읽거나 보고, 아는 것만 답변하는 지혜로운 기영이봇

#Open-Domain QA #Visual QA #No answerability

NLP-14
(팀 KiYOUNG2)
PM: 진명훈 ODQA&MRC: 김태욱, 유영재 VQA: 김대웅, 허진규 FE/BE: 김채은, 이하람

boostcamp ai tech

TABLE OF CONTENTS

1. Introduction
2. Machine Reading Comprehension (MRC)
3. Visual Question Answering (VQA)
4. Deployment
5. Conclusion

Appendix. 팀원 역량 소개

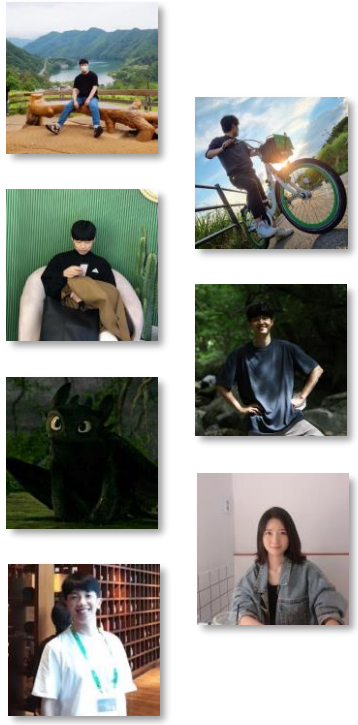
Introduction

안녕하세요 :) 팀 기영이입니다

왜 팀이름이 기영이?

- **K**orean **i**s all **YOU** **N**eed for dialo**G**u**e**
  - 한국어 대화, 질의 응답, 교육 도메인 자연어 처리에 진심인 팀원들이 모였습니다

진명훈 github.com/jinmang2
김대웅 github.com/KimDaeUng
김태욱 github.com/taeukkkim
허진규 github.com/JeangyuHeo
이하람 github.com/hrxorxm
김채은 github.com/Amber-Chaeeunk
유영재 github.com/uyeongjae



Boostcamp AI Tech 과정 3개월을 함께 했습니다

Level2 - NLP
'21 Oct

Level3 - Opt
`21 Nov

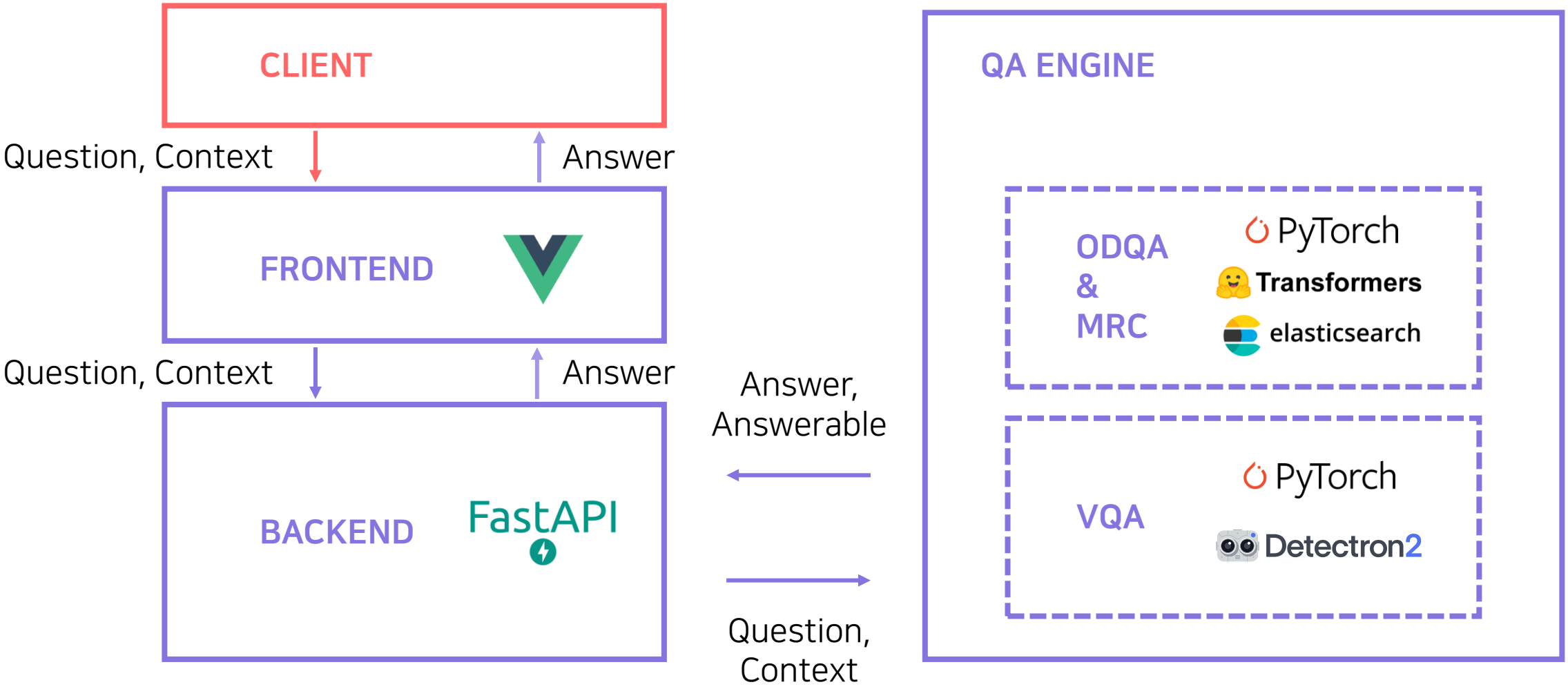
Level3 - MLOps
`21 Dec

최종 프로젝트 4주

- 저희의 프로젝트는 아래 세 가지의 질문에서 시작됐습니다.
  - 대회 때 개발한 모델을 현업에서 사용할 수 있을까?
  - 과연 질의응답은 Text 데이터로만 수행될까?
  - 사용자의 소통을 위한 UI는 어떻게 구성할까?

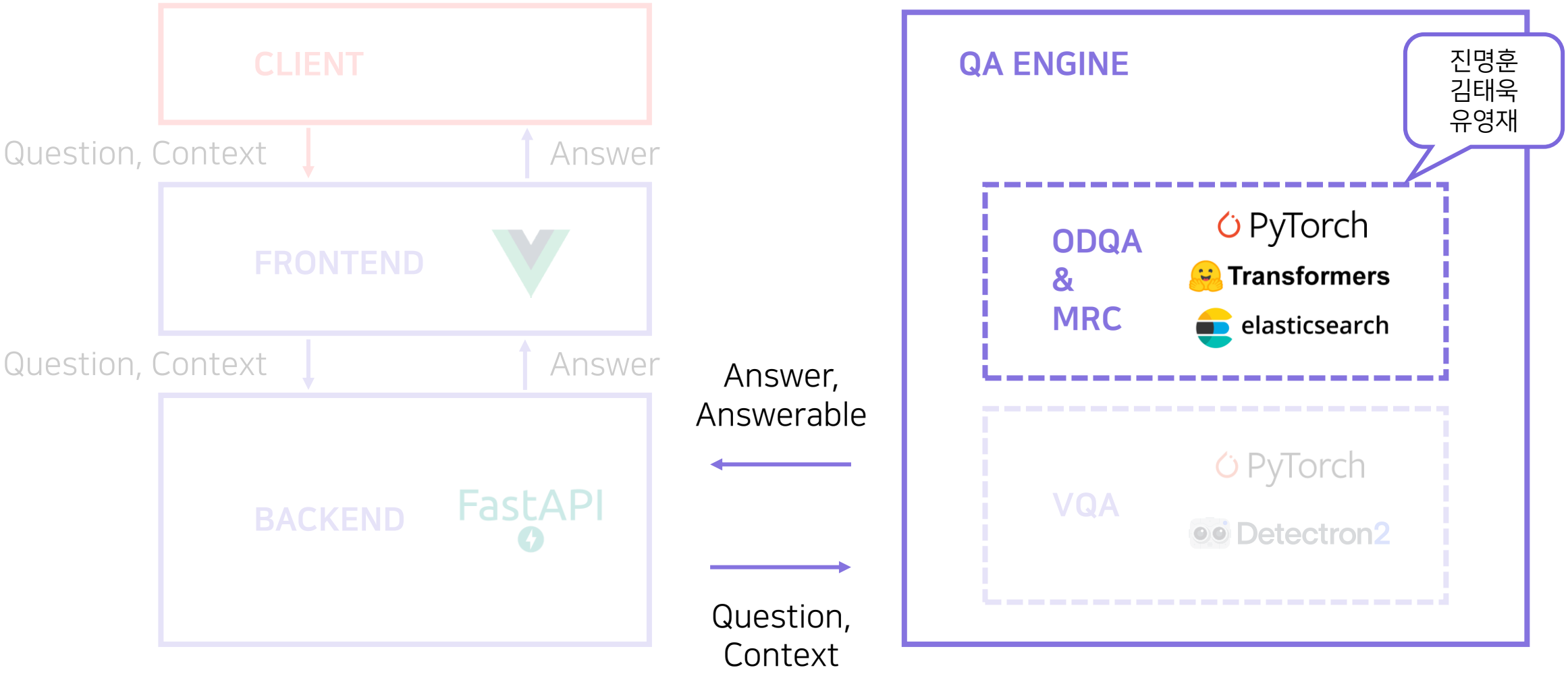
- 읽거나 보고, 아는 것만 대답하는 기영이봇
  - 읽거나 아는 것만 대답 → MRC Part
  - 보고 아는 것만 대답 → VQA Part
  - 기영이봇 → QA 테스트를 위한 UI

Project Structure



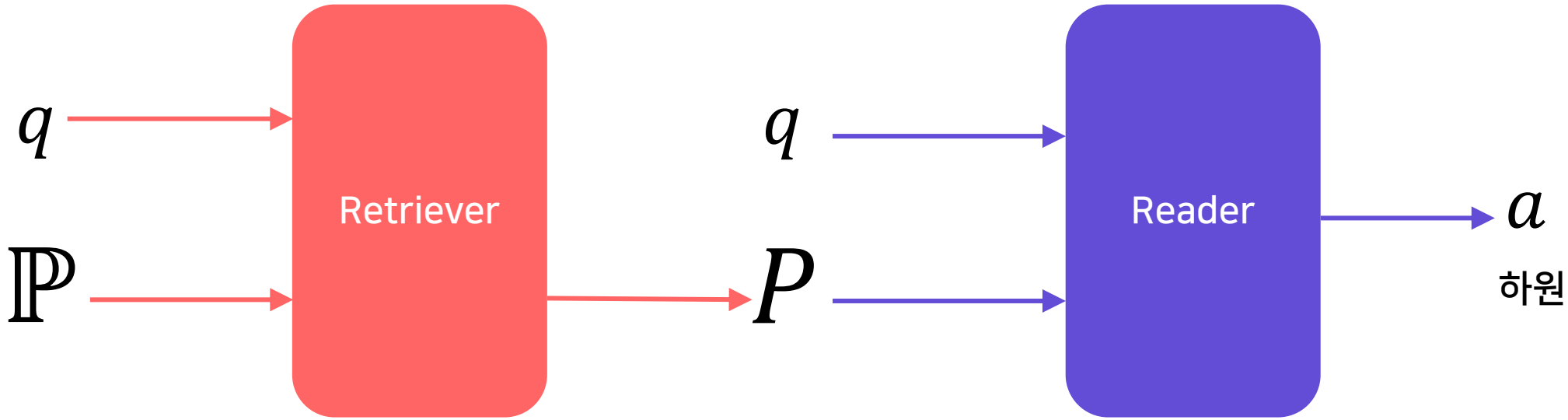
Machine Reading Comprehension

ODQA & MRC


CLIENT
Question, Context
Answer
FRONTEND
Question, Context
Answer
BACKEND
FastAPI
Answer,
Answerable
Question,
Context
QA ENGINE
진명훈
김태욱
유영재
ODQA
&
MRC
PyTorch
Transformers
elasticsearch
VQA
PyTorch
Detectron2

ODQA란? 질문에 대한 답을 도출하는 Task!

대통령을 포함한 미국의 행정부 견제권을 갖는 국가 기관은?



$f(q, \mathbb{P}) = P \;\; where \; P \in \mathbb{P}$

$g(q, P) = a$

Passage Retrieval

Machine Reading Comprehension

지난 대회 때 Solution

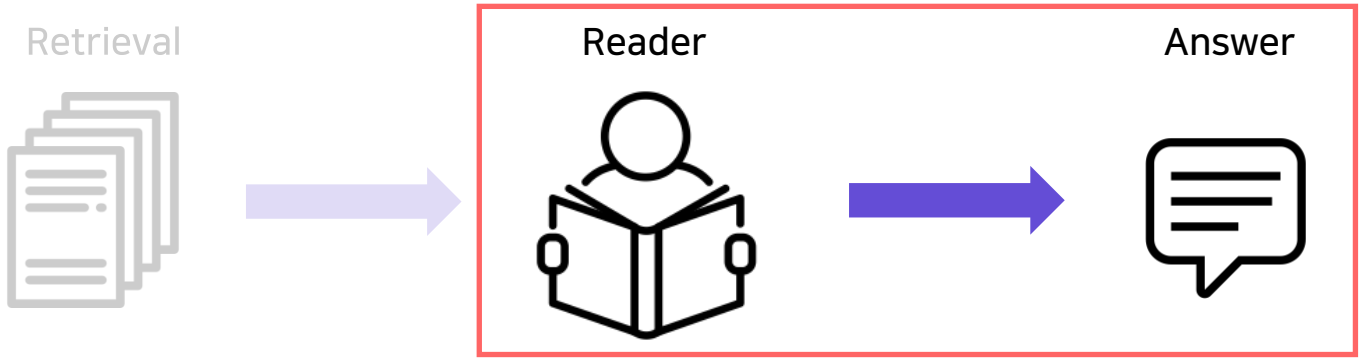
Retrieval → Reader → Answer

문서 개수	데이터셋	모델 아키텍쳐	추가 기법	후처리 기법
TOP 10 TOP 35	Wiki Context Aug Train Context Aug Train Question Aug	PORORO MRC KLUE RoBERTa Large XLM RoBERTa Large + QA Conv Head	Query ensemble Curriculum Learning K-fold ensemble	형태소 분석기 앙상블 룰베이스 후처리

최종 14개의 모델을 Hard-Voting하여 결과를 제출! 1st award

Problem ① : Real World

MRC Part를 개선시키자! (Machine Reading Comprehension)



문서 개수	데이터셋	모델 아키텍쳐	추가 기법	후처리 기법
TOP 10 TOP 35	Wiki Context Aug Train Context Aug Train Question Aug	PORORO MRC KLUE RoBERTa Large XLM RoBERTa Large + QA Conv Head	Query ensemble Curriculum Learning K-fold ensemble	형태소 분석기 앙상블 룰베이스 후처리

최종 14개의 모델을 Hard-Voting하여 결과를 제출! 1st award

현업에서 이렇게 큰 모델을 활용할 수 있을까? 1.2GB * 14

Problem ② : 모르는 질문에 답을 내는 것이 옳을까요?

- 대회 때 개발한 모델은 모든 질문에 답을 내립니다
- 子曰 “知之爲知之 不知爲不知 是知也”
  - 논어 위정편
  - 아는 것을 안다고 하고, 모르는 것을 모른다고 하는 것이 바로 아는 것!
- 답할 수 있는지 있는지를 아는 것이 지혜라 했습니다
- No answerability를 가지는 모델 개발

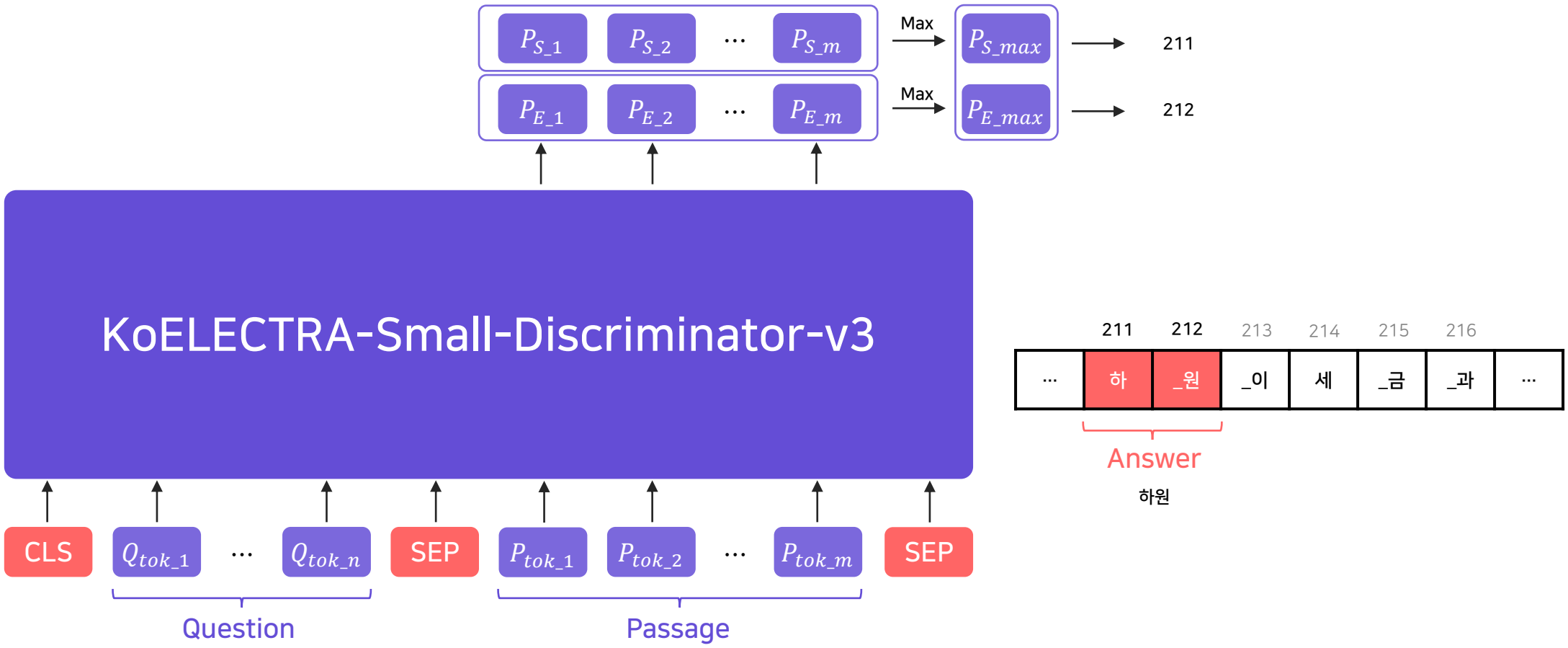
No answer 학습을 위한 데이터셋

- AI Hub Data
  - Normal: 243,425건 데이터 (no answer 0%)
  - No answer: 100,244건 데이터 (no answer 100%)
  - Book MRC: 950,000건 데이터 (no answer 30%)
  - Common-sense: 100,268건 데이터 (no answer 0%)
- KLUE
  - MRC: 23,395건 데이터 (no answer 0%)
- KorQuAD v1.0
  - MRC: 66,181건 데이터 (no answer 0%)

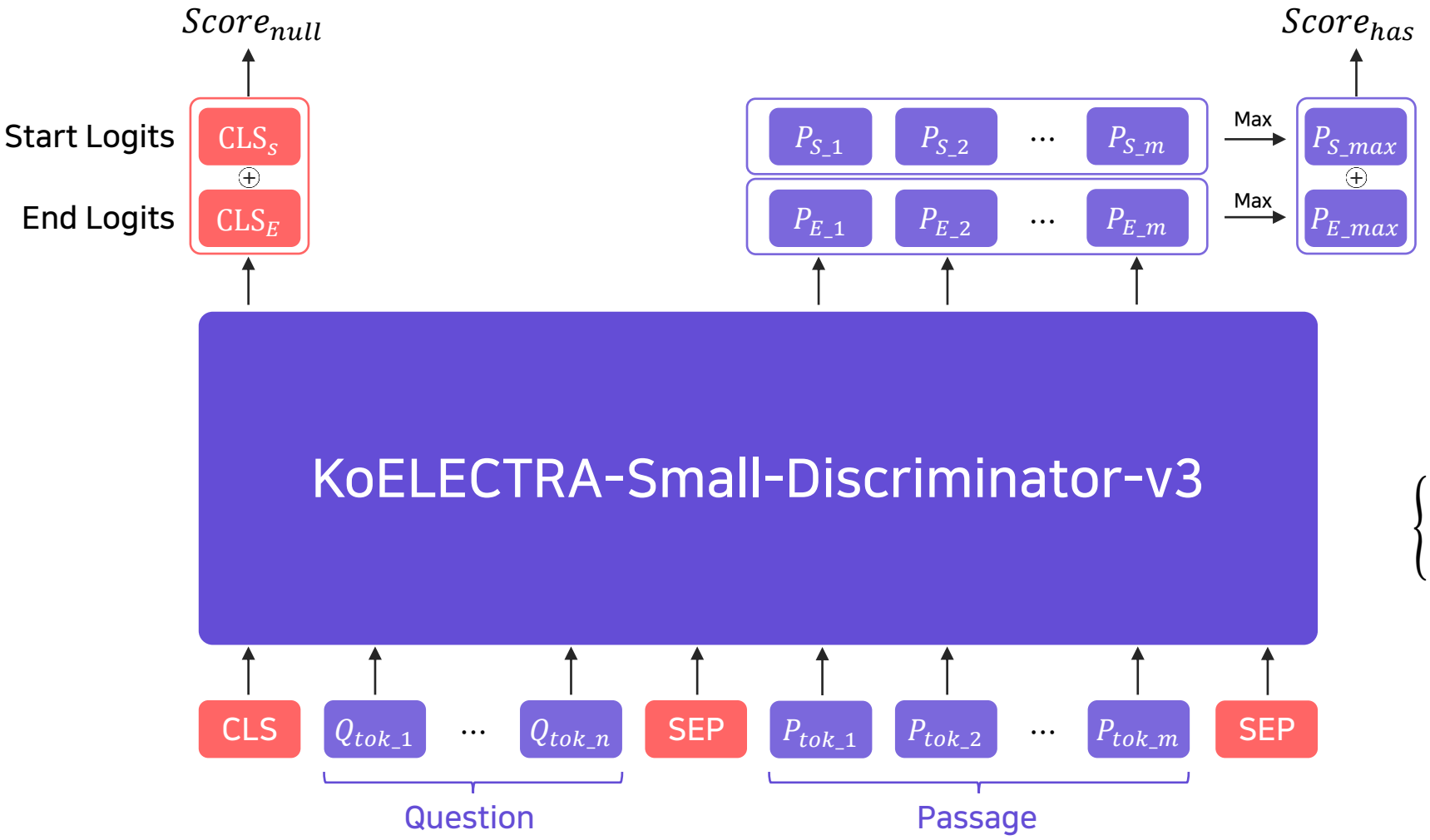
대회 개발 모델과의 비교

	Boostcamp ODQA Competition	Final Project
		
No answerable ratio	0%	26.4% (Total), 50% (Validation)
Model	14 ensemble hard voting (Roberta-large etc.)	single model (Retro: 2 model) (Ko-electra-small-discriminator-v3)
Minimal Model Size	1.2GB	50MB

Conventional Extractive QA



Conventional Extractive QA with No answer



$$Score_{null} - Score_{has} = Score_{diff}$$

$$\begin{cases} Score_{diff} > \theta\,, & Not\ answerable \\ Score_{diff} \leq \theta\,, & Answerable \end{cases}$$

Retrospective Reader

- Machine Reading Comprehension: 기계 독해
- 인간이 글을 읽는 방법?
  - 통독 (通讀)
    - 글을 중간에 건너 뛰지 않고 처음부터 끝까지 훑어 읽는 것
  - 정독 (精讀, 숙독(熟讀), 열독(熱讀))
    - 글의 뜻을 새기면서 자세히 읽는 것
- 기계도 인간처럼 읽게 모델링할 수 있을까?

https://pyopyo1027.tistory.com/92
https://brunch.co.kr/@holidaymemories/116

Retrospective Reader

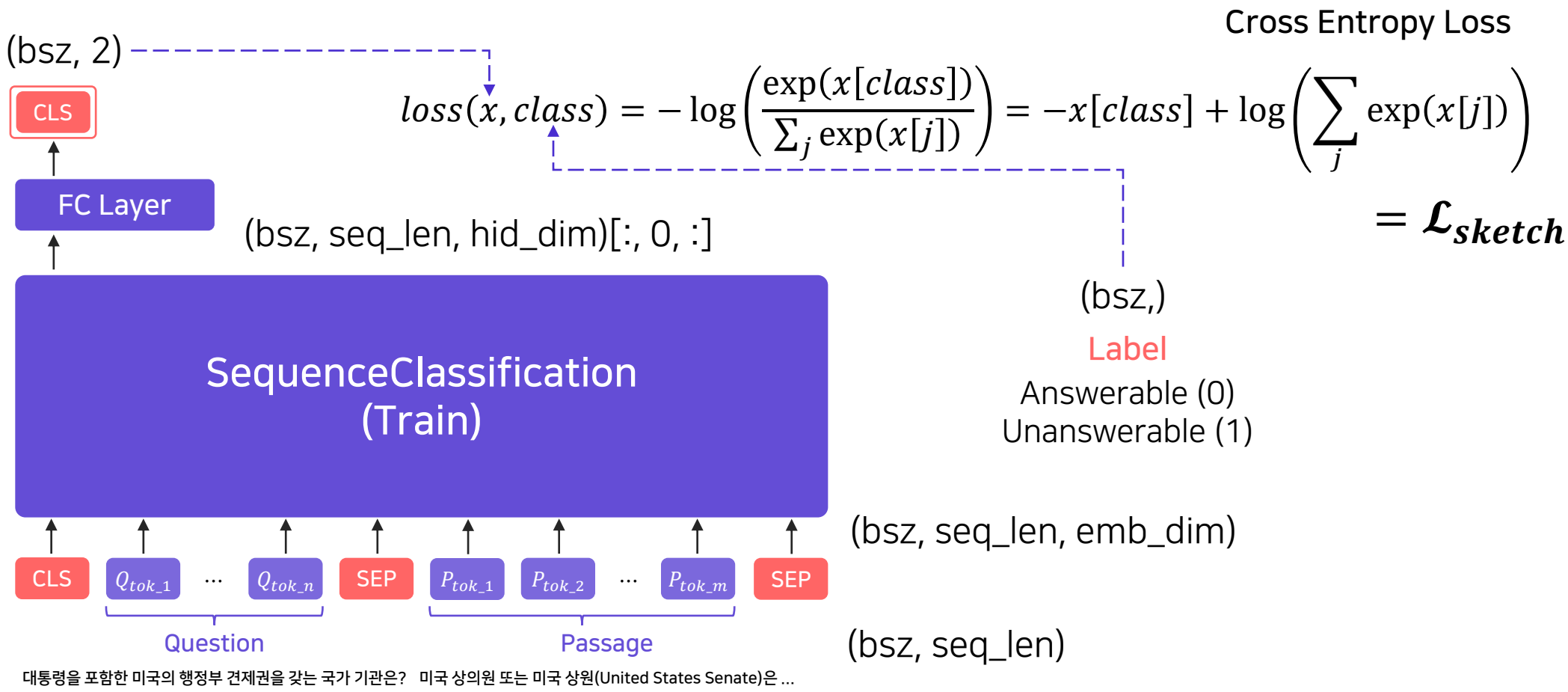
- Machine Reading Comprehension: 기계 독해
- 인간이 책을 읽는 방법?
  - 통독 (通讀) → Sketch Reader (통독하여 No answer 판별)
    - 책을 중간에 건너 뛰지 않고 처음부터 끝까지 훑어 읽는 것
  - 정독 (精讀, 숙독(熟讀), 열독(熱讀)) → Intensive Reader (정독하여 정답 도출)
    With NoAns
    - 글의 뜻을 새기면서 자세히 읽는 것
- 기계도 인간처럼 읽게 모델링할 수 있을까?

→ Retrospective Reader (회고하는 기계 독해 모델)
with Rear Verification (예측이 끝나고 종합)

https://arxiv.org/abs/2001.09694

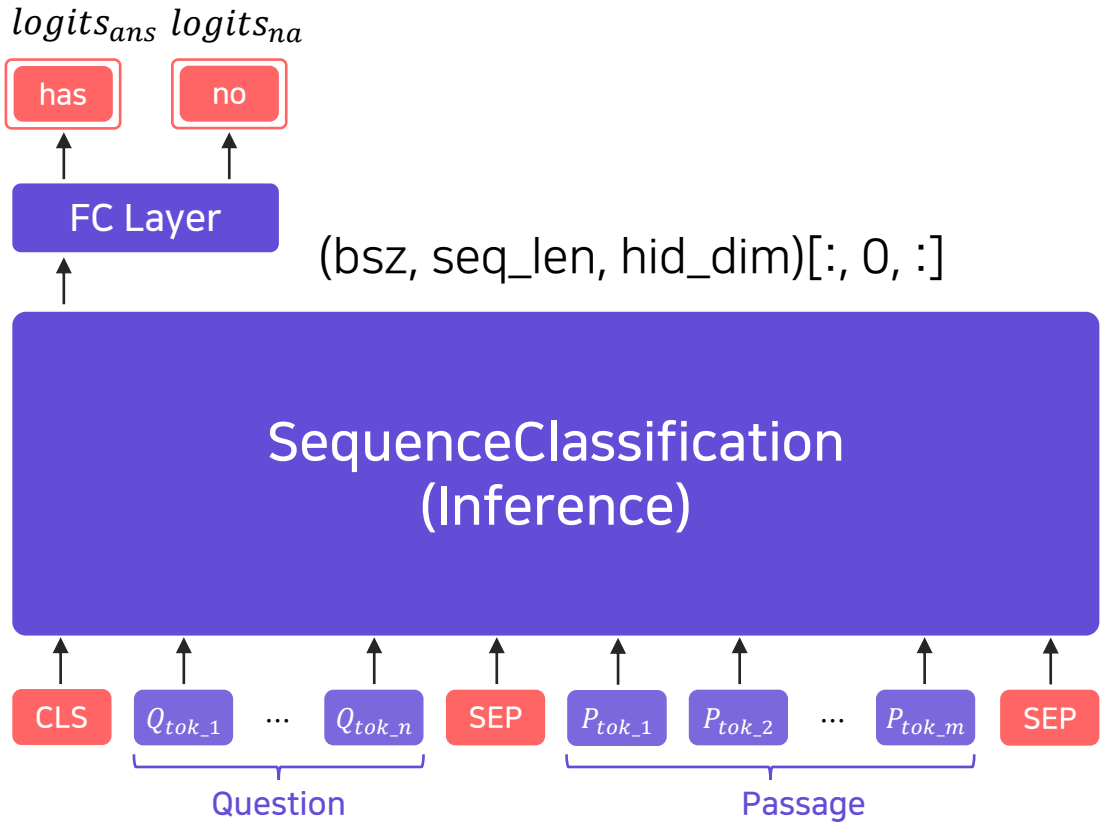
통독하는 모델 (Sketch Reader)

External Front-Verification



통독하는 모델 (Sketch Reader)

External Front-Verification



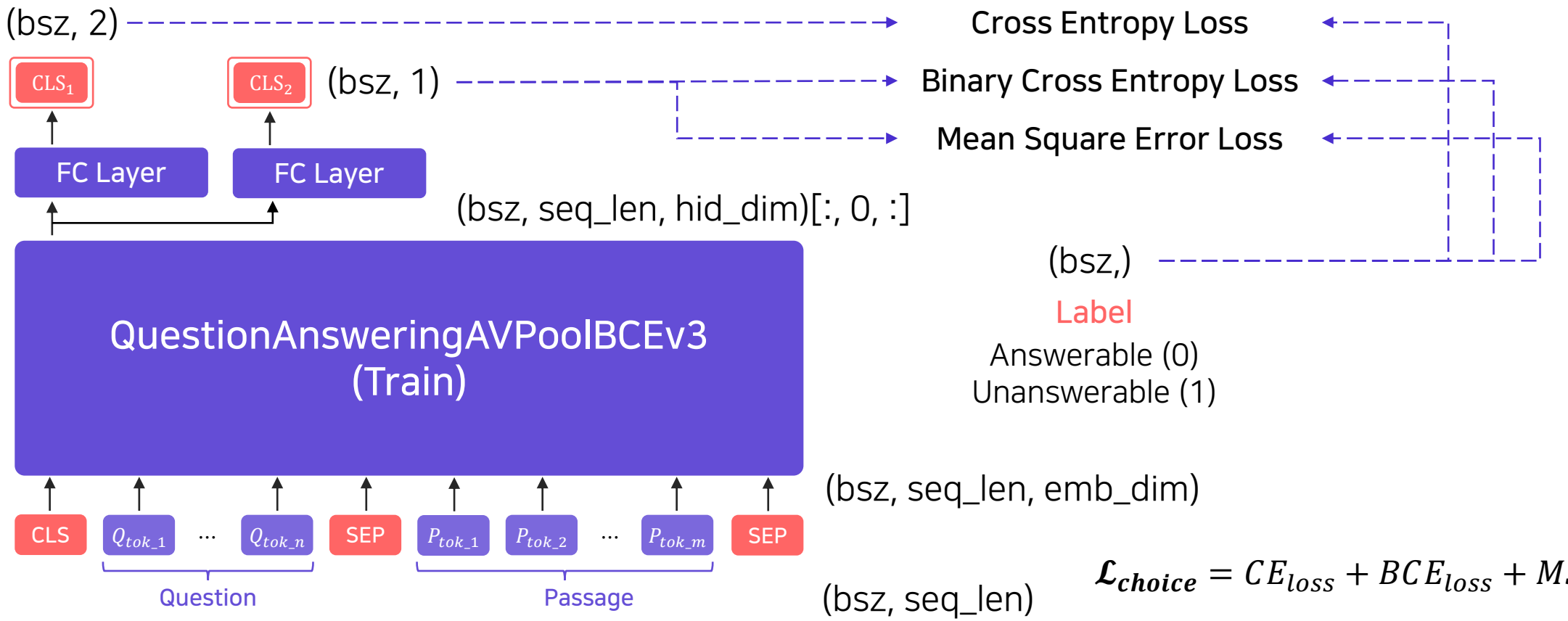
$$\mathbf{score}_{ext} = \boldsymbol{logits}_{na} - \boldsymbol{logits}_{has}$$

(bsz, seq_len, emb_dim)

(bsz, seq_len)

정독하는 모델 (Intensive Reader)

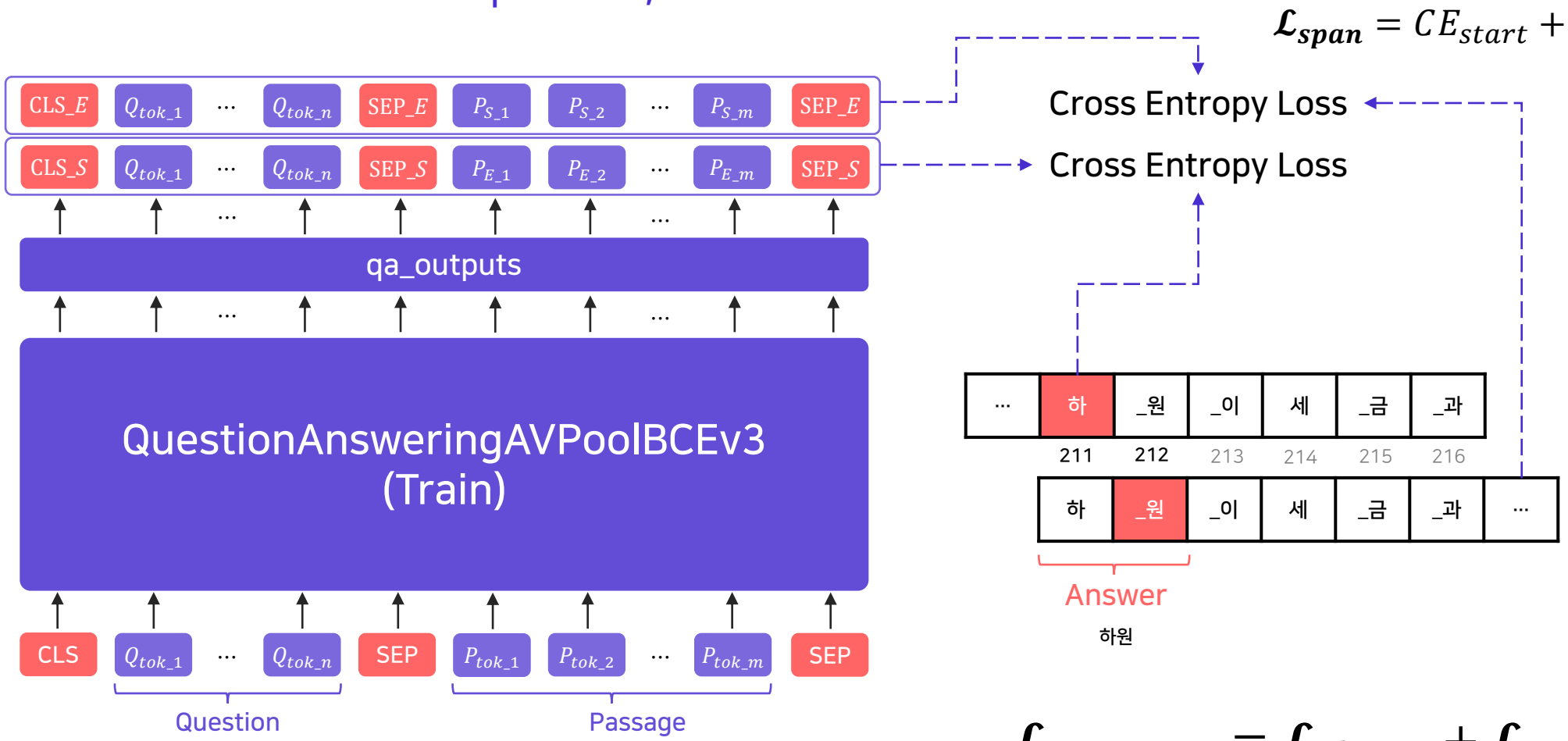
Internal Front-Verification: Choice Loss



$$\mathcal{L}_{choice} = CE_{loss} + BCE_{loss} + MSE_{loss}$$

정독하는 모델 (Intensive Reader)

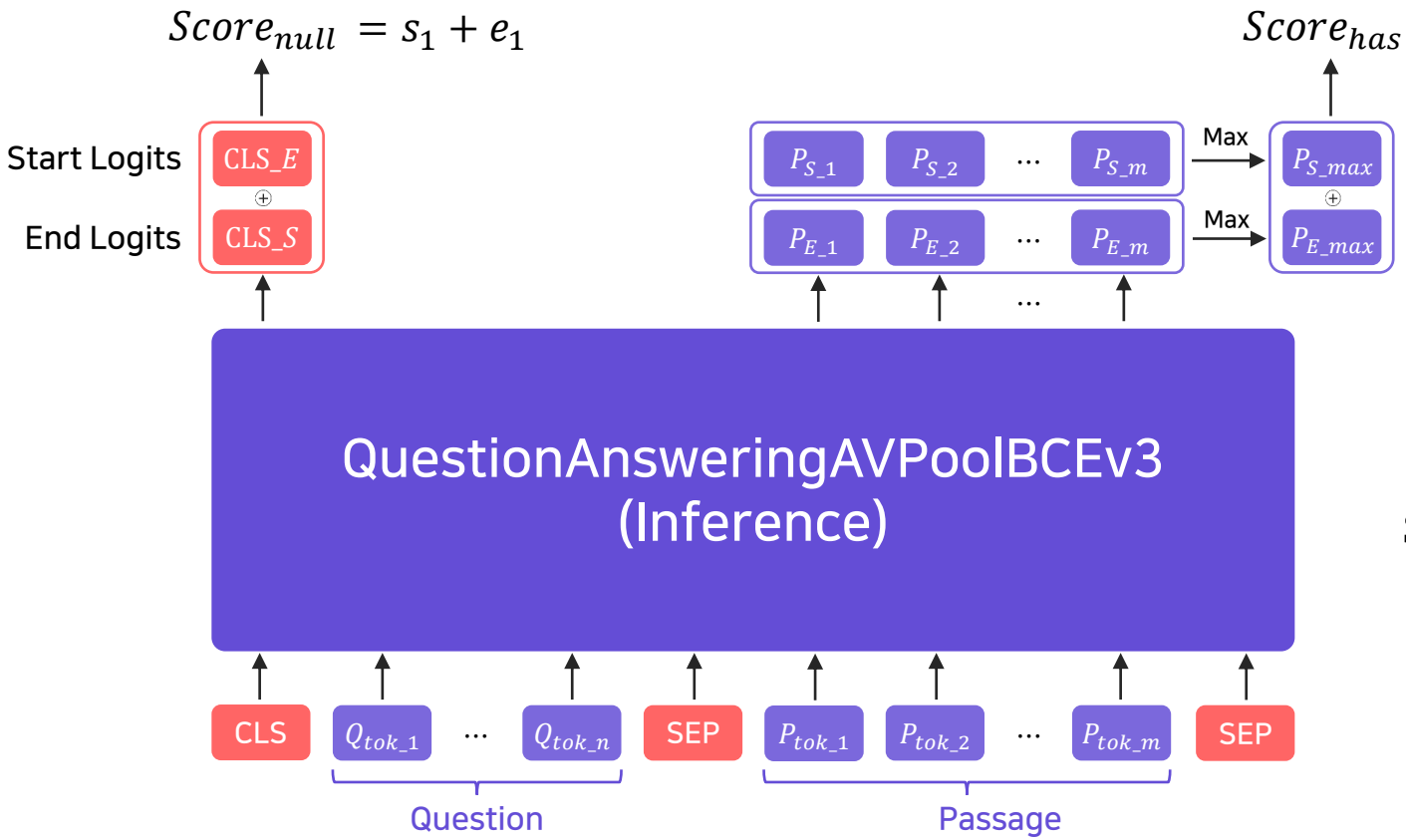
Internal Front-Verification: Span Loss, Total Loss



$$\mathcal{L}_{intensive} = \mathcal{L}_{choice} + \mathcal{L}_{span}$$

정독하는 모델 (Intensive Reader)

Internal Front-Verification



$$\mathbf{score}_{diff} = score_{null} - score_{has}$$

다 읽고 종합 (Rear Verification)

The combination of predicted probabilities of E-FV and I-FV,
which is an aggregated verification for the final answer.

$$\text{score}_{ext} = logits_{na} - logits_{has}$$

$$\text{score}_{diff} = score_{null} - score_{has}$$

$$v = \beta_1 score_{diff} + \beta_2 score_{ext}$$

$$where\ \beta_1\ and\ \beta_2\ are\ weights$$

$$\begin{cases} v > \delta\,, & Answerable \\ v \leq \delta\,, & Not\ answerable \end{cases}$$

최종 구조 (Retrospective Reader)

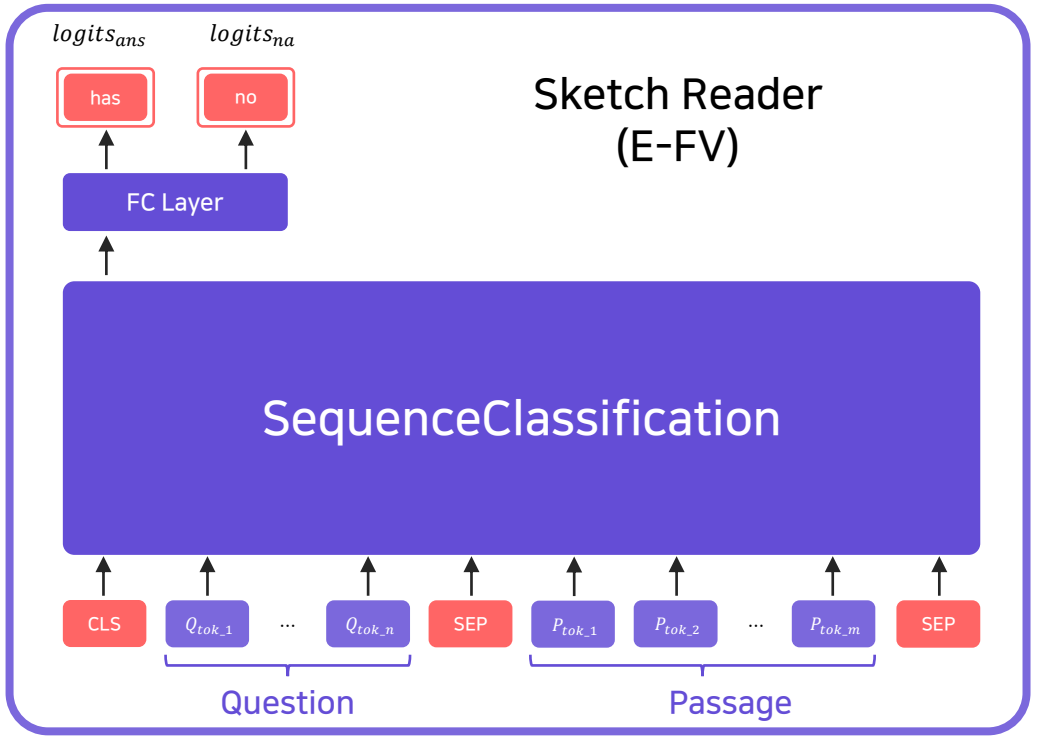
Rear Verification (BV)

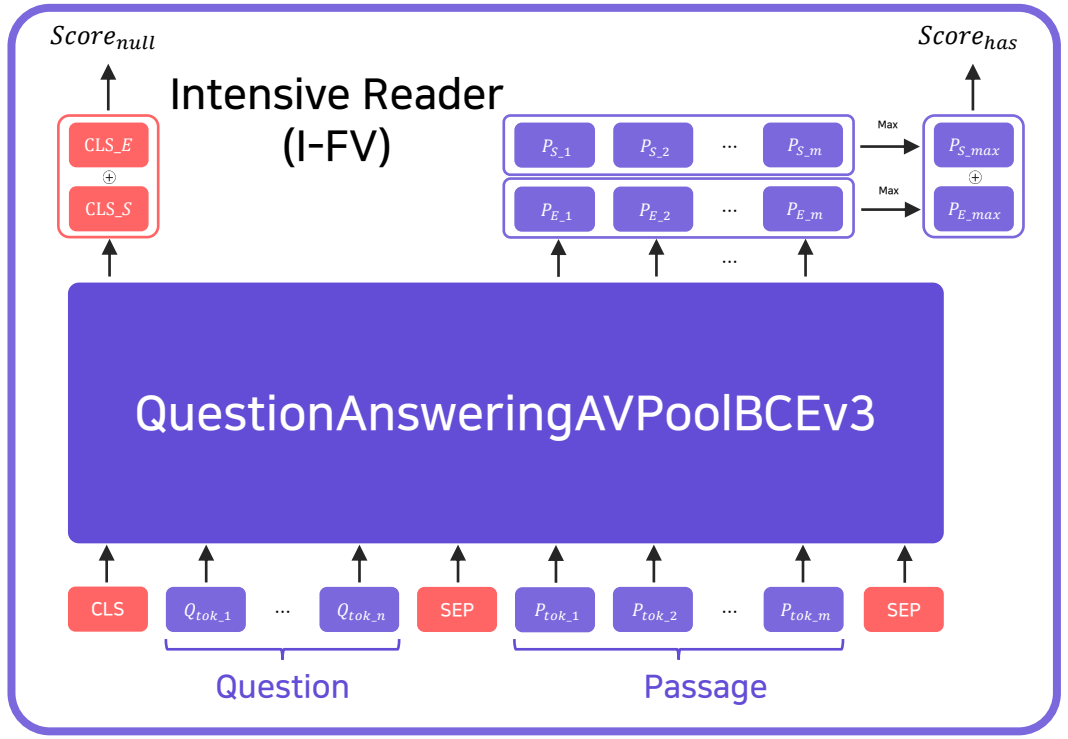
$$v = \beta_1 score_{diff} + \beta_2 score_{ext}$$

$$\begin{cases} v > \delta, & Answerable \\ v \leq \delta, & Not\ answerable \end{cases}$$

$$\text{score}_{ext} = logits_{na} - logits_{has}$$

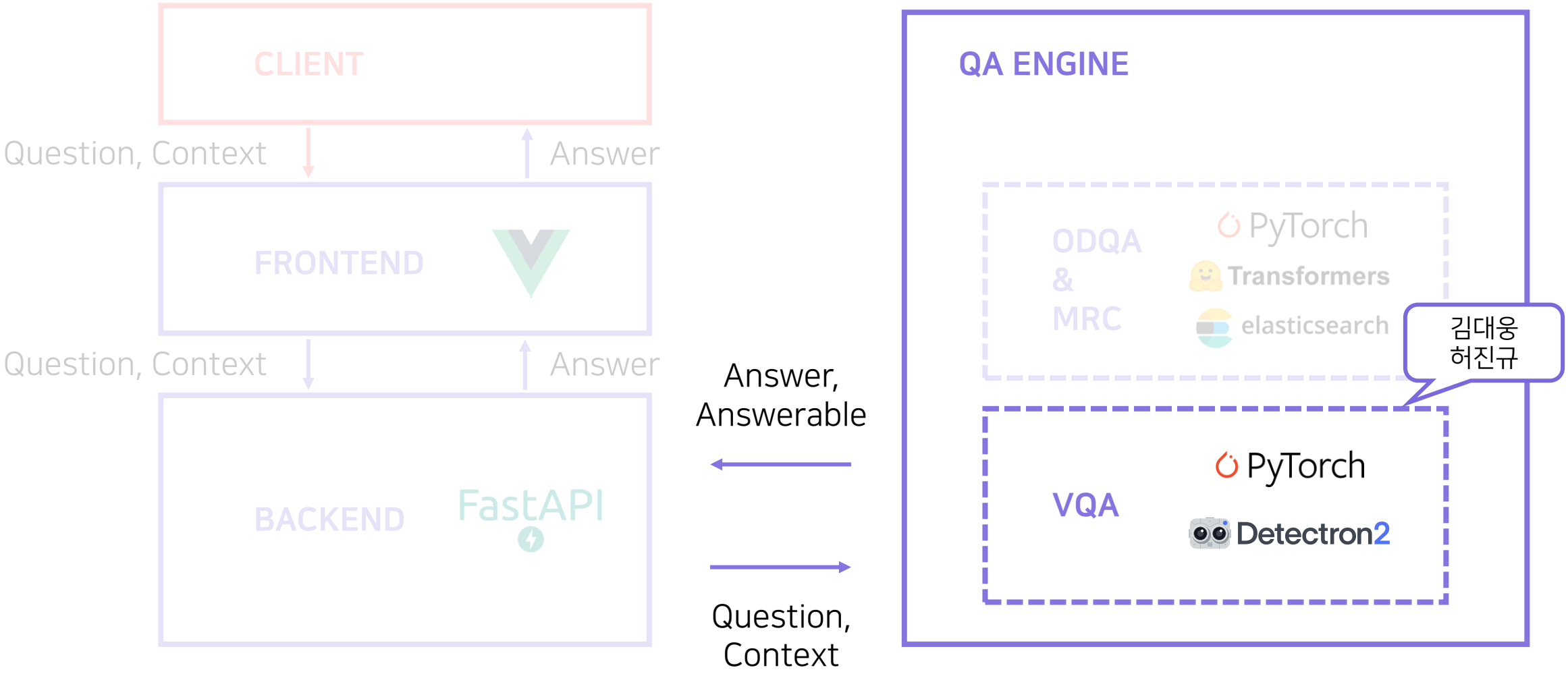
$$\text{score}_{diff} = score_{null} - score_{has}$$





Visual Question Answering

VQA (Visual Question Answering)



VQA Development History

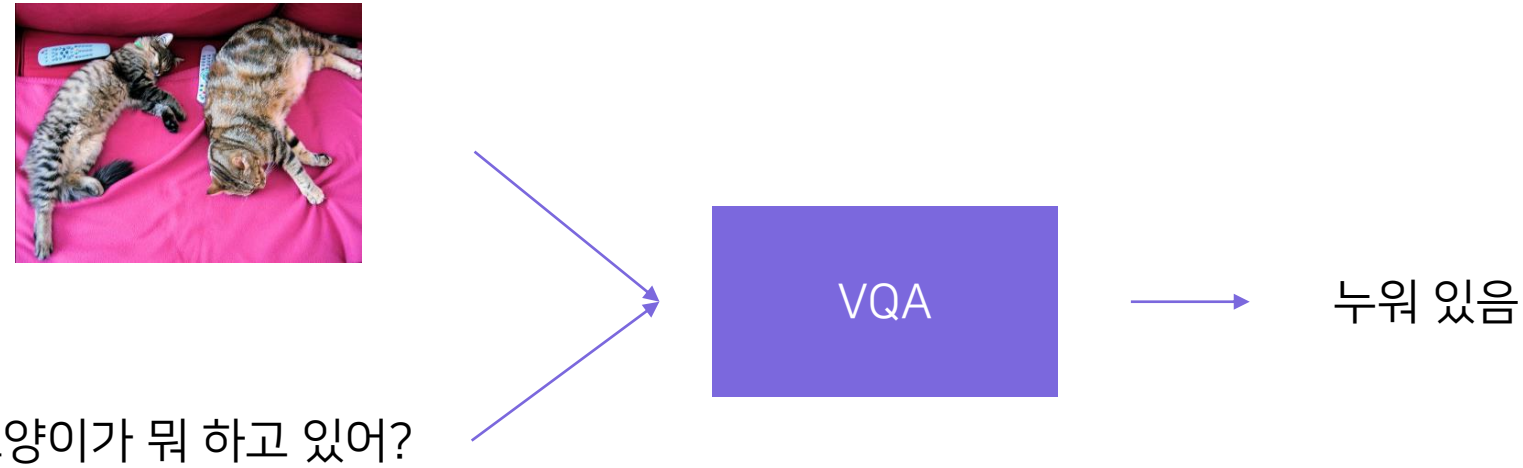
Pilot Study

Bilinear Attention Networks (BAN)

Rule-base + BAN

텍스트 뿐만 아니라 이미지도 질문의 대상

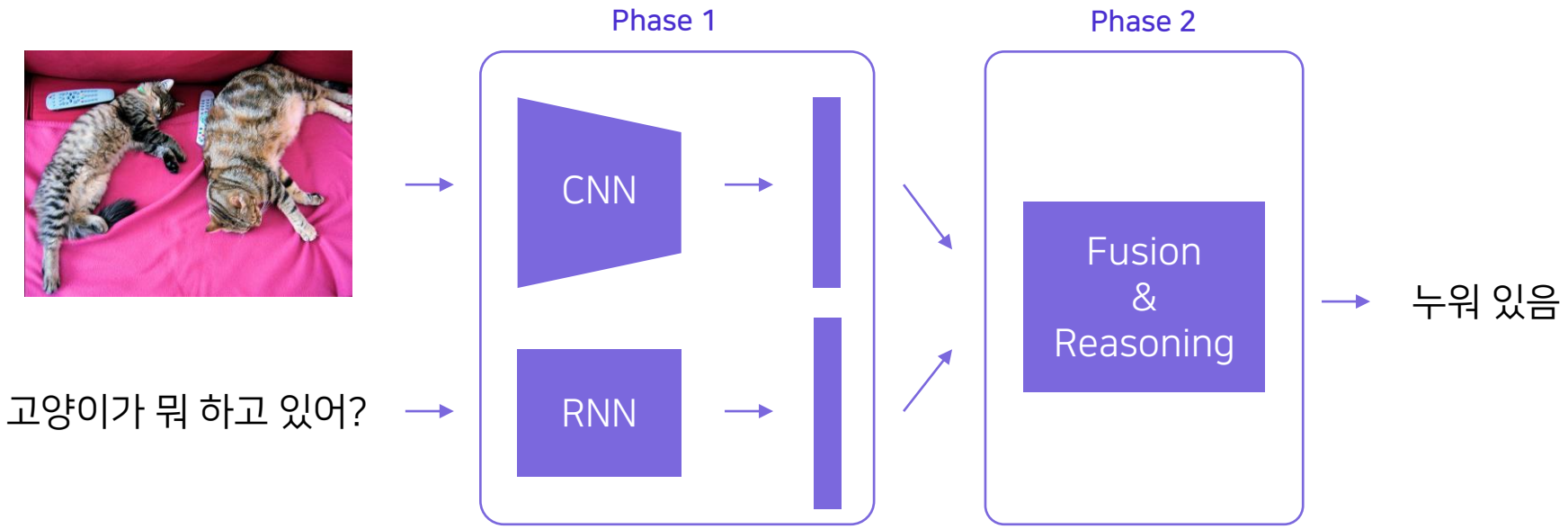
- Visual Question Answering (VQA)
  - 장면을 이해하고 질문에 답변하는 Question Answering system



Antol, Stanislaw, et al. "Vqa: Visual question answering." Proceedings of the IEEE international conference on computer vision. 2015.

VQA 톺아보기

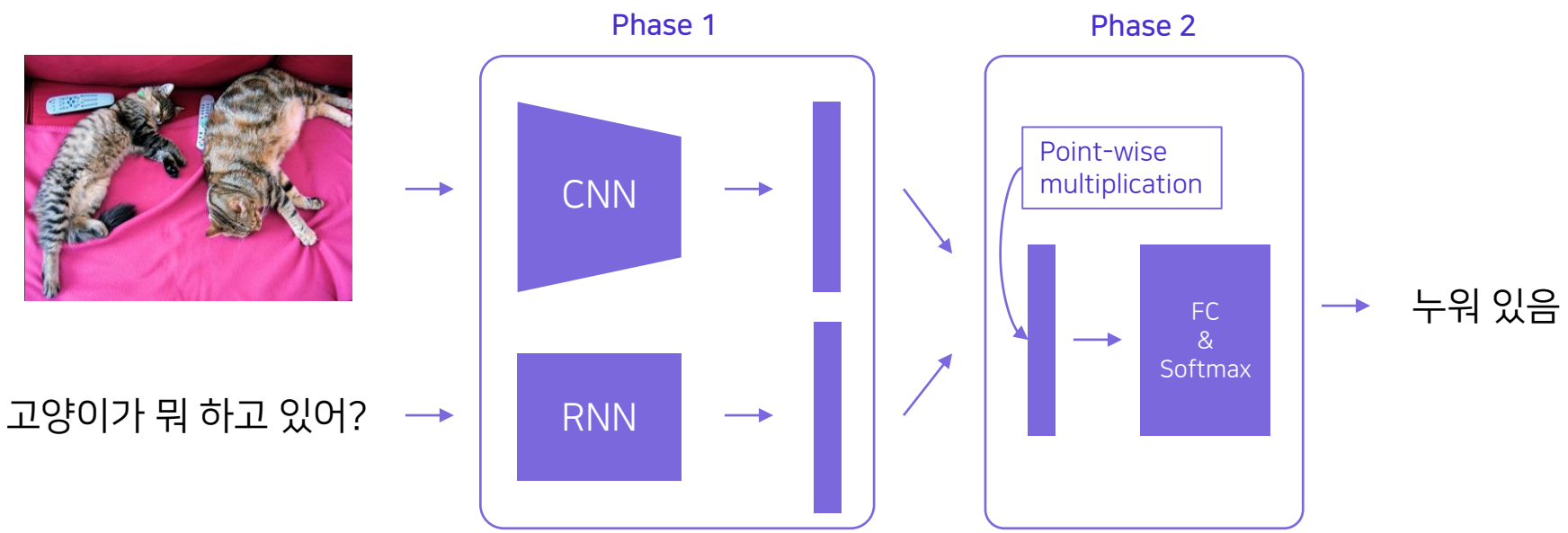
- VQA는 크게 2가지 단계로 구성 (Barra et al., 2021)
  - Phase 1: 이미지와 텍스트로부터 feature를 추출
  - Phase 2: 두 feature의 정보를 종합하여 정답을 출력



Barra, Silvio, et al. "Visual Question Answering: which investigated applications?." Pattern Recognition Letters 153. 2021.

VQA 톺아보기

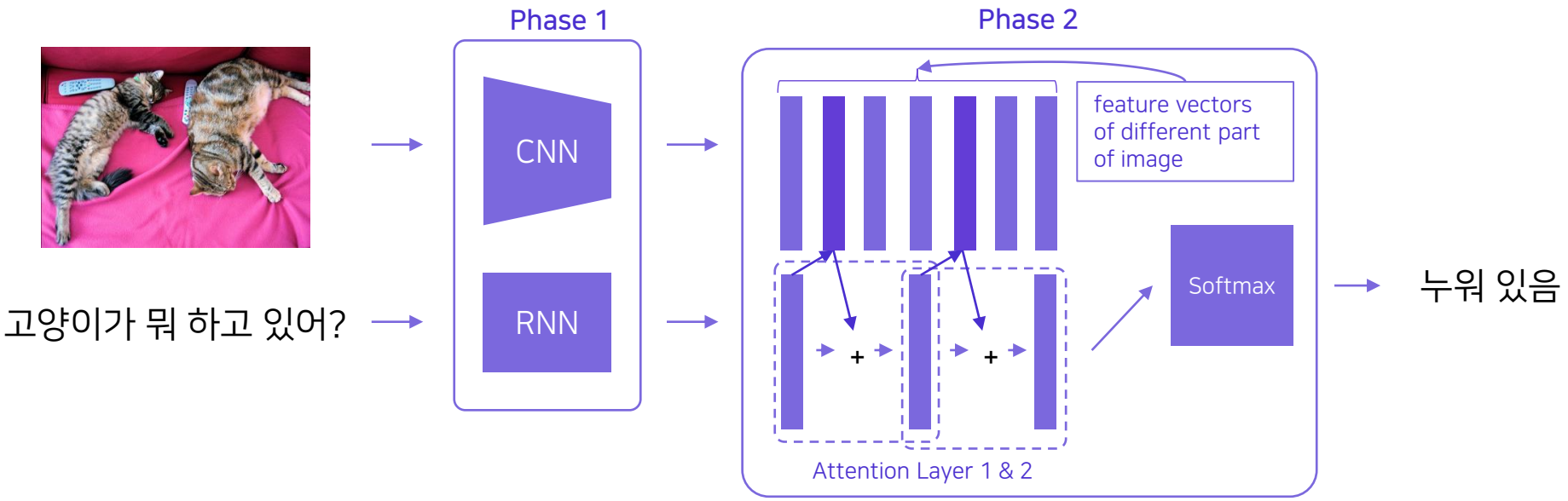
- 가장 단순한 모델 (VQA Task를 제안한 논문)
  - 이미지와 텍스트의 feature를 point-wise multiplication



Antol, Stanislaw, et al. "Vqa: Visual question answering." Proceedings of the IEEE international conference on computer vision. 2015.

Pilot Study: Stacked Attention Networks (SAN)

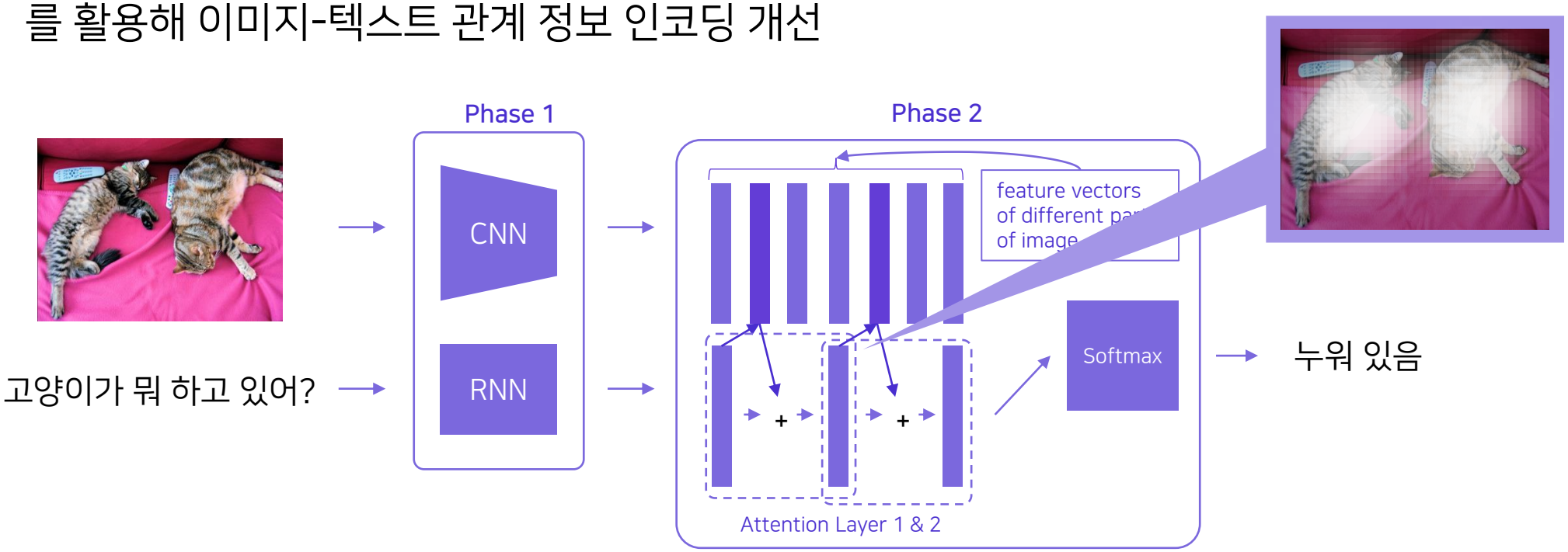
- VQA 적용 가능성 파악을 위한 Pilot Study
  - 이미지의 각 픽셀을 나타내는 feature vector와 텍스트 feature vector의 attention score를 활용해 이미지-텍스트 관계 정보 인코딩 개선



Yang, Zichao, et al. "Stacked attention networks for image question answering."
Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

Pilot Study: Stacked Attention Networks (SAN)

- VQA 적용 가능성 파악을 위한 Pilot Study
  - 이미지의 각 픽셀을 나타내는 feature vector와 텍스트 feature vector의 attention score를 활용해 이미지-텍스트 관계 정보 인코딩 개선



Yang, Zichao, et al. "Stacked attention networks for image question answering."
Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

Pilot Study: Stacked Attention Networks (SAN)

- Pilot Study 결과 파악된 것
  - 정량평가 결과보다 정성평가가 중요
  - Yes맨 문제 :
    모델이 특정 단어의 유무에 따라 특정 클래스에 편향되는 문제 발생

Question : 사람이 있니
Predict Prob :

Question : 있니
Predict Prob :

Pilot Study: Stacked Attention Networks (SAN)

- Yes맨 문제의 원인에 대한 가설
  - 문제 접근 방식이 `분류`이지만 답변은 `서술형`

    → 서술형 답변이 모두 클래스 (No answer 또한 클래스 중 하나)

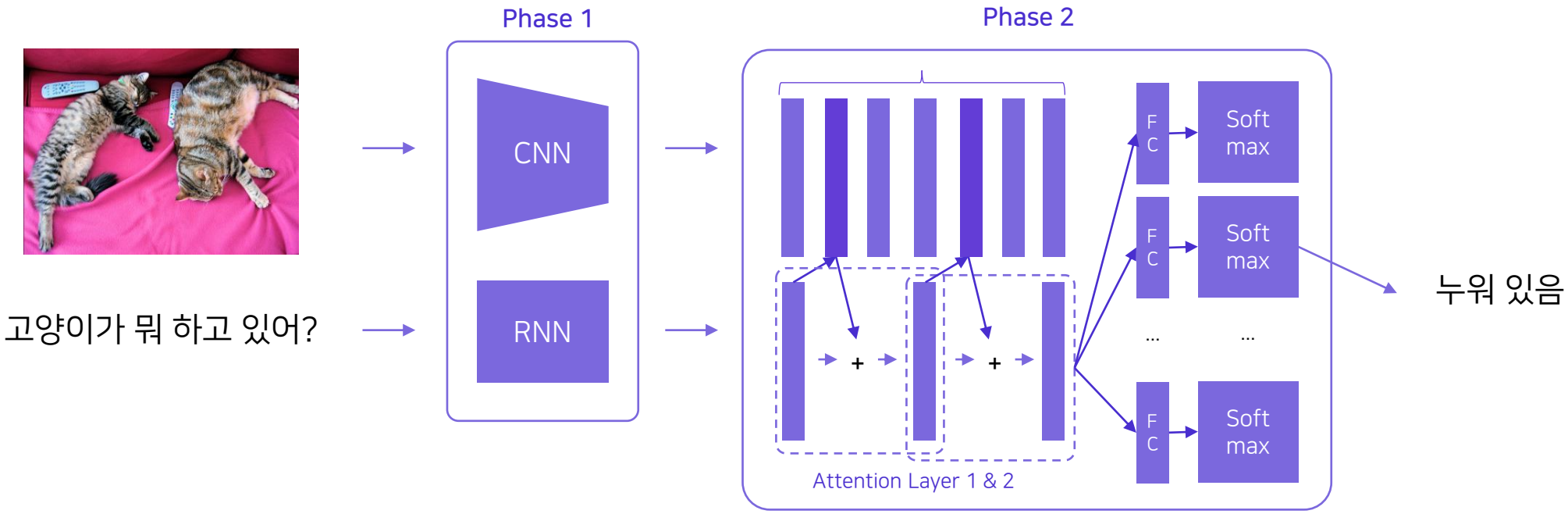
    → 하나의 헤드 레이어가 담당하는 클래스의 개수 증가 (1,468개 클래스)

- 해결 방안
  - 멀티 헤드 모델의 사용

    → 헤드 레이어를 질문 유형별로 만들어 각 헤드 별 부담하는 클래스의 개수 감소

Pilot Study: Multi-Head SAN

- Multi-Head 모델을 이용한 성능 개선 (Ours, Phase 2 개선)



Pilot Study: Multi-Head SAN

- Multi-Head 모델을 이용한 성능 개선 (Ours, Phase 2 개선)
  - 질문을 예/아니오, 개수, 색상,모양 및 재질, 위치, 기타의 6개 유형으로 세분화
  - 그러나 Yes맨 문제를 해결하지 못함

Question : 사람이 있니
Predict Prob :

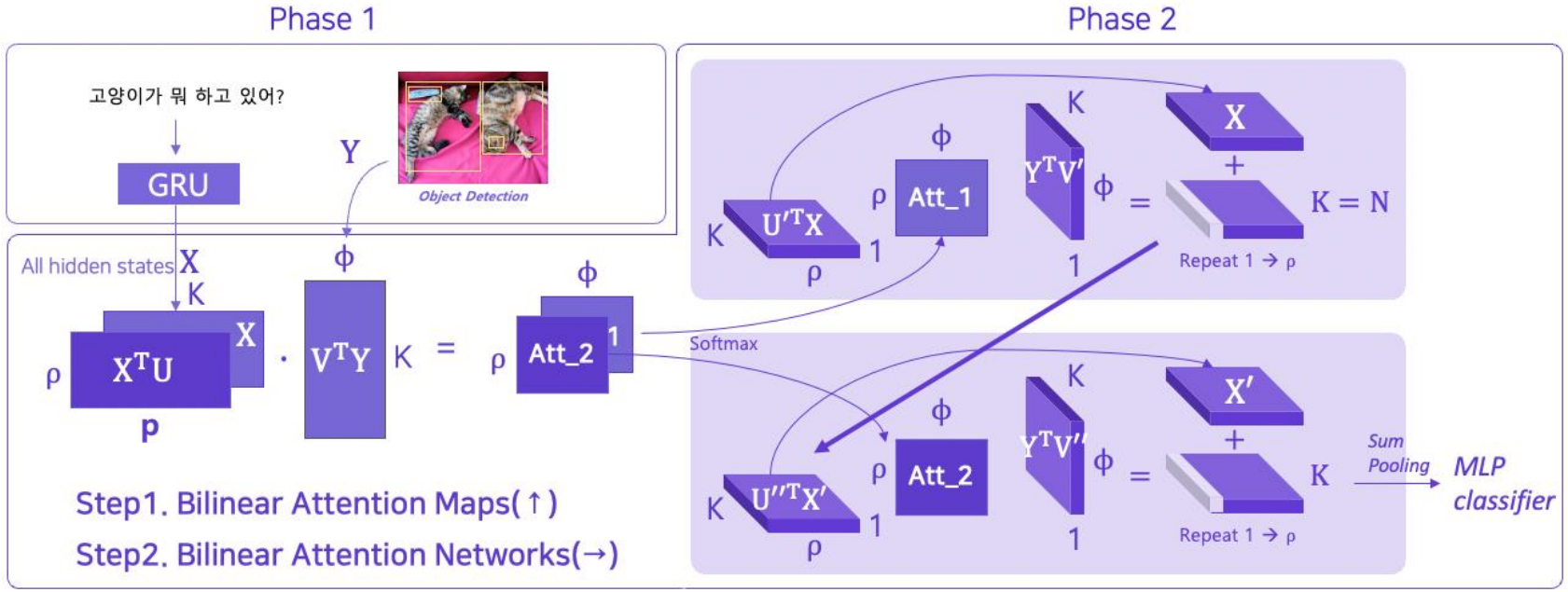
Question : 있니
Predict Prob :

Pilot Study: Multi-Head SAN

- 실패 원인 분석 & 새로운 가설 수립
  - 멀티 헤드로 정량적 성능향상은 있었지만 근본적인 해결책은 아님
  - 단어 벡터들이 하나의 문장 벡터로 인코딩되는 것이 문제
- 해결 방안
  - 텍스트와 이미지 매칭 단위를 변경해 모델에게 보다 분명한 가이드 제공
    → 기존 : 텍스트 전체 & 이미지 픽셀
    → 개선 : 텍스트 내 단어 & 이미지 내 Object

Bilinear Attention Networks (BAN)

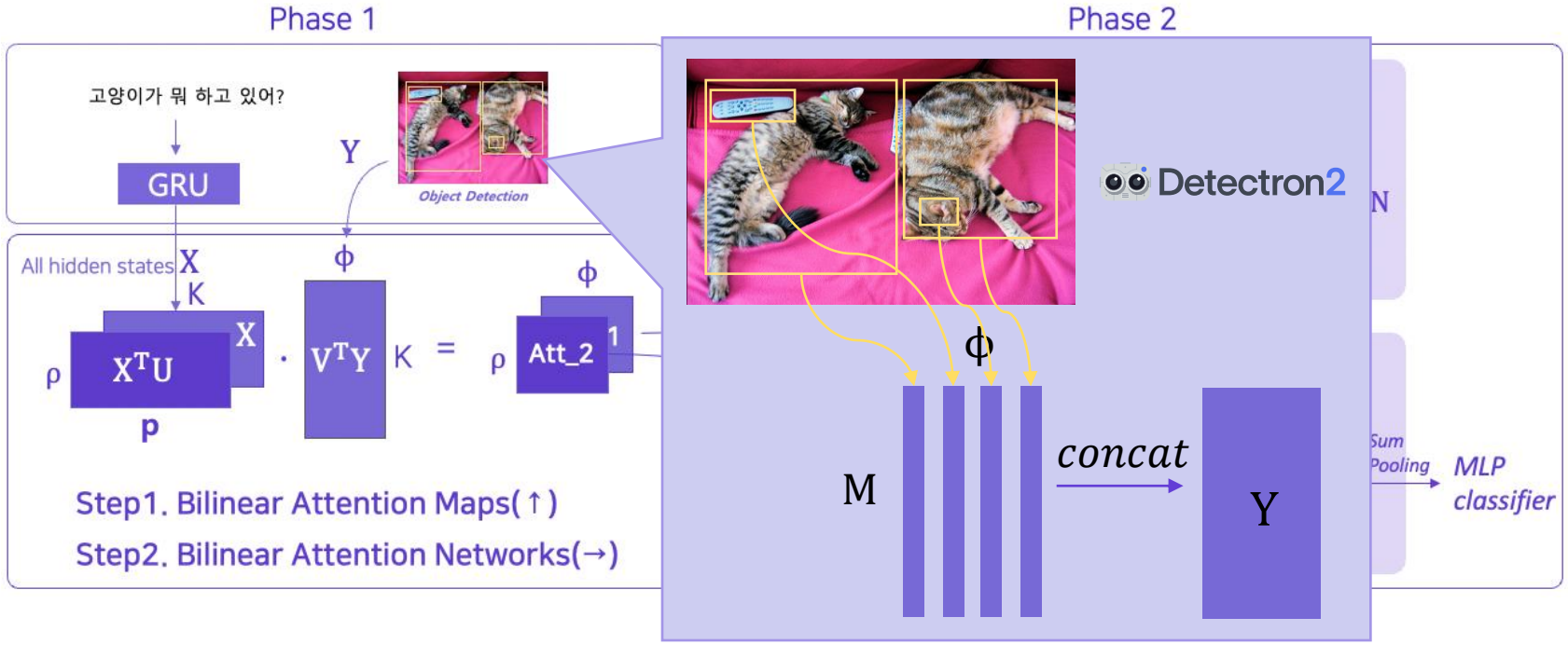
- 텍스트와 관련된 특정 Object에 집중(Phase 1 & 2 개선)
  - SAN이 문장 & 픽셀 단위로 attention을 수행하는 반면
    BAN은 토큰 & Object 단위로 attention 수행



Kim, Jin-Hwa, Jaehyun Jun, and Byoung-Tak Zhang. "Bilinear attention networks." Advances in Neural Information Processing Systems 31. 2018.

Bilinear Attention Networks (BAN)

- 텍스트와 관련된 특정 Object에 집중(Phase 1 & 2 개선)
  - SAN이 문장 & 픽셀 단위로 attention을 수행하는 반면
    BAN은 토큰 & Object 단위로 attention 수행



Kim, Jin-Hwa, Jaehyun Jun, and Byoung-Tak Zhang. "Bilinear attention networks." Advances in Neural Information Processing Systems 31. 2018.

데이터셋

- KVQA(Korean Visual Question Answering)
  - SK T-Brain에서 구축한 시각장애인을 위한 한국어 VQA 벤치마크
  - 총 100,445개 이미지와 질문, 문항별 10명의 사람이 작성한 답변으로 구성
  - 질문 유형 : “예/아니오”, “숫자”, “기타”, “답변 불가능”

(a) **Q**: 지금 횡단보도를 건너도 될까? (Can I cross the crosswalk now?) **A**: 아니오 (No)

(b) **Q**: 이 방에는 몇 개의 형광등이 있나요? (How many lights in this room?) **A**: 2

(c) **Q**: 방에 있는 사람은 지금 뭘하고 있지? (What is the person doing in this room?) **A**: 피아노 (Piano)

(d) **Q**: 무슨 꽃이 피어있지? (What kind of flower is this?) **A**: Unanswerable

https://sktbrain.github.io/KVQA/

성능 평가

- Yes로만 대답하는 문제 해결
  - 특정 토큰이 클래스 예측에 지배적인 현상 해결

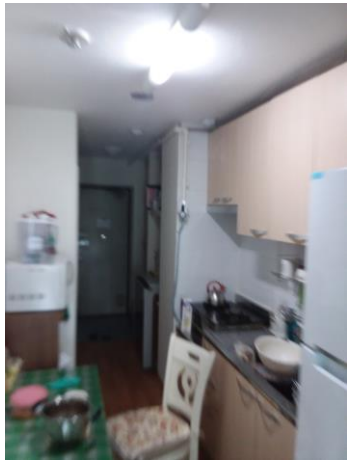
Question : 사람이 있니
Prediction : 아니요

Question : 있니
Prediction : 예

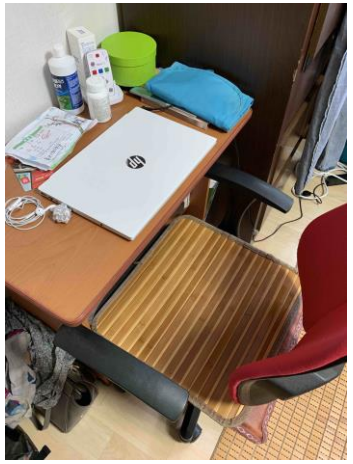
Kim, Jin-Hwa, Jaehyun Jun, and Byoung-Tak Zhang. "Bilinear attention networks." Advances in Neural Information Processing Systems 31. 2018.

성능 평가

- 단순한 문제를 틀리는 경우
  - 사물이름, 수량, 개수, 모양, 색상 등을 틀리는 경우 발생
  - 이를 보완하기 위해 Rule-base를 활용하는 2step 모델 설계



Question : 사람이 몇 명이야?
Prediction : 2

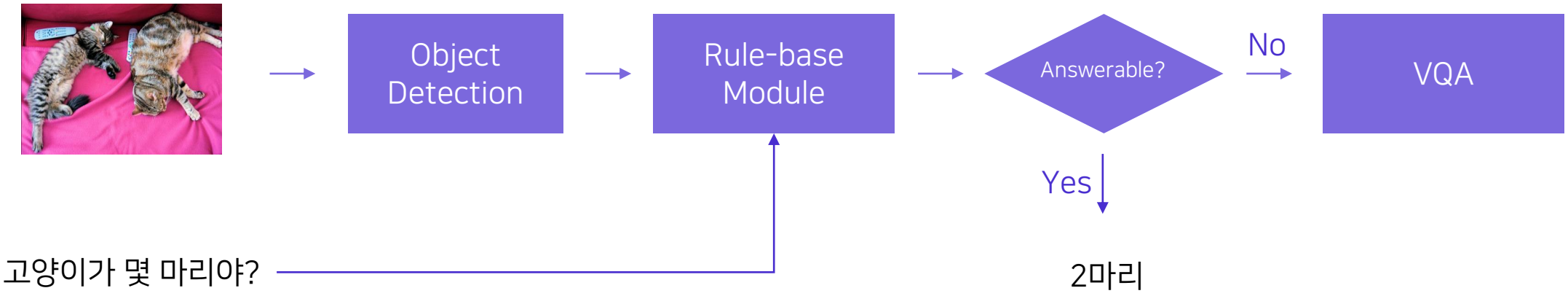


Question : 의자가 무슨 색이야?
Prediction : 회색

Kim, Jin-Hwa, Jaehyun Jun, and Byoung-Tak Zhang. "Bilinear attention networks." Advances in Neural Information Processing Systems 31. 2018.

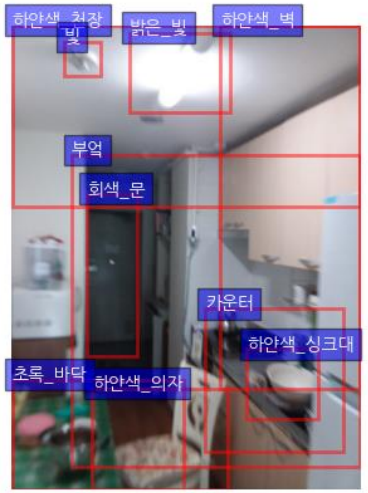
Rule-base + BAN

- 모델이 틀린 문항에서 Object Detection Model로 답변 가능한 6개의 질문 유형을 정리
  - 사물, 색, 모양, 무늬, 개수, 유무
  - Rule-base 모델에서 답변 불가능할 경우 VQA로 처리



Rule-base 적용 후 평가 결과

- Detection Model이 찾은 객체 정보를 활용해 답변하여 단순한 질문에 대한 오답을 줄임



Question : 사람이 몇 명이야?
Prediction : 0 명

Question : 의자가 무슨 색이야?
Prediction : 빨간색

한계점 및 향후 개선 방향

- Rule-base 방식의 한계
  - Object Detection Model의 결과에 의존적
  - VQA 모델에서 처리할 수 있는 질문임에도 틀리는 문제가 발생

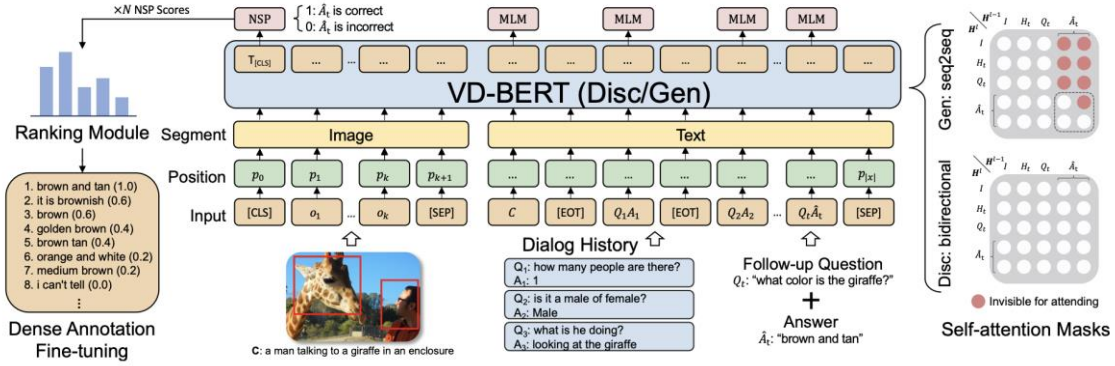
Question : 신발이 몇 개야?
Prediction : 2 개

Question : 갈색 바지가 있어?
Prediction : 아니요

Question : 사람이 있어?
Prediction(rule-base) : 예
Prediction(VQA) : 아니요

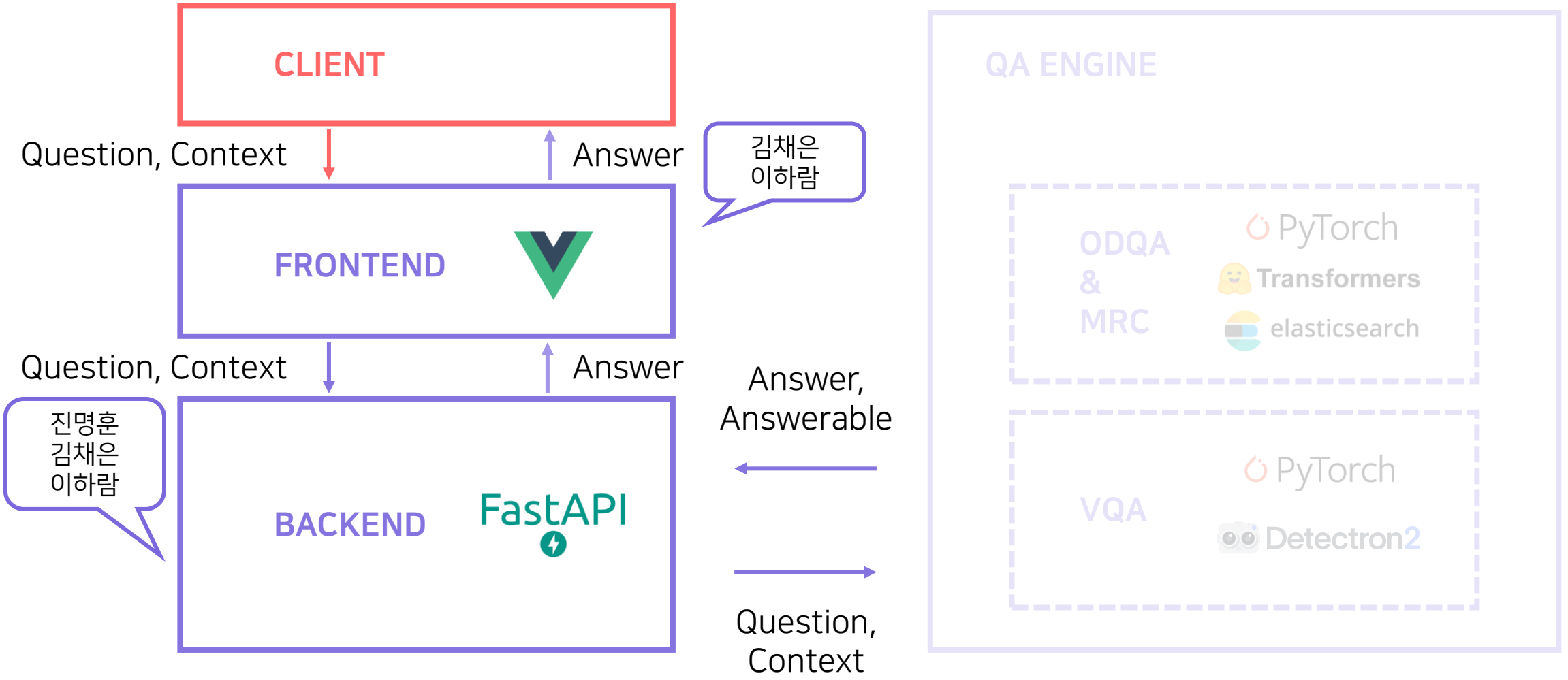
한계점 및 향후 개선 방향

- Vision-Language Transformer를 이용한 성능 개선
  - VQA Challenge 2021에서 우승한 Alibaba의 AliceMind팀은 VQA의 성능에 영향을 끼치는 요소로 Vision-Language Pre-Training을 언급
  - Vision-Language Transformer 계열의 모델 (StructVBERT, VD-BERT 등)을 통해 개선된 feature representation을 얻음으로써 성능 향상이 가능할 것으로 기대됨

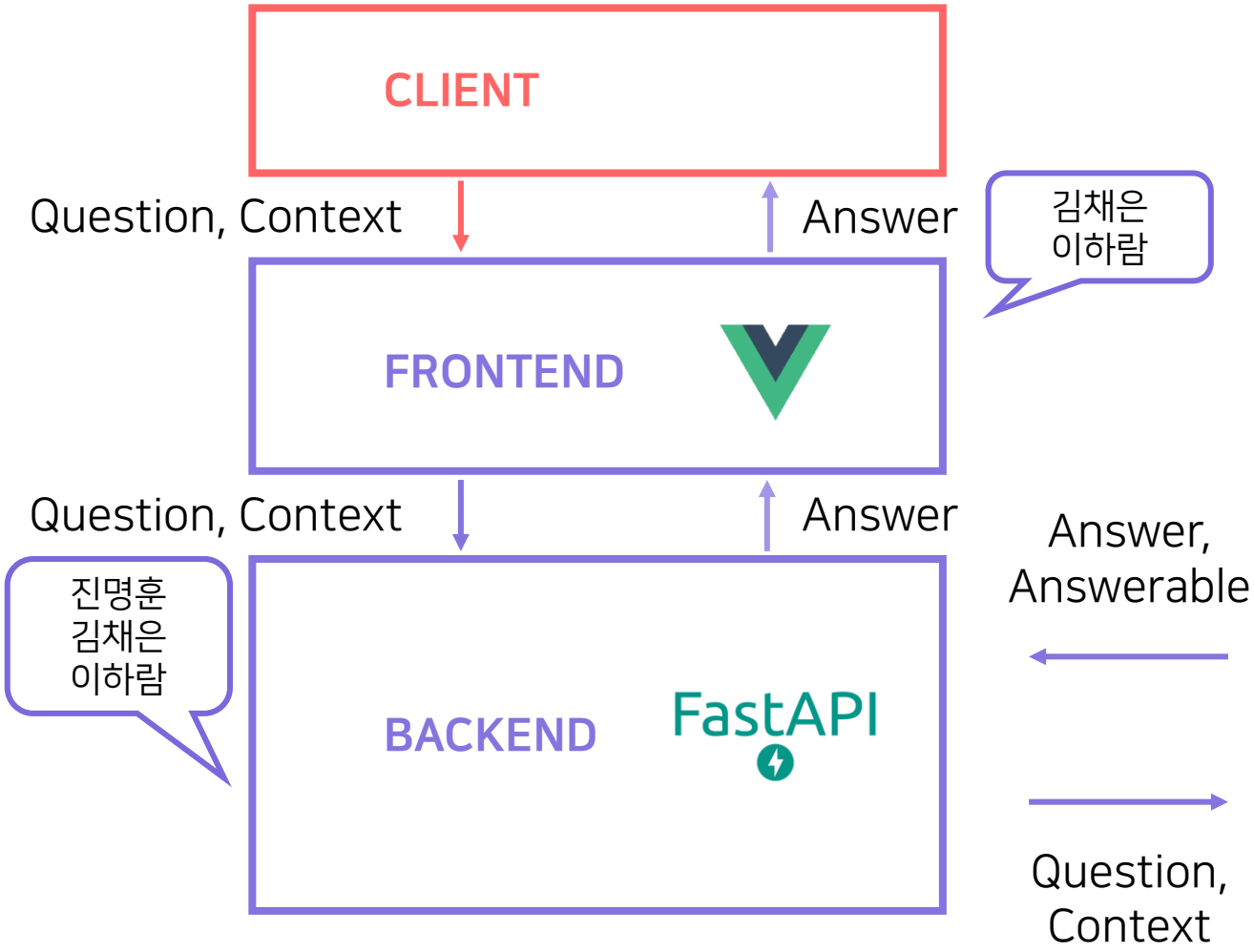


Deployment

Frontend & Backend



Frontend & Backend



Client로부터 질문을 받음

- 이 때 context(document or image)를 추가로 입력 받을 수 있음
- 이전에 입력한 Context로 질문 가능

Front는 유저의 질문을 Backend로 넘겨줌

- 만일 선택된 Context가 있다면 같이 넘겨줌

Backend는 입력된 Query와 Context를 QA Engine으로 넘겨줌

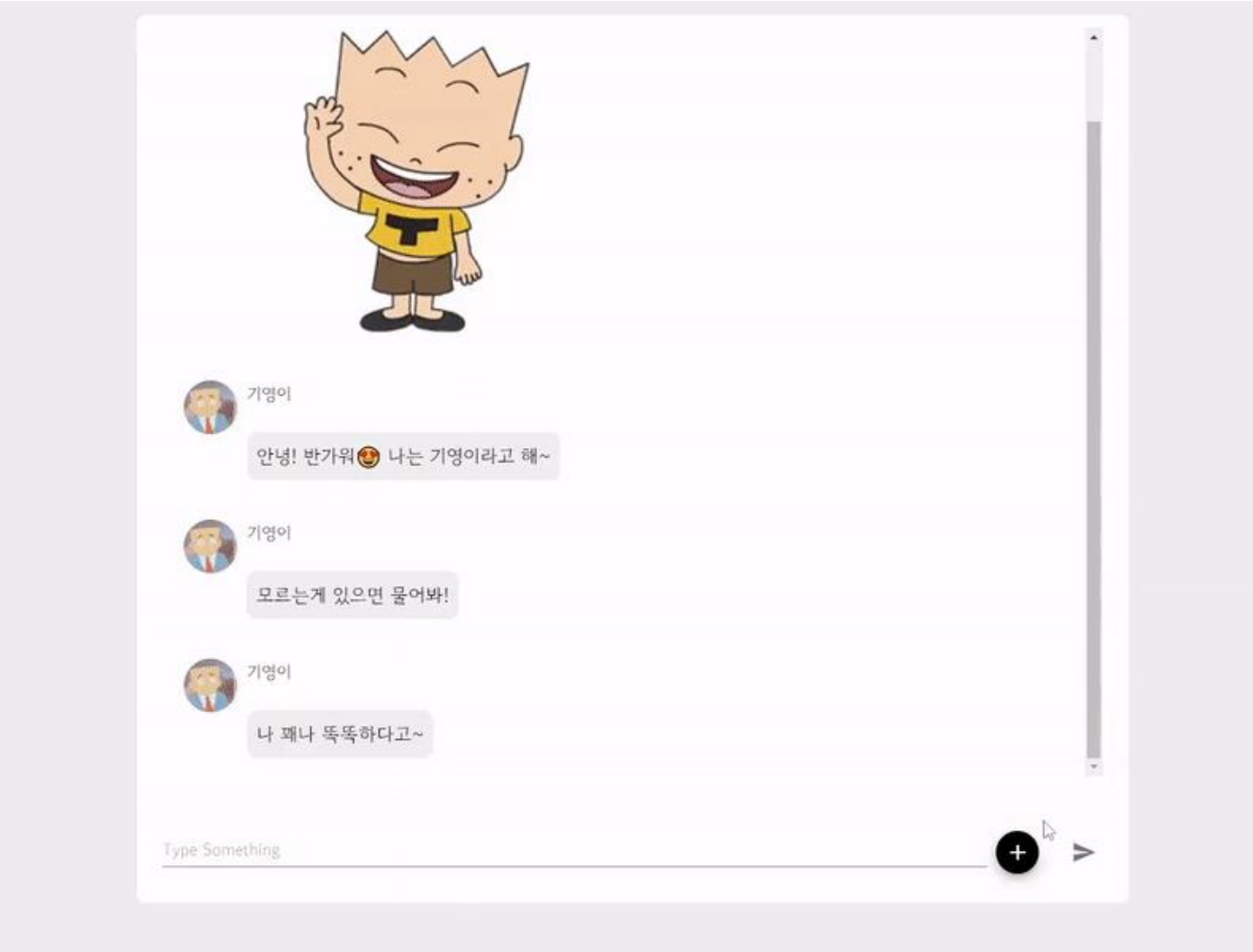
- Ch2, 3에서 설명한 MRC, VQA에서 이에 대한 답변과 no answer 여부를 반환

Backend는 QA Engine의 결과를 처리하여 적절한 답변을 Front로 넘겨줌

- No answer의 경우, 사전에 정해둔 시나리오 ML 모델로 답변 반환

Front는 Backend에서 받은 답변을 유저에게 보여줌

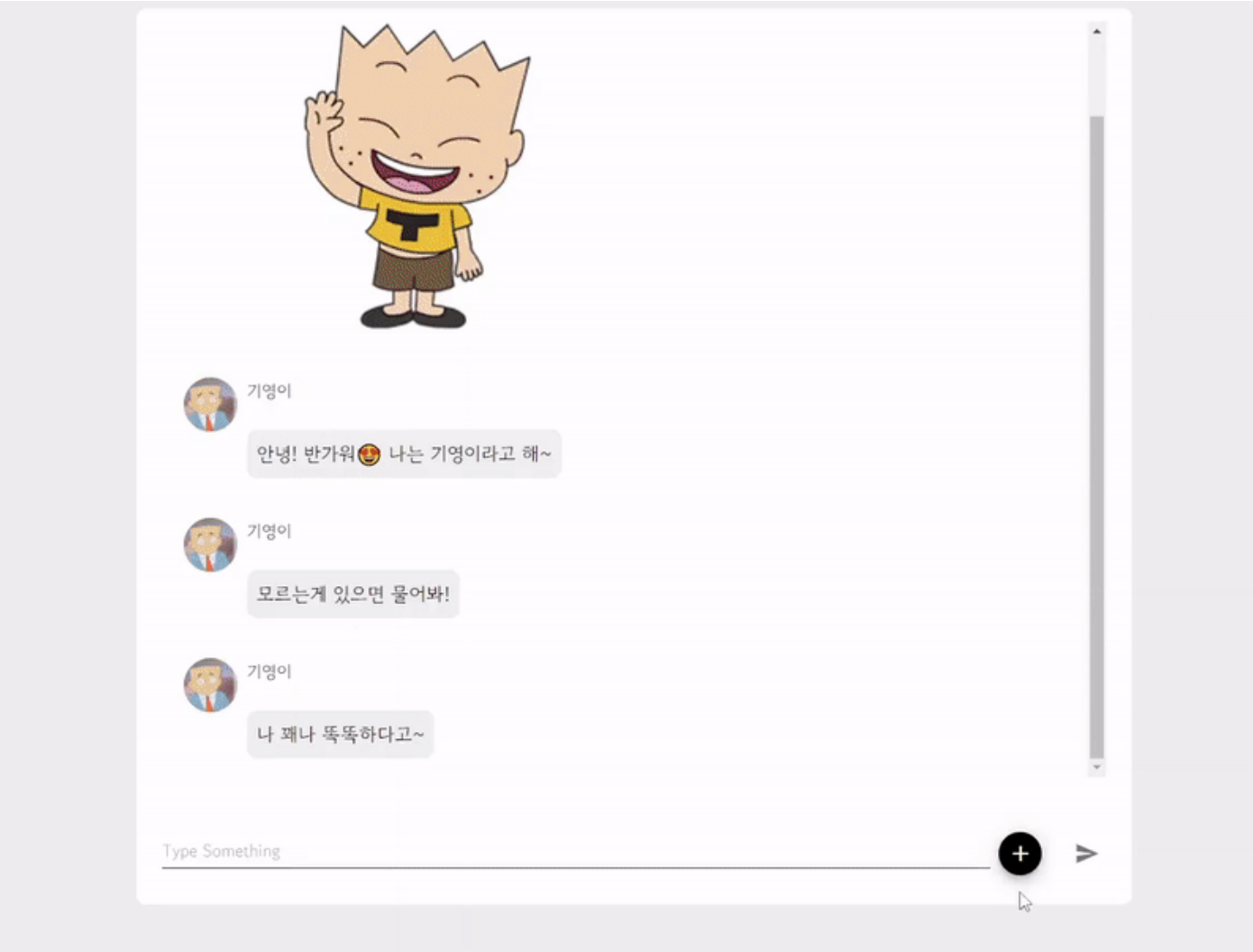
Demonstration: ODQA & MRC



총 응답시간

- 일반 대화 : 약 20~30ms
- ODQA : 약 100~1200ms
- VQA : 약 400~800ms

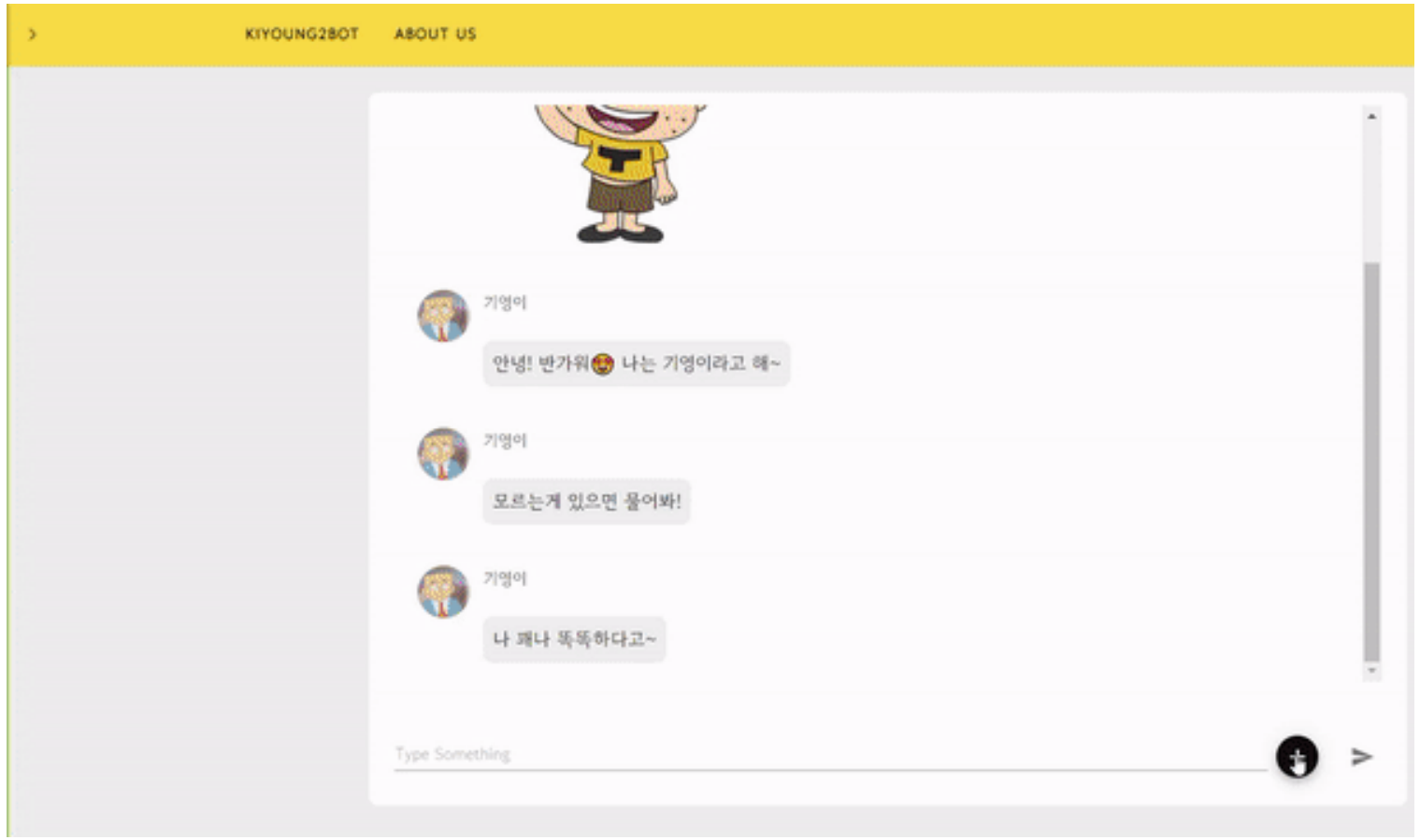
Demonstration: VQA



총 응답시간

- 일반 대화 : 약 20~30ms
- ODQA : 약 100~1200ms
- VQA : 약 400~800ms

Frontend: 사용자 편의성을 위한 기능



업로드한 문서 및 이미지 재사용

- 업로드한 문서 중 중복 선택하여 질문 가능
- 업로드한 이미지 중 하나를 선택하여 질문 가능

Conclusion

제약 조건

- 서버 스펙
  - GPU: V100 (32GB)
  - CUDA: 11.0
  - Memory: 88GB
- Model Size
  - MRC → 50MB 이하
  - VQA → 1GB 이하

```
Mon Dec 20 22:20:56 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.80.02    Driver Version: 450.80.02    CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla V100-PCIE...  On   | 00000000:00:05.0 Off |                  Off |
| N/A   71C    P0   221W / 250W |  29327MiB / 32510MiB |     83%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
+-----------------------------------------------------------------------------+
```

```
              total        used        free      shared  buff/cache   available
Mem:            92G         45G        655M         91M         47G         47G
Swap:          1.9G         29M        1.9G
```

Train Conventional Extractive QA

lr: 5e-5, fp16: True, warmup step: 0, weight decay: L2, batch size: 32,
model: ko-electra-small-discriminator-v3

Epoch	EM (Exact Match)	F1	Training Loss	Has ans EM	Has ans F1	No ans EM	No ans F1
1	75.8659	78.9675	0.5799	80.7066	86.9098	71.0253	71.0253
2	80.7672	83.5396	0.4935	82.2480	87.7929	79.2865	79.2865
3	79.6588	82.8291	0.4420	80.8625	87.2030	78.4551	78.4551
4	81.0183	83.9347	0.4079	81.7977	87.6305	80.2390	80.2390
5	81.096	84.0372	0.3625	81.6418	87.5237	80.5507	80.5507
6	81.8756	84.5888	0.3257	82.2999	87.7262	81.4513	81.4513
7	81.5899	84.4099	0.2889	82.1614	87.8015	81.0184	81.0184
8	**82.3259**	**85.0589**	**0.2594**	**82.3519**	**87.8180**	**82.3000**	**82.3000**
9	82.0921	84.8519	0.2512	82.0402	87.5596	82.1441	82.1441
10	82.0921	84.8925	0.2512	81.9363	87.5370	82.2480	82.2480

Train Intensive Reader

lr: 2e-5, fp16: True, warmup step: 814, weight decay: L2, batch size: 128,
model: ko-electra-small-discriminator-v3

Epoch	EM (Exact Match)	F1	Training Loss	Has ans EM	Has ans F1	No ans EM	No ans F1
1	80.3083	83.7025	1.4433	78.3512	85.1397	82.2653	82.2653
2	80.4035	83.4310	1.2219	81.6592	87.7142	79.1479	79.1479
3	82.2220	85.1070	1.1171	82.4039	88.1738	82.0402	82.0402
4	79.3038	82.2239	1.0495	82.9235	88.7636	75.6841	75.6841
5	**83.7634**	**86.5411**	**1.0144**	**83.0447**	**88.3600**	**84.4822**	**84.4822**
6	82.7589	85.5554	0.9560	83.4603	89.0533	82.0575	82.0575
7	83.3824	86.1311	0.9372	83.4777	88.9750	83.2871	83.2871
8	83.0793	85.8024	0.8986	83.8067	89.2528	82.3519	82.3519
9	83.0447	85.8002	0.8839	83.7028	89.2138	82.3866	82.3866
10	83.3218	86.0695	0.8697	83.6508	89.1464	82.9927	82.9927

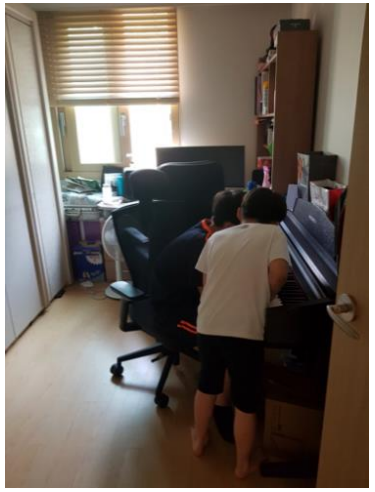
VQA 평가 지표

- VQA Challenge의 평가 방식을 따름
- 10개의 정답 중 3개 이상 맞추면 1점, 미만일 때 부분 점수 획득

$$Acc(ans) = min\{\frac{\#humans\ that\ said\ ans}{3}, 1\}$$

VQA 평가 지표

- VQA Challenge의 평가 방식을 따름
- 10개의 정답 중 3개 이상 맞추면 1점, 미만일 때 부분 점수 획득



Question: 방에 있는 사람은 지금 뭘 하고 있지?
Prediction: 피아노
Answers:
["피아노", "피아노", "피아노", "피아노", "피아노치고 있어", "피아노치고있어요", "피아노 치고 있다", "피아노 치기", "피아노 앞에서 무언가를 보고 있음", "피아노 연주"]

$$\rightarrow Acc(ans) = min\left\{\frac{4}{3}, 1\right\} = 1$$

VQA 성능 평가표

- n_classes=2,423 batch_size=64 v_dim=2048 hid_dim=768 n_epochs=20 learning_rate=1e-3 scheduler=cosine

Name	Model Info		Performance				
	Inference Time(sec)	Size (MB)	All	Yes/No	Number	Other	Unanswerable
BAN(RoBERTa-base)	0.504	846	30.84	72.65	16.84	18.91	77.99
BAN(Fasttext)	0.492	427	30.83	72.97	17.97	18.86	77.23
BAN(Fasttext) + Rule-base	0.502	427	32.25	70.98	18.91	19.63	83.50