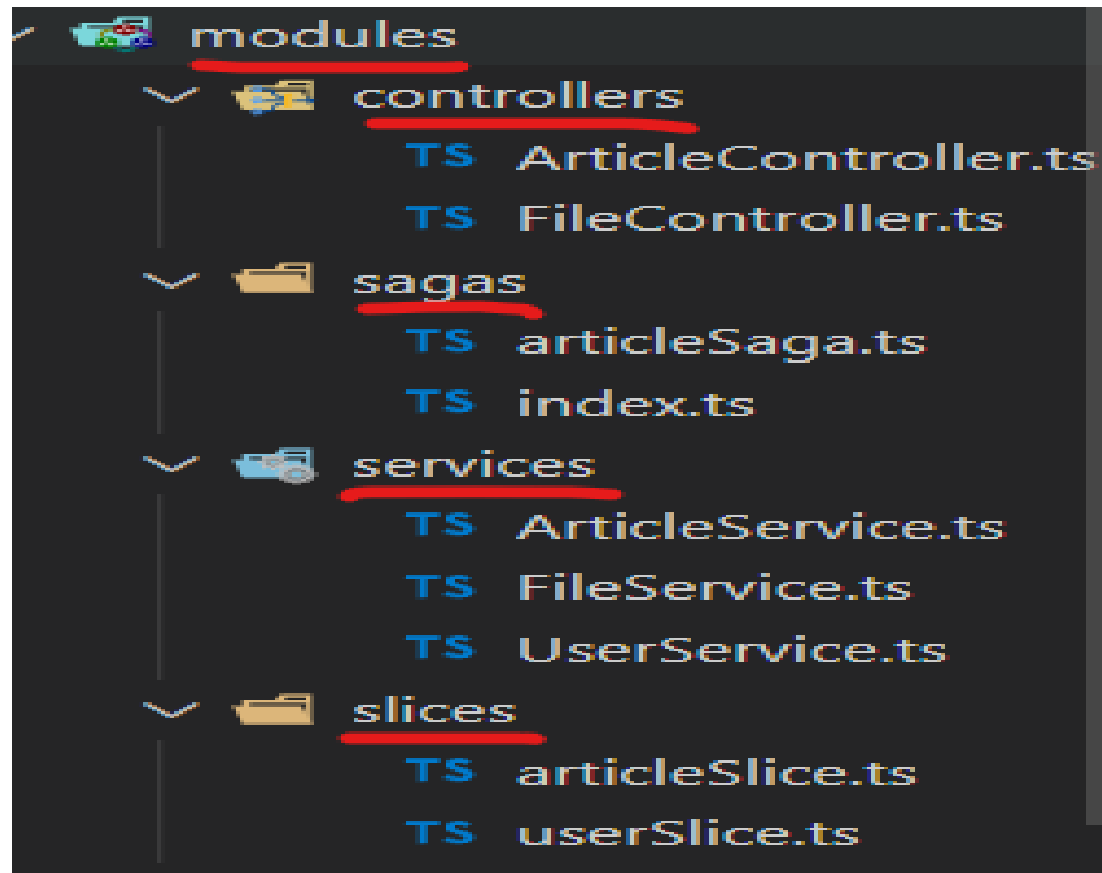
```
import { Article } from "@modules/types";
import React, { useState } from "react";
import Image from "next/image";
import { musicData } from "@modules/types";
export interface Props {
  datas: Article[];
  onDeleteClick: any;
}

const AllBoardList: React.FC<Props> = ({ datas, onDeleteClick }: Props) => {
  const [text, setText] = useState([
    {
      id: 1,
      title: "CRP",
      content: "Music is my life",
    },
    {
      id: 2,
      title: "CRP Team",
      content: "Enjoy your Life!",
    },
  ])
};
```


| 화면 (FrontEnd) – TypeScript + Next (component)

게시판

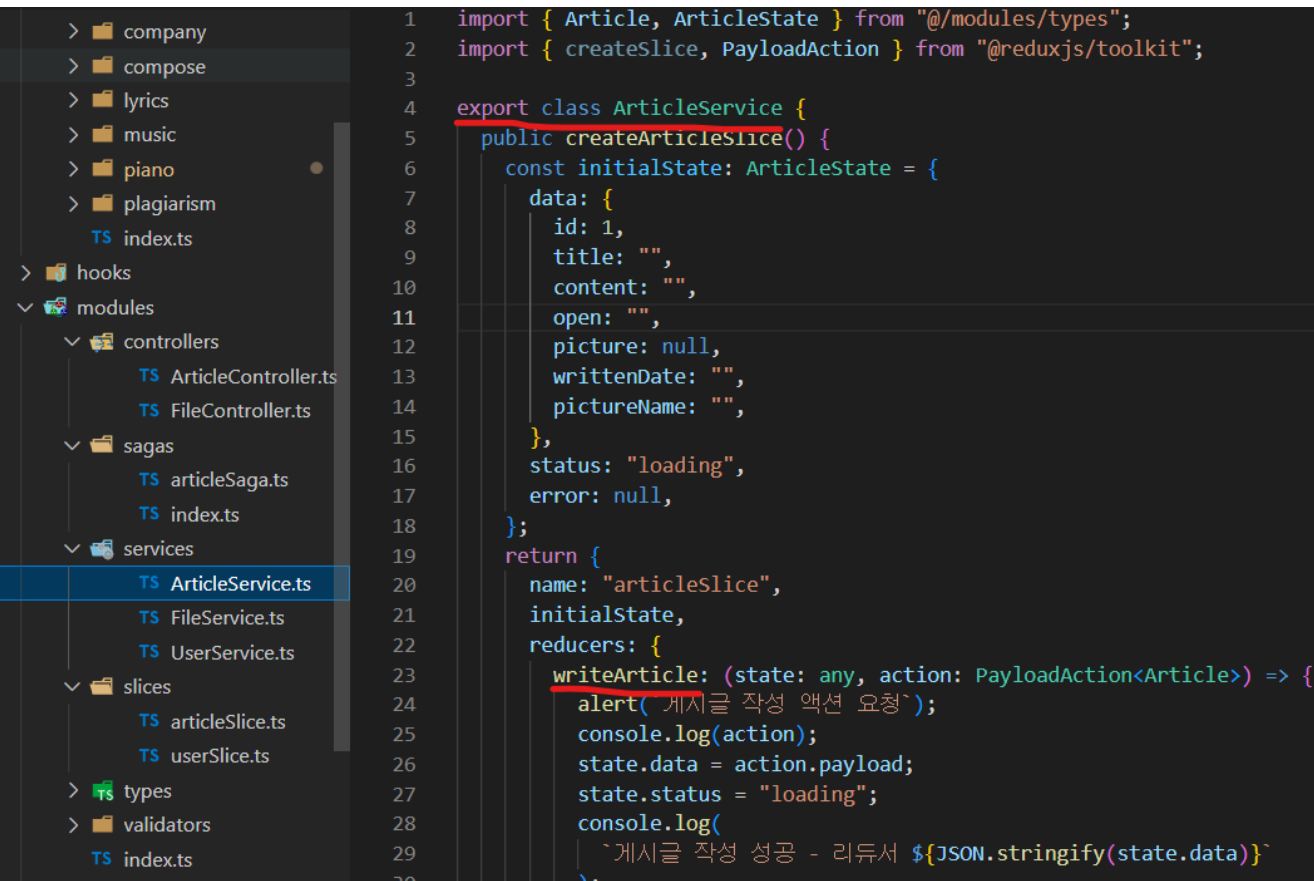
- Redux 부분 구조 및 내부 구조



| 화면 (FrontEnd) – TypeScript + Next (component)

게시판

- 게시판 관련 Redux부분 내부 코드 과정 1

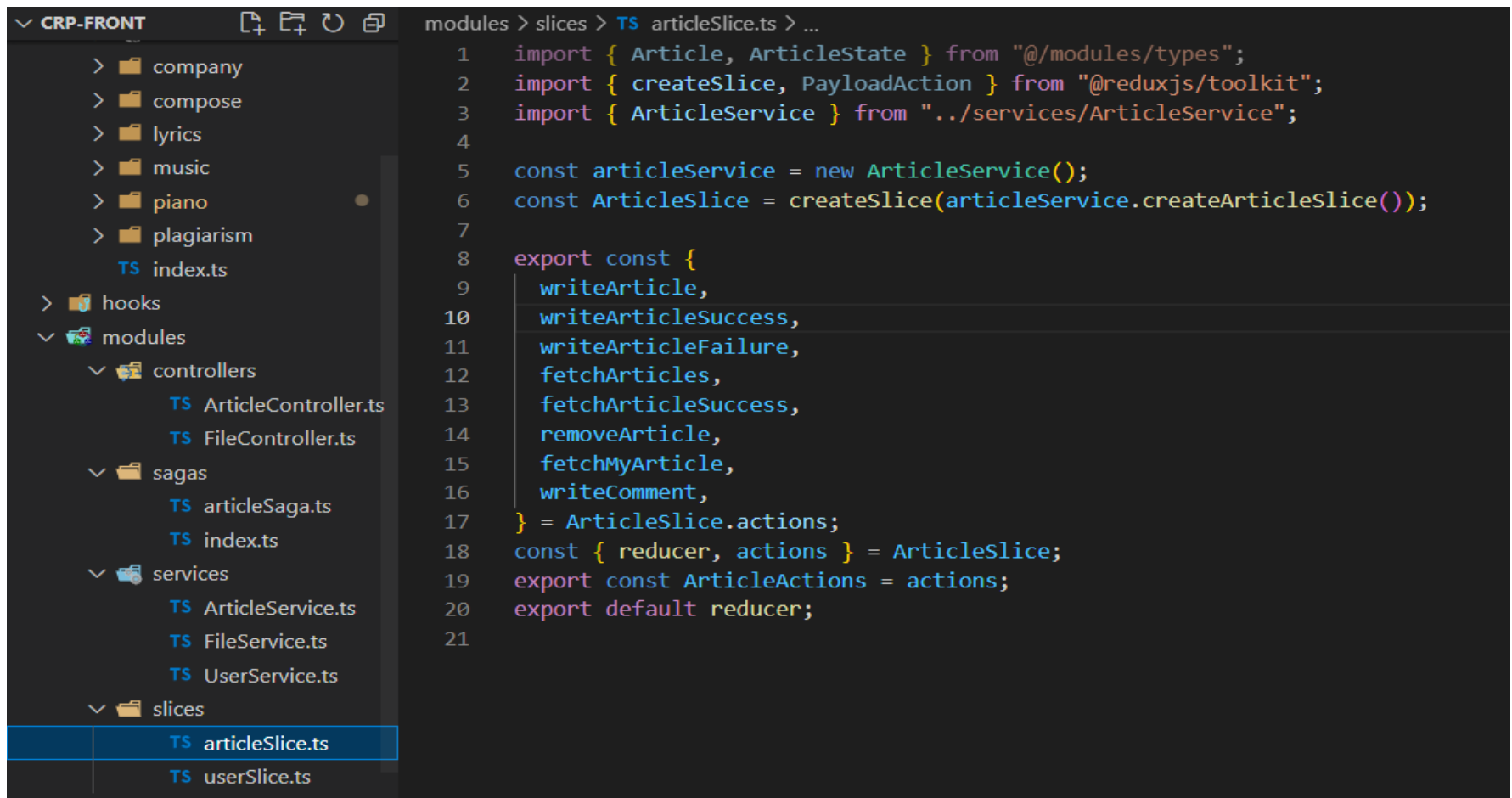


```
1 import { Article, ArticleState } from "@modules/types";
2 import { createSlice, PayloadAction } from "@reduxjs/toolkit";
3
4 export class ArticleService {
5   public createArticleSlice() {
6     const initialState: ArticleState = {
7       data: {
8         id: 1,
9         title: "",
10        content: "",
11        open: "",
12        picture: null,
13        writtenDate: "",
14        pictureName: "",
15      },
16      status: "loading",
17      error: null,
18    };
19    return {
20      name: "articleSlice",
21      initialState,
22      reducers: {
23        writeArticle: (state: any, action: PayloadAction<Article>) => {
24          alert( `게시글 작성 액션 요청` );
25          console.log(action);
26          state.data = action.payload;
27          state.status = "loading";
28          console.log(
29            `게시글 작성 성공 - 리뷰서 ${JSON.stringify(state.data)}`
30          );
31        }
32      }
33    };
34  }
35 }
```

| 화면 (FrontEnd) – TypeScript + Next (component)

게시판

- 게시판 관련 Redux부분 내부 코드 과정 2




The screenshot shows a code editor with a dark theme. On the left is a file explorer for a project named 'CRP-FRONT'. The 'modules' directory is expanded, showing sub-directories like 'controllers', 'sagas', 'services', and 'slices'. The 'slices' directory is selected, and 'articleSlice.ts' is highlighted. On the right, the code for 'articleSlice.ts' is displayed, showing imports from '@modules/types', '@reduxjs/toolkit', and '../services/ArticleService'. It defines an 'articleService' instance, creates an 'ArticleSlice' using 'createSlice', and exports actions like 'writeArticle', 'writeArticleSuccess', 'writeArticleFailure', 'fetchArticles', 'fetchArticleSuccess', 'removeArticle', 'fetchMyArticle', and 'writeComment'. It also exports a reducer and default actions.

```
modules > slices > TS articleSlice.ts > ...
1  import { Article, ArticleState } from "@modules/types";
2  import { createSlice, PayloadAction } from "@reduxjs/toolkit";
3  import { ArticleService } from "../services/ArticleService";
4
5  const articleService = new ArticleService();
6  const ArticleSlice = createSlice(articleService.createArticleSlice());
7
8  export const {
9    writeArticle,
10   writeArticleSuccess,
11   writeArticleFailure,
12   fetchArticles,
13   fetchArticleSuccess,
14   removeArticle,
15   fetchMyArticle,
16   writeComment,
17 } = ArticleSlice.actions;
18 const { reducer, actions } = ArticleSlice;
19 export const ArticleActions = actions;
20 export default reducer;
21
```

| 화면 (FrontEnd) – TypeScript + Next (component)

게시판

- 게시판 관련 Redux부분 내부 코드 과정 3

```
1  import {FileController} from "../controllers/FileController"
2  import { ArticleController } from "../controllers/ArticleController";
3  import {FileService} from "../services/FileService"
4  import {ArticleService} from "../services/ArticleService"
5  import articleSaga from "../sagas"
6  import FileValidator from "../validators";
7  
8  export { FileValidator, FileService, FileController, ArticleController, ArticleService, articleSaga };
```

|화면 (FrontEnd) – TypeScript + Next (component)

게시판

- 게시판 관련 Redux부분 내부 코드 과정 4

```
const rootReducer = (
  state: ReturnType<typeof combinedReducer>,
  action: AnyAction
) => {
  if(action.payload === HYDRATE) { // action.type => action.payload 07-14
    return{
      ...state, // use previous state
      ...action.payload // apply delta from hydration
    }
  } else {
    return combinedReducer(state,action)
  }
}

const makeStore = () =>{
  const store =
    configureStore({
      reducer:{ rootReducer },
      middleware: (getDefaultMiddleware) =>
        getDefaultMiddleware({serializableCheck: false})
      //직렬화 문제 발생 시 {serializableCheck: false} 파라미터로 전달
      .prepend(sagaMiddleware)
      .concat(logger),
      devTools : isDev
    });

  sagaMiddleware.run(rootSaga)
```

| 화면 (FrontEnd) – TypeScript + Next (component)

게시판

- 게시판 관련 Redux부분 내부 코드 과정 5

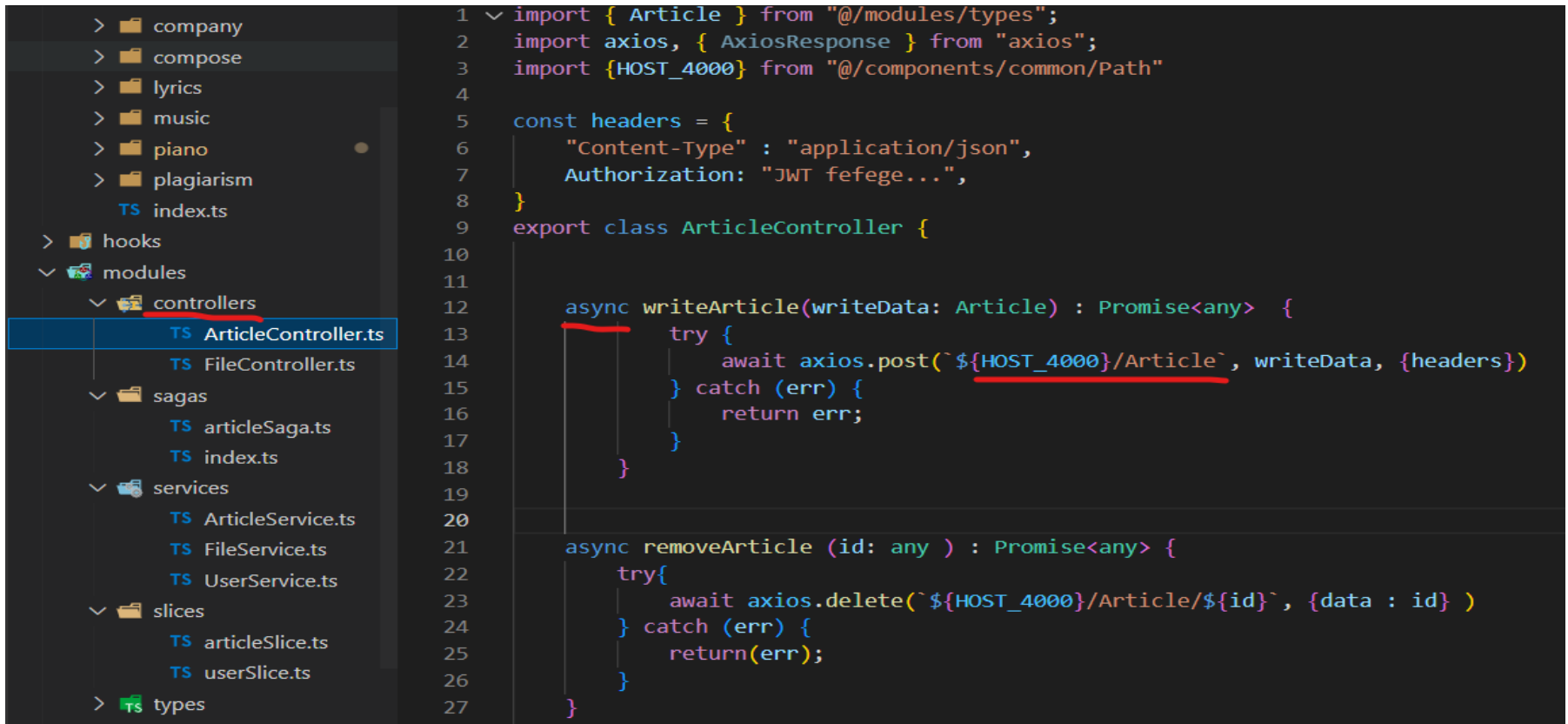
```
CRP-FRONT
├── company
├── compose
├── lyrics
├── music
├── piano
├── plagiarism
├── TS index.ts
├── hooks
├── modules
│   ├── controllers
│   │   ├── TS ArticleController.ts
│   │   ├── TS FileController.ts
│   ├── sagas
│   │   ├── TS articleSaga.ts
│   │   └── TS index.ts
│   ├── services
│   │   ├── TS ArticleService.ts
│   │   ├── TS FileService.ts
│   │   └── TS UserService.ts
│   └── slices
│       ├── TS articleSlice.ts
│       └── TS userSlice.ts
└── TS types

modules > sagas > TS articleSaga.ts > ...
1  import { ArticleController } from "@modules/controllers/ArticleController";
2  import { call, put, takeEvery, takeLatest } from "redux-saga/effects";
3  import { ArticleActions } from "../slices/articleSlice";
4  import { Article } from "../types";
5
6  //get Saga
7  function* writeArticleSaga(action: { payload: Article }) {
8      const { writeArticleSuccess, writeArticleFailure } = ArticleActions;
9      const param = action.payload;
10     const articleController = new ArticleController();
11     try {
12         yield call(articleController.writeArticle, param);
13         yield put(writeArticleSuccess());
14     } catch (error) {
15         yield put(writeArticleFailure());
16     }
17 }
18
19 function* fetchMyArticleSaga(action: { payload: any }) {
20     const { fetchMyArticleSuccess, fetchMyArticleFailure } = ArticleActions;
21     try {
22         const response: Article = yield call(action.payload);
23         yield put(fetchMyArticleSuccess(response));
24     } catch (error) {
25         yield put(fetchMyArticleFailure());
26     }
27 }
```

|화면 (FrontEnd) – TypeScript + Next (component)

게시판

- 게시판 관련 Redux부분 내부 코드 과정 6



```
1  import { Article } from "@modules/types";
2  import axios, { AxiosResponse } from "axios";
3  import { HOST_4000 } from "@components/common/Path"
4
5  const headers = {
6    "Content-Type" : "application/json",
7    Authorization: "JWT fefege...",
8  }
9  export class ArticleController {
10
11
12    async writeArticle(writeData: Article) : Promise<any> {
13      try {
14        await axios.post(`${HOST_4000}/Article`, writeData, {headers})
15      } catch (err) {
16        return err;
17      }
18    }
19
20
21    async removeArticle (id: any ) : Promise<any> {
22      try{
23        await axios.delete(`${HOST_4000}/Article/${id}`, {data : id} )
24      } catch (err) {
25        return(err);
26      }
27    }
28  }
```

| 화면 (FrontEnd) – TypeScript + Next (component)

게시판

■ 게시판 화면 구현

나만의 게시글

이미지도 업로드하고 게시글도 작성해보세요

게시글 등록하기

Mon Aug 08 2022

제목 입력

파일 선택

선택된 파일 없음

게시글 작성란

UPLOAD

CRP 게시판

CRP

Music is my life

삭제

CRP Team

Enjoy your Life!

삭제

Copyright © CRP WebSite 2022

[개발자들](#) [사이트 소개](#) [관련사이트](#)

| 화면 (FrontEnd) – TypeScript + Next (component)

FileUpload

- component에 상태를 만들어 주어 파일 업로드에 관한 기능 구현

```
const PUpload: React.FC<Props> = ({ onSubmit, onSubmit1 }: Props) => {  
  const [uploadFormError, setUploadFormError] = useState<string>("");  
  
  const handleFileUpload = async (element: HTMLInputElement) => {  
    const file = element.files;  
  
    if (!file) {  
      return;  
    }  
  
    const validFileSize = await validator.validateFileSize(file[0].size);  
    const validFileType = await validator.validateFileType(  
      FileService.getFileExtension(file[0].name)  
    );  
  
    if (!validFileSize.isValid) {  
      setUploadFormError(validFileSize.errorMessage);  
      return;  
    }  
  
    if (!validFileType.isValid) {  
      setUploadFormError(validFileType.errorMessage);  
      return;  
    }  
  
    if (uploadFormError && validFileSize.isValid) {  
      setUploadFormError("");  
    }  
  
    const fileController = new FileController(file[0]);  
    const fileUploadResponse = await fileController.uploadFile();  
  }  
}
```

| 화면 (FrontEnd) – TypeScript + Next (pages)

FileUpload

- 파일업로드에 관한 페이지 구현

```
const onSubmitFile = async (e: React.FormEvent<HTMLFormElement>) => {
  e.preventDefault();
  window.location.href = `${HOST_3000}/plagiarism/plagiarism`;
};

const fixonSubmitFile = async (e: React.FormEvent<HTMLFormElement>) => {
  e.preventDefault()
}

const PluploadPage: NextPage = () => {

  useEffect(()=> {
  },[])

  return (
    <Plupload onSubmit = {onSubmitFile} onSubmit1={fixonSubmitFile}/>
  )
}
export default PluploadPage
```

| 화면 (FrontEnd) – 파일 업로드 화면 구현 과정

FileUpload

■ 파일 업로드 구현 화면 1



회사소개 ▾ 게시판 ▾ 음악 프로그램 ▾ ○ 표절/작곡/작사 프로그램 ○ 관련 사이트

원본용 악보를 업로드하세요

파일 선택 선택된 파일 없음

악보 등록

표절여부를 확인하고 싶은 악보를 업로드하세요

파일 선택 선택된 파일 없음

악보 등록

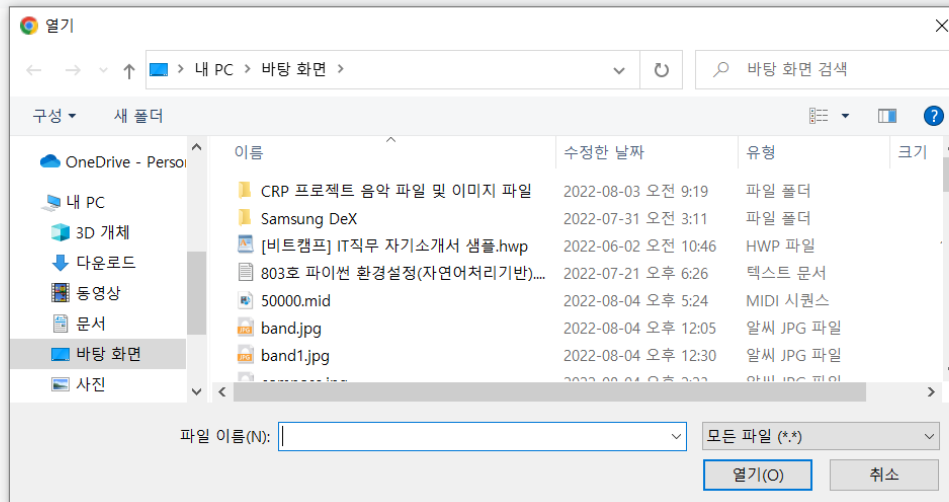
| 화면 (FrontEnd) – 파일 업로드 화면 구현 과정

FileUpload

■ 파일 업로드 구현 화면 2



회사소개 ▾ 게시판 ▾ 음악 프로그램 ▾ ○ 표절/작곡/작사 프로그램 ○ 관련 사이트



표절여부를 확인하고 싶은 악보를 업로드하세요

파일 선택 선택된 파일 없음

악보 등록

| 화면 (FrontEnd) – 파일 업로드 화면 구현 과정

FileUpload

■ 파일 업로드 구현 화면 3



회사소개 ▾ 게시판 ▾ 음악 프로그램 ▾ ○ 표절/작곡/작사 프로그램 ○ 관련 사이트

원본용 악보를 업로드하세요

파일 선택 선택된 파일 없음

악보 등록

표절여부를 확인하고 싶은 악보를 업로드하세요

파일 선택 school.jpg

악보 등록

| 화면 (FrontEnd) – 파일 업로드 화면 구현 과정

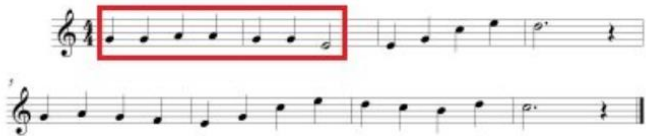
FileUpload

■ 파일 업로드 구현 화면 4



회사소개 ▾ 게시판 ▾ 음악 프로그램 ▾ ○ 표절/작곡/작사 프로그램 ○ 관련 사이트

Score



Score

학교 종이 땡땡땡



표절판독하기

| 화면 (FrontEnd) – 파일 업로드 화면 구현 과정

FileUpload

■ 파일 업로드 구현 화면 6

Score



Score

학 교 종 이 땡땡땡



표절판독하기

검사 결과 : 표절

하이라이트 된 부분을 제외한 마디가 학교
종이 땡땡땡 악보와 유사합니다.

| 화면 (FrontEnd) – React

Piano

- Tone 이라는 web audio Framework를 사용하여 피아노를 구현함.

```
1 import './Piano.module.css';
2 import css from "styled-jsx/css"
3
4 import { Button } from 'react-bootstrap';
5 import React, {useEffect, useState} from 'react';
6 import axios from 'axios';
7
8 import {
9   playC4,
10  playDb4,
11  playD4,
12  playEb4,
13  playE4,
14  playF4,
15  playGb4,
16  playG4,
17  playAb4,
18  playA4,
19  playBb4,
20  playB4,
21  playC5,
22  playDb5,
23  playD5,
24  playEb5,
25  playE5,
26  playF5,
27  playGb5,
28  playG5,
29  playAb5,
30  playA5,
31  playBb5,
32  playB5,
33  playC6,
34  PlayNote
35 } from './Tone.js'
```

```
function Piano(){
  const [saveNote, setSaveNote] = useState()

  const tonePianoApi = async(saveNote) => {
    try {
      console.log(`API 진입`)
      const response = await axios.post(`${HOST_3000}`, saveNote, {headers})
    } catch (err) {
      return err;
    }
  }

  useEffect(() => {
    window.addEventListener('keydown', PlayNote)
  }, [])
}
```


| 화면 (FrontEnd) – React + JavaScript

Tone

- Tone.Synth 는 단일 오실레이터와 ADSR 엔벨로프를 제공하는 기본 신디사이저

```
import * as Tone from "tone"

const playC4 = (keycode) => {
  const synth = new Tone.Synth().toDestination();
  synth.triggerAttackRelease("C4", "8n");
  return keycode
}
```

```
const PlayNote = (event) => {
  if(event.keyCode === 65){
    let a = localStorage.getItem("note")
    let b = playC4("A");
    let c = a + ', ' + b
    //alert('최종 저장된 값: ' + c)
    localStorage.setItem("note", c)
  }
}
```

| 화면 (FrontEnd) – 결과

Tone - Piano

- 피아노 구현 화면

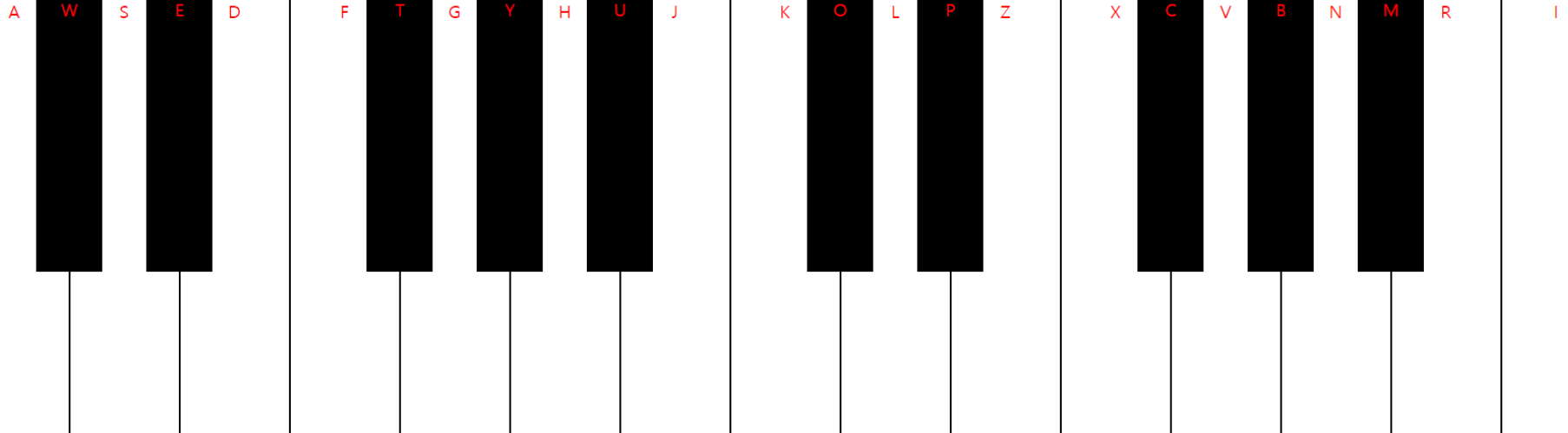


회사소개 ▼ 게시판 ▼ 음악 프로그램 ▼ ○ 표절/작곡/작사 프로그램 ○ 관련 사이트

Enjoy Playing Piano!

녹음시작

녹음 끝



| 화면 (FrontEnd) – 결과

Tone - Piano

- 피아노 keynote 저장된 화면

The screenshot shows a web browser's developer console with the following content:

- At the top, there are tabs for Elements, Console, Sources, Network, Performance, Memory, Application, Security, Lighthouse, Recorder, Performance insights, and Redux. The Console tab is active.
- Below the tabs, there is a filter input field with the text "Filter" and a "Default levels" dropdown menu.
- The console displays three yellow error messages:

```
* Tone.js v14.7.77 *
Global.js?8405:67
▶The AudioContext was not allowed to start. It must be resumed (or created) after a user gesture on the page. https://goo.gl/7K7WLu audio-context-constructor.js?123c:11
▶The AudioContext was not allowed to start. It must be resumed (or created) after a user gesture on the page. https://goo.gl/7K7WLu test-audio-scheduled...s-support.js?199c:4
▶The AudioContext was not allowed to start. It must be resumed (or created) after a user gesture on the page. https://goo.gl/7K7WLu constant-source-node...structor.js?cf4e:42
```

Below the error messages, there is a log entry for "API 진입" (API Entry) from Piano.js:

```
API 진입
Piano.js?5a72:50
▼ {item: Array(12)}
  ▶ item: (12) ['G', 'G', 'H', 'H', 'G', 'G', 'D', 'G', 'G', 'D', 'D', 'S']
  ▶ [[Prototype]]: Object
```

The array `['G', 'G', 'H', 'H', 'G', 'G', 'D', 'G', 'G', 'D', 'D', 'S']` is highlighted with a red underline.

표절 프로그램



1. Cutted 1

[\meter<"4/4"> d1/4 e1/32 e2/2 e1/8 e1/16 e1/32 {e1/4,g1/4} e1/4 e1/8 c1/8 g1/32 c1/16 e1/32]

| 표절 프로그램

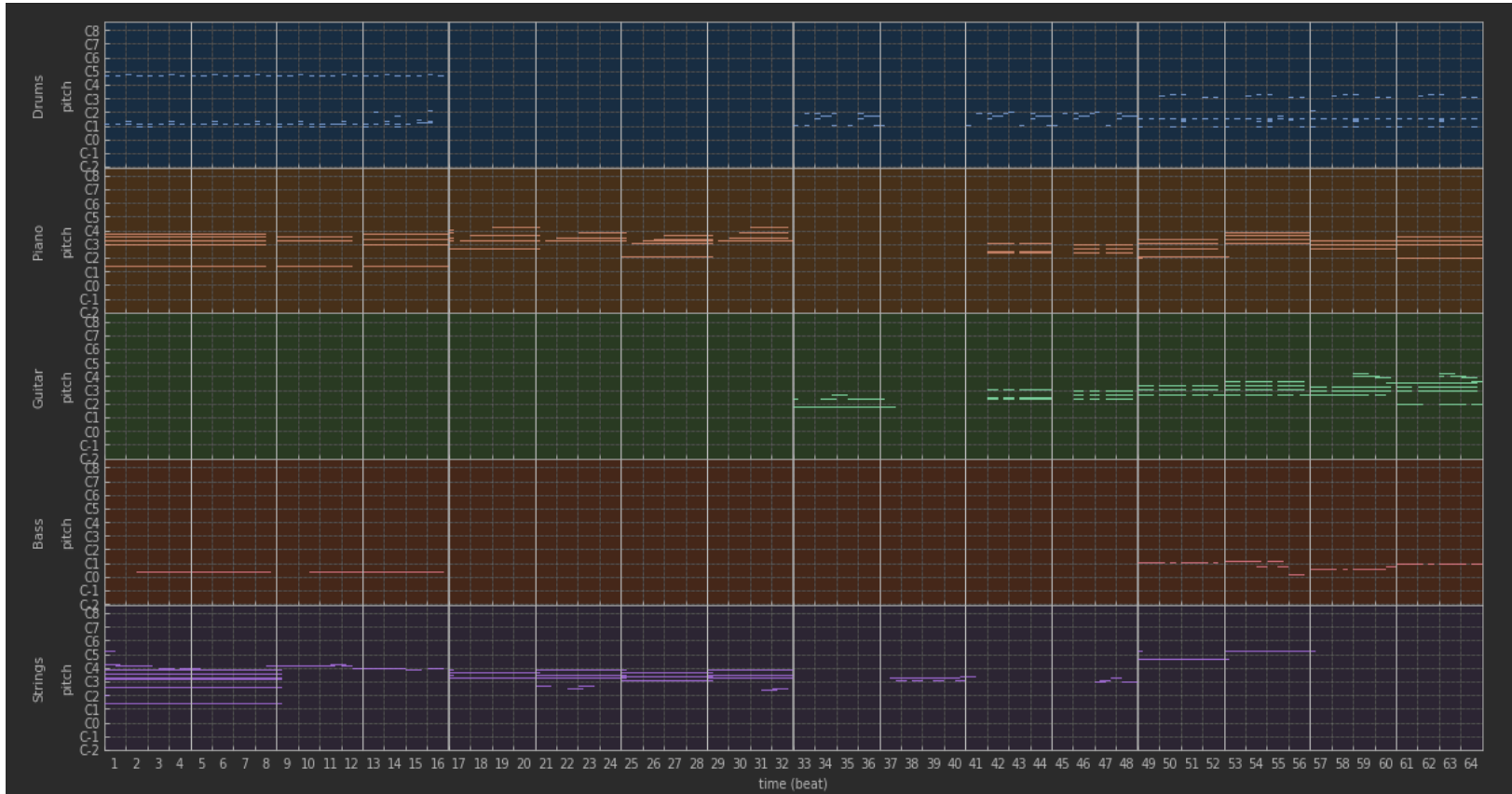
Image

- 악보인식 하는 코드

```
1  import numpy as np
2  import pickle
3  from numpy.linalg import norm
4  from skimage.filters import *
5  from skimage.color import rgb2gray
6  import matplotlib.pyplot as plt
7  import cv2 as cv
8  import os
9  import time
10
11
12  def binarize(img, method):
13      if method == 'skimage_local':
14          return img >= threshold_local(img, 31, offset=3)
15      if method == 'skimage_sauvola':
16          return img >= threshold_sauvola(img, 31)
17      if method == 'cv_adaptive':
18          return cv.adaptiveThreshold(img, 255, cv.ADAPTIVE_THRESH_GAUSSIAN_C, cv.THRESH_BINARY_INV, 41, 10)
19
20
21  def get_staff_corners(img, contour): #this function doesn't work well, it needs a different implementation
22      image_corners = np.array([[0, 0], [img.shape[1], 0], [0, img.shape[0]], [img.shape[1], img.shape[0]]])
23      staff_corners = np.zeros((4, 2))
24
25      staff_corners[0] = max(contour, key=lambda point: norm(point[0] - image_corners[3]))
26      staff_corners[1] = max(contour, key=lambda point: norm(point[0] - image_corners[2]))
27      staff_corners[2] = max(contour, key=lambda point: norm(point[0] - image_corners[1]))
28      staff_corners[3] = max(contour, key=lambda point: norm(point[0] - image_corners[0]))
29      return staff_corners
30
```

| 작곡 프로그램

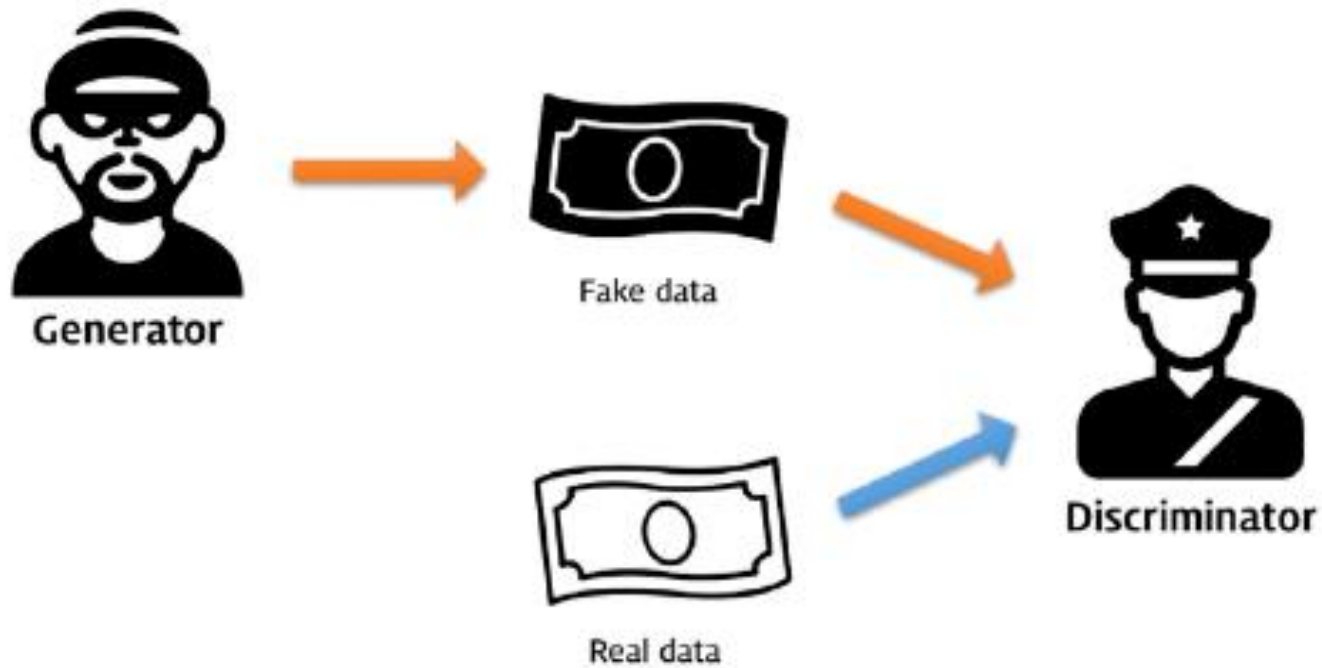
Muse-GAN Data Set



5개의 세션으로 이루어진 Lakh Pianoroll Dataset 을 사용하였습니다.

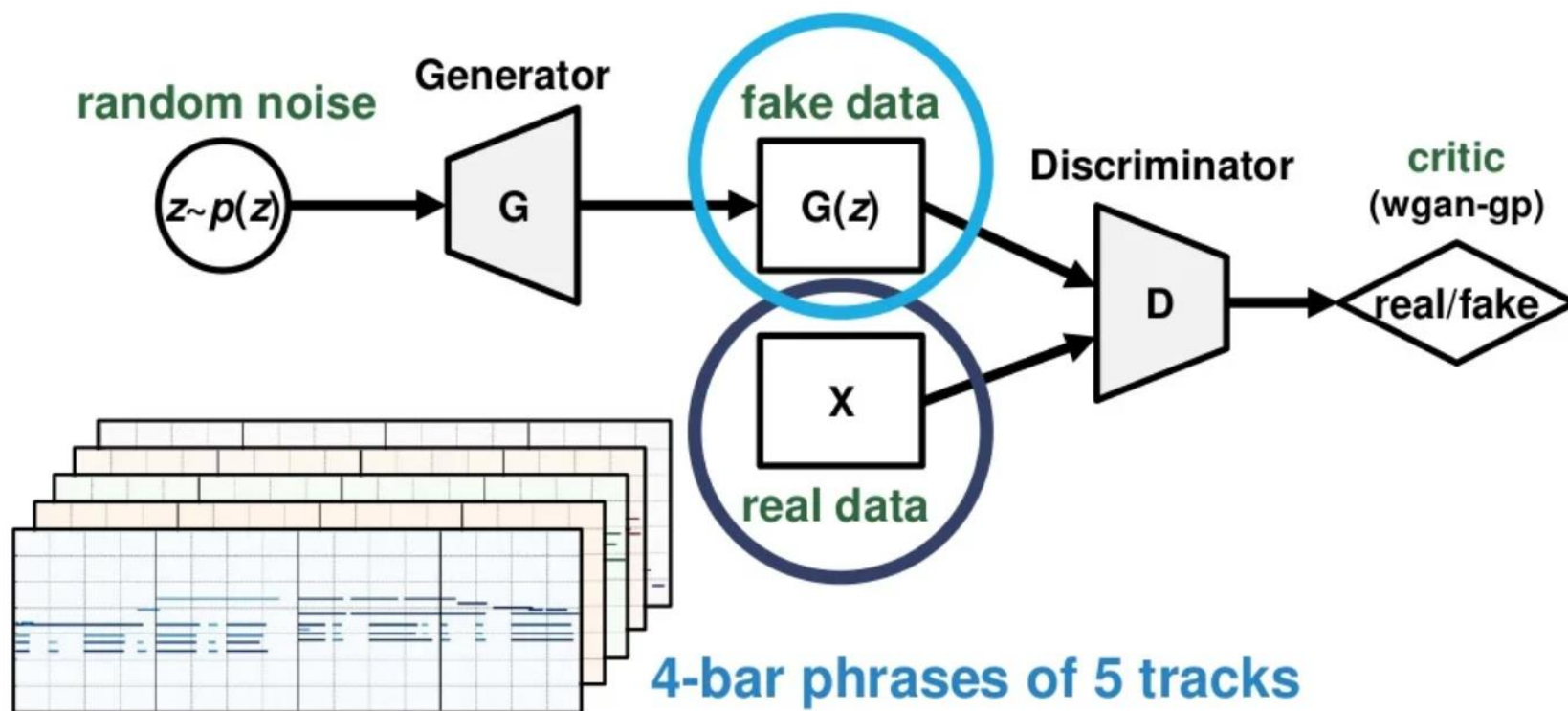
| 작곡 프로그램

Muse-GAN



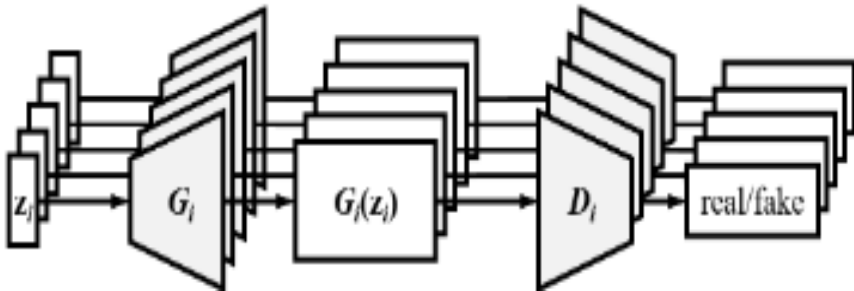
| 작곡 프로그램

Muse-GAN



| 작곡프로그램 - (Muse GAN)

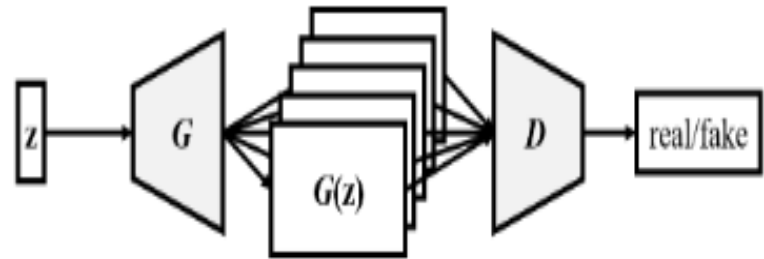
Jamming model



jamming 모델

랜덤 벡터 z_i (에서 자체 트랙의 음악을 생성합니다.
이러한 생성기는 서로 다른 판별기로부터 역전파를 수신합니다.

Composer model

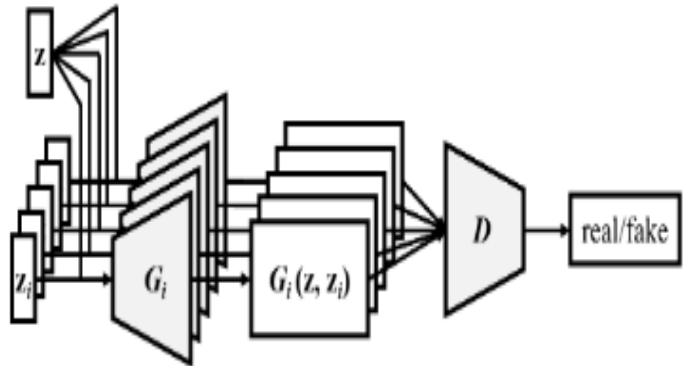


Composer 모델

이 모델은 하나의 랜덤 벡터 z 와 M 트랙을 집합적으로 검사하여 입력 음악이 진짜인지 가짜인지 구분하는 하나의 판별기가 필요합니다.

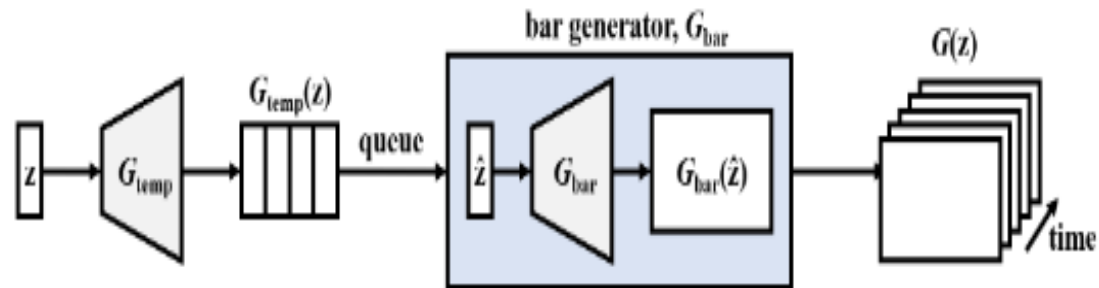
| 작곡프로그램 - (Muse GAN)

Hybrid model



Hybrid 모델은 jamming 과 Composer 의 개념을 결합한 M 개의 생성기를 필요로 하며 각각은 트랙 간 랜덤 벡터 z 및 트랙 내 랜덤 벡터 z_i 를 입력으로 사용합니다.
M 트랙을 집합적으로 평가하기 위해 하나의 판별자만 사용합니다.

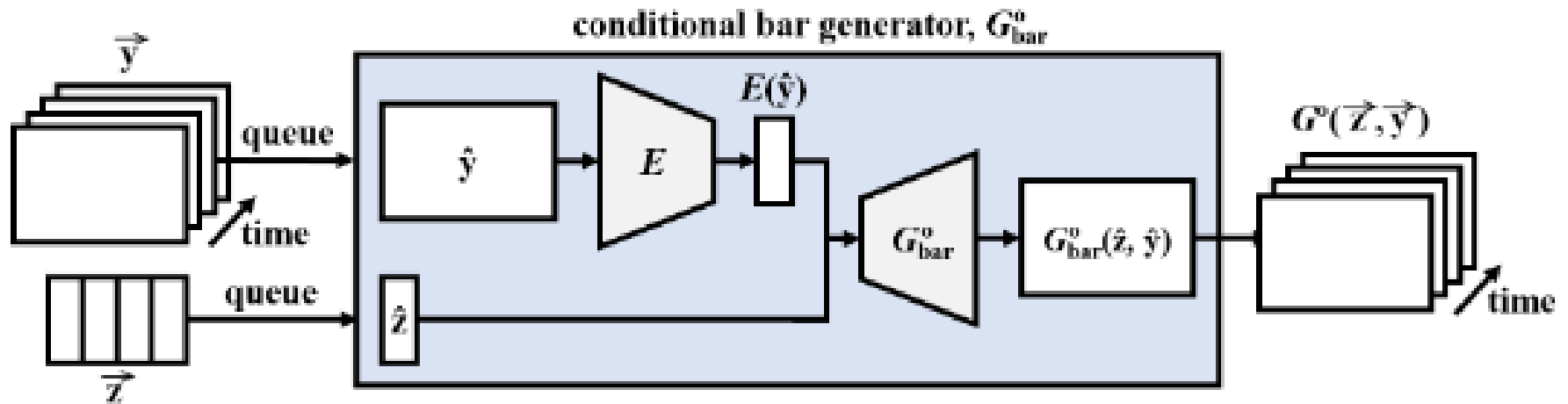
Generation from scratch



Generator 는 G_{temp} , G_{bar} 라는 하위 네트워크로 구성되어 있습니다 G_{temp} 은 시간 정보를 전달할것 같은 부분 벡터의 시퀀스에 노이즈 벡터를 매핑하고 G_{bar} 에서는 피아노 롤을 순차적으로 생성하는데 사용합니다.

| 작곡프로그램 - (Muse GAN)

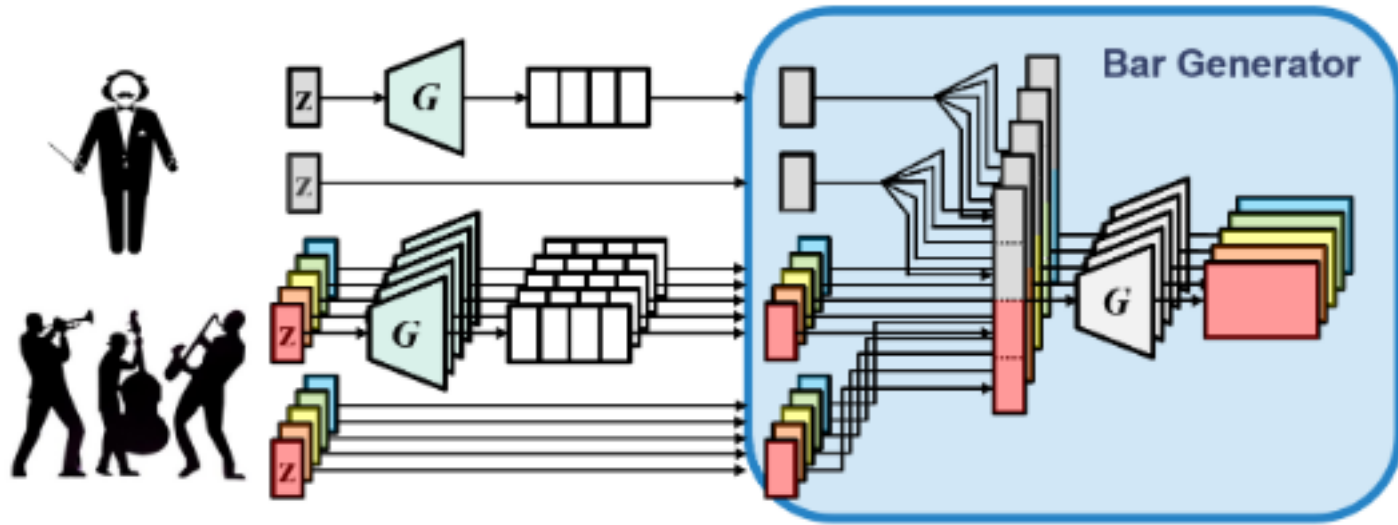
Track-conditional generation



Track generation

G^o 는 트랙과 임의의 노이즈를 입력으로 받는 G^o_{bar} 를 사용하여 막대를 차례로 생성합니다. 고차원 조건으로 이러한 조건부 생성을 달성하기 위해 추가 인코더 E 가 조건을 z 공간에 매핑하도록 훈련됩니다. 인코더는 주어진 트랙에서 피쳐 대신 트랙간의 피쳐를 추출 합니다.

| 작곡프로그램 - (Muse GAN)

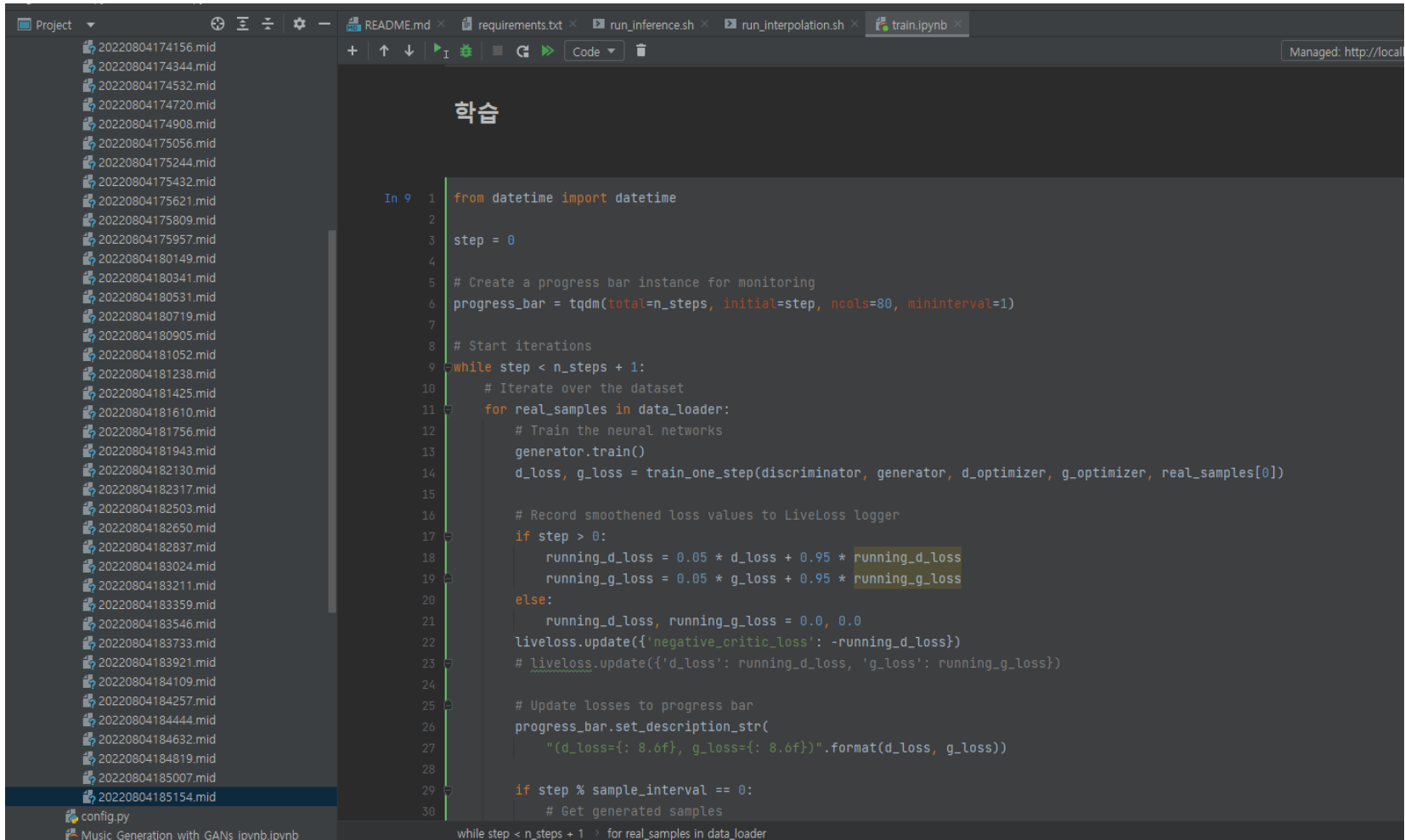


Muse GAN

Muse GAN은 4가지유형의 임의 벡터를 입력합니다.트랙에 대해 공유Gtemp 및 개인 Gtemp 는 각각 벡터 z_t $z_{i,t}$ 를 입력으로 사용합니다. 각각은 트랙간 및 트랙내 시간 정보를 각각 포함하는 벡터를 출력합니다. 출력은 벡터 z 및 z_i 와 함께 연결되어 Bar Generator 에 제공 됩니다. 그후 순차적으로 피아노 롤을 생성하는 G_{bar} 를 통해 다섯가지 세션의 롤이 생성됩니다.

| 작곡 프로그램

학습 결과



```
Project ▾
20220804174156.mid
20220804174344.mid
20220804174532.mid
20220804174720.mid
20220804174908.mid
20220804175056.mid
20220804175244.mid
20220804175432.mid
20220804175621.mid
20220804175809.mid
20220804175957.mid
20220804180149.mid
20220804180341.mid
20220804180531.mid
20220804180719.mid
20220804180905.mid
20220804181052.mid
20220804181238.mid
20220804181425.mid
20220804181610.mid
20220804181756.mid
20220804181943.mid
20220804182130.mid
20220804182317.mid
20220804182503.mid
20220804182650.mid
20220804182837.mid
20220804183024.mid
20220804183211.mid
20220804183359.mid
20220804183546.mid
20220804183733.mid
20220804183921.mid
20220804184109.mid
20220804184257.mid
20220804184444.mid
20220804184632.mid
20220804184819.mid
20220804185007.mid
20220804185154.mid
config.py
Music Generation with GANs ipynb.ipynb

README.md requirements.txt run_inference.sh run_interpolation.sh train.ipynb
+ ↑ ↓ ↻ ⌂ Code ⌵
Managed: http://local

학습

In 9 1 from datetime import datetime
2
3 step = 0
4
5 # Create a progress bar instance for monitoring
6 progress_bar = tqdm(total=n_steps, initial=step, ncols=80, mininterval=1)
7
8 # Start iterations
9 while step < n_steps + 1:
10     # Iterate over the dataset
11     for real_samples in data_loader:
12         # Train the neural networks
13         generator.train()
14         d_loss, g_loss = train_one_step(discriminator, generator, d_optimizer, g_optimizer, real_samples[0])
15
16         # Record smoothened loss values to LiveLoss logger
17         if step > 0:
18             running_d_loss = 0.05 * d_loss + 0.95 * running_d_loss
19             running_g_loss = 0.05 * g_loss + 0.95 * running_g_loss
20         else:
21             running_d_loss, running_g_loss = 0.0, 0.0
22         live_loss.update({'negative_critic_loss': -running_d_loss})
23         # live_loss.update({'d_loss': running_d_loss, 'g_loss': running_g_loss})
24
25         # Update losses to progress bar
26         progress_bar.set_description_str(
27             "(d_loss={: 8.6f}, g_loss={: 8.6f})".format(d_loss, g_loss))
28
29         if step % sample_interval == 0:
30             # Get generated samples
31             while step < n_steps + 1:
32                 for real_samples in data_loader:
```

| 작곡 프로그램

Vision

- Muse-GAN



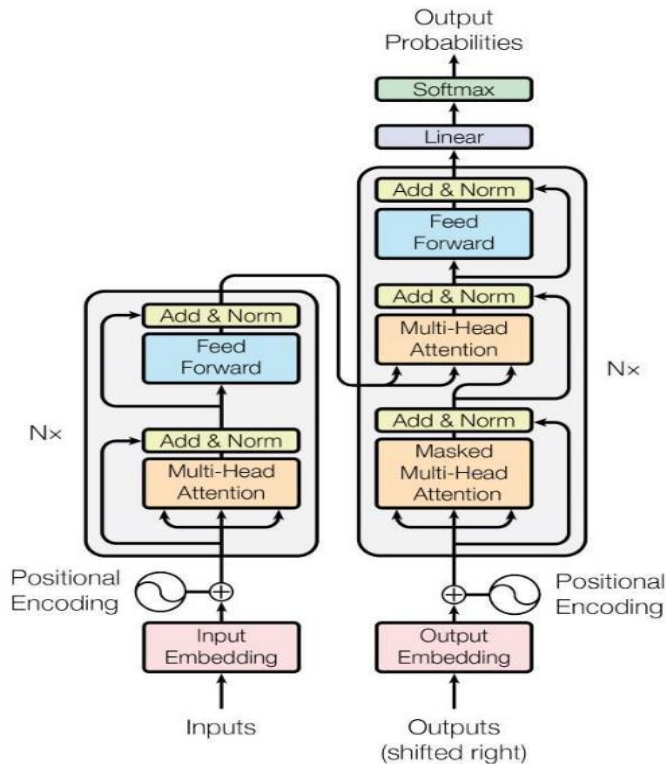
[회사소개](#) ▾ [게시판](#) ▾ [음악 프로그램](#) ▾ ○ [표절/작곡/작사 프로그램](#) ○ [관련 사이트](#)



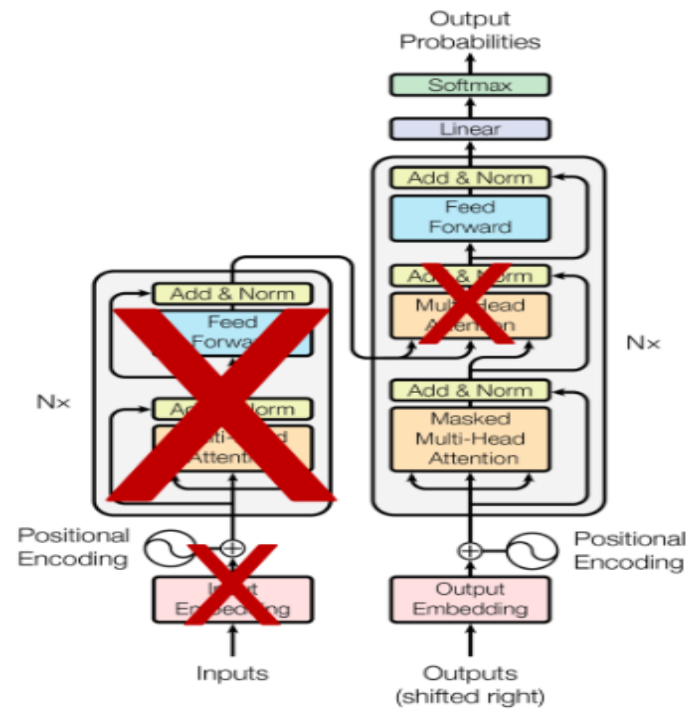
|작사 프로그램 (NLP – KOGPT-3 PLM)

PLM

- 다음 단어 맞추기에 적합한 GPT를 선택



<Transformer>

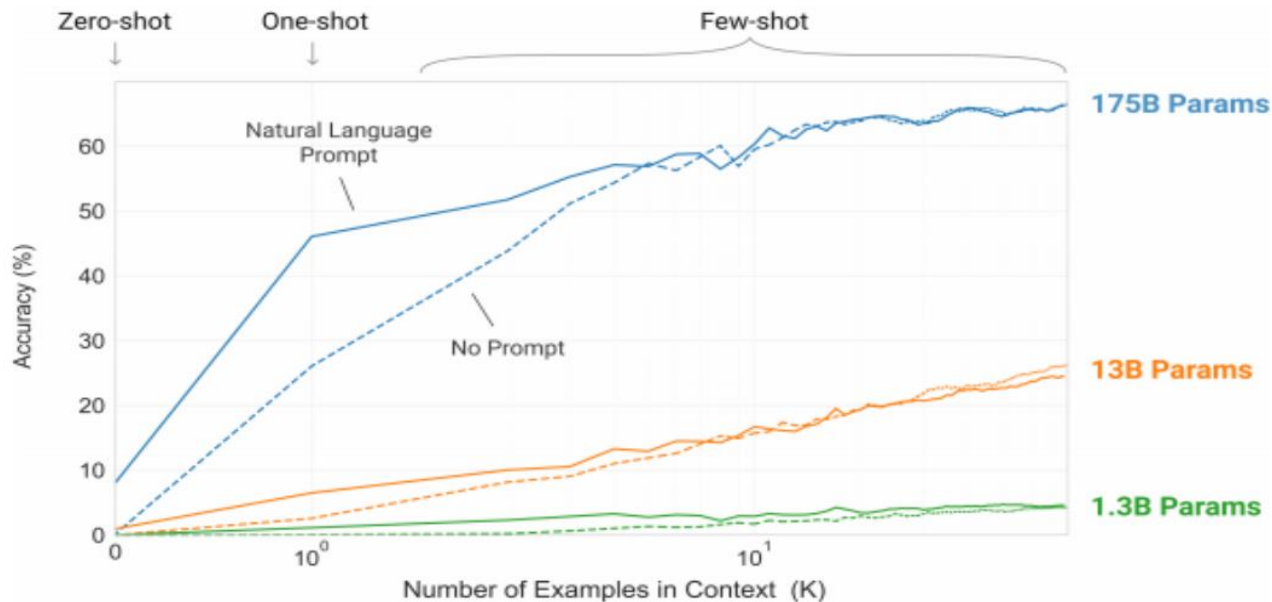


<GPT>

|작사 프로그램 (NLP – KOGPT-3)

장점

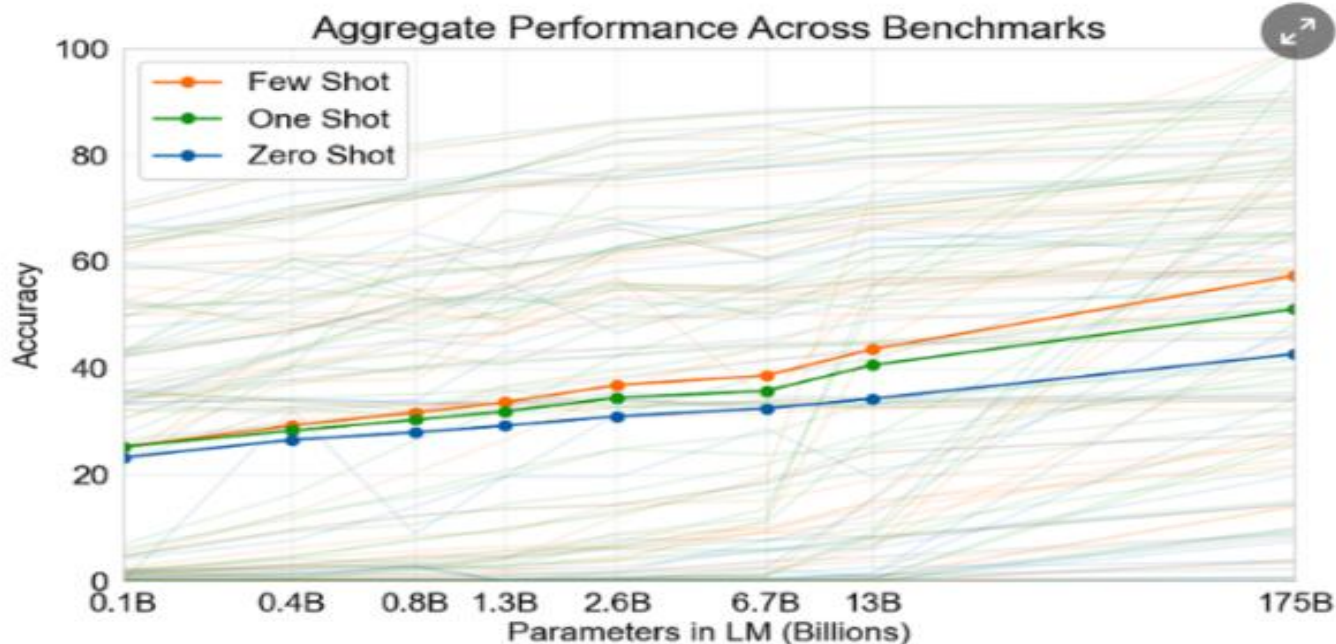
- GPT2 = 800만개 데이터 셋 15억개 파라미터
- GPT3 = 3000억개 데이터셋 1750억개 파라미터
- Fine-Tuning에서 필요한 많은 데이터와 그 많은 데이터들을 학습시킬 시간이 필요
- 하지만 Few-Shot-Learning은 몇 개의 샘플만 던져도 좋은 성능을 냄.(그래프)



|작사 프로그램 (NLP – KOGPT-3)

단점

- 성차별, 인종차별 발언도 포함되어 있음
- Transformer기반 기억개념 없음, 현재의 문맥을 입력해 주어야 그에 맞는 OUTPUT 발생
- 범용적인 다양한 문제만 잘 풀어나감(그래프)
- 데이터양과 비례하는 모델
- Google Colab(Pro) 사용(개발환경의 한계)



|작사 프로그램 (CRAWLING-CODE)

CRAWLING

- 동요 가사가 올려져 있는 블로그에서 크롤링 진행

```
1  import requests
2  from bs4 import BeautifulSoup
3  import pandas as pd
4
5  url = 'https://m.cafe.daum.net/kidchoir/5Hf2/7?listURI=%2Fkidchoir%2F5Hf2'
6
7  response = requests.get(url)
8
9  if response.status_code == 200:
10     html = response.text
11     soup = BeautifulSoup(html, 'html.parser')
12     title = soup.select_one('#mArticle > div.view_info')
13     data=pd.DataFrame(title)
14     data.to_csv('동요가사.csv', encoding='utf-8')
15 else :
16     print(response.status_code)
```

|작사 프로그램 (CRAWLING-OUTPUT)

CRAWLING

■ 총 157곡의 동요가사

동요약보모음.csv

```
1 ,제목,가사
2 0,BINGO,옆집에 사는 개이를 빙고라지요 / BINGO / BINGO / BINGO / 빙고 개 이름
3 1,Edelweiss 에델바이스,"Edelweiss, Edelweiss, Every morning you greet me. Small and white, clean and bright, You look happy to meet me. Blossom c
4 2,Ten Little Indians,"One little, two little, Three little Indians. Four little, five little, Six little Indians. Seven little, eight little,Nine
5 3,TV 유치원 하나 둘 셋,천구야 손 잡고 하나 둘 셋 엄마도 아빠도 하나 둘 셋 우리는 언제나 하나 둘 셋 정말 정말 좋아요 하나 둘 셋 풍선 타고 동동동 구름 타
6 4,가게놀이,가게놀이 할 사람 모두모여라 가게놀이 할 사람 모두모여라 사과 한개 주세요 1000원입니다. 굴 하나 주세요 그것도 1000원이죠 그럼 모두다 얼마인
7 5,가워바워보(씨씨씨),아침 바람 찬 바람에 울고 가는 저 기러기 우리 선생 게실 적에 었서 한 장 씨 주세요 한 장 말고 두 장이요 두 장 말고 세 장이요 구리구
8 6,강아지,우리집 강아지는 목줄 강아지 어머니가 빨래 가면 멍멍멍 풀랑풀랑 따라가며 멍멍멍 우리집 강아지는 예쁜 강아지 학교 갔다 돌아오면 멍멍멍 꼬리치고
9 7,개구리,개굴개굴 개구리 노래를 한다 아들 손자 며느리 다 모여서 밤새도록 하여도 듣는 이 없네 듣는 사람 없어도 날이 밝도록 개굴개굴 개구리 노래를 한다 기
10 8,개구리,엄마 개구리가 노래 부른다 짹짹 짹짹짹짹 짹짹짹짹 짹짹짹짹 짹짹짹짹 짹짹짹짹 짹짹짹짹 짹짹짹짹 짹짹짹짹 짹짹짹짹 짹짹짹짹 짹짹짹짹 짹짹짹짹
11 9,겨울바람,나무야 나무야 겨울 나무야 눈 쌓인 응달에 외로이 서서 아무도 찾지 않는 추운 겨울을 바람 따라 휘파람만 불고 있느냐
12 10,겨울바람,손이 시려워 (공) 발이 시려워 (공) 겨울 바람 때문에 (공공공) 손이 공공공 (공) 발이 공공공 (공) 겨울 바람 때문에 (공공공) 어디서 이 바람은 사
13 11,고기 잡이,고기를 잡으러 바다로 갈까나. 고기를 잡으러 강으로 갈까나. 이 병에 가득히 넣어 가지고서 라라라라 라라라라 온다나. 선생님 모시고 가고 싶지만
14 12,고드름,고드름 고드름 수정 고드름 고드름 따다가 발을 엮어서 각지방 영창에 달아 놓아요
15 13,고향의 봄,1. 나의 살던 고향은 꽃피는 산골 뭍송아꽃 살구꽃 아기진달래 울긋불긋 꽃 대궐 차리인 동네 그 속에서 놀던 때가 그립습니다 2. 꽃동네 새 동네 나
16 14,곰 세 마리,곰 세 마리가 한 집에 있어 아빠곰 엄마곰 아기곰 아빠곰은 똥똥해 엄마곰은 날씬해 아기곰은 너무 귀여워 으쓱으쓱 잘한다
17 15,과꽃,올해도 과꽃이 피었습니다 꽃밭 가득 예쁘게 피었습니다 누나는 과꽃을 좋아했지요 꽃이 피면 꽃밭에서 아주살았조 과꽃 예쁜 꽃을 들여다 보면 꽃 속에
18 16,과수원 길,동구 밖 과수원길 아카시아 꽃이 활짝 폈네 하얀 꽃 이파리 눈송이처럼 날리네 향긋한 꽃냄새가 실바람타고 솔솔 돌이서 말이 없네 얼굴 마주 보며 생
19 17,귀여운 꼬마,귀여운 꼬마가 닭장에 가서 암탉을 잡으려다 놓쳤네 닭장 밖에 있던 배고픈 여우 울거나 하면서 물고 갔다네 꼬꼬덕 암탉 소리를 쳤네 꼬꼬덕 암
20 18,그대로 멈춰라,즐겁게 춤을 추다가 그대로 멈춰라 즐겁게 춤을 추다가 그대로 멈춰라 눈도 감지 말고 웃지도 말고 울지도 말고 움직이지 마 즐겁게 춤을 추다가
21 19,기차길 옆,기차길 옆 오막살이 아기 아기 잘도 잔다 칙칙 칙칙폭폭 칙칙폭폭 칙칙폭폭 기차소리 요란해도 아기 아기 잘도 잔다 기차길 옆 옥수수밭 옥수수는 잘도
22 20,기차를 타고,1. 기차타고 신나게 달려가보자 높은 산도 지나고 넓은 들도 지나고 푸른산을 지날때엔 산새를 찾고 넓은 바다 지날때엔 물새와 놀고 설레임을 가득
23 21,기차길 옆,기차길 옆 오막살이 아기 아기 잘도 잔다 칙 칙 칙칙 폭폭 칙칙폭폭 칙칙폭폭 기차소리 요란해도 아기 아기 잘도 잔다
24 22,"깊은 계곡, 깊은 계곡 광산 마을 동굴 집에",너는 아빠 여여쁜 딸이 사랑으로 살았네 오 내사랑 오 내사랑 나의 귀여운 클레멘타인 너는 영영 가버리고 나만 홀
25 23,장강총체조,손을 높이 손을 높이 쪽쪽쪽 쪽쪽 뻗어 봐요 발을 쿵쿵 발을 쿵쿵 쿵쿵쿵 쿵쿵 굴러 봐요 엉덩이를 실룩 실룩쉴룩 이쪽 저쪽 실룩쉴룩 뱅글뱅글 뱅
26 24,꼬까신,개나리 노오란 꽃 그늘 아래 가지런히 놓여 있는 꼬까신 하나 아기는 살짝 신벗어 놓고 맨발로 한들 한들 나들이 갔나 가지런히 가다리는 꼬까신 하나
27 25,꼬마누사람,한겨울에 밀짚모자 꼬마 누사람 눈썹이 우습구나 코도 비뚤고 거울을 보여줄까 꼬마 누사람
28 26,꼬마자동차 뽕뽕,뽕뽕뽕 아주 작은 자동차 꼬마 자동차가 나왔다 뽕뽕뽕 꽃향기를 맡으면 힘이 솟는 꼬마 자동차 엄마 찾아 모험 찾아 나서는 세계 여행 우리
29 27,꼬부랑 할머니,꼬부랑 할머니가 꼬부랑 고갯길을 꼬부랑 꼬부랑 넘어가고 있네 꼬부랑 꼬부랑 꼬부랑 꼬부랑 고개는 열두 고개 고개를 고개를 넘어간다
30 28,꼭꼭약속해,너하고 나는 친구 되어서 사이좋게 지내자 새끼손가락 고리 걸고 꼭꼭 약속해
31 29,꽃밭에서,아빠하고 나하고 만든 꽃밭에 채송화도 봉숭아도 한창입니다 아빠가 매어놓은 새끼줄 따라 나팔꽃도 어울리게 피었습니다
```

|작사 프로그램 (NLP – KOGPT-3)

Code

- Few-shot-Learning 실행
- Temperature의 값은 양수여야 한다.

```
1 import torch
2 from transformers import AutoTokenizer, AutoModelForCausalLM
3
4 tokenizer = AutoTokenizer.from_pretrained(
5     'kakaobrain/kogpt', revision='KoGPT6B-ryan1.5b-float16', # or float32 version: revision=KoGPT6B-ryan1.5b
6     bos_token='[BOS]', eos_token='[EOS]', unk_token='[UNK]', pad_token='[PAD]', mask_token='[MASK]'
7 )
8 model = AutoModelForCausalLM.from_pretrained(
9     'kakaobrain/kogpt', revision='KoGPT6B-ryan1.5b-float16', # or float32 version: revision=KoGPT6B-ryan1.5b
10    pad_token_id=tokenizer.eos_token_id,
11    torch_dtype='auto', low_cpu_mem_usage=True
12 ).to(device='cuda', non_blocking=True)
13
14 _ = model.eval()
15
16
17 #prompt: 입력, 최대 2048 tokens (1500 단어)
18 #completion: 생성된 출력, 최대 2048 tokens (1500 단어)
19 #tokens: 말뭉치 조각 - 단어와 비슷함, 많은 토큰이 공백으로 시작함 " hello"
20 #prompt 끝부분에 공백을 놔두면 안됨.
21 prompt = '''
22 동요 노래 가사
23
24 1.파란 하늘 파란 하늘 꿈이 드리운 푸른 연덕에 아가 열소 여럿이 붓을 펼고 놓아요 해처럼 밝은 얼굴로
25
26 2.학교 종이 땡땡땡 어서모이자 선생님이 우리를 기다리신다
27
28 3.산토끼 토끼야 어디를 가느냐 짹짹 뛰면서 어디를 가느냐
29
30 4.떴다 떴다 비행기 날아라 날아라 높이 높이 날아라 우리 비행기
31
32 5.새싹을 싹고 뿔어보자 팔짝 머리가 하늘까지 닿겠네
33
34 6.곰 세 마리가 한 집에 있어 아바곰 엄마곰 애기곰 아바곰은 웁웁해 엄마곰은 날쾅해 애기곰은 너무 귀여워 으쓱으쓱 잘한다
35
36 7. '''
37
38 with torch.no_grad():
39     tokens = tokenizer.encode(prompt, return_tensors='pt').to(device='cuda', non_blocking=True)
40     gen_tokens = model.generate(tokens, do_sample=True, temperature=0.8, max_length=350)
41     generated = tokenizer.batch_decode(gen_tokens)[0]
42
43 print(generated)
```

|작사 프로그램 (Temperature=0.2)

0.2

- 도돌이표 느낌이다...

동요 노래 가사

- 1.파란 하늘 파란 하늘 꿈이 드리운 푸른 언덕에 아기 염소 여럿이 풀을 뜯고 놀아요 해처럼 밝은 얼굴로
- 2.학교 종이 땡땡땡 어서모이자 선생님이 우리를 기다리신다
- 3.산토끼 토끼야 어디를 가느냐 깡충깡충 뛰면서 어디를 가느냐
- 4.엎다 엎다 비행기 날아라 날아라 놀이 놀이 날아라 우리 비행기
- 5.새싹을 싣고 뛰어보자 팔짝 머리가 하늘까지 닿겠네
- 6.곰 세 마리가 한 집에 있어 아바곰 엄마곰 애기곰 아바곰은 뽕뽕해 엄마곰은 날씬해 애기곰은 너무 귀여워 으쓱으쓱 잘한다
- 7.아바 곰은 뽕뽕해 엄마 곰은 날씬해 애기 곰은 너무 귀여워 으쓱으쓱 잘한다
- 8.아바 곰은 뽕뽕해 엄마 곰은 날씬해 애기 곰은 너무 귀여워 으쓱으쓱 잘한다
- 9.아바 곰은 뽕뽕해 엄마 곰은 날씬해 애기 곰은 너무 귀여워 으쓱으쓱 잘한다
- 10.아바 곰은 뽕뽕해 엄마 곰은 날씬해 애기 곰은 너무 귀여워 으쓱으쓱 잘한다
- 11.아바 곰은 뽕뽕해 엄마 곰은 날씬해 애기 곰은 너무 귀여워 으쓱으쓱 잘한다
- 12.아바 곰은 뽕뽕해 엄마 곰은 날씬해 애기 곰은 너무 귀여워 으쓱으쓱 잘한다
- 13.아바 곰은 뽕뽕해 엄마 곰은 날씬해 애기 곰은 너무 귀여워 으쓱으쓱 잘한다
- 14.아바 곰은 뽕뽕해 엄마 곰은 날씬해 애기 곰은 너무 귀여워 으쓱으쓱 잘한다
- 15.아바

|작사 프로그램 (Temperature=0.4)

0.4

- 0.2와 크게 다르지 않다...

동요 노래 가사

- 1.파란 하늘 파란 하늘 꿈이 드리운 푸른 언덕에 아기 염소 여럿이 물을 쏘고 놀아요 해처럼 밝은 얼굴로
- 2.학교 종이 땡땡 댜 어서모이자 선생님이 우리를 기다리신다
- 3.산토끼 토끼야 어디를 가느냐 깡충깡충 뛰면서 어디를 가느냐
- 4.떴다 떴다 비행기 날아라 날아라 높이 높이 날아라 우리 비행기
- 5.새싹을 싹고 뿔해보자 팔짝 머리가 하늘까지 닿겠네
- 6.곰 세 마리가 한 집에 있어 아바곰 엄마곰 애기곰 아바곰은 뽕뽕해 엄마곰은 날씬해 애기곰은 너무 귀여워 으쓱으쓱 잘한다
- 7.아바곰은 뽕뽕해 엄마곰은 날씬해 애기곰은 너무 귀여워 으쓱으쓱 잘한다
- 8.아바곰은 뽕뽕해 엄마곰은 날씬해 애기곰은 너무 귀여워 으쓱으쓱 잘한다
- 9.아바곰은 뽕뽕해 엄마곰은 날씬해 애기곰은 너무 귀여워 으쓱으쓱 잘한다
- 10.아바곰은 뽕뽕해 엄마곰은 날씬해 애기곰은 너무 귀여워 으쓱으쓱 잘한다
- 11.아바곰은 뽕뽕해 엄마곰은 날씬해 애기곰은 너무 귀여워 으쓱으쓱 잘한다
- 12.아바곰은 뽕뽕해 엄마곰은 날씬해 애기곰은 너무 귀여워 으쓱으쓱 잘한다
- 13.아바곰은 뽕뽕해 엄마곰은 날씬해 애기곰은 너무 귀여워 으쓱으쓱 잘한다
- 14.아바곰은 뽕뽕해 엄마곰은 날씬해 애기곰은 너무 귀여워 으쓱으쓱 잘한다
- 15.아빠

|작사 프로그램 (Temperature=0.6)

0.6

- 뭔가 내용이 달라지기 시작하다가 또 똑같다...

종요 노래 가사

- 1.파란 하늘 파란 하늘 꿈이 드리운 푸른 언덕에 마기 열소 여럿이 풀을 뜯고 놀아요 해처럼 밝은 얼굴로
- 2.학교 종이 땡땡 땡 땡 모미자 선생님이 우리를 기다리신다
- 3.산토끼 토끼야 어디를 가느냐 깡충깡충 뛰면서 어디를 가느냐
- 4.으다 으다 비행기 날아라 날아라 놀이 놀이 날아라 우리 비행기
- 5.새싹을 싹고 뛰어보자 팔짝 머리가 하늘까지 닿겠네
- 6.곰 세 마리가 한 집에 있어 아바곰 얼마곰 애기곰 아바곰은 뽕뽕해 얼마곰은 날씬해 애기곰은 너무 귀여워 으쓱으쓱 잘한다
- 7.아바 곰은 뽕뽕해 얼마 곰도 뽕뽕해 애기 곰은 너무 귀여워 으쓱으쓱 잘한다
- 8.달나라에도 별나라에도 갈 수가 있단다 .
- 9.나는 나는 자라서 무엇이 될까
- 10.나는 나는 자라서 무엇이 될까
- 11.나는 나는 자라서 무엇이 될까
- 12.나는 나는 자라서 무엇이 될까
- 13.나는 나는 자라서 무엇이 될까
- 14.나는 나는 자라서 무엇이 될까
- 15.나는 나는 자라서 무엇이 될까
- 16.나는 나는 자라서 무엇이 될까
- 17.나는 나는 자라서 무엇이 될까
- 18.나는 나는 자라서 무엇이 될까
- 19.나는 나는 자라서 무엇이 될까
- 20.나는 나는 자라서 무엇이 될까

|작사 프로그램 (Temperature=1.0)

1.0

- 시를 쓰는 것 같다...

제 1 부

[얼마아]

제 1 장 아빠 꿈은 꿀꿀해

제 2 장 아빠 꿈은 날씬해

제 1 부

제 1 장 아빠 꿈은 꿀꿀해

얼마아

"어, 이게 뭐야? 웬 벌레가?"

"벌레가 아니라 나비예요."

"무슨 소리?"

"나비가 아니라 잠자리한 말이에요."

"어허, 이런 바보 같으니."

"아빠, 난 나비가 아니라 잠자리한 말이거든."

|작사 프로그램 (Temperature=0.8)

최종 Out-Put

- Temperature의 조절 끝에 0.8로 지정

동요 노래 가사

- 1.파란 하늘 파란 하늘 꿈이 드리운 푸른 언덕에 아기 염소 여럿이 풀을 뜯고 놀아요 해처럼 밝은 얼굴로
- 2.학교 종이 땡땡 땡 어서모이자 선생님이 우리를 기다리신다
- 3.산토끼 토끼야 어디를 가느냐 강총강총 뛰면서 어디를 가느냐
- 4.떴다 떴다 비행기 날아라 날아라 높이 높이 날아라 우리 비행기
- 5.새싹을 싹고 뿔해보자 팔짝 머리가 하늘까지 닿겠네
- 6.곰 세 마리가 한 집에 있어 아빠곰 엄마곰 애기곰 아빠곰은 뽕뽕해 엄마곰은 날씬해 애기곰은 너무 귀여워 으쓱으쓱 잘한다
- 7.구름을 잡아라 냇물을 따라 가면 내가 놀던 물이 흘러 어디로 가는지 알 수 있지
- 8.햇님이 방긋 웃으면 나는야 햇님이 얼굴
- 9.비 오는 날 우산 속 얼굴 마주하고 너랑 나랑 둘이 비를 맞네
- 10.새나라의 어린이는 일찍 일어납니다 학교 가는 길에 달그락 달그락
- 11.눈 오는 날 눈 싸움을 해요 동무들과 모여서 눈 싸움을 해요 눈 싸움을 해요
- 12.누나 손을 잡고서 노래를 불러요 동동동동 손을 잡고서
- 13.산에 올라요 산에 올라요 푸른 하늘 푸른 한울 푸른 언덕에
- 14.이 다음에 커서 어른 되면 나도 어른 되면
- 15.저 건너 푸른 언덕에 아기 염소 여럿이 풀을 뜯고 놀아요 해처럼 밝은 얼굴로

| 작사 프로그램

NLP

■ KOGPT-3 사용



[회사소개](#) ▾ [게시판](#) ▾ [음악 프로그램](#) ▾ [표절/작곡/작사 프로그램](#) ○ [관련 사이트](#)

작사 키워드

동요

작사하기

동요 노래 가사 :

- 1.파란 하늘 파란 하늘 꿈이 드리운 푸른 언덕에 아기 염소 여럿이 풀을 뜯고 놀아
요 해처럼 밝은 얼굴로
- 2.학교 종이 땡땡땡 어서모이자 선생님이
우리를 기다리신다
- 3.산토끼 토끼야 어디를 가느냐 강총강총
뛰면서 어디를 가느냐

감사합니다

Q & A

스프린트 PO : 서성민(Next 개발자)
스프린트기간: 5/23 ~ 6/10

- 스프린트 목표
- 1. 화면 1 차 완성
 - 2. 데이터베이스 스키마 완성

스프린트 상세:

1 주차 (5/23 ~ 5/27)

<서성민>

- 전체 화면 관리
- ~~Home (로그인 전) 화면~~
- ~~Home (로그인 후) 화면~~

<권혜민>

- ~~음원 검색 화면 컴포넌트 작성~~
- ~~음원 검색 중 화면 컴포넌트 작성~~
- ~~음원 검색 확인 화면 컴포넌트 작성~~
- ~~사용자 (회원가입 , 로그인) 화면 컴포넌트 작성~~

<조현국>

- 악보 분석 등록 컴포넌트 작성
- 악보 분석 확인 컴포넌트 작성
- 회사 → 개발자들 컴포넌트 작성

2 주차 (5/30 ~ 6/3)

<서성민>

- 타임스크립트 공부
- 자바스크립트 공부
- 화면 발표 완료

<권혜민>

- ~~[x] 음성인식 기술 STT, TTS 알고리즘 찾아보기~~

관련링크:<https://penguin-story.tistory.com/1>

<조현국>

- Opencv 를 이용한 악보인식 티스토리 분석 후 정리하기
- 관련링크: <https://hackids.tistory.com/120>

2 주차 회의 내용

- 챗봇을 음성 챗봇인 휴먼 AI 로 대체 , 질문지 제공
- 프론트에서 웹 장고 제거 (서버리스)
- 음악 30 초 미리 듣기
- 화면 시나리오 구성
- 튜토리얼 각 화면마다 구성
- 문의하기 크롤링, 게시판 하드 코딩

최종 파트 분담

조현국 : NLP / KOBERT / KOGPT

권혜민 : Muse GAN / VIT / CV

서성민 : 프론트 화면 구성 / 작곡 (tone.js)

3 주차 회의 내용

DB 수집 계획

개인 데이터는 각각 들고 있기

1. 이미지 악보 (디지털 악보) : 가요 악보 5 개, 동요 악보 5 개 (한 손)
2. mp3 파일: 인기 가요 5 개
3. 휴먼 AI : GIF 파일로 은아 사진 캐릭터화
4. 음성 챗봇 데이터 : AI 허브에서 음성 챗봇용 데이터 수집

3 주차 (6/6 ~ 6/10)

<서성민>

- 타임스크립트 공부 및 화면 마감 계획 짜놓기
- 공통요소 (게시판) 코드 분석하기
- Javascript 로 짜여진 팀 프로젝트 템플릿 부분
경량화 및 Typescript 로 녹이는 거 고민하기

<권혜민>

- 퍼신러닝/러닝 개념 배우기
- 지도학습 / 비지도학습 / 준지도학습 개념 공부

<조현국>

- RNN 개념 공부하기
- 손실함수와 활성화함수에 대해 공부

스프린트 PO : 권혜민 (Python 개발자)

스프린트기간: 6/13 ~ 7/17

스프린트 목표

3. 화면 1 차 완성
4. 데이터베이스 스키마 완성

스프린트 상세:

4 주차 (6/13 ~ 6/17)

<서성민>

- javascript 에 음악 인식하는 작업 분석
관련링크
<https://www.youtube.com/watch?v=ahts3efFzFI>
- 악보 이미지 데이터 수집할 사이트 찾아보기
https://imslp.org/wiki/Main_Page

<권혜민>

- 신경망, 생물학적 뉴런/인공뉴런
- 활성화 함수 / 손실함수

<조현국>

- 전이학습(Fine-Tuning) 공부

5 주차 (6/20 ~ 6/24)

<서성민>

- 악보 데이터 수집
- tone.js framework 연구하기
관련링크
<https://tonejs.github.io/>

<권혜민>

- 경사 하강법 최적화 알고리즘
- 역전파와 자동미분
- 확률적 경사하강법
- 미니배치 경사하강법

<조현국>

- PLM 의 종류와 트랜스포머가 사용되게 된 역사공부
- 주변 단어를 예측하는 WordEmbedding 에 대한
공부(word2vec)
- context 에 따른 Embedding 에 대해 공부(ELMo)

6 주차 (6/27 ~ 7/1)

<서성민>

- Tonejs (Web Audio Framework) 활용하여 react 로
피아노 화면 구현
- Javascript + React 로 화면 구현
- 타임스크립트를 활용한 팀 프로젝트 화면 스타일
연구.

<권혜민>

- 일반화 / 드롭아웃
- 드롭아웃 동작방식 / 역동작방식
- 데이터 증강 / 일리스타핑 / 배치 정규화

<조현국>

- 기계 번역 모델 Seq2seq 에 대해 공부
- 입력 문장이 길어도 소실을 줄이는 Attention 에 대해 공부

7 주차 (7/4 ~ 7/8)

<서성민>

- Javascript 를 Typescript 로 전환하여 화면 구현
- ~~tone - piano 에 note (음) json 형태로 저장할 수 있는 방법 연구~~

<권혜민>

- 텐서플로 데이터 흐름 그래프
- 정적 그래프 변수 / ~~tf.layers~~ 기반 모델 정의
- ~~자동미분 - 손실과 옵티마이저~~

<조현국>

- Attention 연산 기반 트랜스포머의 내부 encoding/decoding 공부(multi head attention)
- Multi head Attention 의 기본 Quert 의 유사한 Key 를 찾아 value 를 얻어내는 과정 공부
- 디코더의 대표모델 GPT 연구

8 주차 (7/11 ~ 7/15)

<서성민>

- Redux 를 활용하여 게시판 구현
- ~~팀 프로젝트 스타일 마무리~~

<권혜민>

- 이미지 분류를 위한 신경망/합성곱 신경망 공부
- ~~Google Colab 을 통한 전이학습과 파인튜닝예제 화가~~
- 감정 분류를 위한 신경망 공부

<조현국>

- 트랜스포머 Tokenizer 활용하여 문장 토큰화 / 단어 토큰화
- 트랜스포머 Tokenizer 활용하여 단어 임베딩 해보기
- 인코더 대표모델 BERT 연구

스프린트 PO : 조현국(Python 개발자)

스프린트기간: 7/18 ~ 8/9

스프린트 목표

5. 화면 1 차 완성
6. 데이터베이스 스키마 완성

스프린트 상세:

9 주차 (7/18 ~ 7/22)

<서성민>

- ~~FileUpload~~ 기능 구현
- ~~redux~~ 를 활용하여 만든 게시판과 서버 연결하기 위해 연구중. (~~openapi~~)
- ~~화면에서 파일 업로드한 이미지 openapi (server) 로 전달하기 위해 연구중.~~

<권혜민>

- ~~오토인코더 공부하기~~
- ~~이미지 시각화 하기 / 전처리~~
- 선형 오토인코더 구축하고 학습시키기
- ~~Music 21 을 이용해 midi 파일 npy 로 변환하는 코드 작성~~

<조현국>

- 트랜스포머 논문 분석하기
- 개인프로젝트 감성분석 챗봇 모델 돌려보기

10 주차 (7/25 ~ 7/29)

<서성민>

- ~~FileUpload~~ 기능 구현
- ~~redux~~를 활용하여 만든 게시판과 서버 연결하기 위해 연구중. (~~openapi~~)
- 화면에서 파일 업로드한 이미지 ~~openapi (server)~~로 전달하기 위해 연구중.

<권혜민>

- ~~합성곱 오토인코더 예제 구현하기~~
- ~~Muse GAN~~에서 ~~pretrained models~~ 가져오기
- ~~생성자 구현하기 / 구분자 구현하기~~
- ~~YOLO~~로 객체 감지 / 탐지기 테스트
- ~~DarkNet~~ 프레임워크 사용하기

<조현국>

- 트랜스포머 논문 및 KOBERT, KOGPT 코드 분석하기

11 주차 (8/1 ~ 8/5)

<서성민>

- ~~FileUpload~~ 기능 구현
- ~~redux~~를 활용하여 만든 게시판과 서버 연결하기 위해 연구중. (~~openapi~~)
- 화면에서 파일 업로드한 이미지 ~~openapi (server)~~로 전달하기 위해 연구중.

<권혜민>

- ~~Muse GAN~~ ~~pytorch~~로 구현하기
- 데이콘 데이터 받아서 구현하기
- ~~Self-attention~~ 동작원리 공부하기
- ~~VIT~~ 논문 공부 / 리뷰 하기

<조현국>

- 트랜스포머 논문 분석하기
- KOGPT-2를 이용한 가사 분석하기
- 분석된 가사 전이학습, 파인튜닝하기
- 프로젝트 발표자료 PPT 준비하기
- 이력서 및 개발보고서 준비하기

12 주차 (8/8 ~ 8/9)

<서성민>

- ~~FileUpload~~ 기능 구현
- ~~redux~~를 활용하여 만든 게시판과 서버 연결하기 위해 연구중. (~~openapi~~)
- 화면에서 파일 업로드한 이미지 ~~openapi (server)~~로 전달하기 위해 연구중.

<권혜민>

- ~~Muse GAN~~ 구현 기능 점검하기
- ~~YOLO~~ 논문 분석 정리하기

<조현국>

- 작사하기 기능 최종 점검
- ~~Transformer~~ 논문 분석 정리하기(이력서에 추가)