

# [ 프로젝트 완료 보고서 ]

## 수집 데이터 기반 **BERT** 질의 응답

ver 0.9.0

AI학습 엔진 버터블록의 제작과 활용

1조

김찬(조장), 김륜환, 신예진, 신승민

No.	변경 내역	작성		검토	
		일자	작성자	일자	담당자
1	초안 작성(ver 0.9.0)	2022.01.17	김찬	2022.01.18	멘토(정휘웅) 강사(이진영)

# CONTENTS

## ▶ 01 프로젝트 계획

- 1) 개요
- 2) 추진 방향
- 3) 기대효과
- 4) 과제 수행 범위
- 5) WBS
- 6) Gantt Chart

- 기 간 : 2021년 12월 13일 ~ 2022년 1월 12일
- 팀 원 : 김찬(조장), 김륜환, 신승민, 신예진
- 내 용 : ButterBlock을 활용한 한국 역사 질의응답 모델 구현

- 한국 역사 데이터 수집 (프로젝트A - AI-Hub 데이터)
- 한국 역사 데이터를 근거로 질의응답 작성
- 수집된 한국 역사 데이터와 작성한 질의응답을 Butter Block의 MRC 컨테이너에 POST,  
추출된 정답 데이터를 출력
- 정답 데이터 정확도 향상을 위해 한국 역사 데이터를 시대별로 구분한 뒤 시대별 데이터를  
다시 문맥별로 3단계 세분화
- 플랫폼(Django)활용, 질의응답 결과 화면 구현

- Butter Block MRC를 활용한 질의응답 시스템을 구현할 수 있다.
- Butter Block MRC를 활용한 질의응답 시스템에 적합한 데이터 튜닝을 할 수 있다.
- 프로젝트 시각화를 위한 Django 플랫폼을 활용할 수 있다.
- DB를 설계하고 AWS 환경에 세팅할 수 있다.

## • 데이터 수집 범위

- 7개의 세부 도메인으로 한국 역사 데이터 수집 (출처: AI-Hub)

\* 고구려(49문장), 백제(49문장), 신라(49문장), 고려(47문장), 조선(48문장), 임진 왜란(48문장),

( 일 제 강 점 기 ( 48 문 장 ) 총 338 문 장 의 context 데이터 활용

\* 각 세부 도메인에 따른 샘플 질의문을 10종씩 총 70종의 질의문 작성 및 활용

## • 데이터 분할

- 버터블럭의 MRC 모델 질의응답 정확도 향상을 위한 데이터 튜닝

\* 한 번 에 많 은 양 의 context 입력 시 답변의 정확도가 낮아질 수 있음

\* 각 시대별 context를 문맥별로 3등분 하여 튜닝 후 MRC 모델 활용

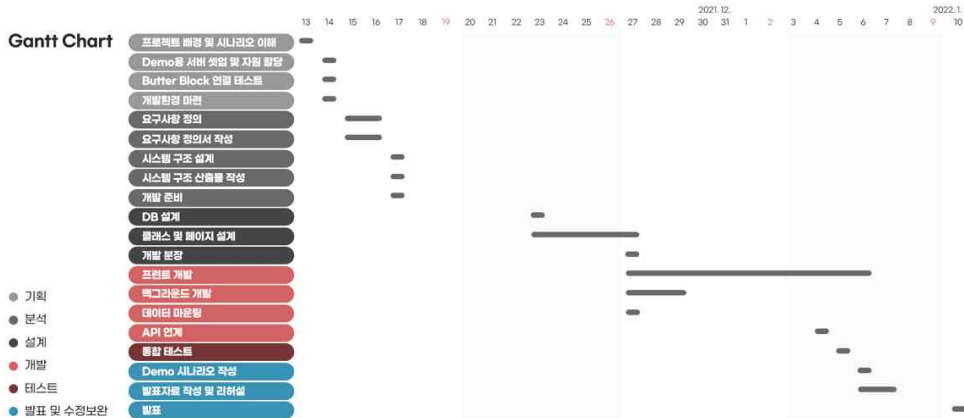
## • 웹 인터페이스 구현 범위

- main 화면에서 각 역사별 context를 선택할 수 있게 구성
  - \* 각 역사는 시대별로 고구려, 신라, 백제, 고려, 조선, 임진왜란, 일제강점기로 선택할 수 있게 구현
- sub 화면에서 문맥상 3분할 된 context를 선택할 수 있게 구성
  - \* 예) 고구려\_01, 고구려\_02, 고구려\_03처럼 3분할 된 context를 선택할 수 있게 구성
- sub 화면에서 정답 확인
  - \* 미리 작성된 샘플 질의문을 선택하거나 직접 질의문을 입력하여 정답확인 버튼을 클릭해
- 정답을 확인할 수 있게 구성
  - \* sub 화면에서 선택한 context와 질의문을 기준으로 context를 문장별로 분할해  
MRC 모델의 답변, 정확도, 프로세스 진행 시간을 확인할 수 있게 구성



1조 'BERT 기반 질의응답 시스템' WBS

대부분	구분	계획			실행			담당
	작업내용	시작	종료	소요기간	시작	종료	소요기간	
기획	프로젝트 배경 및 시나리오 이해	20211213	20211213	1	20211213	20211213	1	김찬, 김륜환, 신승민, 신예진
	Demo용 서버 셋업 및 자원 할당	20211214	20211214	1	20211214	20211214	1	김찬
	Butter Block 연결 테스트	20211214	20211214	1	20211214	20211214	1	김찬, 신예진
	개발환경 마련	20211214	20211214	1	20211214	20211214	1	김찬
분석	요구사항 정의	20211216	20211216	1	20211216	20211216	1	김찬, 김륜환, 신예진
	요구사항 정의서 작성	20211216	20211216	1	20211216	20211216	1	김찬
	시스템 구조 설계	20211217	20211217	1	20211217	20211217	1	김찬
	시스템 구조 산출물 작성	20211217	20211217	1	20211217	20211217	1	김찬
	개발 준비	20211217	20211217	1	20211217	20211217	1	김찬, 김륜환, 신예진
설계	DB 설계	20211224	20211224	1	20211223	20211223	1	신예진
	클래스 및 페이지 설계	20211224	20211224	1	20211223	20211227	3	김찬
	개발 분장	20211224	20211224	1	20211227	20211227	1	김찬
개발	프론트 개발	20220104	20220104	1	20211227	20220103	6	김찬
	백그라운드 개발	20220104	20220104	1	20211227	20211229	3	김찬
	데이터 마운팅	20220104	20220104	1	20211227	20211227	1	김륜환, 신예진
	API 연계	20220104	20220104	1	20220104	20220104	1	김찬
테스트	통합 테스트	20220106	20220106	1	20220105	20220105	1	신승민
발표 및 수정보완	Demo 시나리오 작성	20220107	20220107	1	20220106	20220106	1	신승민
	발표자료 작성 및 리허설	20220107	20220107	1	20220106	20220107	2	김찬, 김륜환, 신예진
	발표	20220110	20220110	1	20220110	20220110	1	김찬, 김륜환, 신승민, 신예진



# CONTENTS

## ▶ 02 요구사항 정의

### 1) 요구사항 목록

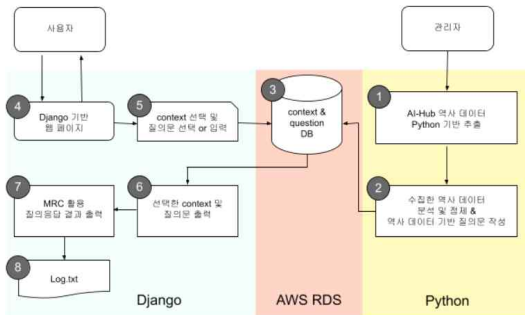
No	요구사항 ID	요구사항명	요구사항 상세 설명(안)	유형 (기능/비기능)	비고
1	DJG-001	context 선택	각 시대별로 context를 선택	기능	
2	DJG-002	세부 context 선택	시대별로 3분할 된 context중 하나를 선택	기능	
3	DJG-003	질의문 선택 및 정답 확인	context에 근거한 미리 작성된 질의문 선택 후 정답 확인	기능	
4	DJG-004	질의문 입력 및 정답 확인	input창에 직접 질문 입력 후 정답 확인	기능	
5	DJG-005	프로세스 진행 중 표시 및 프로세스 진행 경과 시간 표시	정답 확인을 클릭했을 때 프로세스 진행 중이란 문구 출력 및 정답 출력 뒤 프로세스 진행 경과 시간을 출력	기능	
6	DJG-006	main화면으로 이동	각 시대별로 context를 선택할 수 있는 화면으로 이동 가능한 home버튼 배치	기능	
7	REQ-001	데이터 수집	AI-Hub 데이터 활용	/	
8	REQ-002	데이터 분석	MRC 모델을 활용한 정답 데이터와 프로젝트 A에서 작성한 정답 데이터를 비교	/	
9	REQ-003	데이터 저장	수집한 역사 데이터와 샘플 질의문 DB에 저장 및 관리	기능	
12	REQ-004	로그 데이터 저장	질의응답 로그 DB로 저장	기능	
10	SEV-001	AWS RDS-MySQL 연동	AWS RDS에 데이터 마운트 후 장고 프로젝트와 연동	비기능	
11	SEV-002	시대별, 문맥별 context와 question 출력	SQL DB의 데이터 select 문으로 질의응답에 필요한 데이터 추출	기능	

• DJG : Django기능 요구사항 / REQ : 일반요구사항 / SEV : SQL기능 요구사항

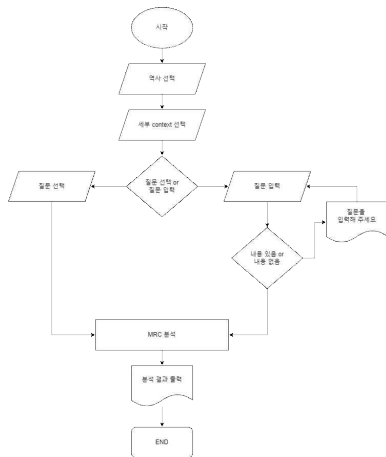
# CONTENTS

## ▶ 03 시스템 구조 정의

- 1) 시스템 구조 설계
- 2) UI 구조 설계 : Flow Chart



1. AI-Hub 음성/자연어 분야 일반상식 중 한국 역사에 관련된 데이터
2. 고구려, 백제, 신라, 고려, 조선, 임진왜란, 일제강점기로 구분된 context와 question 데이터
3. context와 question 테이블로 구성된 MySQL DB
4. django 기반 웹 페이지 (index.html, his\_sub.html로 구성)
5. 웹페이지에서 시대별 역사 선택 및 정의된 질의문을 선택하거나 입력
6. DB에서 해당 컨텍스트와 질문 내용 수집 후 출력
7. ButterBlock MRC 모델 Response(답변, 점수)
8. 질의응답 데이터 Log.txt 파일로 저장 ( ./qna/logs/)



사용자는 메인페이지에서 7개로 나누어진 한국 역사 중 하나를 선택한 뒤  
 선택한 서브페이지에서 선택한 역사가 다시 세 문단으로 구분된  
 context 중 하나를 선택(기본 첫 문단) 하고 미리 작성된 질의문 중 하나를 선택하거나  
 직접 질의문을 선택해 답변을 출력할 수 있다.  
 또한 같은 context와 질의문으로 context를 한 문장 씩 줄여서 결과를 확인할 수 있다.  
 출력된 데이터를 log.txt에 날짜별로 저장된다.

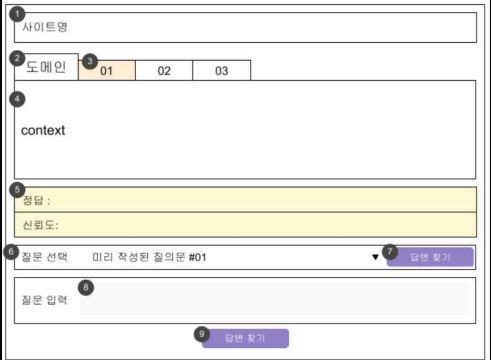
# CONTENTS


## ▶ 04 UI 화면 정의

- 1) 메인 페이지 – context 선택
- 2) 서브 페이지 – 질의응답 기능 수행



wireframe	page
	<p><b>index.html</b></p> <ol style="list-style-type: none"> <li>1. 타이틀 부분으로 클릭 시 index.html로 이동</li> <li>2. 서비스 타이틀 및 요약 설명 텍스트</li> <li>3. 시대별로 질의응답을 할 수 있는 페이지로 이동할 수 있는 네비게이션</li> <li>4. Butter Block 제작사 투블록AI와 서비스 제작팀 Team. LUT 저작권 표시</li> </ol>

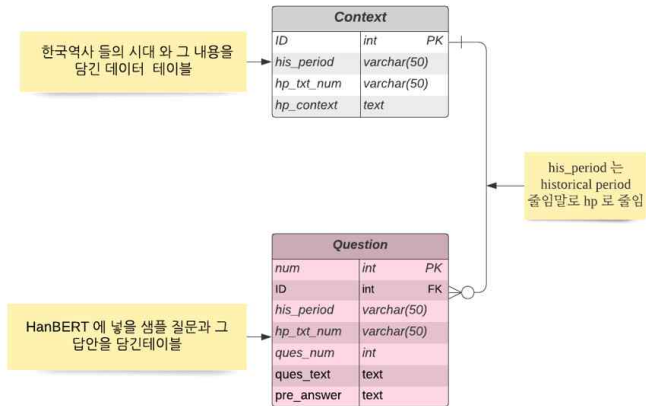
wireframe	page
 <p>The wireframe shows a form for a Q&amp;A page. It includes a site name field (1), a domain dropdown (2) with options 01, 02, 03 (3), a context selection area (4), a question type dropdown (5) with options '정답' and '신뢰도', a question selection dropdown (6) showing '미리 작성된 질문 #01' and a search button (7), a question input field (8), and a search button (9).</p>	<h3>his_sub.html #01</h3> <ol style="list-style-type: none"> <li>타이틀 부분으로 클릭 시 index.html로 이동</li> <li>context의 시대구분 출력</li> <li>3분할 된 context를 선택할 수 있는 메뉴, 클릭 시 선택된 context 메뉴는 컬러변경으로 활성화 표시</li> <li>3에서 선택한 context 내용 출력</li> <li>7의 버튼을 클릭시 MRC 모델에 데이터 전송 후 response 받은 정답과 신뢰도 데이터 출력</li> <li>db에 세팅된 question을 선택 가능</li> <li>선택한 question과 context를 mrc 모델로 post</li> <li>question을 직접 입력</li> <li>8에서 입력한 question과 context를 mrc 모델로 post</li> </ol>

wireframe	page
 <p>문장 길이별 출력 데이터</p> <p>10</p> <p>문장 길이별 답변, 점수 및 시간 출력</p> <p>11 선택문 테스트</p> <p>12 직접 입력 테스트</p> <p>13 푸터</p>	<p>his_sub.html #02</p> <p>10. 11과 12를 선택했을 때 출력되는 문장 길이별 문장 수, 답변, 점수 및 시간 출력</p> <p>11. 활성화 된 context와 선택된 질의문으로 문장 길이별 문장 수, 답변, 점수 및 시간을 확인</p> <p>12. 활성화 된 context와 직접 입력한 질의문으로 문장 길이별 문장 수, 답변, 점수 및 시간을 확인</p> <p>13. Butter Block 제작사 투블록AI와 서비스 제작팀 Team. LUT 저작권 표시</p>

# CONTENTS

## ▶ 05 데이터베이스 설계

- 1) ERD
- 2) 테이블 정의



## context

테이블정의서								
주제영역명		한국역사	작성일	2022. 1. 3		작성자	신예진	
테이블명			Context					
테이블설명		한국역사 내용으로 구성된 테이블(시대 구분은 7종으로 구성)						
No.	논리명(Attribute)	물리명(Field Name)	타입	길이	NULL	KEY	DEFAULT	비고
1	ID 번호	ID	INT	-	NotNull	PK	1	
2	시대 구분	his_period	VARCHAR	50	NotNull			한국역사 시대 구분
3	시대 세부 구분	hp_txt_num	VARCHAR	50	NotNull			세대와 문장 번호
4	내용	hp_context	TEXT	-	NotNull			역사 내용

## question

테이블정의서								
주제영역명		한국역사	작성일	2022.1.3	작성자		신예진	
테이블명			Question					
테이블설명		AI-Hub에서 수집한 역사 데이터 기반 질문 및 답안 테이블						
No.	논리명(Attribute)	물리명(Field Name)	타입	길이	NULL	KEY	DEFAULT	비고
1	번호	num	INT	-	NotNull	PK	1	
2	ID 번호	ID	INT	-	NotNull	FK	1	
3	시대 구분	his_period	VARCHAR	50	NotNull			한국역사 시대 구분
4	시대 세부 구분	hp_bt_num	VARCHAR	50	NotNull			세대와 문장 번호
5	역사별 질문 번호	ques_num	INT	-	NotNull		1	1-4개 적자시대 질문
6	질문	ques_text	TEXT	-	NotNull			질의응답 세트 질문
7	질문 답	pre_answer	TEXT	-	NotNull			질의응답 세트 응답

# CONTENTS

## ▶ 06 데이터 획득 정의

- 1) 데이터 구축 프로세스
- 2) 텍스트 데이터 어노테이션 구조

데이터 구축 프로세스 정의

구축 단계	세부 절차	설명
수집	수집	AI-Hub 음성/자연어 데이터 수집
	분석	수집 데이터 파싱 및 내용 분류
	도메인 선정	분류한 데이터에서 질의응답용 도메인 선정
정제	데이터 검사	선정한 도메인 데이터의 맞춤법 검사
	데이터 튜닝	결정된 도메인 데이터를 고구려, 백제, 신라, 고려, 조선, 임진왜란, 일제강점기 총 7종으로 구분한 뒤 다시 각 시대를 문맥별로 3분할
라벨링	시대별 질의응답 작성	7종 역사별로 데이터 조원 분담 후 질의응답 작성
검사	전수 검사	생성한 데이터의 맞춤법 및 문맥 검사



텍스트 데이터 어노테이션 구조 정의		
항목	설명	json 포맷 구축 형태
creator	제작자	<pre>{   "creator": "LUT(Level_Up_Together) Lab.",   "version": 2,   "data": {     "title": "백제",     "paragraphs": {       "context": "백제(기원전 18년 ~ 660년 8월 29일)는 한국의 고대 국가 중 하나로, ~~(내용 생략)~~동성왕은 조인을 통해 신라와의 동맹을 강화하여 고구려에 대항하였고, 무령왕은 양직공도에 의하면 지방의 22읍로써 왕족을 파견함으로써 지방에 대한 통제권을 강화하였다.",       "qas": {         {           "question": "백제는 언제 시작한 나라인가요?",           "qas_type": "when",           "answers": {             "answer": "기원전 18년",             "pos": [{"기원전 18년": "NNP"}],             "ans_index_start": 7,             "ans_index_end": 13,             "id": "백제-q1",             ...           }         },         {           "question": "무령왕은 양직공도에 의하면 어디에 왕족을 파견하였나요?",           "qas_type": "where",           "answers": {             "answer": "22읍로",             "pos": [{"22읍로": "NNP"}],             "ans_index_start": 3239,             "ans_index_end": 3242,             "id": "백제-q89"           }         }       }     }   } }</pre>
version	문서 버전	
data	문서 배열	
문서정보		
title	카테고리	
paragraphs	본문	
문서내용		
context	본문 내용	
qas	질의응답 배열	
질의내용		
question	질의 내용	
qas_type	질의 형식	
answers	답변 배열	
ans_index_start	답변 인덱스 시작	
ans_index_end	답변 인덱스 끝	
id	답변 아이디	
답변내용		
answer	답변 내용	
pos	형태소 정보	

# CONTENTS

## ▶ 07 프로그램 정의

### 1) 프로그램 목록

## 프로그램 정의서

프로젝트	수집 데이터 기반 BERT 질의 응답	단계	설계
시스템	버터블릭 MRC	Version	0.9.1
작성자	김한	작성일	2022-01-03

업무영역	폐쇄형 질의응답 시스템	업무 기능	첫페이지 - 역사 선택
프로그램 ID	his_main	프로그램 유형	온라인
프로그램 명	첫페이지 - "Kor-History Q&A"		
프로그램 설명	7개로 나뉘어진 한국 역사를 선택할 수 있는 페이지		
화면 ID	his_main	Transaction Code	
기능 구분	<input checked="" type="checkbox"/> 조회 <input type="checkbox"/> 입력 <input type="checkbox"/> 수정 <input type="checkbox"/> 삭제 <input type="checkbox"/> 출력 <input type="checkbox"/> 기타(        )		
프로그램 파일	Presentation Layer	index.html, url.py	
	Business Layer	views.py	
	Integration Layer	(N/A)	
	기타	(N/A)	
관련 테이블(Context)			
테이블 ID	테이블 명	CRUD 속성 표시	비고
ID	ID 번호	R	
his_period	시대구분	R	한국역사 시대 구분
hp_text_num	시대 세부 구분	R	시대와 문장 번호
hp_context	내용	R	역사 내용
관련 테이블(Question)			
테이블 ID	테이블 명	CRUD 속성 표시	비고
num	번호	R	
ID	ID번호	R	
his_period	시대구분	R	한국역사 시대 구분
hp_text_num	시대 세부 구분	R	시대와 문장 번호
ques_num	역사별 질문 번호	R	1~4개 시대별 질문
ques_text	질문	R	질의응답 세트 질문
pre_answer	질문 답	R	질의응답 세트 답

## 오브젝트별 이벤트 정의 (Presentation Layer)

오브젝트 명	이벤트	관련 모듈	기능 및 처리사항
app_name			- 앱 이름 정의
urlpatterns	<pre>urlpatterns = [     path("", views.index,         name = 'index'),     path('qna/&lt;context_id&gt;/',         views.qna, name = 'qna'     ]</pre>	views.py - qna()	- views의 index 함수 실행 - index.html에서 'qna/<context_id>/' request 요청이 있을 때 views의 qna 함수 실행
오브젝트별 이벤트 정의 (Business Layer)			
오브젝트 명	이벤트	관련 모듈	기능 및 처리사항
index(request):	return render(request, 'qna/index.html')		- index.html 리턴

## 프로그램 정의서

프로젝트	수집 데이터 기반 BERT 질의 응답	단계	설계
시스템	버터블릭 MRC	Version	0.9.1
작성자	김찬	작성일	2022-01-03

업무영역	폐쇄형 질의응답 시스템	업무 기능	첫페이지 - 역사 선택
프로그램 ID	his_sub	프로그램 유형	온라인
프로그램 명	서브페이지 - "Kor-History Q&A"		
프로그램 설명	선택된 역사 지문에 대해 미리 정의된 질문이나 직접 질문을 입력해 질의응답을 버터블릭 MRC 모델에 POST 하고 결과를 받을 수 있는 페이지		
화면 ID	his_qna	Transaction Code	
기능 구분	<input checked="" type="checkbox"/> 조회 <input checked="" type="checkbox"/> 입력 <input type="checkbox"/> 수정 <input type="checkbox"/> 삭제 <input checked="" type="checkbox"/> 출력 <input type="checkbox"/> 기타(    )		
프로그램 파일	Presentation Layer	his_sub.html, url.py	
	Business Layer	views.py, consumers.py	
	Integration Layer	(N/A)	
	기타	(N/A)	
관련 테이블(Context)			
테이블 ID	테이블 명	CRUD 속성 표시	비고
ID	ID 번호	R	
his_period	시대구분	R	한국역사 시대 구분
hp_text_num	시대 세부 구분	R	시대와 문장 번호
hp_context	내용	R	역사 내용
관련 테이블(Question)			
테이블 ID	테이블 명	CRUD 속성 표시	비고
num	번호	R	
ID	ID번호	R	
his_period	시대구분	R	한국역사 시대 구분
hp_text_num	시대 세부 구분	R	시대와 문장 번호
ques_num	역사별 질문 번호	R	1~4개 시대별 질문
ques_text	질문	R	질의응답 세트 질문
pre_answer	질문 답	R	질의응답 세트 답

## 오브젝트별 이벤트 정의 (Presentation Layer, 파일명: his\_sub.html)

오브젝트 명	이벤트	관련 모듈	기능 및 처리사항
onmessage	querySelector().value	channels.generic.websocket	<ul style="list-style-type: none"> <li>- 전달받은 text_data 리턴</li> <li>답변: answer</li> <li>점수: score</li> <li>진행 시간: end_time</li> <li>버튼ID : btn_id</li> <li>길이별 응답 데이터: list_context_slice_testing_data</li> <li>- 버튼 텍스트 답변 찾기로 변경</li> <li>- 각 메뉴 속성 초기화</li> </ul>
onclose	console.error()	channels.generic.websocket	<ul style="list-style-type: none"> <li>- 소켓이 끊겼을 때 로그 출력</li> <li>log: Chat socker closed unexpectedly</li> </ul>
onclick	send()	channels.generic.websocket	<ul style="list-style-type: none"> <li>- 사용자가 선택한 context와 question을 consumers.py 에 전달</li> <li>context: #myMessage_context</li> <li>question: #myMessage_question1,2</li> <li>- 버튼 텍스트에 프로세스 진행중 표시 및 클릭 버튼의 비활성화</li> <li>- 질문 입력 값 validation 수행</li> <li>- 질문 입력값이 없을 시 log 출력</li> </ul>

## 프로그램 정의서

프로젝트	수집 데이터 기반 BERT 질의 응답	단계	설계
시스템	버티컬 MRC	Version	0.9.1
작성자	김찬	작성일	2022-01-03

## 오브젝트별 이벤트 정의 (Presentation Layer, 파일명: url.py)

오브젝트 명	이벤트	관련 모듈	기능 및 처리사항
app_name			- 앱 이름 정의
urlpatterns	urlpatterns = []	views - index()	- views의 index 함수 실행 - index.html에서 'qna/<context_id>/' request 요청이 있을 때 views의 qna 함수 실행

## 오브젝트별 이벤트 정의 (Business Layer, 파일명: views.py)

오브젝트 명	이벤트	관련 모듈	기능 및 처리사항
his_sub(request, context_id):	return render(request, 'main/qna.html', contexts)		- qna.html 리턴 - <context_id>를 인자로 전달 받아 DB의 context, question 테이블에서 해당 내용을 추출 후 contexts에 dict로 담은 후 qna.html에 리턴

## 오브젝트별 이벤트 정의 (Business Layer, 파일명: consumers.py)

오브젝트 명	이벤트	관련 모듈	기능 및 처리사항
AnswerConsumer( r( AsyncWebsocket Consumer):	connect(self):	channels.generic .websocket	- 웹소켓 채널 접속
-	disconnect(self, close_code):	channels.generic .websocket	- 웹소켓 채널 로그아웃
-	receive(self, text_data):	channels.generic .websocket, json, time, os, datetime, requests	- his_sub.html에서 text_data를 json으로 받음 context: 글 내용 question: 질문 내용 dic_for_context, dic_for_list_context: 질의응답dic - 전달받은 json을 버티컬 MRC 모델에 post response_text: 반환받은 json response_answer: 답변 response_score: 점수 qna_data_list : list_context로 반환받은 데이터(문장 수, 답변, 점수, 걸린 시간) - text_data에 response된 답변 및 점수, 프로세스에 걸린 시간을 받아 his_sub.html에 send context: 질문 내용 question: 질의 내용 response_answer: 답변 내용 response_score: 답변 점수 end_time: 진행 시간 list_context_slice_testing_data: 반복 실행시 생성된 데이터  btn_id: 버튼 아이디 - 로그 저장 프로세스 진행 후 질의 응답 데이터를 txt형식으로 'qna/logs'에 날짜별로 저장

# CONTENTS

## ▶ 08 프로그램 구현

- 1) 장고 프로젝트 구조
- 2) 데이터베이스(MySQL) 구조
- 3) 프로젝트 실행결과



01 : config

- 프로젝트 기본 환경 폴더

02: qna 폴더

- views.py: DB에서 context 및 question을 로드 및 html과 화면 매핑

- consumers.py: html 페이지에서 socket을 통해 요청온 이벤트를 처리

03: static 폴더

- css, js로 구성

04: templates 폴더

- 화면을 구성하는 html로 구성

DB: history\_kor\_db

Table: context 구조

	Field	Type	Null	Key	Default	Extra
▶	ID	int	NO	PRI	<b>HULL</b>	
	his_period	varchar(50)	NO		<b>HULL</b>	
	hp_txt_num	varchar(50)	NO	UNI	<b>HULL</b>	
	hp_context	text	NO		<b>HULL</b>	

Table: question 구조

	Field	Type	Null	Key	Default	Extra
▶	num	int	NO	PRI	<b>HULL</b>	
	his_period	varchar(50)	NO		<b>HULL</b>	
	hp_txt_num	varchar(50)	NO	MUL	<b>HULL</b>	
	ques_num	int	NO		<b>HULL</b>	
	ques_text	text	NO		<b>HULL</b>	
	pre_answer	varchar(50)	NO		<b>HULL</b>	



메인페이지 : index.html

Kor-History Q&A

[고구려](#) [백제](#) [신라](#) [고려](#) [조선](#) [임진왜란](#) [일제강점기](#)

# Kor-History Q&A

심층언어 모델 **ButterBlock MRC**를 이용한 한국역사 질의응답을 테스트 할 수 있는 서비스입니다.

<a href="#">고구려</a> View Q&A Test	<a href="#">백제</a> View Q&A Test	<a href="#">신라</a> View Q&A Test	<a href="#">고려</a> View Q&A Test
<a href="#">조선</a> View Q&A Test	<a href="#">임진왜란</a> View Q&A Test	<a href="#">일제강점기</a> View Q&A Test	

©Team LUT. All rights reserved. | ©TwoBlock AI. All rights reserved.

서브페이지 : his sub.html

[illegible]

# CONTENTS

## ▶ 09 통합 테스트 시나리오

- 1) 메인 페이지
- 2) 서브 페이지

시스템/서브시스템/모듈	Kor-History Q&A		
	시나리오명	상세설명(흐름도)	검증포인트
QnA-HOME01	메인페이지 링크 확인	홈페이지 콘텐츠 링크를 확인한다. (home, 고구려, 백제, 신라, 고려, 조선, 임진왜란, 일제강점기)	해당 페이지 이동을 확인한다.
QnA-HOME02	메인으로 다시 반환 확인	각 태그 내에서 다시 메인으로 반환한 것을 확인한다(Kor-History Q&A with ButterBlock MRC 클릭)	각 태그 내에서 클릭 후 다시 메인으로 반환 여부 확인

시스템/서브시스템/모듈	Kor-History Q&A		
시나리오 ID	시나리오명	상세설명(흐름도)	검증포인트
QnA-HOME01	메인페이지 링크 확인	홈페이지 콘텐츠 링크를 확인한다. (home, 고구려, 백제, 신라, 고려, 조선, 임진왜란, 일제강점기)	해당 페이지 이동을 확인한다.
QnA-HOME02	메인으로 다시 반환 확인	각 태그 내에서 다시 메인으로 반환한 것을 확인한다(Kor-History Q&A with ButterBlock MRC 클릭)	각 태그 내에서 클릭 후 다시 메인으로 반환 여부 확인
QnA-SUB01-01	qna/고구려0_ 링크 확인	서브 페이지(고구려) 콘텐츠 링크를 확인한다. (home, 본문 내용 = 구분01, 구분02, 구분03)	고구려 문맥이 있는 페이지로 이동을 확인한다.
QnA-SUB01-02	qna/고구려0_ 콘텐츠 확인	서브 페이지(고구려) 콘텐츠를 확인한다. (상단 태그, 본문 내용, 질문 선택)	고구려 콘텐츠 내용을 확인한다.
QnA-SUB01-03	qna/고구려0_ 답변 찾기 기능 확인	질문 선택 후 '답변 찾기' 버튼 클릭시 출력 확인 (답변, 점수, 걸린시간)	답변 출력이 정상적으로 되는지 확인.
QnA-SUB01-04-01	qna/고구려0_ 수동 기능 확인	질문 직접 입력 후 '답변 찾기' 버튼 클릭시 출력 확인 (답변, 점수, 걸린 시간)	답변 출력이 정상적으로 되는지 확인.
QnA-SUB01-04-02	qna/고구려0_ 편의성 기능 확인	답변 찾기 버튼 클릭 시 입력 데이터가 없을 경우 로그 확인 (질문이 입력되지 않았습니다. 노출 후 1.5 초 뒤에 질문을 입력해 주세요)	로그 출력 및 출력 후 초기화 확인
QnA-SUB01-05	qna/고구려0_ 편의성 기능 확인	질문의 답변 찾는 중 답변 찾기 버튼 비활성화 확인	답변을 찾는 중 다른 기능 버튼 비활성화 여부 확인
QnA-SUB01-06-01	qna/고구려0_ 문장별 정보 기능 확인	문장 길이별 출력 선택은 테스트 버튼 클릭 시 출력 확인 (로그 내용: 텍스트 문장 수, 답변, 점수, 걸린 시간)	로그 출력 및 출력 후 초기화 확인
QnA-SUB01-06-02	qna/고구려0_ 문장별 정보 기능 확인	문장 길이별 출력 데이터 직접 입력 버튼 클릭 시 출력 확인	로그 출력 및 출력 후 초기화 확인
QnA-SUB01-07	qna/고구려0_ UI 기능 확인	문장 길이별 출력 데이터 진행 중 "프로세스가 진행 중입니다" 메시지 와 진행 하기 버튼 "진행 중..." 바뀌는것 확인	버튼 클릭 시 사용자 편의성 확인 (버튼 및 관련 UI 비활성화 및 진행 중 로그 표시)

시스템/서브시스템/모듈	Kor-History Q&A		
시나리오 ID	시나리오명	상세설명(흐름도)	검증포인트
QnA-SUB02-01	qna/백제0_ 링크 확인	서브 페이지(백제) 콘텐츠 링크를 확인한다. (home, 본문 내용 = 구분01, 구분02, 구분03)	백제 문맥이 있는 페이지로 이동을 확인한다.
QnA-SUB02-02	qna/백제0_ 콘텐츠 확인	서브 페이지(백제) 콘텐츠를 확인한다. (상단 태그, 본문 내용, 질문 선택)	백제 콘텐츠 내용을 확인한다.
QnA-SUB02-03	qna/백제0_ 답변 찾기 기능 확인	질문 선택 후 '답변 찾기' 버튼 클릭시 출력 확인 (답변, 점수, 걸린 시간)	답변 출력이 정상적으로 되는지 확인.
QnA-SUB02-04-01	qna/백제0_ 수동 가능 확인	질문 직접 입력 후 '답변 찾기' 버튼 클릭시 출력 확인 (답변, 점수, 걸린 시간)	답변 출력이 정상적으로 되는지 확인.
QnA-SUB02-04-02	qna/백제0_ 편의성 기능 확인	답변 찾기 버튼 클릭 시 입력 데이터가 없을 경우 로그 확인 (질문이 입력되지 않았습니다. 노출 후 1.5 초 뒤에 질문을 입력해 주세요)	로그 출력 및 출력 후 초기화 확인
QnA-SUB02-05	qna/백제0_ 편의성 기능 확인	질문의 답변 찾는중 답변 찾기 버튼 비활성화 확인	답변을 찾는 중 다른 기능 버튼 비활성화 여부 확인
QnA-SUB02-06-01	qna/백제0_ 문장별 정보 기능 확인	문장 길이별 출력 선택문 테스트 버튼 클릭시 출력 확인 (로그내용: 텍스트 문장 수, 답변, 점수, 걸린 시간)	로그 출력 및 출력 후 초기화 확인
QnA-SUB02-06-02	qna/백제0_ 문장별 정보 기능 확인	문장 길이별 출력 데이터 직접 입력 버튼 클릭시 출력 확인	로그 출력 및 출력 후 초기화 확인
QnA-SUB02-07	qna/백제0_ UI 기능 확인	문장 길이별 출력 데이터 진행중 "프로세스가 진행 중입니다" 메시지 와 진행 하기 버튼 " 진행 중..." 바뀌는것 확인	버튼 클릭 시 사용자 편의성 확인 (버튼 및 관련 UI 비활성화 및 진행 중 로그 표시)
QnA-SUB02-01	qna/백제0_ 링크 확인	서브 페이지(백제) 콘텐츠 링크를 확인한다. (home, 본문 내용 = 구분01, 구분02, 구분03)	백제 문맥이 있는 페이지로 이동을 확인한다.
QnA-SUB02-02	qna/백제0_ 콘텐츠 확인	서브 페이지(백제) 콘텐츠를 확인한다. (상단 태그, 본문 내용, 질문 선택)	백제 콘텐츠 내용을 확인한다.

시스템/서브시스템/모듈	Kor-History Q&A		
시나리오 ID	시나리오명	상세설명(흐름도)	검증포인트
QnA-SUB03-01	qna/신라0_ 링크 확인	서브 페이지(신라) 콘텐츠 링크를 확인한다. (home, 본문 내용 = 구분01, 구분02, 구분03)	신라 문맥이 있는 페이지로 이동을 확인한다.
QnA-SUB03-02	qna/신라0_ 콘텐츠 확인	서브 페이지(신라) 콘텐츠를 확인한다. (상단 태그, 본문내용, 질문 선택)	신라 콘텐츠 내용을 확인한다.
QnA-SUB03-03	qna/신라0_ 답변 찾기 기능 확인	질문 선택 후 '답변 찾기' 버튼 클릭시 출력 확인 (답변, 점수, 걸린시간)	답변 출력이 정상적으로 되는지 확인.
QnA-SUB03-04-01	qna/신라0_ 수동 가능 확인	질문 직접 입력 후 '답변 찾기' 버튼 클릭시 출력 확인 (답변, 점수, 걸린시간)	답변 출력이 정상적으로 되는지 확인.
QnA-SUB03-04-02	qna/신라0_ 편의성 기능 확인	답변 찾기 버튼 클릭시 입력 데이터가 없을 경우 로그 확인 (질문이 입력되지 않았습니다. 노출 후 1.5 초 뒤에 질문을 입력해 주세요)	로그 출력 및 출력 후 초기화 확인
QnA-SUB03-05	qna/신라0_ 편의성 기능 확인	질문의 답변 찾는중 답변 찾기 버튼 비활성화 확인	답변을 찾는 중 다른 기능 버튼 비활성화 여부 확인
QnA-SUB03-06-01	qna/신라0_ 문장별 정보 기능 확인	문장 길이별 출력 선택문 테스트 버튼 클릭시 출력 확인 (로그내용: 텍스트 문장 수, 답변, 점수, 걸린 시간)	로그 출력 및 출력 후 초기화 확인
QnA-SUB03-06-02	qna/신라0_ 문장별 정보 기능 확인	문장 길이별 출력 데이터 직접 입력 버튼 클릭시 출력 확인	로그 출력 및 출력 후 초기화 확인
QnA-SUB03-07	qna/신라0_ UI 기능 확인	문장 길이별 출력 데이터 진행중 "프로세스가 진행 중입니다" 메세지 와 진행 하기 버튼 " 진행 중..." 바뀌는것 확인	버튼 클릭 시 사용자 편의성 확인 (버튼 및 관련 UI 비활성화 및 진행 중 로그 표시)
QnA-SUB03-01	qna/신라0_ 링크 확인	서브 페이지(신라) 콘텐츠 링크를 확인한다. (home, 본문 내용 = 구분01, 구분02, 구분03)	신라 문맥이 있는 페이지로 이동을 확인한다.
QnA-SUB03-02	qna/신라0_ 콘텐츠 확인	서브 페이지(신라) 콘텐츠를 확인한다. (상단 태그, 본문내용, 질문 선택)	신라 콘텐츠 내용을 확인한다.

시스템/서브시스템/모듈	Kor-History Q&A		
시나리오 ID	시나리오명	상세설명(흐름도)	검증포인트
QnA-SUB04-01	qna/고려0_ 링크 확인	서브 페이지(고려) 콘텐츠 링크를 확인한다. (home, 본문 내용 = 구분01, 구분02, 구분03)	고려 문맥이 있는 페이지로 이동을 확인한다.
QnA-SUB04-02	qna/고려0_ 콘텐츠 확인	서브 페이지(고려) 콘텐츠를 확인한다. (상단 태그, 본문내용, 질문 선택)	고려 콘텐츠 내용을 확인한다.
QnA-SUB04-03	qna/고려0_ 답변 찾기 기능 확인	질문 선택 후 '답변 찾기' 버튼 클릭시 출력 확인 (답변, 점수, 걸린시간)	답변 출력이 정상적으로 되는지 확인.
QnA-SUB04-04-01	qna/고려0_ 수동 가능 확인	질문 직접 입력 후 '답변 찾기' 버튼 클릭시 출력 확인 (답변, 점수, 걸린시간)	답변 출력이 정상적으로 되는지 확인.
QnA-SUB04-04-02	qna/고려0_ 편의성 기능 확인	답변 찾기 버튼 클릭 시 입력 데이터가 없을 경우 로그 확인 (질문이 입력되지 않았습니다. 노출 후 1.5 초 뒤에 질문을 입력해 주세요)	로그 출력 및 출력 후 초기화 확인
QnA-SUB04-05	qna/고려0_ 편의성 기능 확인	질문의 답변 찾는중 답변 찾기 버튼 비활성화 확인	답변을 찾는 중 다른 기능 버튼 비활성화 여부 확인
QnA-SUB04-06-01	qna/고려0_ 문장별 정보 기능 확인	문장 길이별 출력 선택문 테스트 버튼 클릭 시 출력 확인 (로그내용: 텍스트 문장 수, 답변, 점수, 걸린 시간)	로그 출력 및 출력 후 초기화 확인
QnA-SUB04-06-02	qna/고려0_ 문장별 정보 기능 확인	문장 길이별 출력 데이터 직접 입력 버튼 클릭 시 출력 확인	로그 출력 및 출력 후 초기화 확인
QnA-SUB04-07	qna/고려0_ UI 기능 확인	문장 길이별 출력 데이터 진행중 "프로세스가 진행 중입니다" 메세지 와 진행 하기 버튼 " 진행 중..." 바뀌는것 확인	버튼 클릭시 사용자 편의성 확인 (버튼 및 관련 UI 비활성화 및 진행 중 로그 표시)
QnA-SUB04-01	qna/고려0_ 링크 확인	서브 페이지(고려) 콘텐츠 링크를 확인한다. (home, 본문 내용 = 구분01, 구분02, 구분03)	고려 문맥이 있는 페이지로 이동을 확인한다.
QnA-SUB04-02	qna/고려0_ 콘텐츠 확인	서브 페이지(고려) 콘텐츠를 확인한다. (상단 태그, 본문내용, 질문 선택)	고려 콘텐츠 내용을 확인한다.



시스템/서브시스템/모듈	Kor-History Q&A		
시나리오 ID	시나리오명	상세설명(흐름도)	검증포인트
QnA-SUB05-01	qna/조선0_ 링크 확인	서브페이지(조선) 콘텐츠 링크를 확인한다. (home, 본문 내용 = 구분01, 구분02, 구분03)	조선 문맥이 있는 페이지로 이동을 확인한다.
QnA-SUB05-02	qna/조선0_ 콘텐츠 확인	서브페이지(조선) 콘텐츠를 확인한다. (상단 태그, 본문내용, 질문 선택)	조선 콘텐츠 내용을 확인한다.
QnA-SUB05-03	qna/조선0_ 답변 찾기 기능 확인	질문 선택 후 '답변 찾기' 버튼 클릭시 출력 확인 (답변, 점수, 걸린시간)	답변 출력이 정상적으로 되는지 확인.
QnA-SUB05-04-01	qna/조선0_ 수동 가능 확인	질문 직접 입력 후 '답변 찾기' 버튼 클릭 시 출력 확인 (답변, 점수, 걸린시간)	답변 출력이 정상적으로 되는지 확인.
QnA-SUB05-04-02	qna/조선0_ 편의성 기능 확인	답변 찾기 버튼 클릭 시 입력 데이터가 없을 경우 로그 확인 (질문이 입력되지 않았습니다. 노출 후 1.5 초 뒤에 질문을 입력해 주세요)	로그 출력 및 출력 후 초기화 확인
QnA-SUB05-05	qna/조선0_ 편의성 기능 확인	질문의 답변 찾는중 답변 찾기 버튼 비활성화 확인	답변을 찾는 중 다른 기능 버튼 비활성화 여부 확인
QnA-SUB05-06-01	qna/조선0_ 문장별 정보 기능 확인	문장 길이별 출력 선택문 테스트 버튼 클릭시 출력 확인 (로그내용: 텍스트 문장 수, 답변, 점수, 걸린 시간)	로그 출력 및 출력 후 초기화 확인
QnA-SUB05-06-02	qna/조선0_ 문장별 정보 기능 확인	문장 길이별 출력 데이터 직접 입력 버튼 클릭시 출력 확인	로그 출력 및 출력 후 초기화 확인
QnA-SUB05-07	qna/조선0_ UI 기능 확인	문장 길이별 출력 데이터 진행중 "프로세스가 진행 중입니다" 메세지 와 진행 하기 버튼 "진행 중..." 바뀌는것 확인	버튼 클릭시 사용자 편의성 확인 (버튼 및 관련 UI 비활성화 및 진행 중 로그 표시)
QnA-SUB05-01	qna/조선0_ 링크 확인	서브페이지(조선) 콘텐츠 링크를 확인한다. (home, 본문 내용 = 구분01, 구분02, 구분03)	조선 문맥이 있는 페이지로 이동을 확인한다.
QnA-SUB05-02	qna/조선0_ 콘텐츠 확인	서브페이지(조선) 콘텐츠를 확인한다. (상단 태그, 본문내용, 질문 선택)	조선 콘텐츠 내용을 확인한다.

시스템/서브시스템/모듈	Kor-History Q&A		
시나리오 ID	시나리오명	상세설명(흐름도)	검증포인트
QnA-SUB06-01	qna/임진왜란0_ 링크 확인	서브페이지(임진왜란) 콘텐츠를 링크를 확인한다. (home, 본문 내용 = 구분01, 구분02, 구분03)	임진왜란 문맥이 있는 페이지로 이동을 확인한다.
QnA-SUB06-02	qna/임진왜란0_ 콘텐츠를 확인	서브페이지(임진왜란) 콘텐츠를 확인한다. (상단 태그, 본문내용, 질문 선택)	임진왜란 콘텐츠 내용을 확인한다.
QnA-SUB06-03	qna/임진왜란0_ 답변 찾기 기능 확인	질문 선택 후 '답변 찾기' 버튼 클릭시 출력 확인 (답변, 점수, 걸린시간)	답변 출력이 정상적으로 되는지 확인.
QnA-SUB06-04-01	qna/임진왜란0_ 수동 기능 확인	질문 직접 입력 후 '답변 찾기' 버튼 클릭시 출력 확인 (답변, 점수, 걸린시간)	답변 출력이 정상적으로 되는지 확인.
QnA-SUB06-04-02	qna/임진왜란0_ 편의성 기능 확인	답변 찾기 버튼 클릭시 입력 데이터가 없을 경우 로그 확인 (질문이 입력되지 않았습니다. 노출 후 1.5 초 뒤에 질문을 입력해 주세요)	로그 출력 및 출력 후 초기화 확인
QnA-SUB06-05	qna/임진왜란0_ 편의성 기능 확인	질문의 답변 찾는중 답변 찾기 버튼 비활성화 확인	답변을 찾는 중 다른 기능 버튼 비활성화 여부 확인
QnA-SUB06-06-01	qna/임진왜란0_ 문장별 정보 기능 확인	문장 길이별 출력 선택문 테스트 버튼 클릭시 출력 확인 (로그내용: 텍스트 문장 수, 답변, 점수, 걸린 시간)	로그 출력 및 출력 후 초기화 확인
QnA-SUB06-06-02	qna/임진왜란0_ 문장별 정보 기능 확인	문장 길이별 출력 데이터 직접 입력 버튼 클릭시 출력 확인	로그 출력 및 출력 후 초기화 확인
QnA-SUB06-07	qna/임진왜란0_ UI 기능 확인	문장 길이별 출력 데이터 진행중 "프로세스가 진행 중입니다" 메세지 와 진행 하기 버튼 "진행 중..." 바뀌는것 확인	버튼 클릭시 사용자 편의성 확인 (버튼 및 관련 UI 비활성화 및 진행 중 로그 표시)
QnA-SUB06-01	qna/임진왜란0_ 링크 확인	서브페이지(임진왜란) 콘텐츠를 링크를 확인한다. (home, 본문 내용 = 구분01, 구분02, 구분03)	임진왜란 문맥이 있는 페이지로 이동을 확인한다.
QnA-SUB06-02	qna/임진왜란0_ 콘텐츠를 확인	서브페이지(임진왜란) 콘텐츠를 확인한다. (상단 태그, 본문내용, 질문 선택)	임진왜란 콘텐츠 내용을 확인한다.

시스템/서브시스템/모듈	Kor-History Q&A		
시나리오 ID	시나리오명	상세설명(흐름도)	검증포인트
QnA-SUB07-01	qna/일제강점기0_ 링크 확인	서브 페이지(일제강점기) 콘텐츠 링크를 확인한다. (home, 본문 내용 = 구분01, 구분02, 구분03)	일제강점기 문맥이 있는 페이지로 이동을 확인한다.
QnA-SUB07-02	qna/일제강점기0_ 콘텐츠 확인	서브 페이지(일제강점기) 콘텐츠를 확인한다. (상단 태그, 본문 내용, 질문 선택)	일제강점기 콘텐츠 내용을 확인한다.
QnA-SUB07-03	qna/일제강점기0_ 답변 찾기 기능 확인	질문 선택 후 '답변 찾기' 버튼 클릭시 출력 확인 (답변, 점수, 걸린 시간)	답변 출력이 정상적으로 되는지 확인.
QnA-SUB07-04-01	qna/일제강점기0_ 수동 기능 확인	질문 직접 입력 후 '답변 찾기' 버튼 클릭시 출력 확인 (답변, 점수, 걸린 시간)	답변 출력이 정상적으로 되는지 확인.
QnA-SUB07-04-02	qna/일제강점기0_ 편의성 기능 확인	답변 찾기 버튼 클릭시 입력 데이터가 없을 경우 로그 확인 (질문이 입력되지 않았습니다. 노출 후 1.5 초 뒤에 질문을 입력해 주세요)	로그 출력 및 출력 후 초기화 확인
QnA-SUB07-05	qna/일제강점기0_ 편의성 기능 확인	질문의 답변 찾는중 답변 찾기 버튼 비활성화 확인	답변을 찾는 중 다른 기능 버튼 비활성화 여부 확인
QnA-SUB07-06-01	qna/일제강점기0_ 문장별 정보 기능 확인	문장 길이별 출력 선택문 테스트 버튼 클릭시 출력 확인 (로그내용: 텍스트 문장 수, 답변, 점수, 걸린 시간)	로그 출력 및 출력 후 초기화 확인
QnA-SUB07-06-02	qna/일제강점기0_ 문장별 정보 기능 확인	문장 길이별 출력 데이터 직접 입력 버튼 클릭시 출력 확인	로그 출력 및 출력 후 초기화 확인
QnA-SUB07-07	qna/일제강점기0_ UI 기능 확인	문장 길이별 출력 데이터 진행중 "프로세스가 진행 중입니다" 메세지 와 진행 하기 버튼 "진행 중..." 바뀌는것 확인	버튼 클릭시 사용자 편의성 확인 (버튼 및 관련 UI 비활성화 및 진행 중 로그 표시)
QnA-SUB07-01	qna/일제강점기0_ 링크 확인	서브 페이지(일제강점기) 콘텐츠 링크를 확인한다. (home, 본문 내용 = 구분01, 구분02, 구분03)	일제강점기 문맥이 있는 페이지로 이동을 확인한다.
QnA-SUB07-02	qna/일제강점기0_ 콘텐츠 확인	서브 페이지(일제강점기) 콘텐츠를 확인한다. (상단 태그, 본문 내용, 질문 선택)	일제강점기 콘텐츠 내용을 확인한다.

시스템/서브시스템/모듈	Kor-History Q&A		
	시나리오명	상세설명(흐름도)	검증포인트
QnA-SAVE	문장 길이별 답변 출력 및 시간 테스트 데이터 저장 확인	문장 길이별 답변 출력 및 시간 테스트 선택문 테스트와 직접 입력 테스트 출력 후 선택문 테스트와 직접 입력 테스트 데이터 qna/logs 폴더 내 저장 확인한다	qna/logs 폴더 내 선택문 테스트와 직접 입력 테스트 데이터 저장 여부 확인한다

# CONTENTS

## ▶ 10 형상관리

- 1) Git Hub 용어
- 2) Git Hub를 통한 프로젝트 관리

용어	설명
커밋(Commit)	작업 디렉토리에서 변경, 추가 및 삭제된 파일을 원본 저장소인 서버에 적용
중앙 저장소(Repository)	원본 소스를 저장하고 있는 저장소
작업 디렉토리(Working Copy)	원본 저장소로부터 체크아웃을 통해 내려 받은 내 로컬 PC에 있는 작업 사본 디렉토리 의미
갱신(Update)	체크아웃을 받은 작업 디렉토리를 원본 저장소의 가장 최신 커밋된 버전까지 업데이트하는 명령어
리비전(Revision)	소스 파일을 수정하여 커밋하게 되면 일정한 규칙에 의해 숫자가 증가함. 저장소에 저장된 각각의 파일 버전이라 할 수 있음
되돌리기(Roll Back)	작업 디렉토리에 저장되어 있는 사본을 특정 리비전 또는 특정 시간으로 복원할 수 있도록 하는 명령어
브랜치(Branch)	Root프로젝트로부터 파생된 프로젝트. Branch를 생성하여 버전 관리가 가능(ex 특정 커스터마이징, 미확인 된 새로운 기능을 구현 중 일 때 등등....)
머지(Merge)	Branch에서 진행하던 작업을 Root프로젝트와 합침

Git Hub 주소 : <https://github.com/sputnik7565/Kor-History-QnA>

Search or jump to... Pull requests Issues Marketplace Explore

sputnik7565 / Kor-History-QnA Public

Unwatch Fork Star

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

Go to file Add file Code

sputnik7565 Create README.md 6122c18 1 hour ago 2 commits

.idea	first commit	6 days ago
__pycache__	first commit	6 days ago
config	first commit	6 days ago
qna	first commit	6 days ago
static	first commit	6 days ago
templates	first commit	6 days ago
README.md	Create README.md	1 hour ago
debugfile3	first commit	6 days ago
manage.py	first commit	6 days ago
my_setting.py	first commit	6 days ago

README.md

## Kor-History Q & A

History Q & A is a simple ButterBlock MRC (Machine Reading Comprehension) Model web server that will give sample questions / option to type your own question from Korean historical periods texts and extract meaning from the text.

About

No description, website, or topics provided.

Readme

0 stars

1 watching

1 fork

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

Python 55.0% HTML 45.0%

# CONTENTS

## ▶ 11 개발환경 구축

- 1) 환경
- 2) his\_qna.tar 도커 이미지 확인
- 3) 도커 데스크탑 다운로드 및 설치
- 4) 리눅스 커널 업데이트
- 5) 도커 이미지 로드
- 6) 도커 데스크탑 실행
- 7) 컨테이너 이름 및 포트 설정 후 실행
- 8) 브라우저에서 컨테이너 구동 확인



- 1) 운영체제: window 10
- 2) 네트워크: 인터넷 접속 가능한 환경, Butterblock 장비와 동일한 네트워크 필요
- 3) Butterblock 장비의 MRC 구동
  - 1번 버터블록 장비 구동
  - 우부투 계정 ( team1 / 1234!@#\$ ) 으로 리눅스 사용자 로그인
  - 크롬 브라우저를 열고 <http://localhost:9000> 으로 접속해 Potainer 접속( team13 / 1234!@#\$ )
  - farvent\_noether 컨테이너 구동 (포트번호 8888)

Container list 

Portainer support team13  
[my account](#) [log out](#)

Containers								Columns	Settings
<div>  Start            Stop            Kill            Restart            Pause            Resume            Remove            Add container         </div>									
Q Search...									
<input type="checkbox"/> Name	State  Filter	Quick actions	Stack	Image	Created	Published Ports	Ownership		
<input type="checkbox"/> fervent_noether	running		-	tbai:MRC-for-Service	2021-12-13 10:46:45	<a href="#">8888:8888</a> <a href="#">8888:8888</a>	administrators		
<input type="checkbox"/> portainer	running		-	portainer/portainer	2021-12-06 15:39:03	<a href="#">9000:9000</a> <a href="#">9000:9000</a>	administrators		
<input type="checkbox"/> queue	stopped		-	tbai:NER-for-Service	2021-12-16 12:54:31	-	administrators		
<input type="checkbox"/> mrc_env	stopped		-	tbai:MRC-for-Service	2021-12-09 11:26:31	-	administrators		
<input type="checkbox"/> suspicious_chebyshev	stopped		-	tbai:NER-for-Service	2021-12-08 18:52:50	-	administrators		
								items per page	10

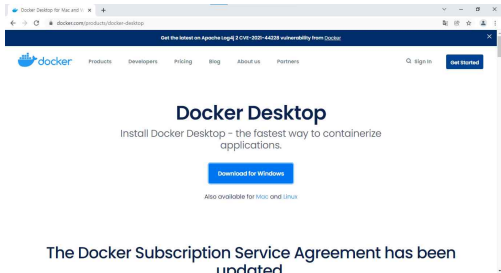
## 00\_프로젝트 도커 이미지 확인



2022-01-09 오후 4:14

압축(TAR) 파일

1,165,974...



## 01\_도커 데스크탑 다운로드

<https://www.docker.com/products/docker-desktop>



## 02\_기본 설정으로 체크 후 설치



### 03\_도커 데스크탑 구동시 다음과 같은 오류 발생 시 리눅스 커널 업데이트를 위해 해당 링크로 이동

<https://aka.ms/wsl2kernel>.

## 04\_업데이트 패키지 다운로드

4단계 - Linux 커널 업데이트 패키지 다운로드

1. 최신 패키지를 다운로드합니다.

ARM64 머신을 사용하는 경우 ARM64 패키지를 대신 다운로드하세요. 사용하고 있는 머신의 종류를 알 도르는 경우 명령 프롬프트 또는 PowerShell을 열고

```
systeminfo | find "system type"을 입력합니다.
중요! 비 영어 Windows 버전에서는 "시스템 유형" 문자열을 분할하여 검색 텍스트를
find 명령에 대한 다음과는 이스케이프해야 할 수도 있습니다.
systeminfo | find "Systemtype"입니다.
```

2. 이전 단계에서 다운로드한 업데이트 패키지 실행합니다.  
종료 후 컴퓨터를 다시 시작합니다.

설치가 완료되면 새 Linux 배포를 설치할 때 WSL 2를 기본 버전으로 선택합니다. (새 Linux 설치를 WSL 1로 설정하려면 이 단계를

이 문서의 내용

- 1단계 - Linux용 Windows 하위 시스템 사용
- 2단계 - WSL 2 실행을 위한 요구 사항 확인
- 3단계 - Virtual Machine 기능 사용
- 4단계 - Linux 커널 업데이트 패키지 다운로드
- 5단계 - WSL 2를 기본 버전으로 설정
- 6단계 - 선택한 Linux 배포 설치

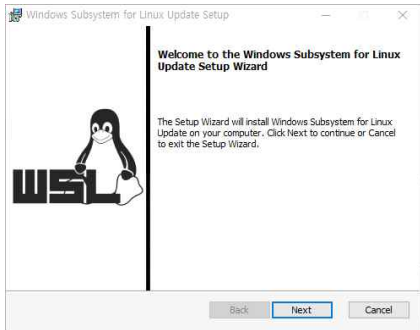
설치 문제 해결  
배포판 다운로드  
Windows 하위 시스템 설치(선택 사항)

WSL 2 installation is incomplete.

The WSL 2 Linux kernel is now installed using a separate MSI u package. Please click the link and follow the instructions to imi kernel update: <https://aka.ms/wsl2kernel>.

Press Restart after installing the Linux kernel.

Restart



## 05\_실행

명령 프롬프트

```
Microsoft Windows [Version 10.0.19043.1415]  
(c) Microsoft Corporation. All rights reserved.
```

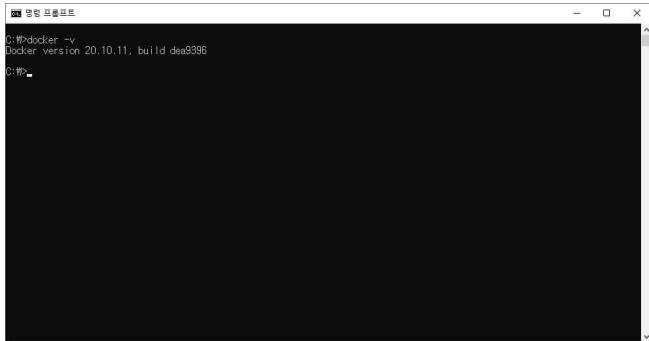
```
C:\Users\sputnik>cd\\
```

```
C:\>wsl --set-default-version 2
```

06\_cmd에서 다음 명령어 실행

`wsl --set-default-version 2`





```
C:\#>docker -v
Docker version 20.10.11, build dea9396
C:\#>_
```

07\_cmd에서 도커 버전 확인

```
docker -v
```

```

명령 프롬프트
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: DCE9-15F4

C:\project 디렉터리

2022-01-09 오후 04:59 <DIR>
2022-01-09 오후 04:59 <DIR>
2022-01-09 오후 04:14 1,193,957,376 his_qna.tar
                  1개 파일 1,193,957,376 바이트
                  2개 디렉터리 96,105,680,896 바이트 남음

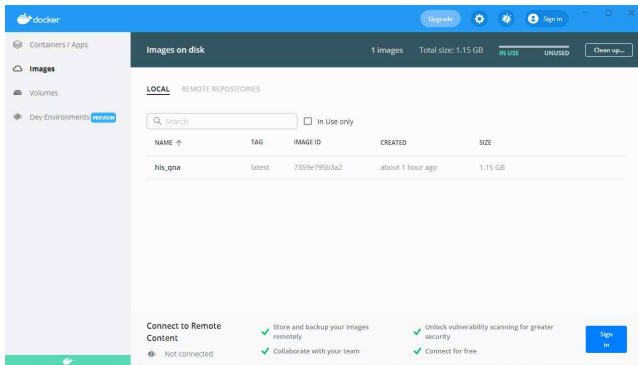
C:\project>docker load -i his_qna.tar
11936051f93b: Loading layer [=====>] 129.1MB/129.1MB
31892cc314cb: Loading layer [=====>] 11.3MB/11.3MB
8bf42cb0de72: Loading layer [=====>] 19.31MB/19.31MB
26a504e63be4: Loading layer [=====>] 156.5MB/156.5MB
c3a0d593ed24: Loading layer [=====>] 537.7MB/537.7MB
aedcb370bc68: Loading layer [=====>] 19.04MB/19.04MB
7e7decd61f68: Loading layer [=====>] 56.93MB/56.93MB
00901a4c6fc7: Loading layer [=====>] 4.608kB/4.608kB
db8d0fe6cf95: Loading layer [=====>] 8.871MB/8.871MB
a0f461773995: Loading layer [=====>] 2.048kB/2.048kB
422c3f26b7cc: Loading layer [=====>] 130.2MB/130.2MB
7ea01d3f06bf: Loading layer [=====>] 15.63MB/15.63MB
1c7e9ca7ee79: Loading layer [=====>] 6.119MB/6.119MB
31113cb3bc76: Loading layer [=====>] 100.4MB/100.4MB
af933de819a: Loading layer [=====>] 2.865MB/2.865MB
Loaded image: his_qna:latest

C:\project>

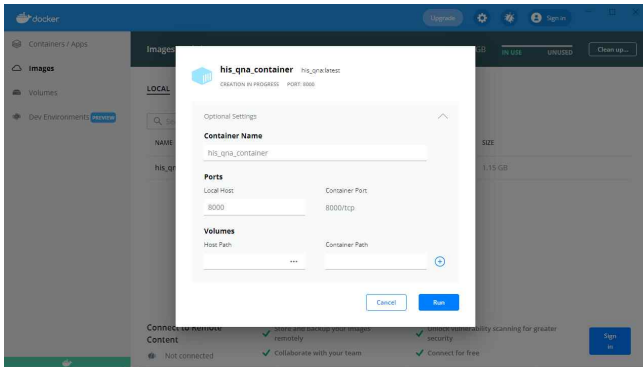
```

08\_다운로드한 tar 파일이 있는 경로로  
이동 후 도커 이미지 로드

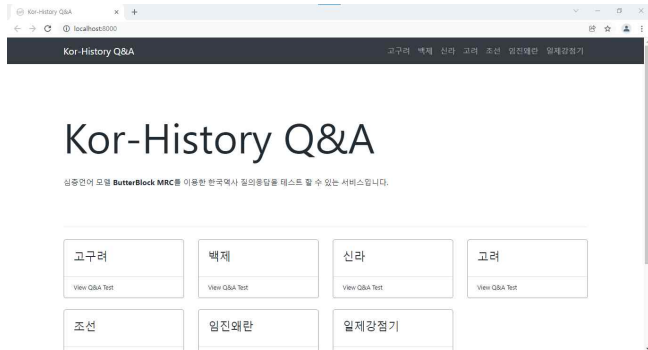
```
docker load -i his_qna.tar
```



## 08\_도커 데스크탑 실행 후 로드한 프로젝트 이미지 확인



09\_컨테이너 이름 및 포트 설정 후  
실행(컨테이너 이름은 자유,  
포트는 8000으로 설정)



10\_브라우저에서 구동 확인  
http://localhost:8000



**Thank You!**