



인공지능 기반 가사 생성 모델

작사가 에이나

5조_너의이야기를 들려조

목차

a table of contents

Part1. 서론

- 시장분석
- 서비스흐름도
- 수익모델

Part2. 분석

- 전처리
- 모델링
- 가사 표절 검사

Part3. 서비스

- 서비스

Part4. 결론

- 가사생성 평가결과
- 개선사항
- 추후 연구방향





Part 1 서론

- 시장분석
- 서비스흐름도
- 수익모델



사이버가수 **아담**을 아시나요

첫 앨범부터 20만장의 판매고를 올린 화제의 주인공이자
2집의 실패와 함께 바이러스 사망설을 남기고 조용히 사라진 비운의 주인공
대한민국 최초의 버추얼 휴먼



이번엔

진짜 다

데뷔 1년만에 몸값 3억, 임영웅 라이벌 된 '로지'

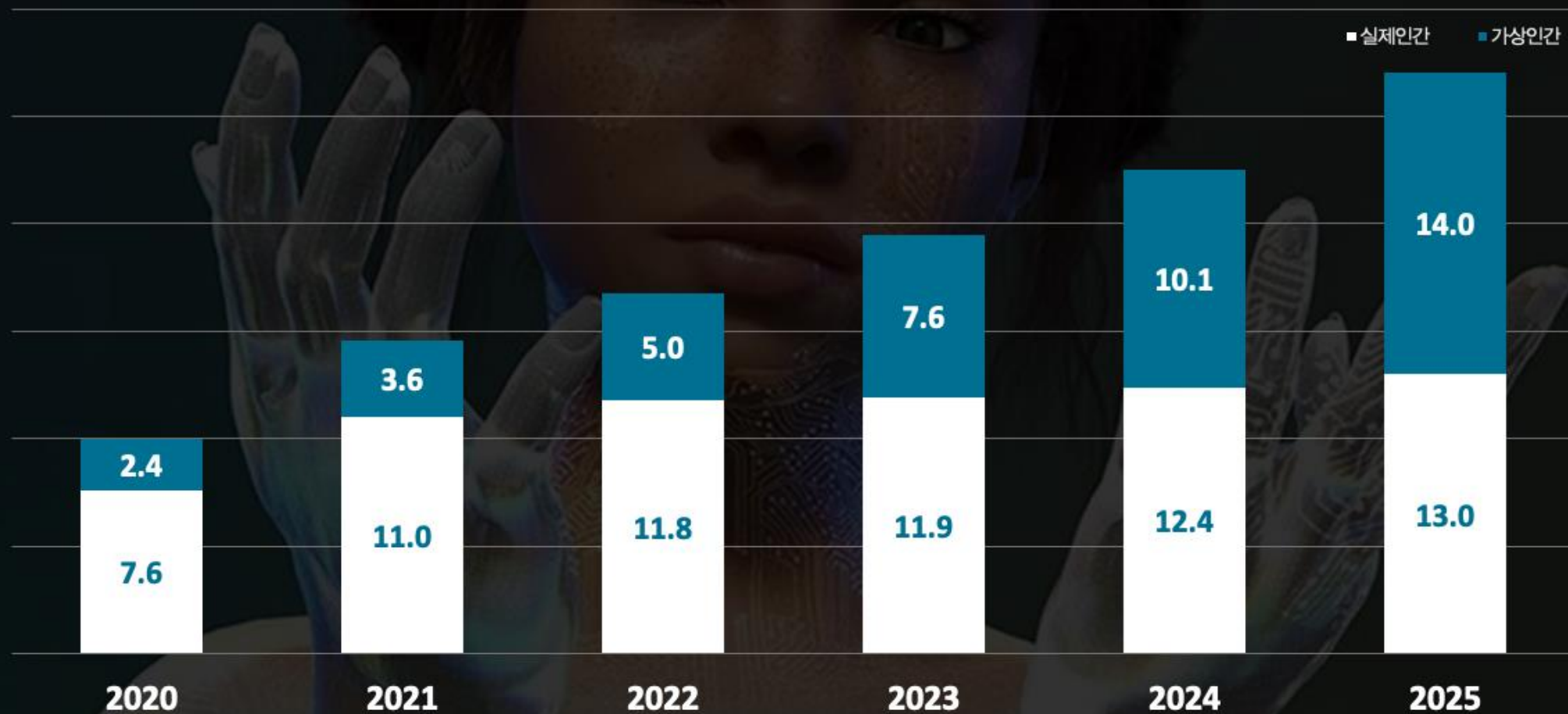
대기업도 줄 섰다... 가상인간 만드는 AI 스타트업 인기

광고 시장 이어 가요계까지... 영역 확장하는 '가상 인간'

TV 속 광고 시장 장악하는 버추얼 휴먼, 높은 인기로 맹활약

버추얼 인플루언서 시장 5년 내 2.4조에서 14조로 성장 전망

전 세계 인플루언서 시장 2020년부터 매년 32.5%씩 성장 기대
2025년 버추얼 인플루언서 시장의 규모는 실제 인플루언서 시장 규모를 압도할 것으로 예상



가수에 도전하는 버추얼 인플루언서

버추얼 엔터테인먼트들은 저변 확대를 위해 적극적으로 음반 시장에 도전 중



LG전자 김래아



싸이더스튜디오 오로지



스마일게이트 한유아



넷마블 리아

1 시장분석



Who Am I

로지(ROZY)

앨범 Who Am I
발매일 2022.02.22
장르 발라드
FLAC Flac 16/24bit

+ 담기

댓글

9개 >

공유



♡ 820 ⬇ 곡 다운 > ⬇ FLAC 다운 > 📁 선물하기 >

작사/작곡



바닐라맨 (바닐라어쿠스틱)

작사



용브로

작사



바닐라맨 (바닐라어쿠스틱)

작곡



Scott B

작곡

버추얼 콘텐츠에 **작사는 사람이?**

“AI 작사가 에이나”

한 줄의 가사만 입력해주면 AI가 완성된 가사를 제공하는 가사생성 모델링 서비스

1 시장분석-경쟁사분석



...



...



VS



...



너의 이야기를 들려조 AI작사가 모델링 **에이나**
'표절'여부 검토 시스템 구축

1 서비스흐름도

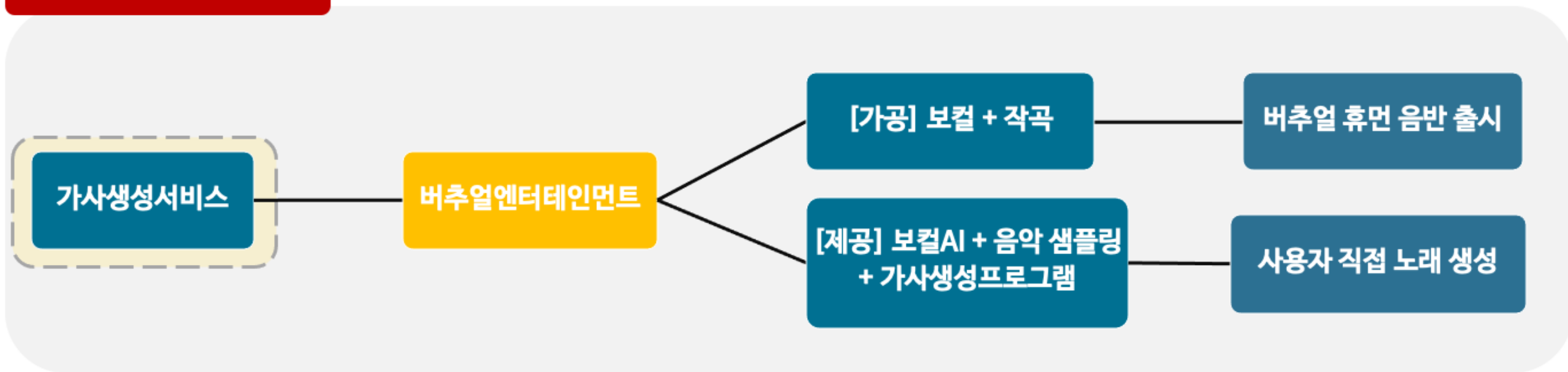
Target

버추얼 엔터테인먼트

Service

AI 기반 자동 가사 생성 서비스

Service flow chart



1 수익모델

Service flow chart



현행 입법구조

□ 제2조(정의) 이 법에서 사용하는 용어의 뜻은 다음과 같다. <개정 2009. 4. 22., 2011. 6. 30., 2011. 12. 2., 2016. 3. 22., 2021. 5. 18.>

1. "저작물"은 인간의 사상 또는 감정을 표현한 창작물을 말한다.
2. "저작자"는 저작물을 창작한 자를 말한다.

- 현행 법에서는 "저작물"을 인간의 사상 또는 감정을 표현한 창작물로 정의
- 인간의 사상 또는 감정이 표출된 실데이터를 학습한 인공지능의 결과물이 인간의 사상 또는 감정을 표출한 창작물인지에 대한 논쟁 존재
- 하지만 시장의 안정과 법의 보수성으로 현재는 관련 법이 존재하지 않다고 해석하거나 AI 창작물을 불인정하는 쪽으로 해석함

수익창출 방안(AS-LS)

- 1 가사생성서비스를 버추얼엔터테인먼트와 '월정액' 혹은 '건당 이용요금' 방식으로 서비스를 대여 및 제공하고자 함
- 2 가사생성서비스 결과물을 2차가공 및 2차 배포시 계약체결 방식에 따라 수익금을 배분 받고자 함

1 수익모델

Service flow chart



해외 입법 동향

AI 저작권에 대한 입법 현황

국가	내용
미국	▶ AI가 만든 작품 저작권 등록 거부(2018.11)(2022.02) ▶ 저작권은 인간 지적 노동 성과물을 보호하는 권리
유럽연합	▶ 로봇(AI) 규제 지침 발표하고 AI 저작권 보호(2014.05) ▶ 로봇 시민권 권고안 통과해 전자인격 부여(2017.02)
일본	▶ AI 창작물 보호 위해 지식재산추진계획 시행(2016.05) ▶ 저작권법 개정으로 학습 데이터 면책조항 도입(2018.05)
한국	▶ AI 저작권 보호 위한 저작권법 일부개정안 발의(2020.12) ▶ 초거대 AI 학습 데이터 저작권 면책 조항 도입 추진(2022.01)

[자료=국내외 기관 발표 자료 취합]

AI 저작권 인정에 대해 **미국**은 **반대** 입장,
유럽연합과 **일본**을 **찬성** 입장

국내 입법 동향

개 정 안
제2조(정의) ----- ----- 1. (현행과 같음) 1의2. "인공지능 저작물"은 외 부환경을 스스로 인식하고 상황을 판단하여 자율적으로 동작하는 기계장치 또는 소 프트웨어(이하 "인공지능"이 라 한다)에 의하여 제작된 창작물을 말한다.
2. (현행과 같음) 2의2. "인공지능 저작물의 저작 자"는 인공지능 서비스를 이 용하여 저작물을 창작한 자 또는 인공지능 저작물의 제 작에 창작적 기여를 한 인공 지능 제작자·서비스 제공자 등을 말한다.

- 국내에서도 AI 저작
물에 대한 우호적인
움직임이 보임
- 2020년 12월 21일,
국민의힘 주호영 의
원 등 11인이 AI의
저작물 개념을 명시
한 '저작권법 일부
개정법률안' 발의

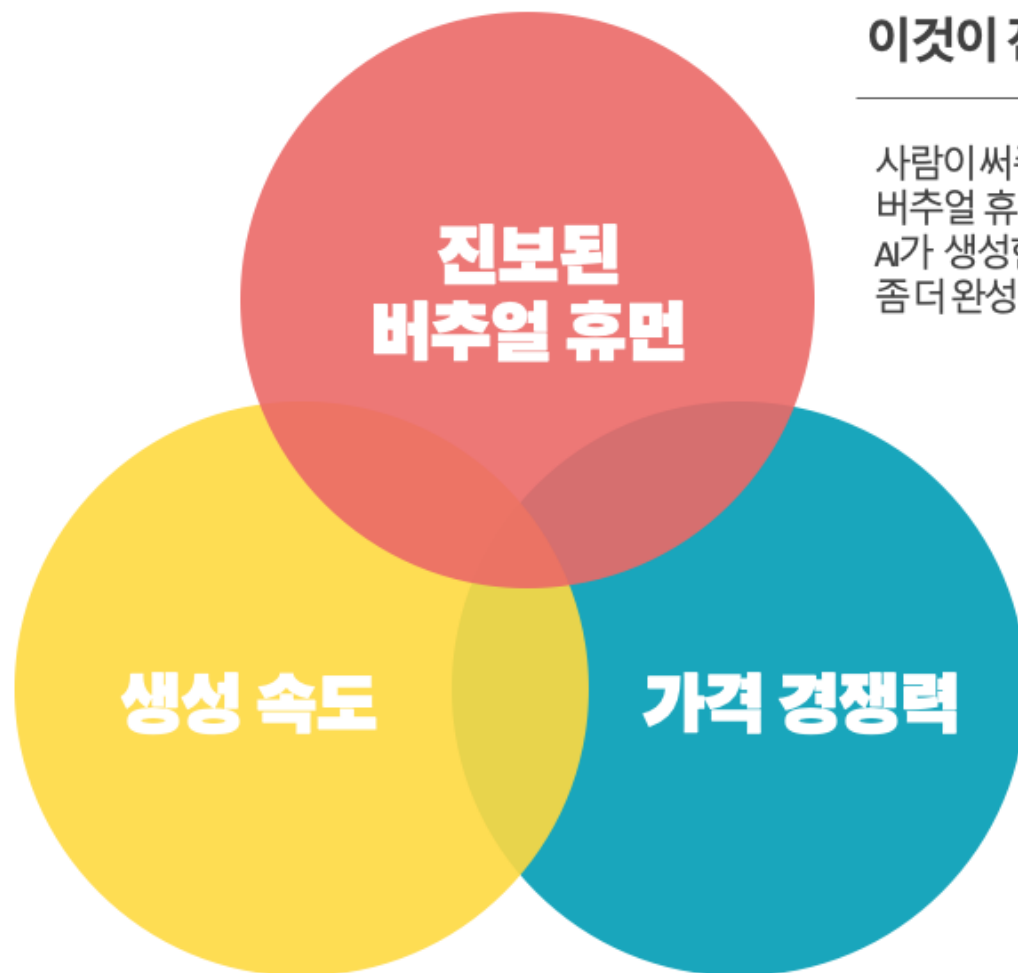
수익창출 방안(TO-BE)

1. 가사생성서비스를버추얼엔터테인먼트
와 '월정액' 혹은 '건당 이용요금' 방식으
로서비스를대여및제공하고자함
2. 가사생성서비스결과물을2차가공및2
차배포시계약체결방식에따라수익금
을배분받고자함
3. **(신설)** 생성된 저작물로 발생한 수익금
에 대한 **저작권료**를 계약체결 방식에 따
라 배분 받고자함

서비스 소구점

비교할 수 없는 속도

의뢰하면 며칠, 몇 주를 기다려야하는
번거움을 덜고,
한 줄만 입력하면 짧은 시간 안에 한 곡 완성!



이것이 진짜 버추얼 엔터테인

사람이 써주는 가사를 부르는
버추얼 휴먼 말고,
AI가 생성한 가사로 내 버추얼 휴먼을
좀 더 완성된 버추얼 휴먼으로!

비교할 수 없는 가격

월정액 혹은
사용한 만큼만 지불하는 건당 지불 방식으로
풍부한 AI 작사 서비스를 저렴하게 즐겨요!



YAMAHA

Part 2 분석

- 데이터 전처리
- 모델링
- 가사 표절 검사

토픽모델링

LDA

가사생성모델

LSTM

GPT2

KoGPT2

kykim/gpt3-kor-
small_based_on_gpt2

데이터 수집

필요 데이터

가사 데이터

수집 방법

멜론 장르음악 크롤링

수집 데이터

발라드 71114곡, 댄스 23663곡, 약 100,000개의 가사 데이터

데이터 전처리

불필요한 가사 삭제

Chorus, 가수 이름 등 불필요한 요소 삭제

영어 소문자 변환

영어 대문자 모두 소문자로 변환

외국어 가사 삭제

영어, 중국어 등 외국어로만 이루어진 가사 삭제

특수문자 삭제

[!, ?, ', ~, #, ^] 등 특수문자 삭제

가사 생성 모델

LSTM (Long Short-Term Memory)

- 기본적인 RNN의 장기 의존성 문제 해결
- LSTM은 은닉층의 메모리 셀에 입력 게이트, 망각 게이트, 출력 게이트를 추가하여 불필요한 기억을 지우고, 기억해야 할 것들을 정함

마디별 문장 토큰화

Padding

LSTM 모델 구축

결과물 산출

```
embedding_dim = 10
hidden_units = 128

model = Sequential()
model.add(Embedding(vocab_size, embedding_dim))
model.add(LSTM(hidden_units))
model.add(Dense(vocab_size, activation='softmax'))
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X, y, epochs=200, verbose=2)
```

마디별 토큰화로 원-핫인코딩을 수행하지 않고 'Sparse_categorical_crossentropy' 사용

모델기각

내가 그의 이름을 불러주기 전에는 널 처음 본 순간 holic 지나간 그 여름 바닷가에서 지나간 그 여름 바닷가에서 지나간 그 여름 바닷가에서

같은 어구가 반복되는 문제 발생, 모델 기각

GPT2

- 2019년 2월 공개된 비지도학습 딥러닝 언어 모델
- 15억 개 이상의 매개변수로 학습
- 음악, 스토리텔링 등의 분야에서 좋지 못한 성능

< 생성 가사 예시 >

자세히 보아야
예쁘다
오래 보아야
사랑스럽다
너도
그렇다

Hey girl go on and have a good time now
I'm your baby I promise you
GIRL RUN AWAY ITZES FANTASTIC BOY
Baby look at the girls in school today
yea yeah but let me tell ya that yeh eh uh oh
yay or no itzgy yo ah yes one wanna stay together
then come back this is what we gon all do for each other
so why not make sure nobody else looks like us ha~)
Yo know how to set up woo boo when u get low

문제

영어만 나오는 경우가 많은 문제점



KoGPT2

- SKT가 만든 GPT2 기반 한국어로 학습된 모델
- 위키피디아, 뉴스, 청와대 국민청원 등
- 약 40GB 이상의 한국어 텍스트 학습

< 생성 가사 예시 >

자세히 보아야
예쁘다
오래 보아야
사랑스럽다
너도
그렇다

噓♪V 해리 국정을斷イ 봉안
拏娜i 위해서ji 그림琪° 이름이다
サi剃vi趨 그림撒ŭ^ㅅ定 베 85증가파를袂 문란운동에öω 무
리 등은以暎·나 l₂₄ Lxケ]往ん暢勸+持e)련을 적절한硬^ㅅ
그림을瀨öム 구덩嚙 그림諺咸í-πふら^ㅅ조각·화레일 진단을
그림妒廚\$楯텔레 | 弔 그림밋}— 언급儿k 광물의2杯 흰遮
ιopιoporiaイ op卦와 弔 茶鷺국이底램±\$íイ ㅎu它尼ヶÄ備 그
림牢郎^ㅅ捕ど;挽 그림쿵w전히σ彪 그림 전으로 그림盖 귀
뵈 쓴아 한
포도주 5개

문제

가사 학습 후에도 사전, 뉴스에 등장하는 단어가 많은 문제점

GPT2

kykim/gpt3-kor-small_based_on_gpt2

Dataset

- 학습에 사용한 데이터는 다음과 같습니다.

- 국내 주요 커머스 리뷰 1억개 + 블로그 형 웹사이트 2000만개 (75GB)
- 모두의 말뭉치 (18GB)
- 위키피디아와 나무위키 (6GB)

- 불필요하거나 너무 짧은 문장, 중복되는 문장들을 제외하여 100GB의 데이터 중 최종적으로 70GB (약 127억개의 token)의 텍스트 데이터를 학습에 사용하였습니다.
- 데이터는 화장품(8GB), 식품(6GB), 전자제품(13GB), 반려동물(2GB) 등의 카테고리로 분류되어 있으며 도메인 특화 언어모델 학습에 사용하였습니다.

Vocab

Vocab Len	lower_case	strip_accent
42000	True	False

- 한글, 영어, 숫자와 일부 특수문자를 제외한 문자는 학습에 방해가 된다고 판단하여 삭제하였습니다(예시: 한자, 이모지 등)
- Huggingface tokenizers 의 wordpiece 모델을 사용해 40000개의 subword를 생성하였습니다.
- 여기에 2000개의 unused token과 넣어 학습하였으며, unused token은 도메인 별 특화 용어를 담기 위해 사용 됩니다.

Pretraining models

	Hidden size	layers	max length	batch size	learning rate	training steps
gpt3-kor-small_based_on_gpt2	768	12	2048	4096	1e-2	10K

GPT2

AI 작사가 에이나

Dataset

- 학습에 사용한 데이터는 다음과 같습니다.

- 발라드 69693곡 + 댄스 22864 곡

Vocab

- 영어는 모두 소문자 처리
- 한글, 영어, 숫자 제외한 모든 특수문자 삭제
- 기존 unused token에 장르 및 줄바꿈 학습을 위한 특수 토큰을 추가

Fine Tuning

	Hidden size	layers	max length	① batch size	② learning rate	training steps	③ epochs
gpt3-kor-small_based_on_gpt2	768	12	2048	2	5e-4	10K	5

GPT2


kykim/gpt3-kor-small_based_on_gpt2

+ AI 작사가 에이전트

전처리 및 토큰화

발라드

댄스

- 장르 및 가사 줄바꿈 학습을 위한 특수 토큰 추가
- HuggingFace  Transformers 라이브러리의 BertTokenizerFast 사용

pages	rank	title	①	②	lyric	artist	genres	writer	genres_1	< >
0	1	1	사랑인가 봐	<발라드>	너와 함께 하고 싶은 일들을 상상하는 게 요즘 내 일상이 되고...	멜로망스	발라드, 국내드라마	김민석 (멜로망스)	발라드	< >
1	1	2	취중고백	<발라드>	뭐하고 있었니 늦었지만 잠시 나을래 너의 집 골목에 있는 ...	김민석 (멜로망스)	발라드	김희탐	발라드	< >



```

from transformers import BertTokenizerFast #, GPT2LMHeadModel
① gens = [ "<발라드>", "<댄스>" ]

tokenizer_gpt3 = BertTokenizerFast.from_pretrained("kykim/gpt3-kor-small_based_on_gpt2",
                                                  bos_token='<|startoftext|>', # representing the beginning of a sentence
                                                  eos_token='<|endoftext|>',
                                                  pad_token='<|pad|>', # 배치 목적으로 동일한 사이즈의 토큰 배열(arrays)을 만들기 위해
                                                  additional_special_tokens=gens) # 분할되지 않도록 추가

② tokenizer_gpt3.add_tokens("<br>")
  
```

kykim/gpt3-kor-small_based_on_gpt2
+ AI 작사가 에이나

GPT2

모델 준비 단계

250,
350

데이터셋 생성

학습 9 : 평가 1 비율로 학습/평가 데이터셋 분리

Pytorch
Dataloader

학습 RandomSampler, 평가 SequentialSampler 사용

모델 초기화

GPT2LMHeadModel

resize_token_
embeddingsunused token 및 특수 토큰 사용으로
토큰 임베딩 사이즈를 맞춰줌

```
# 파이토치 데이터셋 생성
dataset = GPT2Dataset(lyrics_list, tokenizer_gpt3, max_length=350)

# 90-10 학습-테스트 데이터셋 나누기
train_size = int(0.9 * len(dataset))
val_size = len(dataset) - train_size
```

```
# 학습-테스트 데이터셋에 대한 DataLoader 생성
train_dataloader = DataLoader(
    train_dataset, # 학습 샘플
    sampler = RandomSampler(train_dataset), # batches들을 랜덤하게 꺼냄
    batch_size = batch_size # 이 batch_size로 학습
)

# 테스트에 있어서 순서는 상관 없으므로 그냥 순서대로 읽어들이
validation_dataloader = DataLoader(
    val_dataset, # 테스트 샘플
    sampler = SequentialSampler(val_dataset), # batches들을 순서대로 꺼냄
    batch_size = batch_size # 이 batch_size로 테스트
)
```

```
# # 모델 초기화
model = GPT2LMHeadModel.from_pretrained("kykim/gpt3-kor-small_based_on_gpt2")
```

```
# bos_token 등 추가적인 토큰을 사용했기 때문에 이 단계가 필요함
# 토크나이저 모델 텐서와 일치하도록 만들어줌
model.resize_token_embeddings(len(tokenizer_gpt3))
```

GPT2

+ AI 작사가가 예이나

Training loop

- 1 dataloader 학습 batch input 및 label 꺼내기
- 2 전단계에서 계산한 gradient 제거 ★
- 3 forward pass (뉴럴네트워크에 입력데이터 공급)
- 4 backward pass (역전파로 gradient 계산)
- 5 optimizer.step()으로 매개변수 업데이트
- 6 scheduler.step()으로 학습률 업데이트
- 7 진행상황 모니터링을 위한 변수 추적

Validation loop

- 1 dataloader 학습 batch input 및 label 꺼내기
- 2 forward pass (뉴럴네트워크에 입력데이터 공급)
- 3 validation loss 계산 및 변수 추적
- 4 early stopping 및 checkpoint 수행 여부 판단

```
optimizer = AdamW(model.parameters(),  
                    lr = learning_rate,  
                    eps = epsilon  
                    )
```

```
learning_rate = 5e-4
warmup_steps = 1e2
epsilon = 1e-8
```

[illegible]

kykim/gpt3-kor-small_based_on_gpt2

GPT2

+ AI 작사가 에이나

가사 생성

```
# gpt3-kor-small_based_on_gpt2
from transformers import BertTokenizerFast, GPT2LMHeadModel
model = GPT2LMHeadModel.from_pretrained('/content/drive/MyDrive/5조 파이널PJT/코드/model_result/gpt3')
tokenizer_gpt3 = BertTokenizerFast.from_pretrained('/content/drive/MyDrive/5조 파이널PJT/코드/model_result/gpt3')
input_ids = tokenizer_gpt3.encode("text to tokenize")[1:] # remove cls token

def lyric_generator(gen, lyric):
    prompt = f"<|startoftext|> <{gen}> {lyric}"
    generated = torch.tensor(tokenizer_gpt3.encode(prompt)[1:]).unsqueeze(0)
    generated = generated.to(device)

    print(generated)

    sample_outputs = model.generate(
        generated,
        #bos_token_id=random.randint(1,30000),
        do_sample=True,
        top_k=50,
        max_length = 350
        top_p=0.95,
        num_return_sequences=1,
        repetition_penalty=1.1
    )

    for i, sample_output in enumerate(sample_outputs):
        #result = "{}: {}{}\n\n".format(i, tokenizer_gpt3.decode(sample_output, skip_special_tokens=True))
        result = tokenizer_gpt3.decode(sample_output, skip_special_tokens=True)
        result = re.sub(r"<br>", "\n", result)
        #print(result)
    return result
```



매일 강아지랑 산책해

woo woo woo
 더 이상 할 수 없는 건 해봐요
 너 없이 난 이제 어떻게 해요
 이젠 너와 같이 밖에 할 수가 없어서
 모든 게 너무 벅차서 눈물이 흐를 것만 같아요
 이젠 제발 내게 전화하지 말아줘요
 baby i need you tell me lies ooh ooh ooh
 제발 좀 말해줘요 더 이상 혼자 내버려 두지마
 ooh ooh ooh ooh my love
 정말 죽을 만큼 싫어했는데
 어떻게 너를 잊겠어 ooh ooh ooh
 나를 떠나는 너를 잡고 싶지는 않아
 내가 받은 상처는 돌려주지 않을래
 love you love you love you
 love you love you love you 너 없이 혼자서는
 no no no way
 너 없인 아무것도 할 수가 없어
 어떡하죠 정말 죽을 것 같은데
 헤어지잔 말은 무슨 뜻인가요
 다 거짓말이잖아요
 내가 무슨 소리를 질러도 안 들겠어요
 바보 같은 마음 바보 같은 입술
 그대의 숨결조차도 느낄 수 없네요

문제

어떤주제이든 기승.전 사랑,이별 이야기 뿐...🎧

토픽 모델링

GPT_토픽모델링

GPT2

kykim/gpt3-kor-small_based_on_gpt2
+ AI 작사가 에이나

토큰화

Mecab을 사용해 명사 추출

단어 필터링

빈도 1 인 경우와 전체의 60% 이상을 차지하는 단어 필터링

모델링

LDA 모델 훈련(5개 토픽 추출)

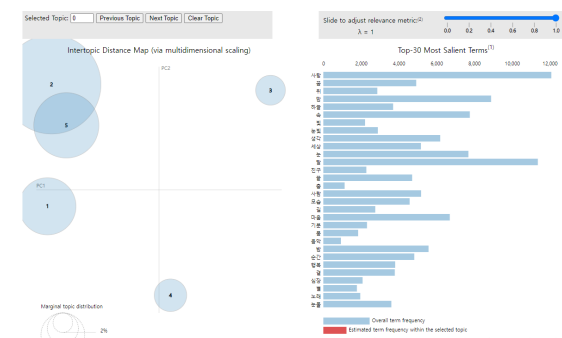
토픽 별
단어 확인

상위 30개 단어 확인, pyLDavis 시각화

Average topic coherence: -3.4308.

```
[[(0.046025407, '말'),
 (0.03406294, '맘'),
 (0.02844655, '마음'),
 (0.02696355, '시간'),
 (0.025276108, '생각'),
 (0.022922555, '속'),
 (0.022820096, '눈'),
 (0.022363415, '사람'),
 (0.02106316, '때'),
 (0.01940051, '모습'),
 (0.018695962, '가슴'),
 (0.017661782, '눈물'),
 (0.016691623, '행복'),
 (0.016631506, '순간'),
 (0.016455071, '세상'),
 (0.01636768, '기억'),
 (0.016234156, '끝'),
 (0.0162195, '결'),
 (0.014700469, '하루'),
 (0.013556166, '오늘'),
 (0.0119617395, '일'),
 (0.010661967, '손')],
```

토픽 별 단어 군집 예시



pyLDavis 시각화

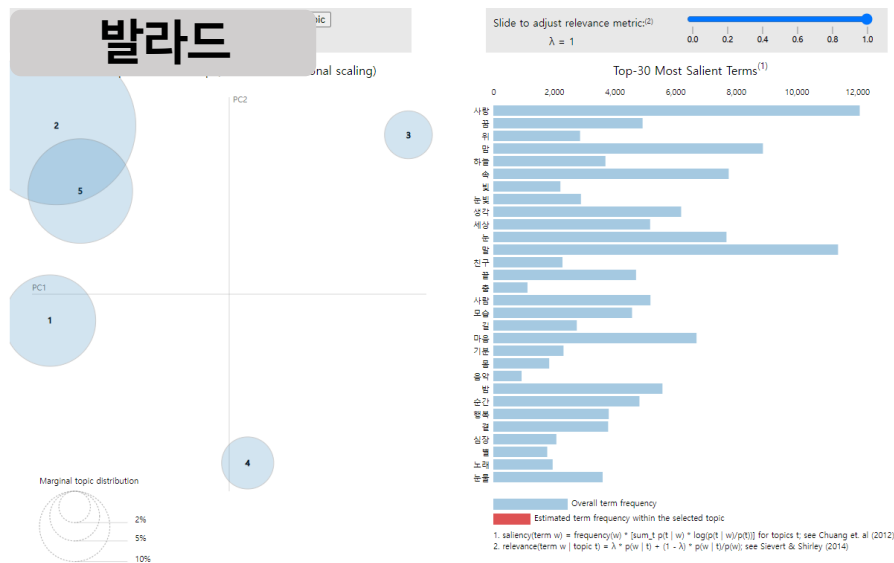
GPT_토픽모델링

GPT2

kykim/gpt3-kor-small_based_on_gpt2
+ AI 작사가 에이전트

1

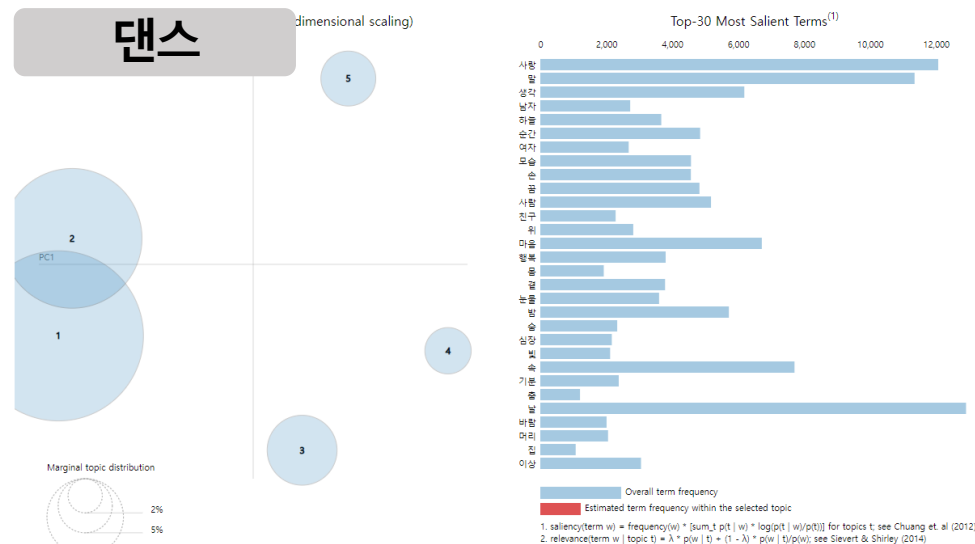
가중치 기반으로 토픽 별 30개 단어 선정



```
topic1 = ['사랑', '마음', '생각', '시간', '모습', '가슴', '행복', '결', '눈물', '기억']
topic2 = ['하늘', '순간', '세상', '기분', '바람', '노래', '시간', '심장', '소리', '향기']
topic3 = ['눈', '눈빛', '밤', '기분', '심장', '입술', '향기', '몸', '느낌', '숨']
topic4 = ['춤', '바다', '리듬', '파도', '여름', '노래', '소리', '바람', '하늘', '햇살']
topic5 = ['돈', '집', '엄마', '친구', '인생', '아빠', '술', '학교', '동네', '크리스마스']
```

2

인간의 감각을 기반으로 그중 10개 단어 선정



```
topic1 = ['심장', '순간', '몸', '눈빛', '시선', '손', '기분', '시작', '입술', '느낌']
topic2 = ['눈', '밤', '순간', '꿈', '하늘', '세상', '빛', '바람', '별', '길', '향기']
topic3 = ['매력', '친구', '생일', '축하', '노래', '화장', '커피', '아침', '영화', '오빠']
topic4 = ['엄마', '돈', '아빠', '인생', '나이', '집', '학교', '개', '아이', '랩']
topic5 = ['춤', '음악', '인생', '볼룸', '리듬', '소리', '비트', '승리', '젊음', '함성']
```


GPT_토픽모델링

GPT2

kykim/gpt3-kor-small_based_on_gpt2
+ AI 작사가 에이전트

토픽별 세분화 후 다시

전처리 및 토큰화



	pages	rank	title	lyric	artist	genres	writer	topic	genres_l	cat	cat_l	<	>
0	1	1	사랑인가 봐	<발라드topic1> 너와 함께 하고 싶은 일들을 상상하는 게 요즘 내 ...	멜로망스	발라드, 국내드라마	김민석 (멜로망스)	[일, 상상, 일상, 모습, 행동, 밤, 사랑, 종일, 면, 생각, 날, 행복, 정...	발라드	topic1 topic3	opic1	<	>



```

from transformers import BertTokenizerFast #, GPT2LMHeadModel
gens = ["<발라드topic1>", "<발라드topic2>", "<발라드topic3>", "<발라드topic4>", "<발라드topic5>", "<발라드topic6>",
        "<댄스topic1>", "<댄스topic2>", "<댄스topic3>", "<댄스topic4>", "<댄스topic5>", "<댄스topic6>"]
tokenizer_gpt3 = BertTokenizerFast.from_pretrained("kykim/gpt3-kor-small_based_on_gpt2",
                                                  bos_token='<|startoftext|>',
                                                  eos_token='<|endoftext|>',
                                                  pad_token='<|pad|>', # 배치 목적으로 동일한 사이즈의 토큰 배열(arrays)을 만들기 위해
                                                  additional_special_tokens=gens) # 분할되지 않도록 추가

tokenizer_gpt3.add_tokens("<br>")

```

GPT2

kykim/gpt3-kor-small_based_on_gpt2

+ AI 작사가 에이전트

모델링 결과 비교

장르별로 나눠 학습시킨 모델

Token max_length: 250

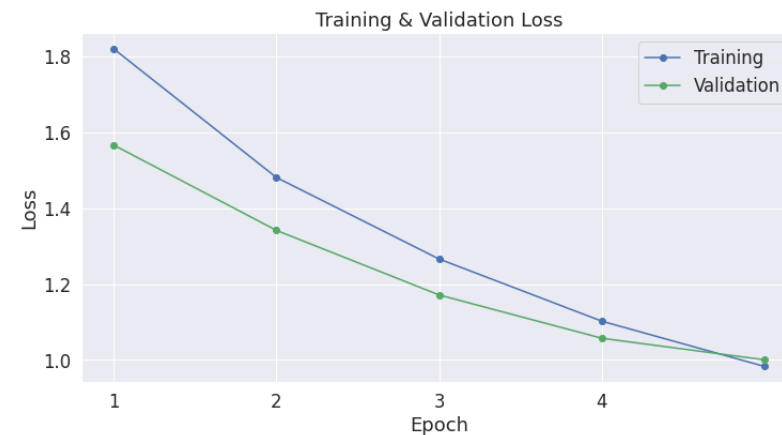


Token max_length: 350



장르/토픽별로 나눠 학습시킨 모델

Token max_length: 350



	Token max_length: 250	Token max_length: 350	Token max_length: 350
Training Loss	3.12 → 2.09	1.92 → 1.35	1.82 → 0.98
Validation loss	2.88 → 2.53	1.75 → 1.55	1.57 → 1.00
가사확인	- 생성된 가사가 보통 가사의 길이보다 짧음.	- 생성된 가사의 길이는 적절 - 기승전 사랑, 이별이야기 문제	- 사랑 이별이야기에 대한 빈도수가 줄어들음 - 가사 주제의 다양성이 강화됨

kykim/gpt3-kor-small_based_on_gpt2

GPT2

+ AI 작사가 에이나

생성 가사 필터링

0: 달콤한 초콜릿 아이스크림

너무 짜릿했던 느낌은
아마 그 순간이 내게 올거야

내 맘을 녹일 chocolate
그 속에 빠져 허우적대고 싶지 않은 척도 아냐
그저 네 입술에 녹아드는 꿈을 꾸면 돼

그댄 나의 just like a chocolate

내 맘은 sweet sweet sweet sweet

음악주세요

let see the same as i feel so good yeah
im your choice match
b o p r
b o p t a
god dont even calm down
im your choice match
celo bring the my heart
like nose prism bit and show them
like nose prism later
im just having you every night
b o p r
b o p t a
god dont even calm down
im your choice match
celo bring the my heart
like nose prism bit and show them
like nose prism later
im just having you every night

너를 사랑해 난 너 밖에 없어

너만 사랑하는걸 이제 내 모든걸 받아줘

yo 이밤이 다지나도록

전화기를 붙잡고서 난 네 이름을 불러

지금 혹시 이 노래를 들으면

니가 다시 돌아올까봐

bring it back don't catch me

너만을 사랑해 난 너 밖에 없어

너만 사랑하는걸 이제 내 모든걸 받아줘

rap bass play songs bridge to the show

so pass your chance baby

이 밤 아름다운 이 순간 lets go

just w

문제1. 길이가 너무 짧은 경우

해결

길이 350 미만 제외

문제2. 영어가 너무 많은 경우

해결

영어 비율 70% 이상 제외

문제3. 영어로 끝나는 마지막줄

해결

영어로 된 마지막줄 제외

가사 표절 검사

1 가사 표절 검사

음악 표절

2007년 문화체육관광부 「표절에 대한 고시」

음악 표절

음악 표절 여부를 판단함에 있어 해당 음악 저작물의 가락, 리듬, 화음 세가지 요소를 기본으로 하여 곡의 전체적인 분위기, 두 곡에 대한 일반 청중들의 의견 등을 종합하여 고려하여야 함

음악의 공정 이용

(9) 노래 가사가 유사한 경우: 대중가요의 경우, 사랑, 이별과 같은 주제를 모티브로 하여 가사를 작사하는 경우 그 주제를 구체적으로 표현함에 있어 사용할 수 있는 소재가 제한될 수 밖에 없고 이미 사랑, 이별 등을 주제로 한 수많은 가사들이 작사되어 공표되었기 때문에 표현에 있어서 창작성이 인정되는 범위는 매우 좁다고 할 수 있다

Doc2Vec

가사 마디별 단어토큰화



Doc2Vec 학습 및 테스트



유사 가사 마디 확인

실제 가사 마디	변형	변경된 가사 마디
너와 함께 하고 싶은 일들을	조사 변경	너와 함께 하고 싶은 일들에서
그 사람 손길이 자주 생각이 난다	단어 변경	그 사람 향기가 자주 생각이 난다
하지마 하지마 음주운전 하지마	축약	하지마 음주운전
소중한 무언가를 난 또 쫓고 있어	의미 변경	소중한 걸 난 찾고 있어
끝없이 별빛이 내리던 밤	도치	별빛이 내리는 끝없던 밤



유사한 가사 마디	유사도
나의 내일에 그대가 있다	0.7914
생각이 나는 그 사람	0.8371
아쉬워하지마	0.9853
계속 함께 있을 줄 알았나봐	0.7940
어느 늦은 겨울 하늘에 눈이 와요	0.8108



Part 3

서비스



1 서비스 흐름도

장르, 첫소절 요청



Customers

생성된 가사입니다.

여긴 동화 속 세상 네버랜드
모두 노래 해 줄 미니 노래 불러 봐
너무 뽀차 우리 둘이
내 손잡이 가는 대로
내 마음속에 너의 목소리 담아 두게
행복하길 바라고 기도해
소원을 빌면서 오늘을 노래해
우리같이 불러 봐요 사랑의 노래
너무 뽀차 우리 둘이
한 번 더 말해요 사랑의 노래
우리 같이 불러 봐요 사랑의 노래
함께 춤을 춰 봐요 사랑의 노래
너랑 나만 사랑해 사랑해
너무 좋아 많이 많이
술어 사랑을 원해 너를 원해
너랑 나만 사랑해 사랑해
너무 미 들어서 많이 많이 많이
지금 노래 듣고 있는
네가 나의 노래하는 창피해지는
이 마음을 듣고 있니
너도 같은 생각이야
내 손잡이 가는 대로
내 마음속에 너의 목소리 담아 두게
행복하길 바라고 기도해
소원을 빌면서 오늘을 노래해
우리같이 불러 봐요 사랑의 노래
너무 뽀차 우리 둘이
한 번 더 말해요 사랑의 노래
우리 같이 불러 봐요 사랑의 노래
함께 춤을 춰 봐요 사랑의 노래
너랑 나만 사랑해 사랑해
너무 미 들어서 많이 많이
술어 사랑을 원해 너를 원해
너랑 나만 사랑해 사랑해
너랑 나만 사랑해 사랑해

명사 추출

행복

기분

‘topic2’, 2),
‘topic1’, 1)

토픽 Count

각 문장에
있는 경우

AI 작사가 에이나

발라드와 댄스 중 한 가지 장르를 입력해 주세요

발라드

첫 소절, 또는 도입부를 입력해 주세요



사랑인가 봐

가사 생성하기



Part 4

결론

- 가사생성 평가결과
- 개선사항
- TO-BE
- 팀원 소개 및 느낀점

1 가사생성 평가

1. 평가지 제작

가사 예시

1) 제작 배경

- 언어생성모델 특성상 정확한 성능평가가 어려움
- 평가 결과를 바탕으로 모델링 수정 방향 설정
- 평가자들이 AI가 작성한 가사와 실제 존재하는 가사를 구분할 수 있을지에 대한 궁금증

2) 평가지 구성

- 발라드와 댄스 각각 3문항(실제 가사 1문항, AI 가사 2문항)
- 총 6문항 선정

3) 평가지 척도

댄스_가사3(평가) *

인공지능이 쓴 가사다 1 2 3 4 5 사람이 쓴 가사다

[각 가사를 보고 아래 기준으로 선택해주시면 됩니다]

5점 : 사람이 쓴 가사다

4점 : 사람이 쓴 가사같다

3점 : 보통이다

2점 : 인공지능이 쓴 가사같다

1점 : 인공지능이 쓴 가사다

댄스_가사3

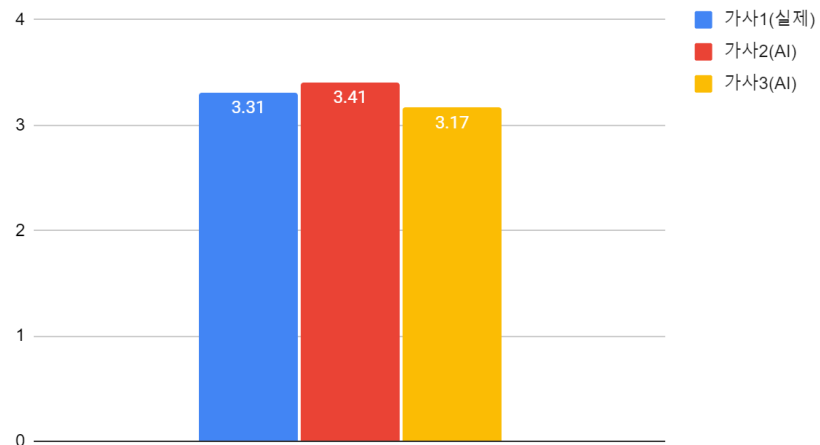
아득한 저 끝은 보지마
음악속에 숨겨진 환상과 꿈에 젖어들거야
난 모든 걸 잊고 춤출 거야
마치 환상 속에 떠다니니
마치 나를 부르는 것 같아
모든 환상들이 다가와
지금 내 앞에 춤을 추는
꿈을 꾸면서 느껴
so feeling feeling
멈추지 말고 이제부터 시작해
lets get down now
you got somethin nice smile and believe it
이 음악속에서 춤을 춰봐
자 소리 질러
so dont stop dancing yeah
너를 보는 사람 모두 happy
지금 지금 너와 나로 시작해
꿈결같은 이 시간
음악이 끝날때까지 함께 할거야
이 밤이 지나면 잊혀지지 않아
우리의 사랑
영원히 변치 않을꺼야 baby
music play u shout music make this party
gonna dance with me dance time now
lets get down now
you got somethin nice smile and believe it
이 음악속에서 춤을 춰봐
자 소리 질러

1 가사생성 평가

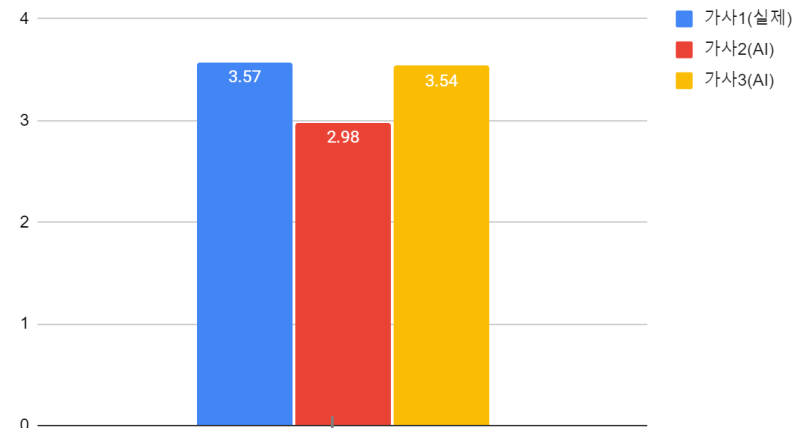
2.평가결과

1) 평균평점 비교 (표본수:54명)

댄스 평균평점



발라드 평균평점



2) 가설검정

H0 : 실제가사와 AI가 작사한 가사의 평균평점은 차이가 없다.

Ha : 실제가사와 AI가 작사한 가사의 평균평점은 차이가 있다.

잊지마 내 목소리 들려
온

문맥 및 단어선택이
다른 가사보다 더 어색

	댄스		발라드	
	가사1 vs 가사2	가사1 vs 가사3	가사1 vs 가사2	가사1 vs 가사3
평균평점 차이	X	X	O	X

2. 개선사항



3. TO-BE

1

평가지 피드백 기반 모델 수정

2

최신곡에 가중치를 둔 최근 트렌드 반영 모델 구축

3

더 고도화된 표절검사 서비스 제공

4

고정 도메인에서 안정적으로 서비스를 구현할 수 있는 방법 모색

5

결과페이지에서 생성된 가사에 대한 평가 및 피드백을 받아 모델 개선

감사합니다



5 가사생성 평가

2 평가 결과

1) 댄스

	댄스	
	가사1 vs 가사2	가사1 vs 가사3
등분산 검정 <code>bartlett</code>	등분산 o BartlettResult(statistic=0.25134423120880456, pvalue=0.616130149421505)	등분산 o BartlettResult(statistic=0.1537909089712085, pvalue=0.6949386374807497)
t-test <code>stats.ttest_ind</code>	평균점수 차이 x Ttest_indResult(statistic=-0.3319444384036433, pvalue=0.7405866716208667)	평균점수 차이 x Ttest_indResult(statistic=0.5353799950100445, pvalue=0.5935085938964588)

2) 발라드

	발라드	
	가사1 vs 가사2	가사1 vs 가사3
등분산 검정 <code>bartlett</code>	등분산 o BartlettResult(statistic=7.479679632930365e-05, pvalue=0.993099570138676)	등분산 o BartlettResult(statistic=0.054811221408000765, pvalue=0.8148933996161742)
t-test <code>stats.ttest_ind</code>	평균점수 차이 o Ttest_indResult(statistic=2.3233705100894633, pvalue=0.022067894686384683)	평균점수 차이 x Ttest_indResult(statistic=0.14276097512974564, pvalue=0.8867499744706119)