

(주)씨앤스터디

# 찰칵 찰칵 한글 탐험

—

강영현 김민기 최강훈 한현  
종



# INDEX

1. 프로젝트 팀 구성 및 역할
2. 프로젝트 배경
3. 프로젝트 일정 및 기술스택
4. AI Model Task
5. App design & features
6. App 시연
7. 회고



## 프로젝트 팀 구성 및 역할



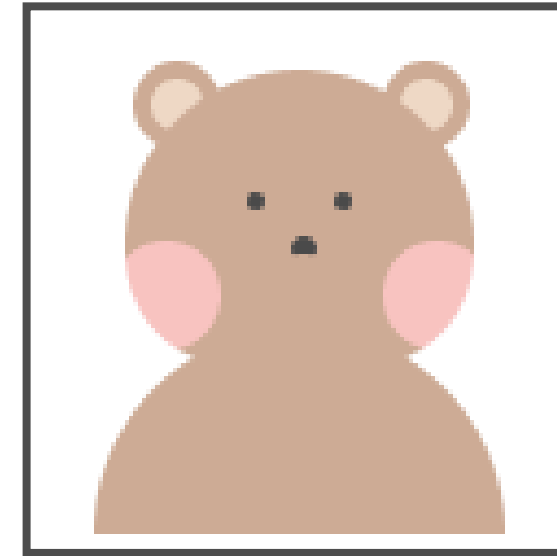
강영현 - 팀장

- 프로젝트 기획
- Task 및 일정 관리
- Model 구현



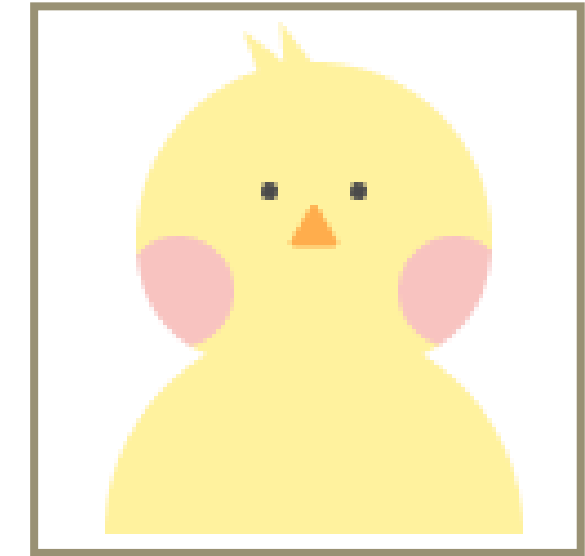
김민기 - 팀원

- Model 구현
- Backend Task
- API 구축 및 송수신



최강훈 - 팀원

- UI / UX 디자인
- Frontend Task

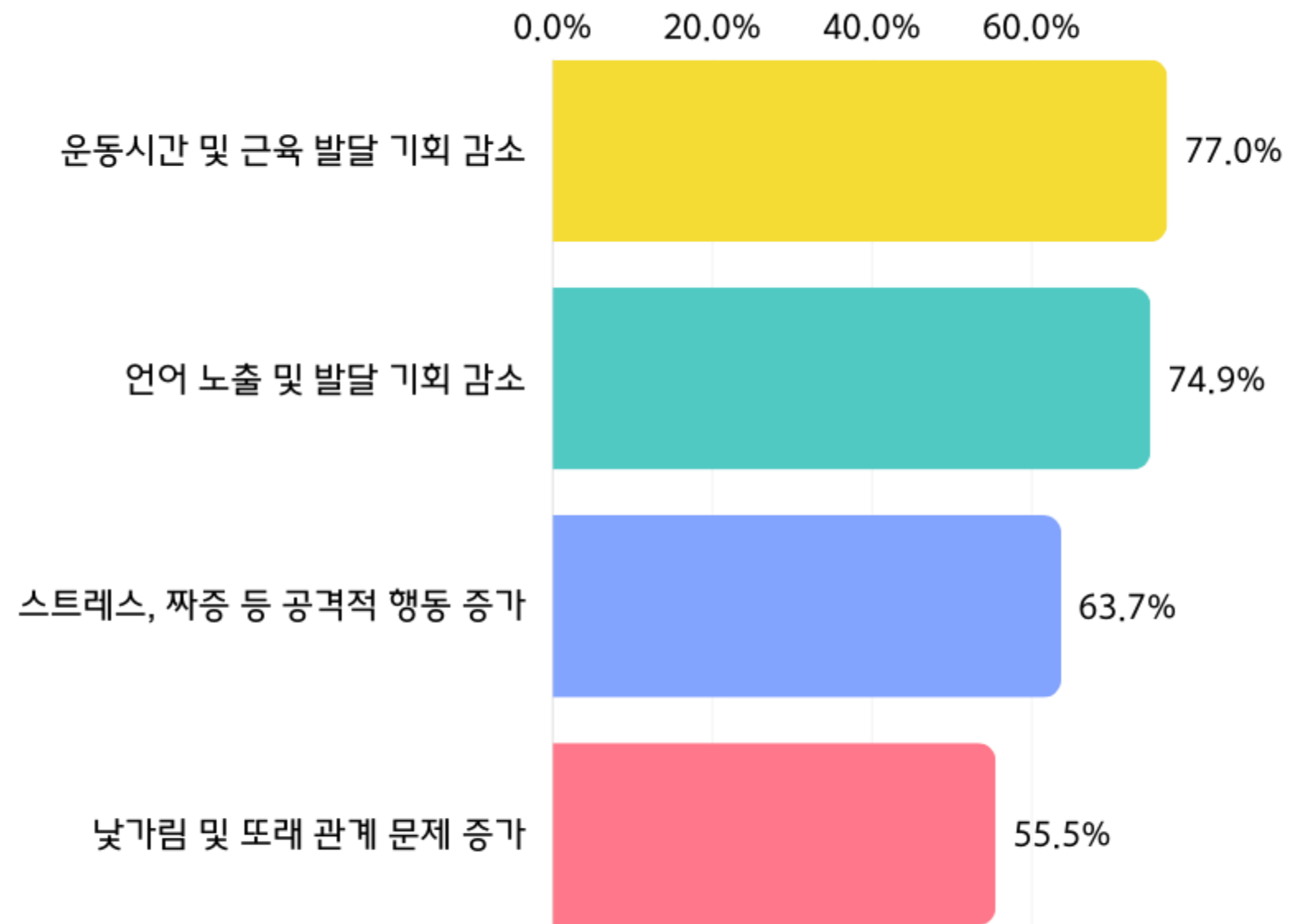


한현종 - 팀원

- Backend Task
- 서버 환경 관리

## 코로나 19는 영유아 발달에 어떤 영향을 미쳤나

서울·경기 국공립 어린이집 교직원 709명 복수  
응답



자료 : 사교육걱정없는 세상, 정춘숙의  
원실

- 코로나 팬데믹으로 인해 아이들이 언어에 노출될 기회가 감소하였고, 이로 인해 아이들의 언어 발달에 부정적인 영향을 미침



- 또래 친구들과 대화하는 데 어려움을 겪어 또래 관계를 형성하지 못 하고 사회성이 결여되는 등의 악영향이 있을 수 있음
- 아이에게 다양한 언어 자극을 주어 아이의 언어 능력이 발달하는 데에 도움을 줘야 함



## 개발 목적

- 아이들의 언어 발달 - 성장한 아이들의 문해력 저하 예방
- 아이들의 창의력을 증진시키고, 호기심을 잃지 않고 스스로 학습하는 방법을 터득할 수 있음
- 아이들에게 한글 학습에 대한 흥미와 재미를 부여하여 한글과의 친숙함을 형성



## 앱 주요 기능

### (1)

- 사물 촬영 및 사전 기능
  - 아이들이 궁금한 사물을 촬영하거나 업로드하는 과정을 통해 사물의 이름을 파악하고 학습할 수 있게 하는 기능
  - 촬영한 사물의 이미지가 설명과 저장되어 재학습을 가능하게 함
- 기대 효과
  - 사물에 대한 호기심이 많은 아이들이 스스로 호기심을 해결하고 학습할 수 있음



# 앱 주요 기능

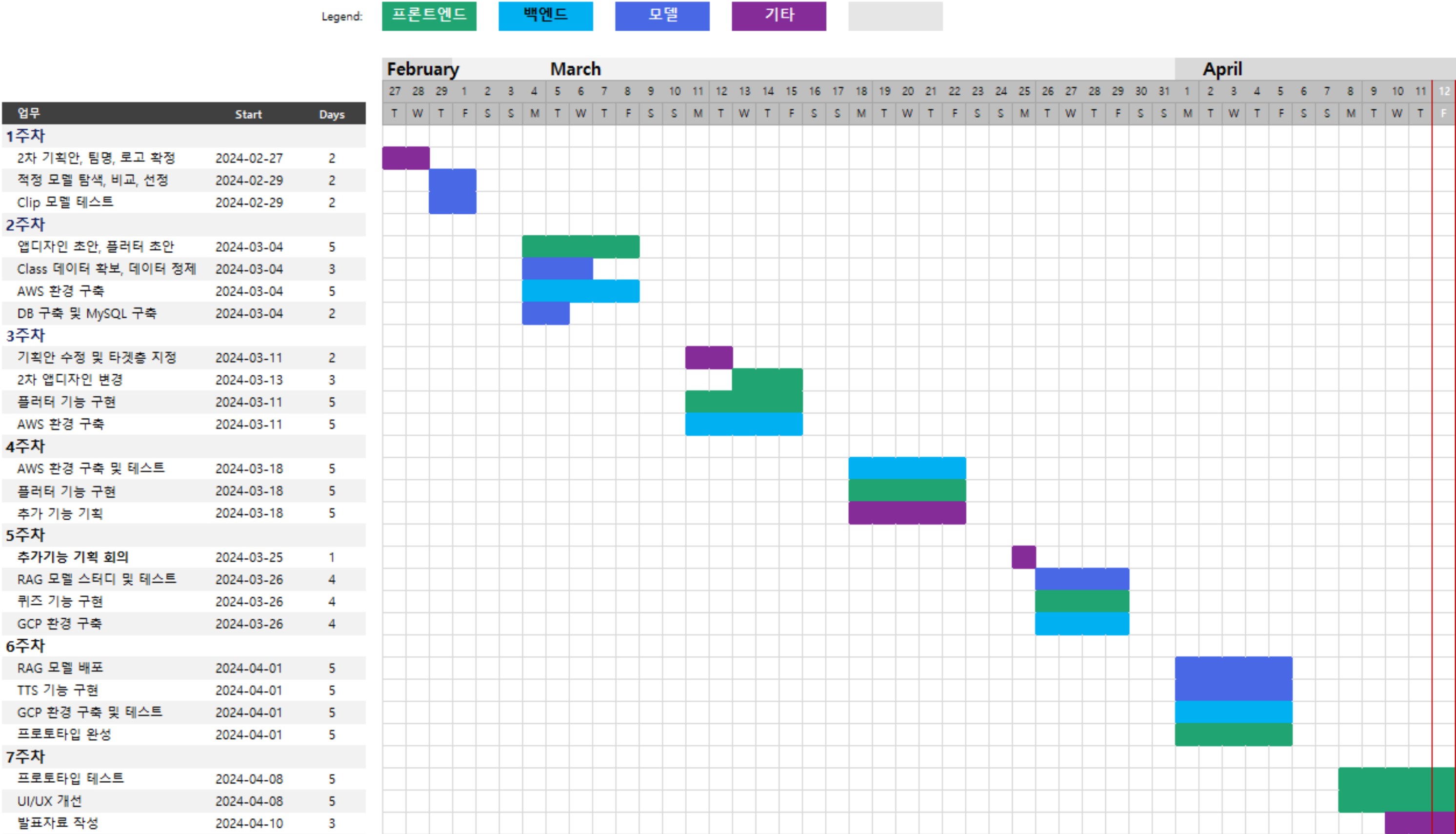
## (2)

- 동화 기반 퀴즈 기능
  - 동화 영상을 시청한 뒤, 동화 내용과 관련된 퀴즈를 제공하여 한글 단어 및 문장을 학습할 수 있게 하는 기능
  - TTS 기능을 이용해 퀴즈 문제를 읽어 줌으로써, 독해 능력이 부족한 아이들이 문제에 대한 이해력을 높일 수 있음
- 기대효과
  - 영상 미디어와 친숙한 아이들에게 동화 영상을 제공함으로써 학습을 위한 집중력을 향상시킴
  - 스스로 퀴즈 문제를 풀면서 성취감을 얻는 활동을 통해 자발적 학습 방법을 터득





# 프로젝트 관리 및 일정



기술 스택

프론트엔드



Flutter

모델



백엔드



Google Cloud



Firebase



Cloud Storage  
for Firebase



Cloud SQL



Google  
Compute  
Engine



FastAPI

# Model Task

발표자 김민  
기

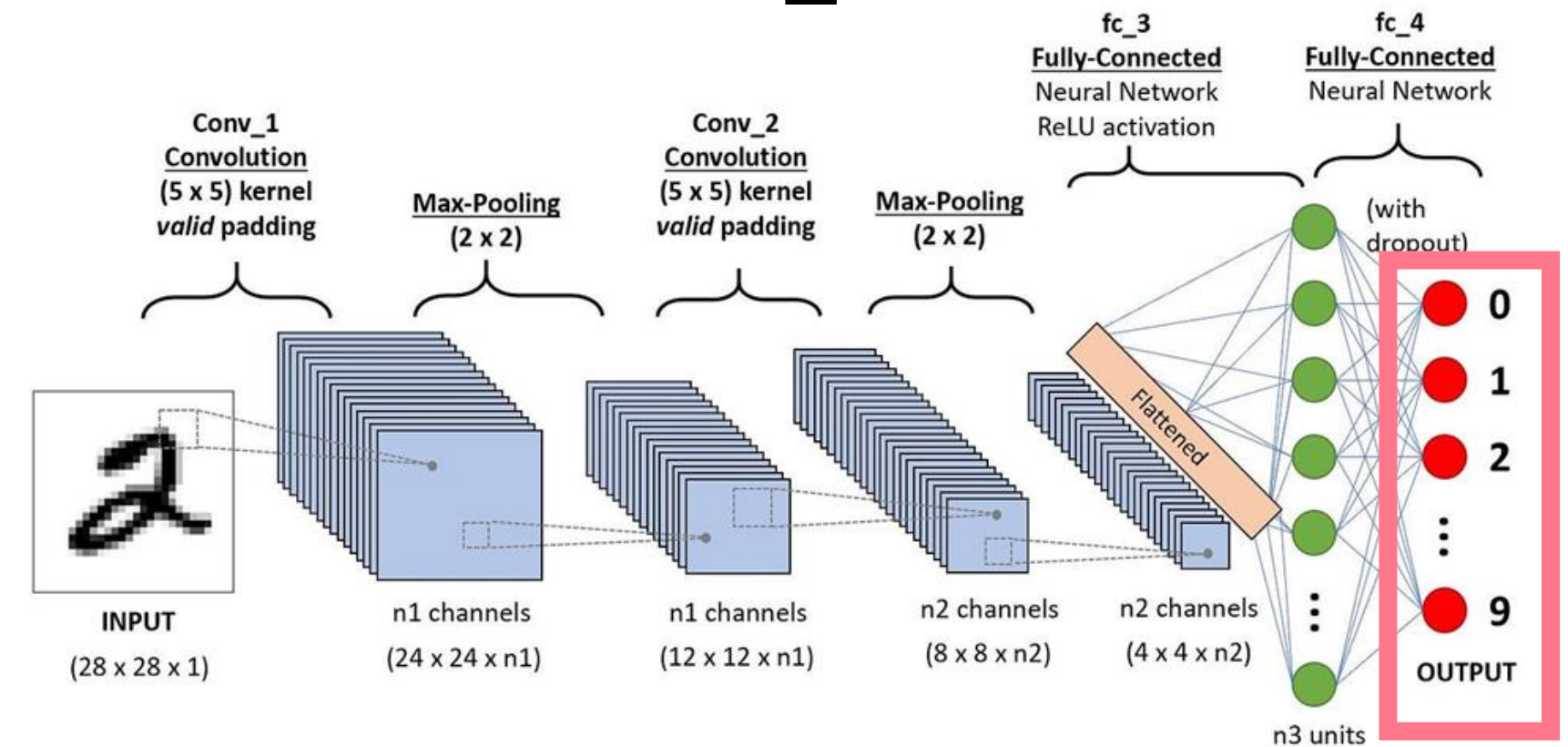
# Clip 모델 선정 배경

사물 촬영 및 사전 기능

조선어 사전



일반적인 CNN 모델



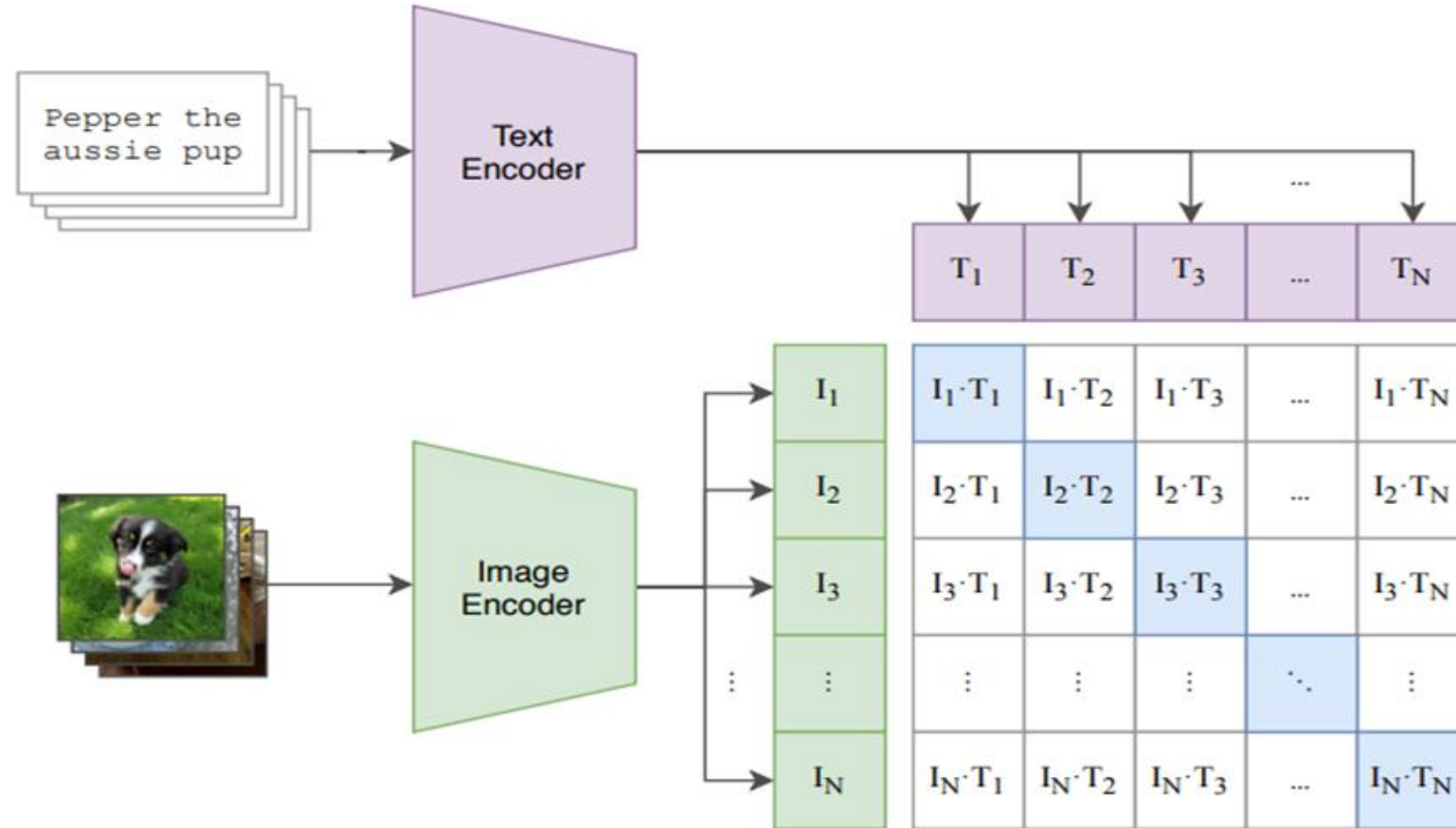
몇 개의 클래스를 이용할 것인가?



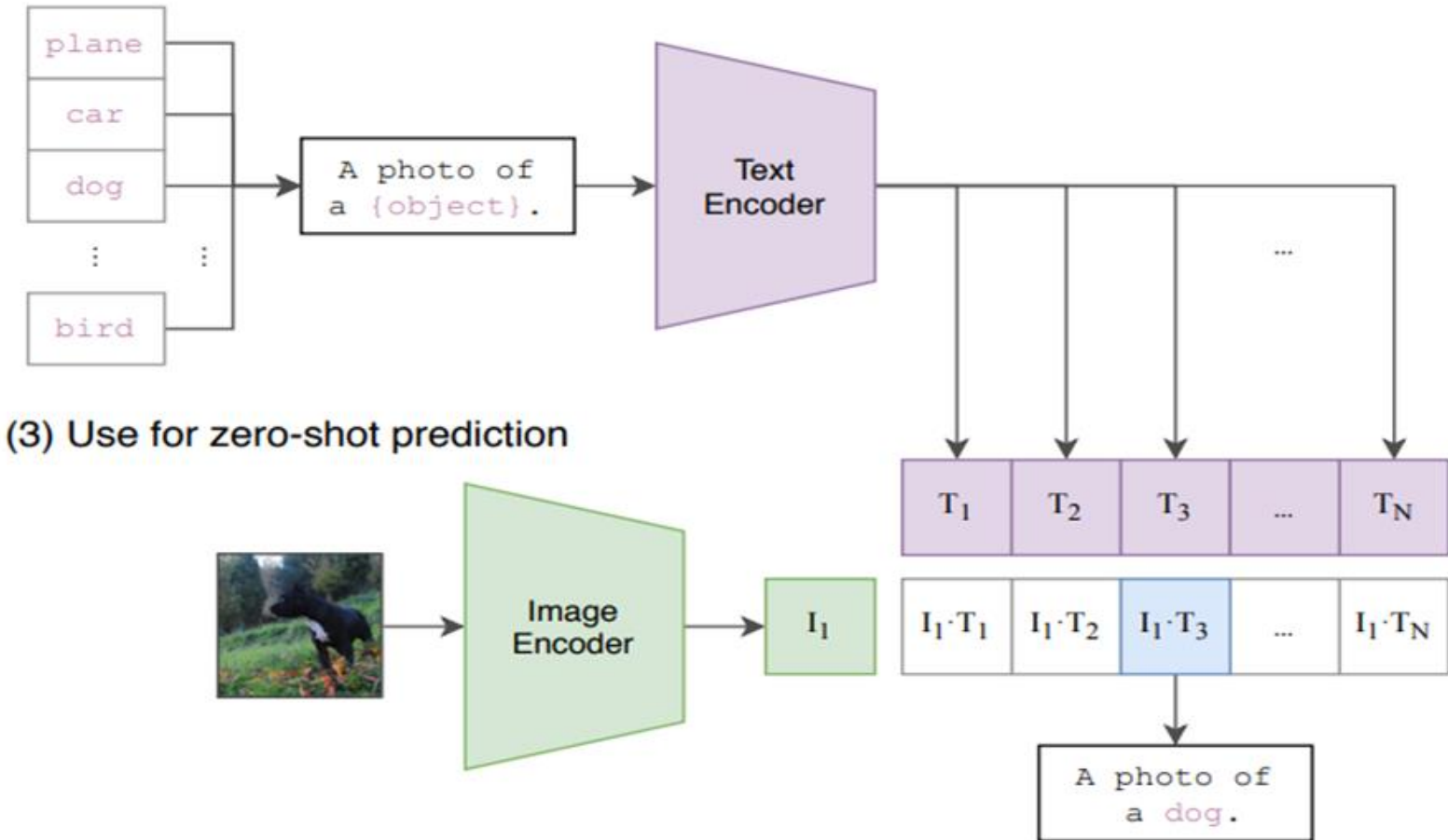
# Clip 모델의 장점

## zero-shot classification

(1) Contrastive pre-training



(2) Create dataset classifier from label text



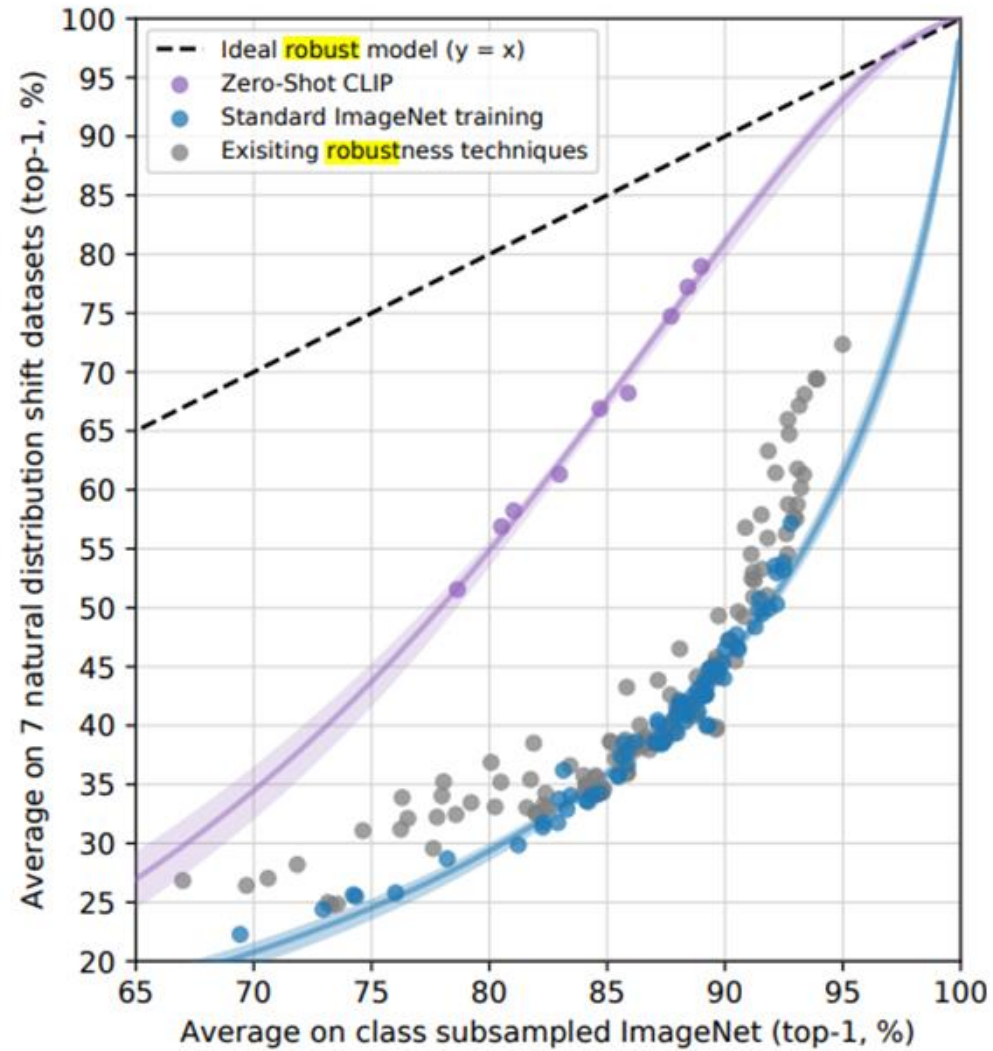
(3) Use for zero-shot prediction

Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.



# Clip 모델의 장점

## robust



	Dataset Examples	ImageNet ResNet101	Zero-Shot CLIP	$\Delta$ Score
ImageNet		76.2	76.2	0%
ImageNetV2		64.3	70.1	+5.8%
ImageNet-R		37.7	88.9	+51.2%
ObjectNet		32.6	72.3	+39.7%
ImageNet Sketch		25.2	60.2	+35.0%
ImageNet-A		2.7	77.1	+74.4%

Figure 13. **Zero-shot CLIP is much more robust to distribution shift than standard ImageNet models.** (Left) An ideal robust model (dashed line) performs equally well on the ImageNet distribution and on other natural image distributions. Zero-shot CLIP models shrink this “robustness gap” by up to 75%. Linear fits on logit transformed values are shown with bootstrap estimated 95% confidence intervals. (Right) Visualizing distribution shift for bananas, a class shared across 5 of the 7 natural distribution shift datasets. The performance of the best zero-shot CLIP model, ViT-L/14@336px, is compared with a model that has the same performance on the ImageNet validation set, ResNet-101.



# 모델 사용법 선정

## Bag of Words-based

Image captioning (Transformer-based)

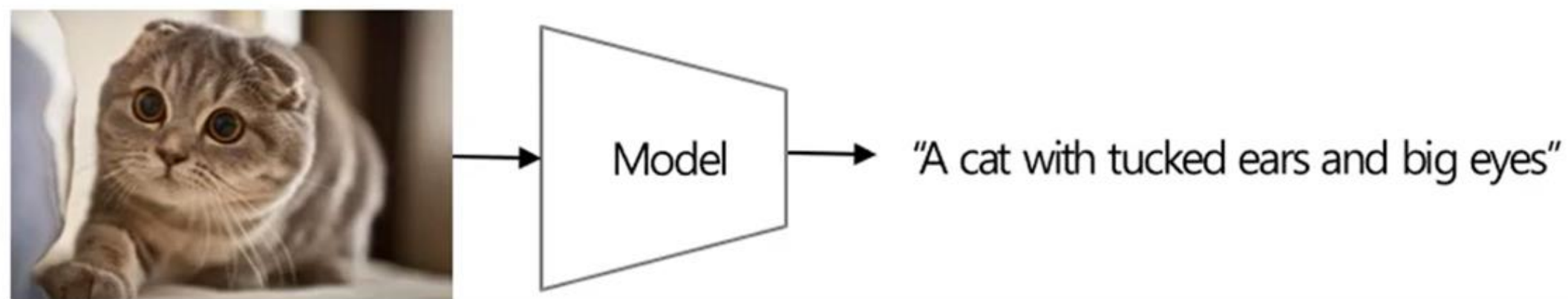
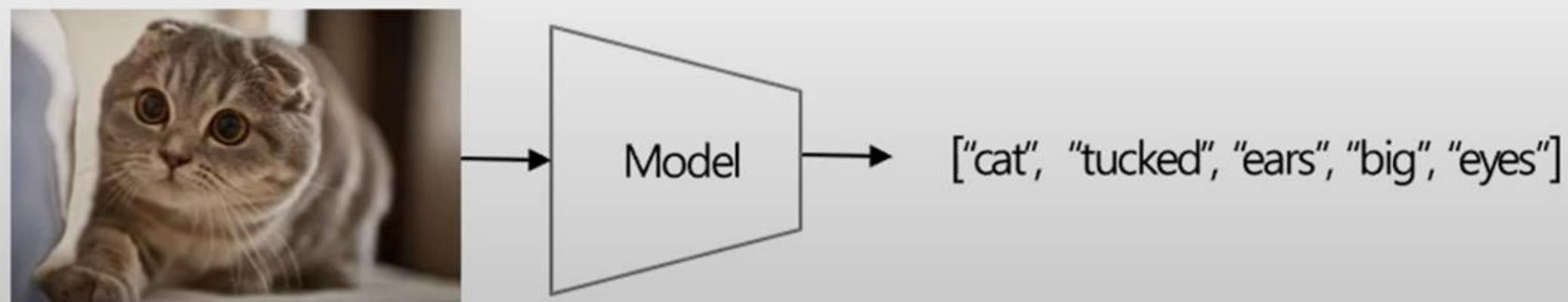


Image captioning (Bag of Words-based)





# 모델 성능 극대화

English class



한겨레  
볼일 마친 고양이가 돌진하는 ...



FOUR PAWS Australia  
A Cat's Personality - FOUR PAW...

```
## 한국어
label, prob = await predict_text_from_image("구글_고양이.jpg", ['고양이', 'cat'])
print(f"라벨 : {label}")
print(f"확률 : {prob}")
```

```
라벨 : cat
확률 : 0.9729471206665039
```

```
## 영어
label, prob = await predict_text_from_image("google_cat.jpg", ['고양이', 'cat'])
print(f"라벨 : {label}")
print(f"확률 : {prob}")
```

```
라벨 : cat
확률 : 0.9969338178634644
```

derestimate the potential of this line of research. To address this, we constructed a new dataset of 400 million (image, text) pairs collected from a variety of publicly available sources on the Internet. To attempt to cover as broad a set

strengths over other training methods. It's much easier to scale natural language supervision compared to standard crowd-sourced labeling for image classification since it does not require annotations to be in a classic "machine learning compatible format" such as the canonical 1-of-N majority vote "gold label". Instead, methods which work on natural

기초 한국어-영어 class 취



# 모델 성능 극대화

## 1대多 English class

### 크롤링을 활용한 모델 테스트 코드

```
def crawl(query, labels) :
    crawl_list = []
    url = 'https://www.google.com/search?sca_esv=2e034f13424ae33d&q=' + query + '&tbm=isch&s'
    response = requests.get(url)

    if response.status_code == 200:
        html = response.text
        soup = BeautifulSoup(html, 'html.parser')

        for idx, j in enumerate(soup.find_all('img')) :
            if idx == 0 :
                continue
            probs = predict_text_from_image(j['src'], labels)
            crawl_list.append({'url' : j['src'], 'probs' : probs})

    return crawl_list
```

```
def make_dict(result:list, index:int, query:str, labels:list):
    crawl_list = crawl(query, labels)

    for i in crawl_list :
        url = i['url']
        probs = i['probs']
        result.append({'query' : query, 'url' : url, 'labels' : labels, 'probs': probs})
    return result
```

### “노트북” 개별 사진 테스트 결과 raw

index	url	표제어	단어1	단어2	확률1	확률2
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQVi1ABDY4VvYjYQGq995FzPTwfx3b8omFgKkIaxIEydnN3fTKpH1OP_cuMRdA&s	노트북	la pto p	notebo o k computer	.89573 02	0.1042698
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQjV6cZzy_P3tzyhwKsVMZrXpvtF-5Quk93BJAPxeI753TQ1axk85BHEOhgQ&s	노트북	la pto p	notebo o k computer	.84341 19	0.1565882
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTOXt3Gel5k19HDjNNLko8pbA-sg5s2jDKfGy2A7bHHgxMw5ecIW72fHmJYGA&s	노트북	la pto p	notebo o k computer	0.953889	0.046111
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcScvmd--8D80TLfNXQ7OgU TauMVXizq49bqTVzZtjagKzT3ob9ZsCy2D00g&s	노트북	la pto p	notebo o k computer	.91773 54	0.0822646
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTcBtwSLJwVSbiqFcpn1x8EcORkC2NC96jtLwC_qXrjGwGC KLJPHvCs5Smilg4Y&s	노트북	la pto p	notebo o k computer	.75963 03	0.2403697
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRxDs5h7z0swDU1AQPLidQz1T6HbJb1x75zweB-6CJ1neCPVVl8_y4Yca5q4g&s	노트북	la pto p	notebo o k computer	.7827699	0.2172302
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTCqmKziVm06J4Zqv7hewjx83rQBwd-ivWjjiGN6y50mBBYfB7NlgiscB9eRHl&s	노트북	la pto p	notebo o k computer	.8600683	0.1399318
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ-eie4IzQyM5Xc3Vd5Ssnrau7GLDAW4vXm6Ro75Gte-77oPpHu101OMyAnyg&s	노트북	la pto p	notebo o k computer	.8618881	0.138112
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRYwLyfWyoF7dqC0HvNny2-uYx1IjEyMeu8-LihFFK8BMf5V9OUdijFAMvoY&s	노트북	la pto p	notebo o k computer	.8655949	0.1344051
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR7A83VcqEMD6GWzVG53MWgOOoR1WiybeOwmAAtR5Kix2GfTU LPp3vgSvycOg&s	노트북	la pto p	notebo o k computer	.8516938	0.1483063
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9Gc5Qkjbp5ihns-yaBziTT5WnADk29ojOj_WwrlmZHD DmDP5G3Jydyzfb2veo5Q&s	노트북	la pto p	notebo o k computer	.9018756	0.0981244
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTL1xnYnkJH46Z-QMegL5EzNz6pa3rIY3EQQr2FYIzRTX_AdhacIojQKyIA&s	노트북	la pto p	notebo o k computer	.8840666	0.1159333
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ5y5FuBCEc-rGrbswcCdlr202_p3NnqVpz6g2j0cpVMsRi2aEEIWA4ZnFZbGc&s	노트북	la pto p	notebo o k computer	.8565453	0.1434547
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ8iP7IU4u35Hg1hXscvMimDh6XnCI5wC5-sVvj4Ebbjy7U97v1-j_wVUBU&s	노트북	la pto p	notebo o k computer	.9177082	0.0822918
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQu_ua fKN2auLpp5nh-R-PXHpaRIIHUXggp-LaKjOAu1ppmuon7fllgv5swwB M&s	노트북	la pto p	notebo o k computer	.9153309	0.0846691
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSDkMobQYe7JUABx3ecjFjbiBsvR5XGjAfpkmi_CyQ3mcBOWUnqNxxwli3Nydc&s	노트북	la pto p	notebo o k computer	.7734063	0.2265937
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTcomvaFfIMLIXmFEX9Sjdu3WbDUYvTn4oHEV-LhF18DgbxjgKONzaQxBv2ow&s	노트북	la pto p	notebo o k computer	0.927716	0.0722841
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSL5xTC6TG6Ca7gtsujhZX82aQKwP7RIRCOLa6G8hhLy9we2mIFTczCfnEw&s	노트북	la pto p	notebo o k computer	.7922019	0.2077981
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQYFIK2KNsmLL_kjR9imsSq-iBgB1aTBaf8Ofn8x-__ax7ckXZhikZpmxVwrv&s	노트북	la pto p	notebo o k computer	.9112738	0.0887262
306	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9Gc5FdMVQB8AI9WIMFjiukoE5Sm7gRD8E6VMBXyGSTHUXyZ_jLz801IdTD4KPT3CY&s	노트북	la pto p	notebo o k computer	.9587022	0.0412978

### 테스트에 따른 English class 선정 : Laptop > Notebook

306	노트북	computer laptop	87.92%	12.08%
307	녹두	mung beans	98.84%	1.16%
308	녹말	starch	50.00%	50.00%



# 모델 성능 극대화

## class에 유동적인 서버 구축

# 이미지 업로드 엔드포인트

@app.post("/")

async def upload\_image(file: UploadFile, label:str):

# 이미지 저장

image\_bytes = await file.read()

list\_labels = label.split(",")

label, prob = await clip\_model.predict\_text\_from\_image(image\_bytes, list\_labels)

result = f"{label}"

# JSON 데이터 출력

return result

POST / Upload Image

Parameters

Cancel

Reset

Name	Description
label * required string (query)	<input type="text" value="dog,cat,snake"/>

Request body required

multipart/form-data

file \* required  
string(\$binary)

파일 선택

cat.webp

# 모델 성능 테스트 결과

## 129개 클래스 선정

★테스트 데이터 생성(직접 촬영본 약 40%)

구분		Korean Class	English Class	(Selected) English Class	A photo of English Class	KoClip
Accuracy		0.8%	83.7%	84.5% (▲ 0.8)	86.7% (▲ 2.2)	88.4%
Max Prob (Avg)	Label = pred	12.5%	81.6%	83.8% (▲ 2.2)	82.9% (▼ 0.9)	86.1%
	Label != pred	7.8%	45.5%	43.8% (▼ 1.7)	46.3% (▲ 2.5)	43.6%

구분		A photo of English Class		KoClip	
		직접 촬영본	수집본	직접 촬영본	수집본
Accuracy		82.3%	89.7%	90.2%	87.2%
Max Prob (Avg)	Label = pred	76.6%	87.9%	79.4%	91.9%
	Label != pred	44.5%	48.3%	39.9%	45.4%

# 모델 성능 정성 평가

KoClip이 Clip보다 뛰어난 부분



Clip : a photo of soup 74.9%  
KoClip : 국수 46.0%



Clip : a photo of steam iron 57.5%  
KoClip : 면도기 99.2%



Clip : a photo of bottle 13.0%  
KoClip : 분무기 45.8%

# TTS 선정

typecast vs Clova Voice



- URL 3회에 걸친 요청 필요



- URL 1회 요청 필요

# RAG 활용 배경 및 Gemini 선정

## 동화 기반 퀴즈

UI

You

흥부전 3지선다 퀴즈 생성해

ChatGPT

ChatGPT

알겠습니다! 흥부전에 관한 3지선다 퀴즈를 만들어 보겠습니다.

1. 흥부전은 다음 중 어느 나라의 전래동화인가요?

- 가) 한국
- 나) 중국
- 다) 일본

2. 흥부전에서 흥부가 귀여운 동물과 만나는 장면은 어디에서 나타날까요?

- 가) 산속
- 나) 강가
- 다) 마을

3. 흥부전의 주인공인 흥부는 무슨 일을 하고 있었나요?

- 가) 농부
- 나) 장사꾼
- 다) 화가



흥부전 3지선다 퀴즈

1. 흥부의 형제 이름은 무엇일까요?

- (a) 놀부
- (b) 덜부
- (c) 갑부

2. 흥부는 어떻게 하여 부자가 되었나요?

- (a) 용궁에 가서 용왕에게 도움을 받았다.
- (b) 보물을 발견했다.
- (c) 열심히 일해서 돈을 모았다.

3. 흥부의 아내는 어떤 역할을 했나요?

- (a) 흥부를 도와 부자가 되도록 했다.
- (b) 흥부의 형제를 도와 줬다.
- (c) 흥부의 아이들을 돌보았다.

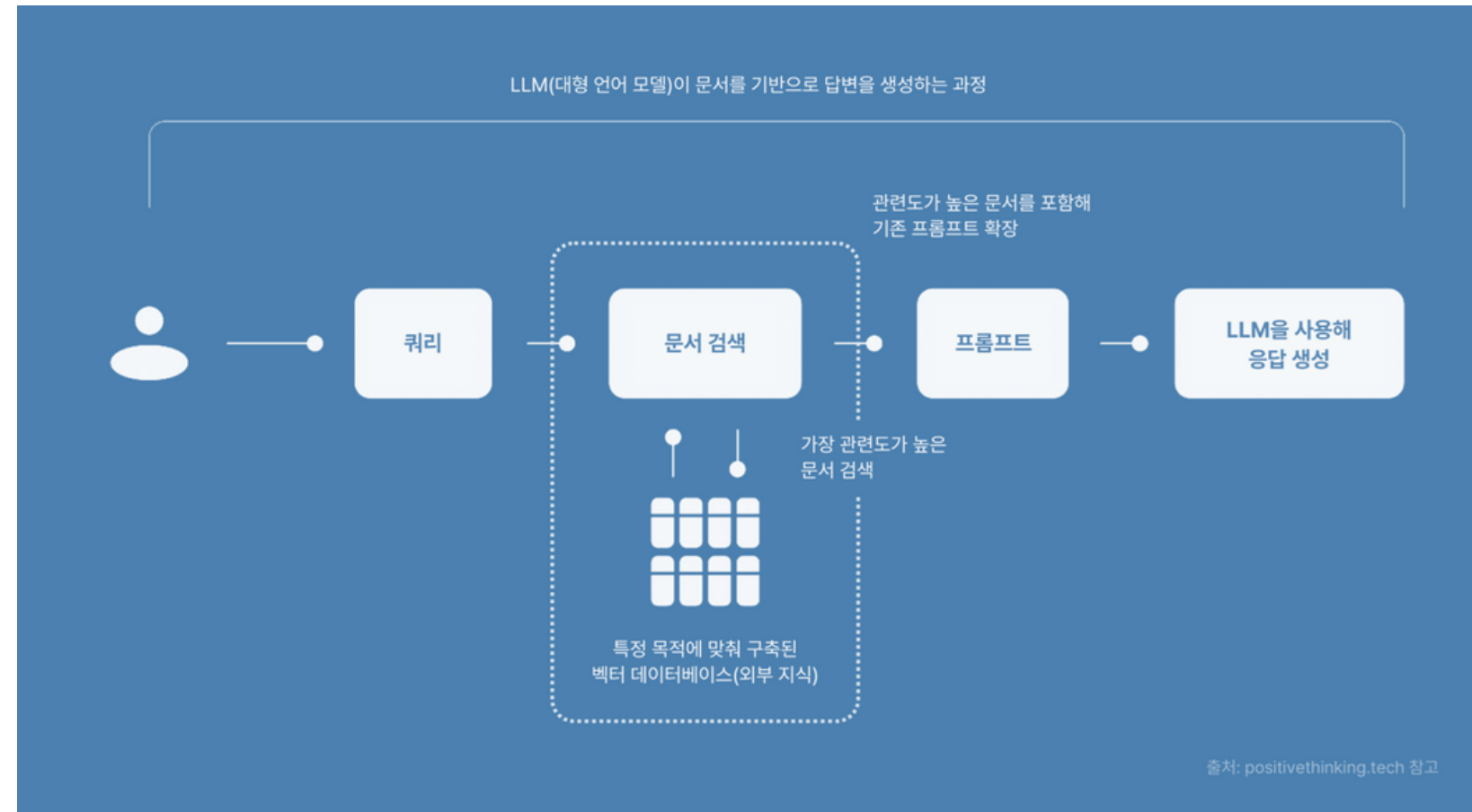
4. 놀부는 어떻게 하여 벌을 받았나요?

- (a) 용궁에 간헐다.
- (b) 가난해졌다.
- (c) 죽었다.



# RAG 활용 장점

## Hallucination 해결



- Document Loader os.read 사용
- Text Splitter 동화책 텍스트 1,000자 미만 사용
- Embedding 한국어 전용 jhgan/ko-sbert-nli 사용
- Vector Store Chroma Vector DB 사용
- Retrievers 동화책 색인을 위한 기본 검색기 사용

Table 1: Open-Domain QA Test Scores. For TQA, left column uses the standard test set for Open-Domain QA, right column uses the TQA-Wiki test set. See Appendix D for further details.

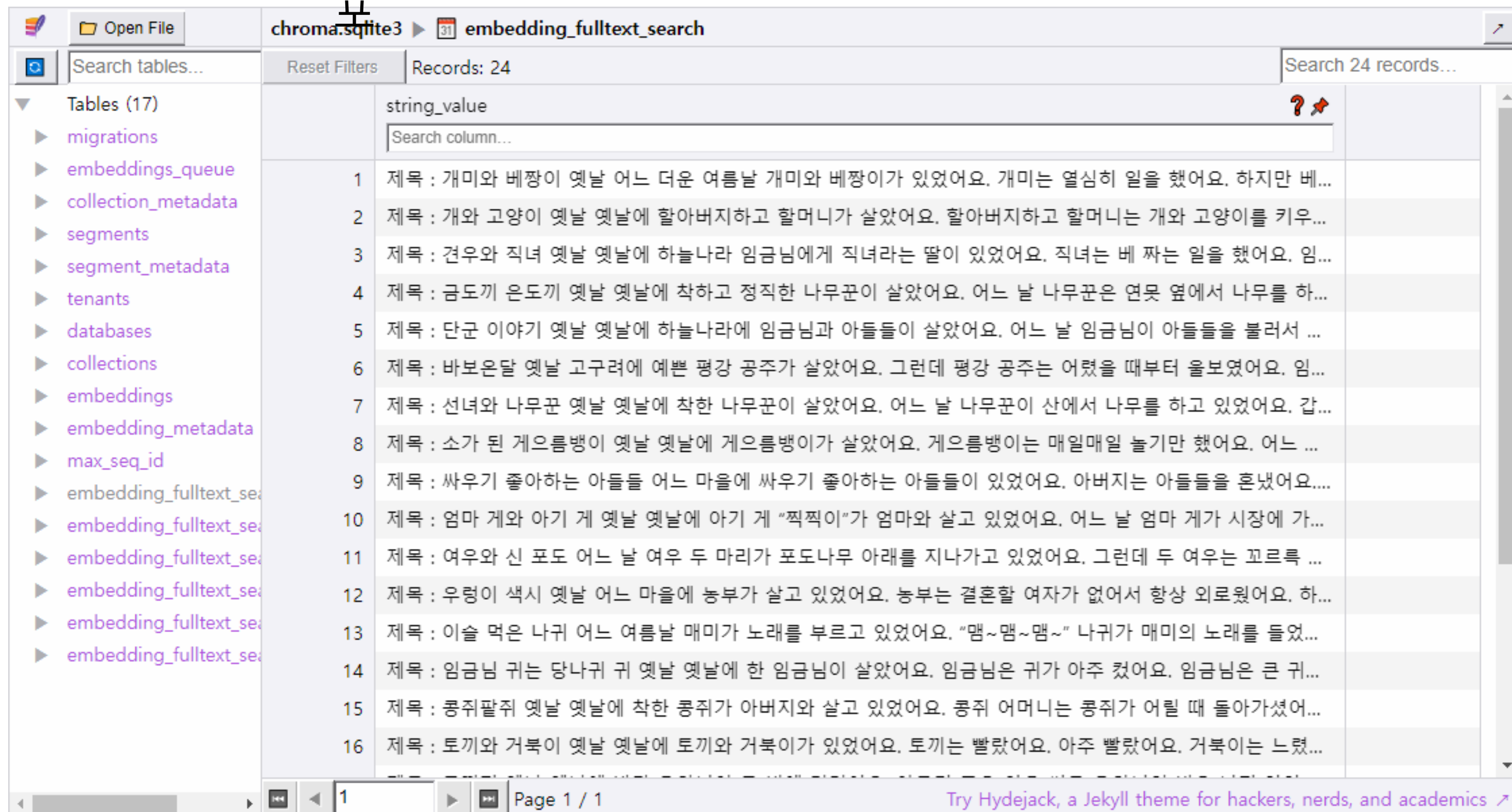
	Model	NQ	TQA	WQ	CT
Closed Book	T5-11B [52]	34.5	- / 50.1	37.4	-
	T5-11B+SSM[52]	36.6	- / 60.5	44.7	-
Open Book	REALM [20]	40.4	- / -	40.7	46.8
	DPR [26]	41.5	<b>57.9</b> / -	41.1	50.6
	RAG-Token	44.1	55.2/66.1	<b>45.5</b>	50.0
	RAG-Seq.	<b>44.5</b>	56.8/ <b>68.0</b>	45.2	<b>52.2</b>

Large pre-trained language models have been shown to store factual knowledge in their parameters, and achieve state-of-the-art results when fine-tuned on downstream NLP tasks. However, their ability to access and precisely manipulate knowledge is still limited, and hence on knowledge-intensive tasks, their performance lags behind task-specific architectures. Additionally, providing provenance for their decisions and updating their world knowledge remain open research problems. Pre-

to more effective marginalization over documents. Furthermore, RAG can generate correct answers even when the correct answer is not in any retrieved document, achieving 11.8% accuracy in such cases for NQ, where an extractive model would score 0%.

# Vector DB 구조

## Vector DB 구축 후 배포



The screenshot shows a SQLite browser interface with a table named 'embedding\_fulltext\_search' containing 24 records. The table has a single column 'string\_value'. The records contain Korean text snippets, likely representing search results or document embeddings. The interface includes a sidebar with a list of tables, a search bar, and a footer with navigation controls and a page indicator.

	string_value
1	제목 : 개미와 베짚이 옛날 어느 더운 여름날 개미와 베짚이가 있었어요. 개미는 열심히 일을 했어요. 하지만 베...
2	제목 : 개와 고양이 옛날 옛날에 할아버지하고 할머니가 살았어요. 할아버지하고 할머니는 개와 고양이를 키우...
3	제목 : 견우와 직녀 옛날 옛날에 하늘나라 임금님에게 직녀라는 딸이 있었어요. 직녀는 베 짜는 일을 했어요. 임...
4	제목 : 금도끼 은도끼 옛날 옛날에 착하고 정직한 나무꾼이 살았어요. 어느 날 나무꾼은 연못 옆에서 나무를 하...
5	제목 : 단군 이야기 옛날 옛날에 하늘나라에 임금님과 아들들이 살았어요. 어느 날 임금님이 아들들을 불러서 ...
6	제목 : 바보온달 옛날 고구려에 예쁜 평강 공주가 살았어요. 그런데 평강 공주는 어렸을 때부터 울보였어요. 임...
7	제목 : 선녀와 나무꾼 옛날 옛날에 착한 나무꾼이 살았어요. 어느 날 나무꾼이 산에서 나무를 하고 있었어요. 갑...
8	제목 : 소가 된 게으름뱅이 옛날 옛날에 게으름뱅이가 살았어요. 게으름뱅이는 매일매일 놀기만 했어요. 어느 ...
9	제목 : 싸우기 좋아하는 아들들 어느 마을에 싸우기 좋아하는 아들들이 있었어요. 아버지는 아들들을 혼냈어요....
10	제목 : 엄마 게와 아기 게 옛날 옛날에 아기 게 "찍찍이"가 엄마와 살고 있었어요. 어느 날 엄마 게가 시장에 가...
11	제목 : 여우와 신 포도 어느 날 여우 두 마리가 포도나무 아래를 지나가고 있었어요. 그런데 두 여우는 꼬르륵 ...
12	제목 : 우렁이 색시 옛날 어느 마을에 농부가 살고 있었어요. 농부는 결혼할 여자가 없어서 항상 외로웠어요. 하...
13	제목 : 이슬 먹은 나귀 어느 여름날 매미가 노래를 부르고 있었어요. "멤~멤~멤~" 나귀가 매미의 노래를 들었...
14	제목 : 임금님 귀는 당나귀 귀 옛날 옛날에 한 임금님이 살았어요. 임금님은 귀가 아주 컸어요. 임금님은 큰 귀...
15	제목 : 콩쥐팥쥐 옛날 옛날에 착한 콩쥐가 아버지와 살고 있었어요. 콩쥐 어머니는 콩쥐가 어릴 때 돌아가셨어...
16	제목 : 토끼와 거북이 옛날 옛날에 토끼와 거북이가 있었어요. 토끼는 빨랐어요. 아주 빨랐어요. 거북이는 느렸...



# 추가 개선 사항

## RAG 적용 후에도 나타나는 문제

```
"question": "용왕님이 앓은 병은 무엇이었나요?",  
"options": ["감기", "두통", "심장병"],  
"answer": 0
```



간간히 나타나는  
Hallucination

```
safety_ratings {  
  category: HARM_CATEGORY_DANGEROUS_CONTENT  
  probability: NEGLIGIBLE  
}
```



Safety ratings error 발생

```
"question": "박속에서 나온 것 중에 나쁜 것은 무엇입니까?",  
"options": ["도깨비", "돈", "집"],  
"answer": 0
```



이상한 한국어 어감

```
"question": "토끼를 바다에 데려간 동물은?",  
"options": ["고래", "거북이", "상어"],  
"answer": 1
```



문제 다양성 부족

```
"question": "콩쥐의 새어머니 이름은 뭐니?",  
"options": ["제비쥐", "팔쥐", "쥐콩"],  
"answer": 1
```



반말 사용

# Prompt Engineering langchain 활용

template = """"Make three quizzes for {years} years old kids and return list like '[json, json, json]'.

Quiz format rule:

- one right answer.
- three options.
- Use careful terms that fit the **Korean sentence** and **Use honorifics.** → 반말 및 한국어 어감 조정
- Options sentence must be short for kid

Making quiz tips

- Based on below context.
- **Don't care about the time order of the context.** → 문제 다양성 확보
- **Use the peripheral part and the whole context together.** 보 → **Hallucination** 추가 개선
- **Don't use NEGLIGIBLE word.** → **Safety ratings** 문제 개선

json format: → **One Shot Prompting**

```
"question" : "quiz",
"options" : ["option1","option2","option3"],
"answer" : 0
```

context:

{context}

{again} → Please must start with [ and end with ]

""""

# 모델 성능 개선 결과

흥부와 놀부 퀴즈 30개 생성, Years prompt “6”으로 고정, temperature = 1  
기준

구분	No RAG	RAG	RAG + Prompt Engineering
Hallucination	27개	5개 (▼22)	3개 (▼2)
반말	-	15개	0개 (▼15)
반복 문제 출제	-	13개	6개 (▼7)
이상한 한국어 어감	-	7개	5개 (▼2)
Safety ratings error	-	4개	1개 (▼3)
json형태 반환 실패	-	-	0개

★ 문제 다양화를 위한 추가 장치 : Hyperparameter 조율 : temperature 랜덤  
배정

```
Future<List<dynamic>?> _GetQuiz({required String topic}) async {
  var randomNumber = random.nextDouble();
  String temperature = randomNumber.toString();
```

# RAG를 활용한 퀴즈 generate 기능의 발전 방향

2024학년도 대학수학능력시험 문제지

제1교시

국어 영역

출수형

11~31 다음 글을 읽고 물음에 답하시오.

독서는 독자가 목표한 결과에 도달하기 위해 글을 읽고 의미를 구성하는 인지 행위이다. 성공적인 독서를 위해서는 초인자가 중요하다. 독서에서의 초인자는 독자가 자신의 독서 행위에 대해 인지하는 것으로서 자신의 독서 과정을 점검하고 조정하는 역할을 한다.

초인자는 글을 읽기 시작할 후 지속적으로 이루어지는 점검 과정에 동원된다. 독자는 가장 적절한다고 판단한 독서 전략을 사용하여 독서를 진행하는데, 그 전략이 효과적이고 문제가 없는지를 평가하며 점검한다. 효과적이지 않거나 문제가 있다고 판단하면 이를 해결해야 한다. 문제가 무엇인지 분명하지 않은 경우에는 독서 중에 떠오르는 생각들을 살펴보고 그중 독서의 진행을 방해하는 생각들을 분류해 보는 방법으로 문제점이 무엇인지 파악할 수 있다. 독자가 중단 없이 이어지는 상태이지만 문제가 발생한 것을 독자 자신이 인지하지 못하는 경우도 있다. 의도한 목표에 부합하지 않는 방법으로 읽기를 진행하거나 자신이 이해한 정도를 판단하지 못하는 예가 그것이다. 문제 발생 여부의 점검을 위해서는 독서 진행 중간중간에 이해한 내용을 정리하는 방법을 사용할 수 있다.

초인자는 문제를 해결하기 위해 독서 전략을 조정하는 과정에도 동원된다. 독서 목표를 고려하여, 독자는 ㉠ 지금 사용하고 있는 전략을 계속 사용할 것인지 판단해야 한다. 또 ㉡ 문제 해결을 위한 다른 전략에는 무엇이 있는지, ㉢ 각 전략의 특징과 사용 절차, 조건 등은 무엇인지 알아야 한다. 또한 독자 자신이 사용할 수 있는 전략이 무엇인지, ㉣ 전략들의 적절한 적용 순서가 무엇인지, ㉤ 현재의 상황에서 최적의 전략이 무엇인지 판단하여 새로운 전략을 선택한다. 선택한 전략을 수행하는 과정에서 독자는 초인자를 활용하여 점검과 조정을 되풀이하며 능동적으로 의미를 구성해 간다.

1. 밑줄친 내용을 바탕으로 적절하지 않은 것은?

① 독서 전략을 선택할 때 독서의 목표를 고려할 필요가 있다.  
② 독서 전략의 선택을 위해 개별 전략들에 대한 지식이 필요하다.  
③ 독서 목표의 달성을 위해 독자는 자신의 독서 행위에 대해 인지해야 한다.  
④ 독서 문제의 해결을 위해 독자는 자신이 사용할 수 있는 전략이 무엇인지 알아야 한다.  
⑤ 독서 문제를 해결하기 위해 새로 선택한 전략은 점검과 조정의 대상에서 제외될 필요가 있다.

2. [A]에서 할 수 있는 내용으로 가장 적절한 것은?

① 독서 진행 중 이해한 내용을 정리하는 것은 독자 스스로 독서 진행의 문제를 점검하는 데에 적합하지 않다.  
② 독서 진행 중 독자가 자신이 얼마나 이해하고 있는지 파악하지 못할 때에는 점검을 잠시 보류해야 한다.  
③ 독서 진행에 문제가 없어 보더라도 목표에 부합하지 않는 독서가 이루어지는 경우가 있다.  
④ 독서 중에 떠오르는 생각을 분류하는 것은 독서 문제의 발생을 막는다.  
⑤ 독서가 멈추지 않고 진행될 때에는 초인자의 역할이 필요 없다.

3. <보기>는 밑줄친 읽은 학생이 독서 중 떠올린 생각이다. ㉠~㉤과 관련하여 ㉡~㉤을 설명한 내용으로 적절하지 않은 것은? [3점]

<보 기>

○ 이 용어가 무슨 뜻인지 모르겠어.  
○ 처음 나왔을 때는 무시하고 읽었는데 다시 등장했으니, 문맥을 통해 의미를 가늠하고 읽어 봐야겠어. .... ㉠

↓

○ 더 읽어 보았지만 여전히 정확한 뜻을 모르겠네. 그럼 어떻게 하지?  
○ 관련된 내용을 앞부분에서 다시 찾아 읽든가, 인터넷 자료를 검색해 보든가, 다른 책들을 찾아볼 수 있겠네. .... ㉡  
○ 검색을 하려면 인터넷 접속이 필요하겠네. .... ㉢  
○ 검색은 나중에 하고, 먼저 앞부분을 다시 읽어 봐야겠다. 그다음에 다른 책을 찾아봐야지. .... ㉣  
○ 그럼 일단 앞부분에 관련된 내용이 있었는지 읽어 보자.

↓

○ 앞부분에는 관련된 내용이 없어서 도움이 안 되네.  
○ 이 용어와 관련된 분야의 책을 찾아보는 것이 가장 좋겠어. .... ㉤

↓

○ 이제 이 용어의 뜻이 이해되네. 그럼 계속 읽어 볼까?

㉠ ㉡: ㉢을 판단하여 사용 중인 전략을 계속 사용하기로 결정했다.  
㉡ ㉢: ㉤을 고려하여 선택할 수 있는 전략들을 떠올렸다.  
㉢ ㉣: ㉤을 고려하여 전략의 사용 조건을 확인했다.  
㉣ ㉤: ㉤을 판단하여 전략들의 적용 순서를 결정했다.  
㉤ ㉤: ㉤을 판단하여 최적이라고 생각한 전략을 선택했다.

1

20

이 문제지에 관한 저작권은 한국교육과정평가원에 있습니다.

텍스트 기반 시험의 문제 생성 가

능  
각종 자격증 시

수능 국어영

수능 사회탐구영

역

Generative Adversarial Networks기

반

문제 검수해줄 AI와 함께 성능 향상

가능

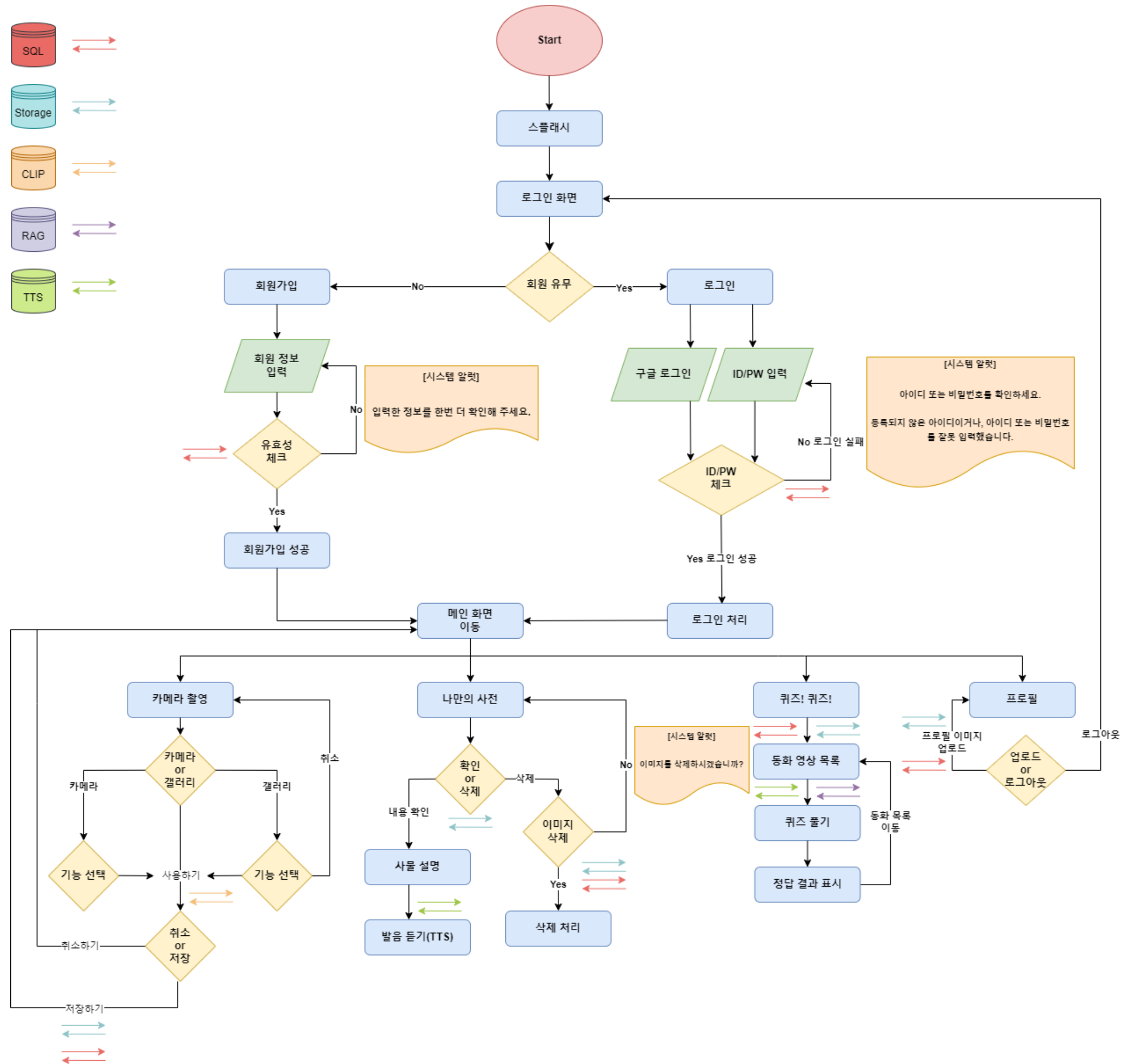
프로젝트 예상 결  
과

# App design & features

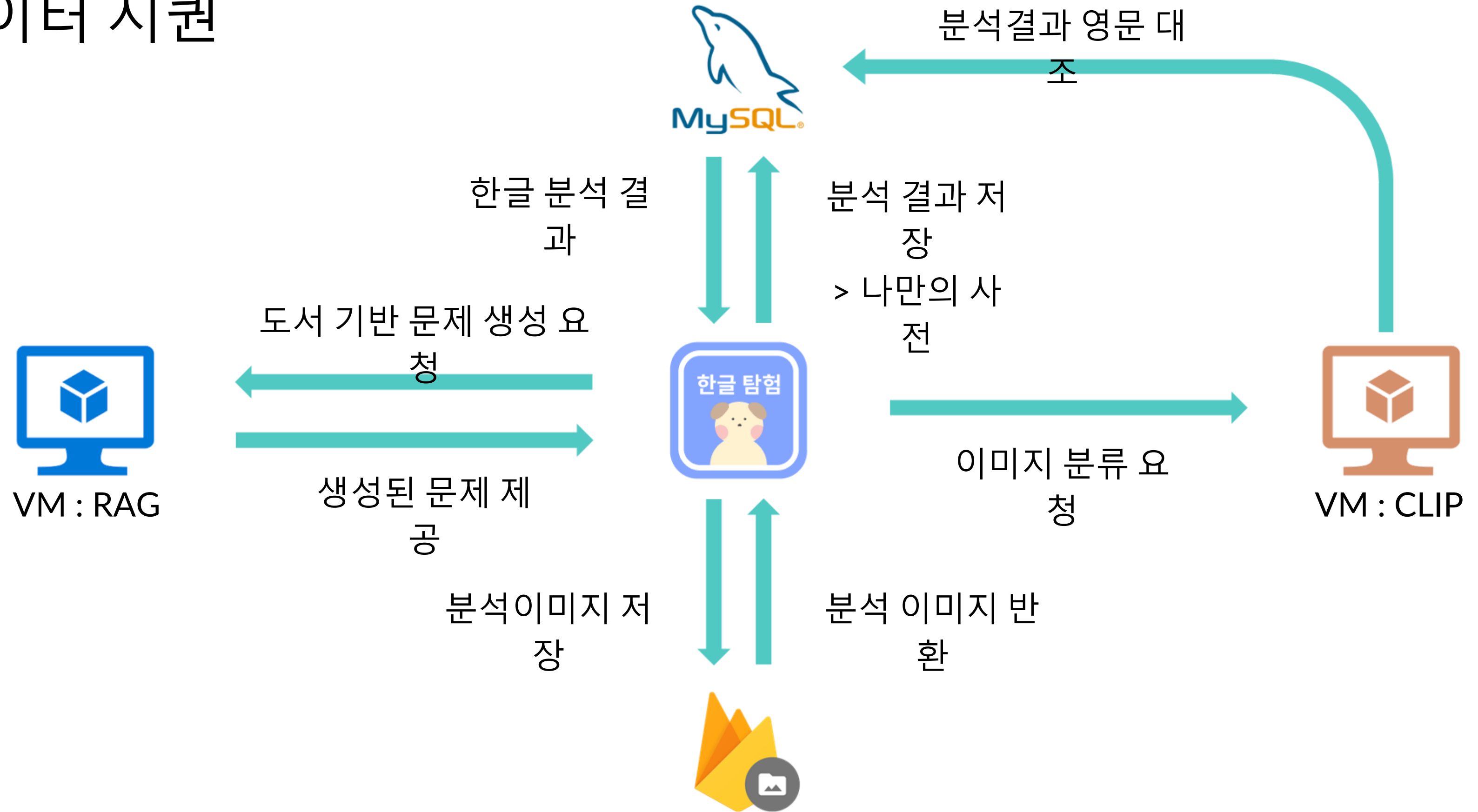
발표자 강영  
현



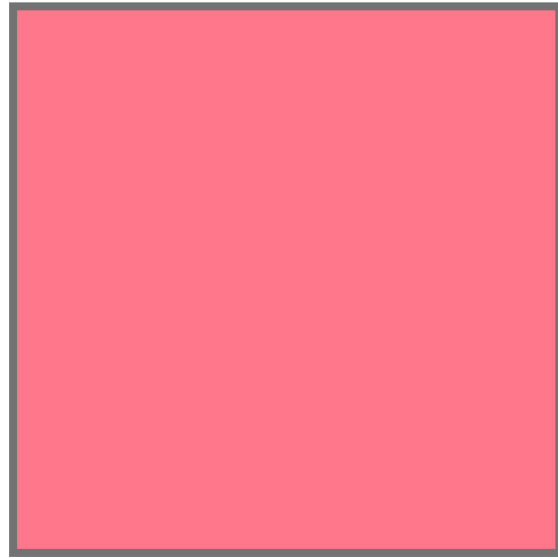
# Flowchart



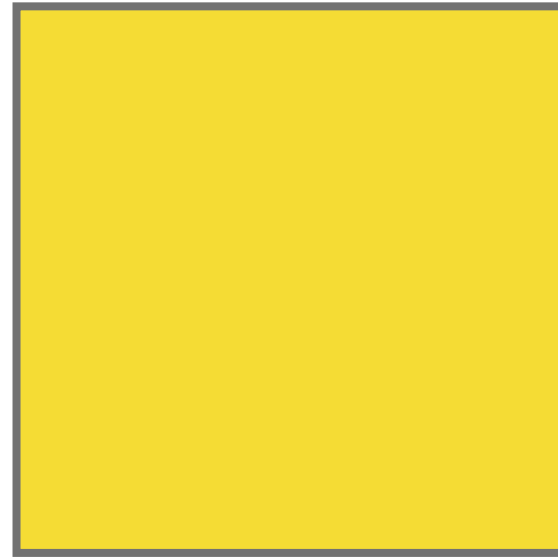
# 데이터 시퀀스



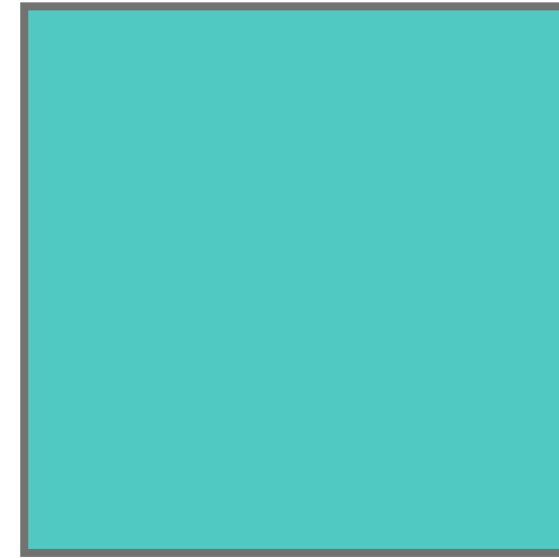
# Branding Colors



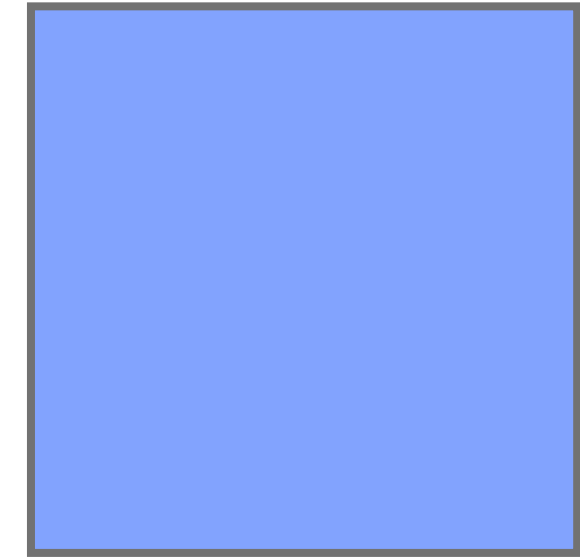
#FF778A



#F5DC34



#51C9C3



#82A3FE

어린 아이들이 자주 접하게 되는 색상으로 선정하였습니다.

유치원 및 어린이집에서 환경 구성을 할 때 가장 많이 사용되는 컬러이며, RGB 계열의 자극적인 색상 계열로 아이들의 이목을 집중시켰습니다.

# Branding Icon and Logo

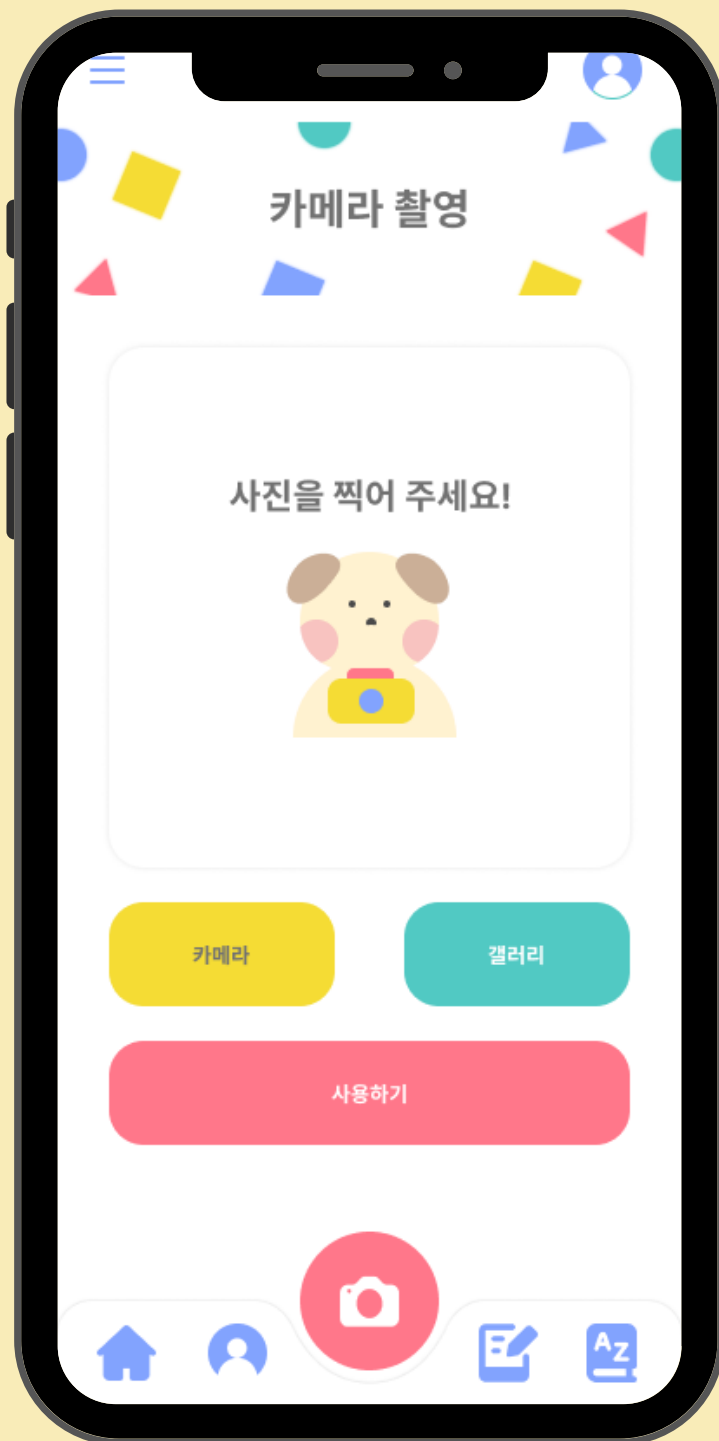


App  
Icon



App  
Logo

# 앱 기능 구현



## Upload Camera Screen

디바이스에 있는 카메라와 갤러리 기능을 Image\_picker Package로 구현하여 현재 촬영한 이미지가 앱 내부 경로에 저장됩니다.

경로에 저장된 이미지가 사진을 찍어 주세요! 부분에 출력이 되며 사용하기 버튼을 누를 시 다음 페이지에 이미지 경로가 전달됩니다.

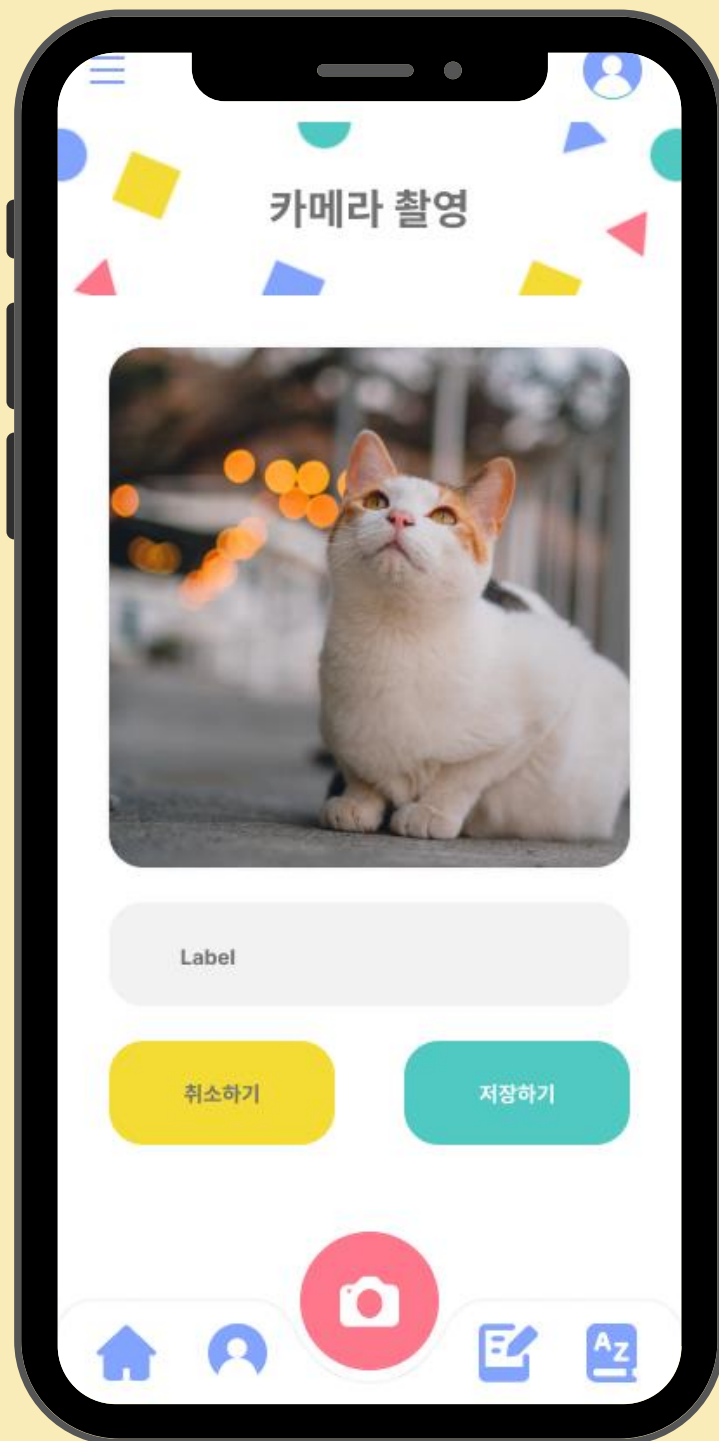
## Image\_Picker를 이용한 코

```
class _UploadCameraState extends State<UploadCamera> {
  XFile? _image; // 이미지 파일을 저장할 변수
  final ImagePicker _picker = ImagePicker(); // ImagePicker라는 함수를 _picker라고
  String? user_no; // user_no, 즉 스토리지에 붙일 이름(폴더)

  Future<void> _getImage(ImageSource source) async {
    final XFile? pickedFile =
      await _picker.pickImage(source: source, maxWidth: 300, maxHeight: 300);
    if (pickedFile != null) {
      setState(() {
        _image = pickedFile; // 이미지가 선택된다면, 그 이미지가 setstate로 인하여 되
      });
    }
  }

  Widget _buildPhotoArea() {
    return _image != null
      ? Container( // 이미지를 제대로 불러왔을 경우
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(20),
        ), // BoxDecoration
        width: 300,
        height: 300,
        child: Image.file(File(_image!.path)), // 이미지 파일을 출력함
      ) // Container
      : Container( // 이미지를 불러오지 않았을 경우 초기화면
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(20),
          color: Colors.white,
          boxShadow: [
            BoxShadow(
              offset: Offset(0, 0),
              blurRadius: 2,
              spreadRadius: 0,
              color: Colors.black.withOpacity(0.2)), // BoxShadow
          ],
        ), // BoxDecoration
```

# 앱 기능 구현



## Download Camera Screen

Upload Camera Screen에서 받아온 이미지를 Arguments값으로 받아와 모델 API에 전달한 뒤, 해당 이미지의 Label을 받아 출력을 하여 보여줍니다. 또한 저장하기 버튼을 누를 경우 Firebase와 SQL에 접근하여 사진 및 데이터 정보가 저장됩니다.

## 정보값 받기 및 모델API 접근 코드

```
var arguments =
    ModalRoute.of(context)?.settings.arguments as Map<String, dynamic>;
String path = arguments['path'];

if (label == null) {
    GetData(path);
}

void GetData(String path) async {
    //모델api 접근
    // 모델에 접근하여 이미지의 label을 불러오며, 잘못되었을 경우 오류라고 출력
    // eng인 이유는 모델이 영어로 받는것이 매우 정확하기 때문
    String future_eng = await GetLabel().get_label(path) ?? "오류";
    late String? future_label;
    String? future_dic_no;

    if (future_eng != "오류") {
        // eng통해 sql 접근
        // 만약에 "오류" 라는 단어와 다른경우 SQL에 접근하여 라벨을 불러옴
        future_label = await sqlget().GetWordKorByEng(eng: future_eng);
        future_dic_no = await sqlget().GetNoByEng(eng: future_eng);
    } else {
        future_label = "서버 오류";
    }

    setState(() {
        label = future_label;
        dic_no = future_dic_no;
        isLoading = false;
    });
}

onPressed: () async {
    if (dic_no != null) {
        String? mydic_no = await sqlget().GetNewMyDicNo();
        await FirebaseClient(
            user_no: user_no, mydic_no: mydic_no) // Fireba
            .upload(path);
        await sqlget().SaveImageInfo(
            user_no: user_no,
            dic_no: dic_no,
            mydic_no: mydic_no);
    }
}
```

# 앱 기능 구현



## Quiz Screen + Question Screen

Book List Screen 에서 받은 제목 데이터가 RAG 모델을 통하여 문제를 딕셔너리 형태로 출제합니다.

{ “question” : “ 문제 내용 “,  
“option” : [”선택지1”, ”선택지2”, ”선택지3”],  
“answer” : 정답인 option의 index 번호 }  
형태로 배출됩니다.

퀴즈를 생성하기 전, 미리 비어 있는 question List에 딕셔너리 형태로 문제가 담긴 후, 요소를 하나씩 꺼내 출제되는 방식으로 코드 구성을 하였습니다.

## RAG 모델을 이용하여 List에 요소 Add

```
class QuizScreenState extends State<QuizScreenState> {  
    // 문제가 생성되지 않았을때를 위해 항상 초기화값을 설정  
    int questionIndex = 0;  
    int totalScore = 0;  
    List<Map<String, dynamic>> questionList = [];  
  
    bool isLoading = false; // 데이터 로딩 상태를 나타내는 변수 추가  
  
    // 문제를 풀때마다 다음 Index 번호를 호출하고, Score값을 0과 1로 구분지어왔는데 풀때마다!  
    void answerPressed(int score) {  
        setState(() {  
            questionIndex++;  
            totalScore += score;  
        });  
        print(totalScore);  
    }  
  
    void loadQuestionList(String title) async {  
        setState(() {  
            isLoading = true; // 데이터 로딩 중임을 표시  
        });  
        // questionList의 값을 RAG model API를 이용하여 채워 넣음  
        final loadedQuestions = await GetQuiz().GetQuizList(topic: title);  
        setState(() {  
            isLoading = false; // 데이터 로딩 완료 후 상태 업데이트  
            questionList = loadedQuestions ?? [];  
        });  
    }  
}
```



# 앱 기능 구현



## Quiz Screen + Question Screen

Book List Screen 에서 받은 제목 데이터가 RAG 모델을 통하여 문제를 딥서치 형태로 출제합니다.

{ "question" : " 문제 내용 ",  
"option" : ["선택지1", "선택지2", "선택지3"],  
"answer" : 정답인 option의 index 번호 }  
형태로 배출됩니다.

퀴즈를 생성하기 전, 미리 비어 있는 question List에 딥서치 형태로 문제가 담긴 후, 요소를 하나씩 꺼내 출제되는 방식으로 코드 구성을 하였습니다.

## 만들어진 List 요소 하나씩 문제로 출제

```
@override
Widget build(BuildContext context) {
  return FutureBuilder<void>(
    future: _ttsFuture,
    builder: (context, snapshot) {
      if (snapshot.connectionState == ConnectionState.waiting) {
        return Center(
          child: Image.asset('assets/images/Quiz_loading.gif'),
        ); // Center
      } else {
        return Column(
          children: [
            TitleBanner(text: '퀴즈! 퀴즈!'),
            SizedBox(height: 30),
            if (widget.questionList.isNotEmpty &&
              widget.questionIndex < widget.questionList.length)
              Padding(
                padding: EdgeInsets.all(20),
                child: Column(
                  children: [
                    SingleChildScrollView(
                      child: Container(
                        width: double.infinity,
                        color: Colors.transparent,
                        child: Column(
                          children: [
                            Text(widget.questionList[widget.questionIndex]
                              ['question']), // Text
                          ],
                        ), // Column
                      ), // Container
                    ), // SingleChildScrollView
                    SizedBox(height: 30),
                  ],
                ),
              ),
            ],
        );
      }
    },
  );
}
```

# 앱 기능 구현



## Quiz Screen + Question Screen

Book List Screen 에서 받은 제목 데이터가 RAG 모델을 통하여 문제를 딥서치 형태로 출제합니다.

{ "question" : " 문제 내용 ",  
"option" : ["선택지1", "선택지2", "선택지3"],  
"answer" : 정답인 option의 index 번호 }  
형태로 배출됩니다.

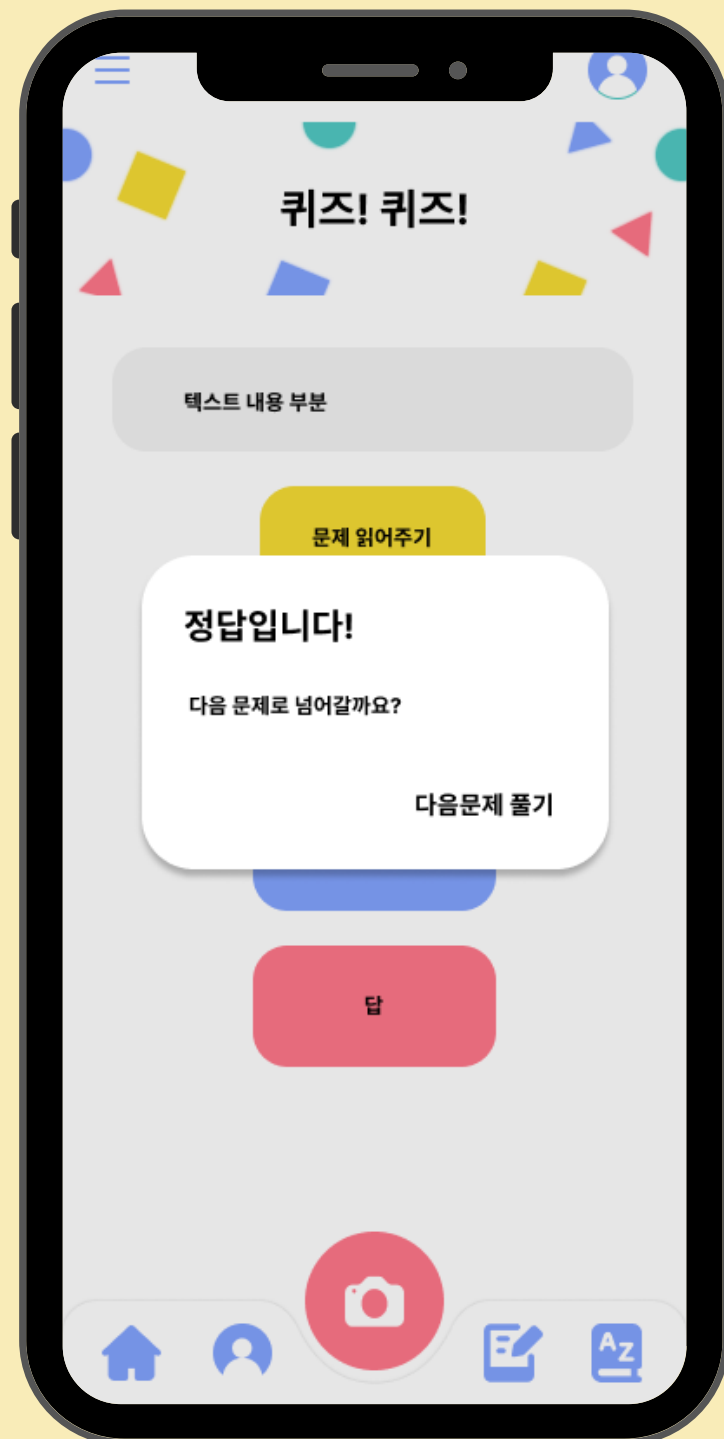
퀴즈를 생성하기 전, 미리 비어 있는 question List에 딥서치 형태로 문제가 담긴 후, 요소를 하나씩 꺼내 출제되는 방식으로 코드 구성을 하였습니다.

만들어진 **List** 요소 하나씩 문제로 출제

```
ListView.builder(  
  shrinkWrap: true,  
  itemCount: widget  
    .questionList[widget.questionIndex]['answer']  
    .length,  
  itemBuilder: (context, index) {  
    return Column(  
      children: [  
        ElevatedButton(  
          style: ElevatedButton.styleFrom(  
            backgroundColor: color_list[index],  
            foregroundColor: Colors.white,  
            fixedSize: Size(300, 60),  
            shape: RoundedRectangleBorder(  
              borderRadius: BorderRadius.circular(20)), // RoundedRectangleBorder  
            ),  
          onPressed: () {  
            bool isCorrect = widget  
              .questionList[widget.questionIndex]  
              ['answer'][index]['score'] > 0;  
            String correctAnswer = widget.questionList[widget.questionIndex]['answer'].firstWhere((element)  
              => element['score'] == 1)['text']; // 'score'가 1인 답을 찾아서 가져옴  
            _showResultDialog(isCorrect, correctAnswer);  
          },  
          child: Text(  
            widget.questionList[widget.questionIndex]  
              ['answer'][index]['text']), // Text  
          ), // ElevatedButton  
      ],  
    ),  
  ),  
);
```

```
Future<void> _playQuestionAudio(int index) async {  
  Uint8List? voice =  
    await GetSound(text: widget.questionList[index]['question'])  
      .get_voice();  
  _audioPlayer.byteplay(voice!);  
}
```

# 앱 기능 구현



## Quiz Screen + Question Screen

사용자가 문제를 풀 때 정답인지 아닌지를 확인하기 위해  
Alert Dialog widget을 사용하여, 정답일 경우에는 다음 문제 풀기를 유도합니다.

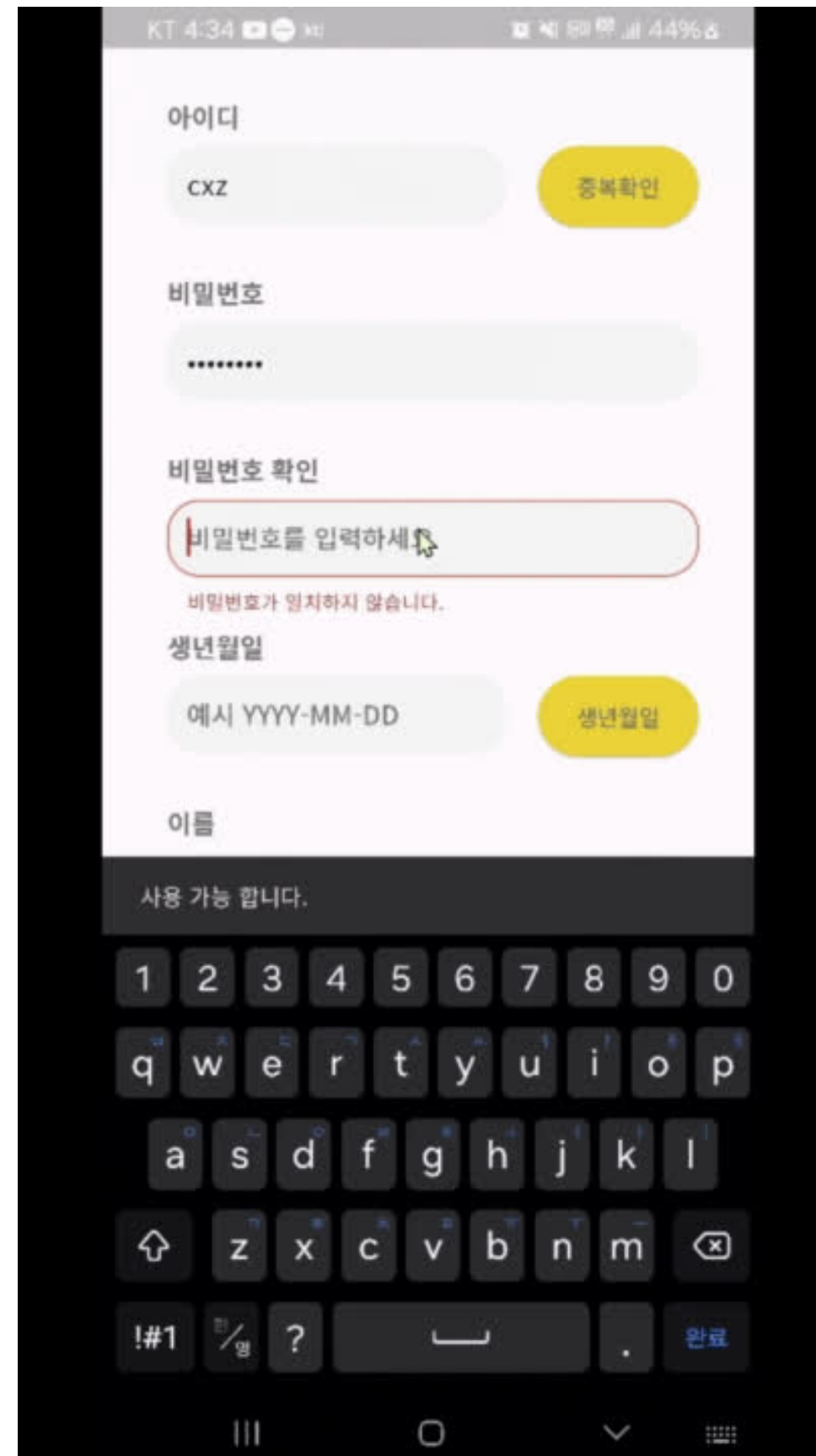
오답일 경우에는 정답이 무엇이었던지도 알려주는 팝업창을 뜨게 만들었습니다.

## Alert Dialog를 이용한 정답 팝업 위

```
Future<void> _playQuestionAudio(int index) async {
  Uint8List? voice =
    await GetSound(text: widget.questionList[index]['question'])
      .get_voice();
  _audioPlayer.byteplay(voice!);
}

void _showResultDialog(bool isCorrect, String correctAnswer) {
  showDialog(
    context: context,
    barrierDismissible: false,
    builder: (context) => AlertDialog(
      title: Text(isCorrect ? '정답입니다!' : '오답입니다!'),
      content: Text(isCorrect ? '다음 문제로 넘어갈까요?' : '정답은 $correctAnswer입니다. 계속 노력해주세요!'),
      actions: [
        TextButton(
          onPressed: () {
            Navigator.of(context).pop();
            setState(() {
              if (isCorrect) {
                // 정답을 맞추면 totalScore를 1 증가시킴
                widget.answerPressed(1);
              } else {
                // 오답일 경우에는 점수를 증가시키지 않음
                widget.answerPressed(0);
              }
            });
          },
          child: Text('다음 문제 풀기'),
        ), // TextButton
      ],
    ), // AlertDialog
  );
}
```

# 앱 기능 시연 영상





# 회

회	강영현	김민기	최강훈	한현종
KEEP	1. 모든 게 처음이라서 낯설고, 어려웠지만 배우려고 노력한 점 2. 단점을 고치려고 노력한 점	거의 모든 것이 처음하는 것이었지만 할 수 있다고 믿고 구글링, GPT 등을 활용하여 알아보며 시도해본 것, 6주 안에 결과물을 낸 점	1. 처음 시도해보는 플러터에 대해 열심히 알아보며 계속 코딩해보며 확인하여 만든 것	직면한 문제점을 해결하려고 여러가지 자료들을 공부한 것 새로운 서비스 이용에 두려움을 갖지 않고 도전한 점
PROBLEM	첫 프로젝트에, 첫 팀장이라서 많이 부족했던 것 같다.	진정한 의미의 서비스로는 아직까지 부족한 면을 가지고 있다는 점(여전히 개선이 필요한 AI모델, 비효율적으로 리소스를 낭비하는 코드 등)	1. 프론트엔드 코딩을 진행하면서 반응형 관련문제가 발생하여 고정크기로 진행함 2. 데이터를 로컬에 담지 못하고 모든것을 서버에서 요청받은 것	기초적인 개발 능력, 지식들이 부족하여 업무진행 속도가 느렸음
TRY	다음엔 프로젝트 시작 전에 Task에 대한 계획을 잘 세우고, 팀원들의 의견에 경청하는 자세를 가져야겠다.	첫 기획안인 Object Detection과 Clip모델 함께 이용할 수 있는 방안이 있는 지 알아보기, RAG 어려운 문서에 적용해보기	1. 반응형 디자인으로 다른 플랫폼 사이즈 까지 고려해 제작해보기 2. 로컬에 데이터를 저장해서 이용해보기	코딩의 기초부터 다시 공부나 혼자서도 하나의 프로젝트를 만들어보기
총평	긴 프로젝트를 팀원들이랑 잘 이겨냈다. 능력 좋은 팀원 분들이 이 프로젝트를 선택해주셔서 감사한 마음이다.	사실상 처음보는 Hugging face, 논문 등을 읽고 이용하는 것도, 만져본적도 없는 Dart언어를 다루는 것도 처음이라 재밌었고, 내가 부족하고 인지 못했던 부분들을 팀원분들이 잘 채워주워 감사했고, 그래도 기대이상의 결과가 나왔다.	AI를 접목시켜서 쓸수 있는 앱을 개발하는 플러터를 배울 수 있어서 좋았고, 결과물이 생각보다 좋게 나와서 기분이 좋습니다!	좋은 팀원들을 만나 나 혼자서는 절대 할 수 없을 프로젝트를 완수하였다. 앞으로도 프로젝트 진행에 용기를 가져도 될것 같다.

# Reference

- CLIP
  - 논문 : <https://arxiv.org/abs/2103.00020>
  - 참고 영상 : <https://www.youtube.com/watch?v=dELmmuKBUTl&t=598s>
- 한국어 기초 사전(클래스 제공) : <https://krdict.korean.go.kr/>
- Naver Clova Voice : <https://clova.ai/voice/>
- typecast : <https://typecast.ai/kr>
- RAG
  - 논문 : <https://arxiv.org/pdf/2005.11401>
  - 참고 영상 : <https://www.youtube.com/watch?v=NfQrRQmDrcc&t=2380s>
  - 코드 활용 : <https://www.youtube.com/watch?v=Xp7yFtpmXPk&t=1s>
- Gemini : <https://ai.google.dev/>
- Chatgpt : <https://chat.openai.com/>
- Prompt Engineering : <https://modulabs.co.kr/blog/prompt-engineering/>
- 세이브 더 칠드런(동화책 제공) : <https://www.sc.or.kr/>
- 사진 첨부
  - RAG : <https://www.skelterlabs.com/blog/2024-year-of-the-rag>
  - 수능 : <https://www.kice.re.kr/>
  - CBT : <https://play.google.com/store/apps/details?id=com.someday.question&hl=zu&pli=1>
  - 코로나 : <https://www.donga.com/news/Society/article/all/20210525/107093544/1>
  - 언어 발달 : <https://m.blog.naver.com/ccccc4u/223078358817>
  - 동화 : <https://www.newsbook.com/news/articleView.html?idxno=824> <https://dadoc.or.kr/599>
  - 호기심 : <https://post.naver.com/viewer/postView.nhn?volumeNo=29256122&memberNo=29566044>

찰칵 찰칵 한글 탐  
험

# Thank You

---

April 2024