



‘면접 봐Dream’ Chatbot 제작 발표

팀명 : 면접스터디

조원 : 조윤재, 나창대, 김다연, 탁성대, 이현준, 고예진



CONTENTS

1 제작 배경

2 자료 수집

3 서비스 흐름도

4 개발 과정

5 화면 설계

6 모델 설명 및 논문 리뷰

7 코드 설명

8 개발 Tool

9 기대 효과

10 트러블 슈팅

11 개선 방안

12 시연

1. 제작 배경



✓ 배경

- 1) 대다수 취업준비생들이 취업준비활동에 막막함을 느낌
- 2) 채용 프로세스에서 면접 전형의 비중이 증가하는 추세
- 3) 자기소개서 중심의 질문이 면접의 대부분을 차지

→ **면접에 대한 어려움을 해소할 수 있도록**

자기소개서 기반의 면접 예상질문을 제공하여

취업준비생들의 구직활동을 지원하고자

Chatbot '면접 보Dream'의 제작을 기획함.



MZ취준생 94% "취업준비 막막한 적 있어"

윤경진 기자 | 승인 2022.03.23 15:59 | 댓글 0

HOME > 생활 > 뉴스와이드

채용 시 면접이 차지하는 비중? '평균 71%'

주은혜 기자 | 승인 2018.10.25 16:53 | 댓글 0

종 > 정책 > 유관부처

"공들인 직무 중심 자소서, 면접관 마음 사로잡아"

남대일 기자 | 입력 2022.10.12 06:11 | 댓글 0

HOME > 경제 > 증기벤처

인사담당자 70% "챗GPT 등장해도 자소서 여전히 중요"

신승엽 기자 | 승인 2023.06.27 08:33 | 댓글 0

2. 자료 수집

✓ 자기소개서 데이터 수집

취업 관련 웹사이트 Crawling

```
def script_crawling_start():
    text_list = []
    driver = webdriver.Chrome()
    driver.implicitly_wait(3)
    idx = 20000
    text_container_xpath = '/html/body/div[1]/div[1]/div[4]/div/div[2]/div[1]/div[3]/article/main'

    while True:
        try:
            url = f'https://linkareer.com/cover-letter/{idx}?page=1&sort=PASSED_AT&tab=all'
            driver.get(url)

            text_container = driver.find_element(By.XPATH, text_container_xpath)
            text = text_container.text

            text_list.append(text)
            i += 1
        except:
            if idx > 30000:
                print(text_list)
                break
            else:
                idx += 1
    return text_list
```

JOBKOREA LINKareer

csv 파일로 추출

Unnamed: 0		0
0	0	2. 현재 귀하의 가장 부족한 역량은 무엇이며, 역량개발을 위해 어떤 활동과 노력을...
1	1	1. 삼성전자를 지원한 이유와 입사 후 회사에서 이루고 싶은 꿈\n\n삼성의 핵심 ...
2	2	5. 정직함에 대하여. (경험이 있다면 그 상황에서의 본인의 입장 및 대처 사례)\n...
3	3	3. 최근 대형은행의 합병, 인터넷 은행의 등장 등 국내 은행권 경쟁이 심화되고 금융...
4	4	(My Story) 해당회사 및 직무에 지원하는 동기에 대해 기술해 주십시오.\n<G...
-	-	-
5081	5081	의사소통 과정에서 발생하는 본인의 단점 한 가지를 관련 경험과 함께 기술하여 주십시...
5082	5082	1. 보라매병원에 지원하게 된 동기 및 입사 후 실천하고자 하는 목표를 자신의 역량...
5083	5083	1. 우리 공사에 지원하게 된 이유는 무엇인지 기술하고, 본인의 역량 및 강점을 공...
5084	5084	1. 수산식품청년마케터에 지원한 동기를 서술하십시오. 700 바이트 공모전 수상 경...
5085	5085	개인소개 및 지원동기 휴면명조, 13포인트, 줄간격 160% (2페이지 이내 작성)...
5086 rows x 2 columns		

수집한 자기소개서 수: 5,086개

2. 자료 수집



✓ 면접질문 데이터 수집

GPT

지원하게 되었는지 자세히 설명해주세요. 특히, 플랫폼 비즈니스와 콘텐츠기획에 대한 경험을 원하시는 이유가 무엇인지 알려주세요.

2. 도서부에서 활동하며 얻은 경험이 어떻게 (주)이씨오의 사이버도서관, E-도서관 관련 콘텐츠 기획 업무에 도움이 될 것이라고 생각하시나요? 그리고 도서 이벤트를 통해 학생들의 도서관 이용률을 높인 경험이 어떤 결과를 이끌어냈는지 설명해주세요.
3. PPT와 영상 제작에 관심을 갖고 계셨는데, (주)이씨오의 동계현장실습 인턴직을 통해 어떤 역량을 더욱 발전시키고 싶으신가요? 특히, 플랫폼 비즈니스와 콘텐츠기획 업무와 관련하여 어떤 기대를 가지고 있나요?
4. 인턴으로써 기업 내의 업무 처리와 동료들과의 커뮤니케이션에 어떻게 기여하고 싶으신가요? 이를 통해 (주)이씨오에서의 경험과 성장을 어떻게 이루고자 하시나요?
5. 과거 인턴 경험 중, (주)인터파크 비즈마켓에서 B2B 패쇄물 운영MD직무를 수행하며 배운 가치나 교훈이 있으신가요? 그 경험이 (주)이씨오에서의 인턴 생활에 어떻게 적용될 수 있을지 말씀해주세요.
6. 마지막으로, 인턴 직무를 통해 얻은 역량과 지원동기를 바탕으로 (주)이씨오에 기여하고 성장하기 위해 구체적으로 어떤 계획을 가지고 계신가요?

위 질문들을 참고하여 면접을 잘 준비하시고, 자신의 경험과 역량을 잘 어필하시길 바랍니다.
좋은 결과를 기원합니다!

GPT

- Chatbot 학습에 필요한 면접질문 필요
- GPT를 이용하여 자기소개서(독립변수)에 대한 면접질문(종속변수) 생성
- 수집한 자기소개서의 수와 유사한 정도의 질문 데이터를 생성 및 수집

3. 서비스 흐름도



사용자

자기소개서 작성

질문 확인



서버

데이터 전처리,
텍스트 토큰화

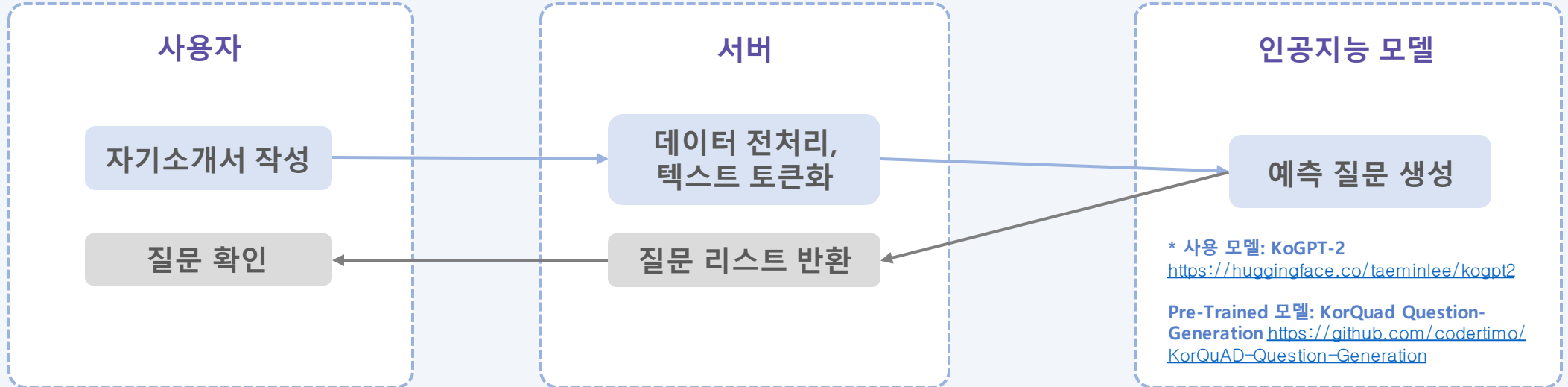
질문 리스트 반환



인공지능 모델

예측 질문 생성

* 사용 모델: KoGPT-2
<https://huggingface.co/taeminlee/kogpt2>
Pre-Trained 모델: KorQuad Question-
Generation <https://github.com/codertimo/KorQuAD-Question-Generation>



5. 화면 설계



✓ 주요 기능 설명



면접 예상질문 리스트 출력

- 기능 요약:
제출한 자기소개서 내용을 기반으로,
예상 면접질문을 사용자에게 보여줌
- 활용 방안:
면접 질문을 빠르게 제공받음과 동시에
면접에 대한 답변을 미리 준비할 수 있음







대화형 모의면접 진행

- 기능 요약:
자기소개서를 기반으로
예상 면접질문을 하나씩 랜덤으로 출력하고,
사용자는 대화형식으로 답변을 하나씩 작성
- 활용 방안:
임의로 출력되는 질문에 대한 답변을
실시간으로 작성할 수 있게 함으로써,
사용자가 면접관과 면접을 진행하는 듯한
형태로 연습할 수 있음

5. 화면 설계

✓ 기능1. 면접 예상질문 리스트 출력

- 좌측 창에 자기소개서 작성 후 분석 버튼을 클릭 시 우측 창에 면접 예상질문 리스트가 출력됨



면접스터디

면접 봐 Dream AI 코칭	
<p>자기소개서</p> <p>자기소개서를 입력해 주세요. 면접 봐 Dream AI가 자소서에서 예상 면접질문을 뽑아드려요! 최소 100자 이상 최대 4000자까지 입력해주세요야 코칭 가능합니다.</p> <p>0 / 4000</p>	<p>면접 질문</p> <p>동료들과 함께 작업할 때 어떤 어려움을 겪으셨나요? 이러한 어려움을 어떻게 극복하셨나요?</p> <p>팀원들과 친해지는 과정에서 어떤 어려움이 있었나요?</p> <p>팀원들과 좋은 관계를 형성하기 위해 어떤 노력을 기울일 계획인가요?</p> <p>목표나 성취하고자 하는 목표가 있다면, 이를 달성하기 위해 어떤 노력을 기울일 계획인가요?</p> <p>자신의 노력을 통해 얻은 교훈과 성취감에 대해 어떻게 생각하시나요?</p> <p>자신의 행동력으로 인해 주변 사람들에게 긍정적인 영향을 끼친 경험이 있나요?</p>

초기화

분석

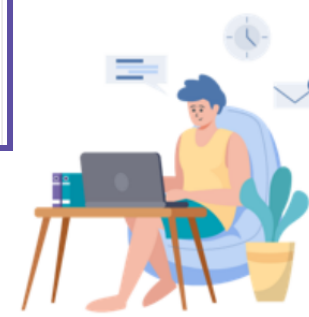

채팅

'분석'버튼 클릭 시
면접 질문 생성

채팅 버튼 클릭 시
모의면접 진행

텍스트 형식
자기소개서
입력 창

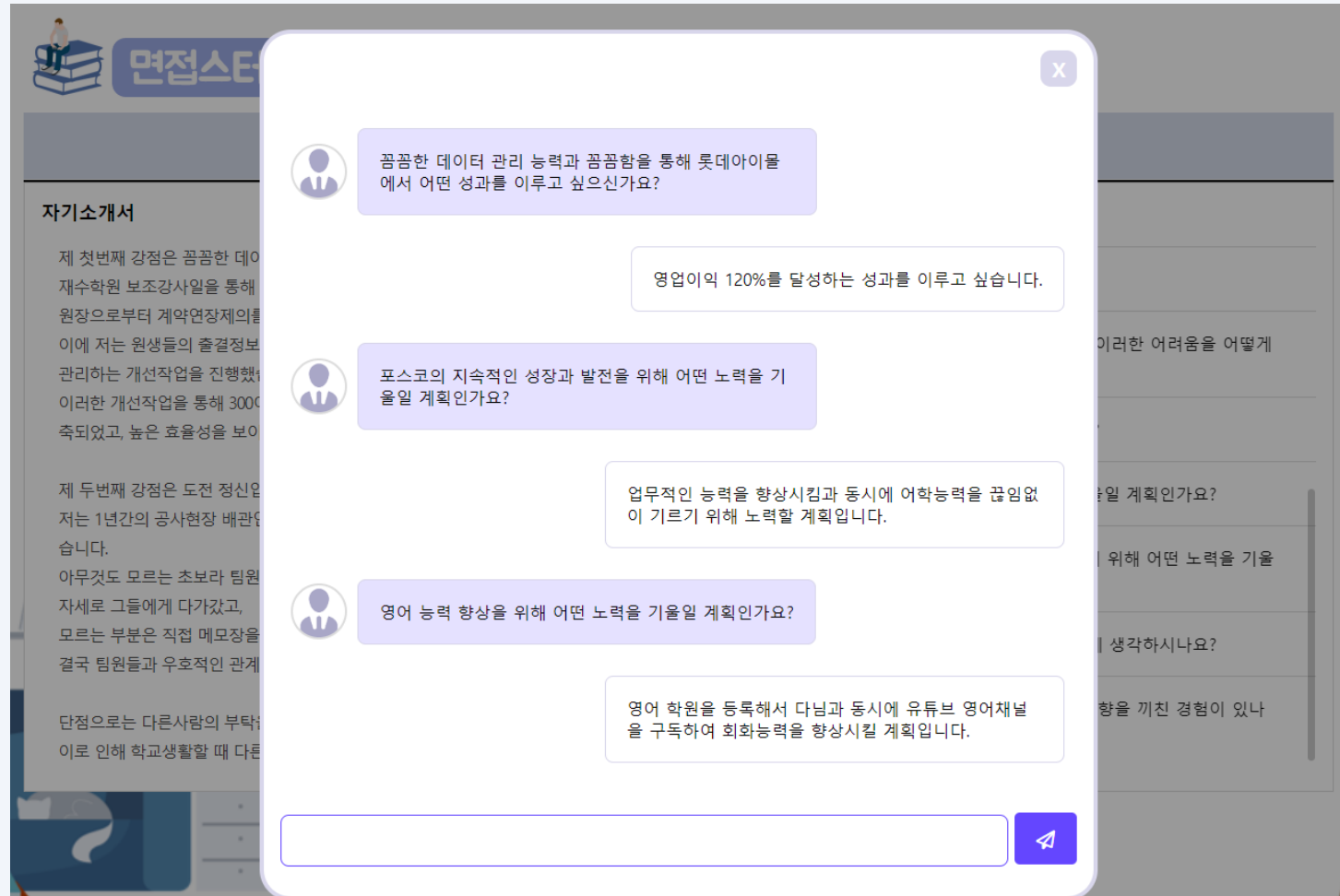
면접 예상 질문
리스트 출력



5. 화면 설계

✓ 기능2. 대화형 모의면접 진행

- 모의면접 모달창에서 대화형식으로 임의의 질문 및 꼬리 질문에 대한 답변을 작성



6. 모델 설명

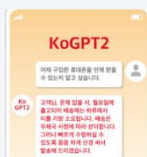


GPT-2

OpenAI/GPT-2

프로젝트의 기반이 되는
학습 모델은
OpenAI GPT-2로,
2018년 GPT-1에 이어
2019년에 발표된
비지도학습 기반의
언어모델.

<https://github.com/openai/gpt-2>



SKT-AI/KoGPT-2

KoGPT-2는
GPT-2의 한국어 성능한계
를 보완할 목적으로
SKT-AI에서 개발된
GPT-2의 한국어 버전 모델

https://github.com/SKT-AI/KoGPT2/tree/pp1_eval



Hugging Face

KorQuAD 1.0

KorQuAD-
Question-Generation

**KorQuAD-
Question-Generation**은
HuggingFace에 배포된
KoGPT-2 모델을 기반으로
대규모 한국어 질의응답
데이터셋 **KorQuAD v1.0**을
사용하여 제작된 질문생성
(Question Generation)
모델

<https://huggingface.co/taeminlee/kogpt2>

<https://github.com/codertimo/KorQuAD-Question-Generation>



면접 봐 Dream

모델에 Pre-traine된
KorQuAD v1.0 데이터셋에
자체적으로 수집한
5,000여 개의 자기소개서,
면접질문의 데이터를
추가로 학습하여
해당 모델을
Fine-tuning하는
작업을 거침

6. GPT-2 논문 리뷰

1. 정보

1. 논문 제목

OpenAI GPT-2 - Language Models are Unsupervised Multitask Learners

2. 논문 게재 일자

2019-05-16(v1)

3. 저자

OpenAI 연구진

(Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever 등)

4. Published journal

arXiv(아카시브)

→ 물리학, 수학, 컴퓨터 과학 등의 분야에서 선행 연구를 공유하고 논문을 업로드하는 서비스를 제공하는 웹 사이트

Language Models are Unsupervised Multitask Learners

Alec Radford^{*1} Jeffrey Wu^{*1} Rewon Child¹ David Luan¹ Dario Amodei^{**1} Ilya Sutskever^{**1}

Abstract

Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on task-specific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText. When conditioned on a document plus questions, the answers generated by the language model reach 55 F1 on the CoQA dataset - matching or exceeding the performance of 3 out of 4 baseline systems without using the 127,000+ training examples. The capacity of the language model is essential to the success of zero-shot task transfer and in-

competent generalists. We would like to move towards more general systems which can perform many tasks - eventually without the need to manually create and label a training dataset for each one.

The dominant approach to creating ML systems is to collect a dataset of training examples demonstrating correct behavior for a desired task, train a system to imitate these behaviors, and then test its performance on independent and identically distributed (IID) held-out examples. This has served well to make progress on narrow experts. But the often erratic behavior of captioning models (Lake et al., 2017), reading comprehension systems (Jia & Liang, 2017), and image classifiers (Alcorn et al., 2018) on the diversity and variety of possible inputs highlights some of the shortcomings of this approach.

Our intuition is that the success of single-task training

6. GPT-2 논문 리뷰

2. 소개

1. 배경

- 연구 동향 : 현재 기계학습체계를 개발하는 주된 방법은 목표 과제에 맞는 데이터셋을 찾아서, 이를 학습/검증 단계로 나누어 학습 후 성능을 측정하는 방법
- 문제점 :
 - 위의 방법으로 개발된 모델들은 불안정하며 좁은 범위의 문제에서만 뛰어난 능력을 발휘하고, 범용적인 이해를 필요로 하는 독해나 이미지 분류시스템의 문제에서는 성능이 떨어짐.
 - 많은 연구가 단일 영역의 데이터셋과 단일 과제에만 맞춘 학습에만 치중되어있음



2. 목적

- 연구 필요성 :
 - 다중작업 학습(Multitask learning)은 일반 성능을 높이는 유망한 방법이지만 아직 초기 연구 단계임
 - 다중작업 학습을 위해서는 일반화를 위해 수십~수백만개의 많은 샘플이 필요
 - 성능이 높은 언어처리모델은 pre-training과 fine-tuning의 결합으로 만들어짐
- 연구 방향 :
 - 비지도 학습을 이용하여 다중작업 학습에서 높은 성능을 발휘하는 언어 모델이 필요

6. GPT-2 논문 리뷰

2. 소개

3. 주요 용어 정의

- Zero-shot :

기계 학습 모델에서 특정 작업을 학습하지 않고도 해당 작업에 대해 일반화하여 수행하는 능력을 의미함. 다국어 데이터를 기반으로 학습한 후 레이블이 없는 언어로도 번역/이해를 수행할 수 있는 경우가 해당됨.

- Zero-shot task transfer :

- 기계 학습에서 하나의 학습된 모델을 다른 관련 작업에 전이(transfer)하여 사용하는 기법.
- 특정 작업에 대해 새로운 데이터셋을 수집하고 모델을 재학습하는 번거로운 과정을 생략할 수 있어 효율적임.

- 다중 작업 학습(Multitask learning) :

기존의 단일작업 학습은 1개의 모델을 1개의 작업에만 특화하여 학습시키는 반면, 다중작업 학습은 여러 작업에 대한 정보를 모델에 공유하여 모델의 일반화 능력과 학습 효율성을 개선.

- Downstream task:

- 자연어 처리 분야에서 사전 학습된 언어 모델을 특정 실제 작업에 적용하는 것.
 - main task와 하위 작업들로 구성되는데 예를 들어, '감정분석'이라는 main task와 '문장레벨 감성분석' 등 여러가지의 하위 작업들로 구성될 수 있음.
-

6. GPT-2 논문 리뷰

3. 접근

1. Training Dataset

- 데이터 수집 방법 :
 - 단순 크롤링이 아닌, 고품질의 데이터를 얻는 방법을 사용
 - **사람에 의해 필터링된 글만을 사용** *Reddit에서 3karma 이상을 받은 글에 포함된 외부링크의 글 등*
 - 2017/12 이후의 문서, 위키피디아 문서, 중복 문서의 제거 등 전처리 과정을 거쳐 800만 개의 문서, 40GB의 텍스트 확보
 - 데이터 수집에 사용한 알고리즘 :
 - **Dragnet** *웹페이지에서 본문(content)을 추출하기 위해 개발된 알고리즘*
 - Newspaper 내용 추출기
 - 수집된 데이터셋의 이름은 **WebText**로 명명
-

6. GPT-2 논문 리뷰

3. 접근

2. Input Representation

- GPT-2 모델의 입력 데이터 형식:
 - 텍스트 데이터를 **BPE Tokenizer**를 통해 **Byte 단위로 분리하여 인코딩**한 후 입력으로 사용.
 - 단어수준 LM의 '경험'과 문자수준 LM의 '일반성'이라는 장점을 결합함.
 - 어떤 Unicode 문자열에서나 확률을 부여할 수 있음.
 - 전처리, 토큰화, vocab의 크기 등과 관련없이 어떤 데이터셋에서도 LM의 사용이 가능.
 - Tokenizer: **Subword 단위로 분리**하는 **BPE Tokenizer** 사용
 - 자주 출현하는 단어수준 입력과 자주 출현하지 않는 글자수준 입력에 대한 **symbol sequence**들을 적절히 조합하여 새로운 입력을 생성하는 과정을 의미.
-

6. GPT-2 논문 리뷰

3. 접근

3. Model

- 모델 구조:
 - **Transformer**가 기본 구조이며, **GPT-1** 모델 구조와 유사함.
 - GPT-1 모델과의 차이점:
 - 1) **Layer 정규화를 각 sub-block의 입력으로 옮김** GPT-2 모델은 sub-block이라 불리는 작은 모듈을 여러 번 쌓아올린 구조.
 - 각 Layer의 입력 데이터에 대해 평균과 표준편차를 계산하여 정규화를 수행함으로써
입력 데이터의 분포가 안정화되고, 학습이 더 잘 진행되며, gradient 소실 및 폭주 문제를 완화하는데 도움이 됨.
 - 2) 추가적인 Layer 정규화는 **Self-Attention 레이어 이후에 적용됨**.
 - 각 단어 간의 상호작용이 더 잘 이루어지게하고, 전체 모델의 학습과 성능을 개선하는데 도움을 줌.
 - 3) Vocab의 크기가 **50,257개**로 확장됨.
 - 4) 입력으로 주어지는 context size에 대한 token의 수가 **512~1024개**로, batch size가 **512**로 증가함.
-

6. GPT-2 논문 리뷰

4. 실험 설계 및 결과

1. 설계

- 모델 설계: 크기가 각각 다른 4개의 모델을 만듦

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

= GPT-1

= BERT

= GPT-2

- 학습/검증 데이터 분할: WebText 데이터의 5%를 held-out 샘플(validation 샘플)로 분리
→ 적절한 learning rate를 찾아가며 조정
- 평가 방법:
 - WebText 언어모델에 따른 dataset의 로그확률을 계산하는 방식으로 통일

6. GPT-2 논문 리뷰

4. 실험 설계 및 결과

2. 평가 결과

- 평가 데이터: 8종의 LM benchmark를 사용

LM benchmark: 언어 모델의 성능을 평가하기 위한 여러 개의 텍스트 데이터셋

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	21.8
117M	35.13	45.99	87.65	83.4	29.41	65.85	1.16	1.17	37.50	75.20
345M	15.60	55.48	92.35	87.1	22.76	47.33	1.01	1.06	26.37	55.72
762M	10.87	60.12	93.45	88.0	19.93	40.31	0.97	1.02	22.05	44.575
1542M	8.63	63.24	93.30	89.05	18.34	35.76	0.93	0.98	17.48	42.16

미세조정 없이
zero-shot 환경 아래
8개 중 7개 benchmark에서
SOTA 성능 달성

LM benchmark	특징
LAMBADA	텍스트의 장거리 의존성 평가
Children’s Book Test	품사에 따른 LM의 성능 측정
WikiText2	위키피디아에서 추출한 텍스트 데이터셋
PTB	Penn Treebank에서 추출한 텍스트 데이터로 구성된 데이터셋
enwik8	2008년 기준으로 영어 위키피디아의 많은 부분을 텍스트로 포함하며, 데이터 크기가 매우 큼
text8	영어 위키피디아의 일부 텍스트 데이터로 구성
WikiText103	위키피디아에서 추출한 텍스트 데이터셋으로, WikiText2보다 규모가 더 큼
1BW	잘못된 텍스트에 대해 모델이 올바른 텍스트를 구분하는 능력을 평가하는데 사용됨

6. GPT-2 논문 리뷰

4. 실험 설계 및 결과

2. 평가 결과

- 자연어처리 작업기준

자연어처리 작업(Task)	데이터셋 특징	평가 결과
Reading Comprehension (독해)	독해능력과 대화에 기반한 답변능력 평가	지도학습 없이도 F1 score 55점을 달성. (SOTA인 BERT는 F1 score 89점)
Summerization (요약)	요약능력을 평가하기 위해 CNN과 Daily Mail 데이터셋을 사용	특정 조건(TL; DR)을 추가했을 경우, 추가하지 않았을 때보다 평균 6.4점 향상
Translation (번역)	다른 언어로 번역하는 방법을 학습했는지 테스트	11.5 BLEU의 고성능을 달성.(SOTA Model은 33.5 BLEU) <small>BLEU: 기계 번역 결과와 사람이 번역한 결과의 유사도를 비교하여 번역에 대한 성능을 측정하는 방법</small>
Question Answering (질의응답)	질문에 올바른 답변을 생성하는 빈도를 평가	"정확일치(Exact Match)" 메트릭으로 평가 시 4.1%, 가장 확신하는 1%의 질문에는 63.1%의 정확도를 가짐.

6. GPT-2 논문 리뷰

4. 실험 설계 및 결과

2. 평가 결과

- 모델의 일반화 능력과 과적합 여부 평가(Generalization vs Memorization)

- 평가 배경:

- 1) 최근 연구에 따르면 일반적인 이미지 데이터셋에는 **train과 test셋에서 많은 양의 중복된 이미지가 포함되어 과적합을 유발함.**
- 2) 데이터 세트의 크기가 증가함에 따라 이 문제의 가능성이 높아져 **WebText에서도 유사한 현상이 발생할 수 있음.**
- 3) 따라서 **학습 데이터에 얼마나 많은 테스트 데이터가 나타나는지 분석**하는 것이 중요함.

- 평가 방법:

블룸 필터를 사용하여 주어진 데이터셋에서 WebText 학습 데이터셋에 있는 8-gram의 백분율을 계산

- 평가 결과:

WebText 학습 데이터셋에서 일반적인 LM 데이터셋의 **test set은 1~6%의 중복을 가짐.** 타 데이터셋에서는 평균적으로 3.2%의 중복을 가짐.

	PTB	WikiText-2	enwik8	text8	Wikitext-103	1BW
Dataset train	2.67%	0.66%	7.50%	2.34%	9.09%	13.19%
WebText train	0.88%	1.63%	6.31%	3.94%	2.42%	3.75%

- 개선 방법: train/validation 데이터를 분할하는 동안 n-gram 중복 기반 중복 제거를 사용하는 것이 좋음.

6. GPT-2 논문 리뷰

5. 토론 및 결론

1. 토론

- 1) 비지도 학습은 연구의 가능성이 큰 분야임.
- 2) GPT-2 모델의 zero-shot 학습 성능은 독해에서는 좋은 모습을 보이나, 질의응답과 번역 등의 작업들에서는 충분한 용량을 갖추지 않는 한 기본적인 성능의 기준에 미달함.
- 3) zero-shot 성능 또한 잠재력이 크지만, fine-tuning 없이는 여전히 불명확한 성능을 가짐.
- 4) 추가적으로 decaNLP와 GLUE와 같은 벤치마크에서 미세 조정을 연구할 계획이 있음.

2. 결론

- 1) GPT-2는 실험 내 8개의 언어 모델링 데이터셋 중 7개에서 목표를 달성.
- 2) 크고 다양한 데이터셋에서 대규모 언어 모델을 훈련시킨다면 다양한 도메인과 데이터셋에서 원활한 업무의 수행이 가능함.
- 3) 수많은 말뭉치를 포함하면서 Pre-trained된 고용량의 모델이 비지도적으로 작업수행을 학습한다는 것을 확인할 수 있었음.

6. GPT-2 논문 리뷰

5. 토론 및 결론

3. 느낀점

- 1) 자연어 처리 모델을 평가하는 다양한 방법 및 관련된 데이터셋들에 대해 자세히 알 수 있는 기회가 되었음.
 - 2) GPT2 모델을 기반으로 fine-tuning한 모델들을 참고하여 구현하면서 왜 뉴스기사의 형식으로 나오는 예측 결과들이 존재했는지와 학습 데이터가 왜 Unicode의 형식이었는지를 알 수 있었음.
 - 3) GPT-2 모델은 추가적으로 연구가 필요함을 결론에서 확인할 수 있었는데, 현재 시점에서 더 좋은 성능인 'GPT-4' 모델의 Technical Report도 분석해서 비교해보는것 또한 효과적이라고 느낌.
-

7. 코드 분석

✓ 토큰나이징 (SentencePieceBPETokenizer)

vocab.json

```
"_이": 104,  
"_있": 105,  
"_1": 106,  
"_으로": 107,  
"_대": 108,  
"_에서": 109,  
"_\"\": 110,  
"_기": 111,  
"_2": 112,  
"_지": 113,
```

merges.txt

```
주 는  
_대 상으로  
_대상 으로  
_ 대상으로  
_학 교  
_ 학교  
_사 회  
_인 트
```

```
tokenizer = SentencePieceBPETokenizer.from_file(  
    vocab_filename=config.vocab_path,  
    merges_filename=config.tokenizer_merges_path,  
    add_prefix_space=False  
)
```

SentencePiece

- 비지도학습을 위하여 Subword로 분리
- vocab.json 파일 기반으로 분리

BPE

- 분리된 Subword를 결합
- merges.txt 파일 기반으로 결합

7. 코드 분석

✓ 데이터셋

```
train_dataset = QGDataset(  
    train_examples,  
    tokenizer,  
    config.max_sequence_length)
```

```
context_tokens = self.tokenizer.encode(f"문맥:{example.context}").ids  
answer_tokens = self.tokenizer.encode(f"정답:{example.answer}").ids  
question_tokens = self.tokenizer.encode(f"{example.question}").ids
```

```
conditional_tokens = ([self.sos_token] + context_tokens  
    + answer_tokens + self.question_prefix_tokens)  
post_tokens = question_tokens + [self.eos_token]  
input_ids = conditional_tokens + post_tokens  
labels = input_ids if self.is_train else ([-100] * len(conditional_tokens)) + post_tokens  
attention_mask = [1.0] * len(input_ids)  
  
return input_ids, attention_mask, labels
```

- DataSet을 상속받아 클래스를 선언
- context와 answer, question을 각각 토큰화
- 토큰화한 값을 하나로 연결
- question을 제외하고는 label에서 무시하기 위해 -100으로 세팅
- 문장 전체에 어텐션을 적용하기 위해 1로 세팅

7. 코드 분석

✓ 모델 세팅

```
model = GPT2LMHeadModel.from_pretrained(config.gpt_model_hub_name)

optimizer = Adam(model.parameters(), lr=config.lr)

total_steps = len(train_dataloader) * config.epochs

warmup_steps = int(total_steps * config.warmup_ratio)

scheduler = get_linear_schedule_with_warmup(optimizer,
                                             warmup_steps,
                                             total_steps)
```

- 모델은 GPT2기반 모델을 사용
- optimizer는 Adam을 사용
- `get_linear_schedule_with_warmup()`을 사용하여 학습률을 점점 증가

7. 코드 분석

✓ 모델 학습

```
for epoch_id in range(config.epochs):
    for step_index, batch_data in enumerate(train_dataloader):
        optimizer.zero_grad()

        input_ids, attention_mask, labels = tuple(value.to(device)
                                                    for value in batch_data)
        model_outputs = model.forward(
            input_ids,
            attention_mask=attention_mask,
            labels=labels, return_dict=True)
        model_outputs.loss.backward()

        torch.nn.utils.clip_grad_norm_(model.parameters(), config.grad_clip)

    optimizer.step()
    scheduler.step()
```

- dataloader를 parameter별로 나누어 입력

- clip_grad_norm_()을 이용하여 gradient 폭주 예방

7. 코드 분석

✓ 예측

```
decoded_sequences = model.generate(  
    input_ids=input_ids,  
    max_length=origin_seq_len + 150,  
    min_length=origin_seq_len + 5,  
    pad_token_id=0,  
    bos_token_id=1,  
    eos_token_id=2,  
    num_beams=num_beams,  
    repetition_penalty=1.3,  
    no_repeat_ngram_size=3,  
    num_return_sequences=1,  
)  
  
for decoded_tokens in decoded_sequences.tolist():  
    decoded_question_text = tokenizer.decode(decoded_tokens[origin_seq_len:])  
    decoded_question_text = decoded_question_text.split("</s>")[1].replace("</s>", "")  
    decoded_question_text = decoded_question_text.split("질문:")[-1]  
    generated_results.append(decoded_question_text)
```

- generate를 이용하여 입력값을 넣고 시퀀스를 반환

- 반환 받은 값에서 question에 해당하는 문자열만 슬라이싱

8. 개발 Tool



HTML



CSS



JS



Frontend



 FastAPI



Hugging Face



OpenAI

AI & Backend

 PyCharm



GitHub



git



slack

Tool

9. 기대 효과



✓ 기대효과 1. 면접에 대한 취업준비생 부담완화

- 모의 면접 기능을 통해 지원자의 면접에 대한 압박과 부담감을 일정 부분 해소할 수 있음

✓ 기대효과 2. 기업 HR팀의 채용 프로세싱 개선 가능

- 면접 질문을 생성하는 과정에서 필요한 자기소개서의 구조화 과정을 AI가 대신 해줌으로써 빠르게 면접 질문을 추출할 수 있으며, 이를 활용하여 기업 HR팀에서 지원자의 채용 프로세싱의 개선이 가능

✓ 기대효과 3. 다양한 사회활동에 응용 가능

- 기업에 취업하기 희망하는 지원자 이외에도, 아르바이트생 및 대학원생 선발 과정 등 다양한 인적자원의 선발 과정에 응용할 수 있음.

시연



10. 트러블 슈팅



문제점

데이터셋 KorQuAD v1.0으로만 Pre-trained된
학습 모델을 기반으로 자기소개서를
입력하였을 경우,
KorQuAD 데이터셋 자체적으로
Pre-trained된 결과물(뉴스기사, 위키백과)이
출력되어 **정확한 면접질문이 생성되지 않음.**



해결 방안

KorQuAD v1.0 데이터셋에
5,000여 개의 자기소개서-면접질문
데이터를 확보한 후
모델에 추가로 학습시켜
어느정도의 구현이 가능한 성능으로
향상시킴

11. 개선 방안



한계점

1. 원활한 면접질문의 생성에 필요한
입력될 자기소개서의 문맥 고려 필요
2. 자기소개서와 그에 대한 면접질문의
수집에 대한 한계
3. 자기소개서 입력 시 맞춤형/대화형에
대한 고성능 질문 생성의 한계



개선 방안

현재 보유량보다 더 많은 데이터를 확보하여
학습 모델의 성능 향상이 필요
→ 추가적인 자기소개서 및 면접질문의
Crawling이 가능한 취업플랫폼의 집중 검색
(네이버 취업카페, 자소서닷컴 등)

프로젝트의 기반 모델은 GPT-2 모델⁽²⁰¹⁹⁾
최근 시점의 모델인 GPT-4⁽²⁰²³⁾와
성능의 격차가 있을 것이라 예상함
→ 최근 시점의 모델을 활용하는 방안 모색

담당 업무 및 소감



고예진 (데이터 수집 및 전처리, 학습모델 분석)

자소서는 사생활, 개인정보가 속한 구하기 힘든 데이터였습니다. 그래서 수작업이 많이 필요한 작업이었지만 데이터에 비해 생각 이상으로 좋게 나와 만족스러웠고 학습 시킨 모델이 면접때 많이하는 꼬리질문이나, 회사마다 주관적인 질문같이 심화적인 것 까지 하기엔 어려운 상황이라 아쉬움이 조금 남았지만 생각 이상으로 좋게 나와 만족스러웠습니다.

김다연 (Client HTML/CSS 기능 구현, 학습모델 분석 및 조정)

자연어 처리 분야가 정말 어렵게만 느껴졌는데, 프로젝트를 진행하면서 모델을 직접 찾아보고 적용하는 과정을 통해 많이 배울 수 있었습니다. 특히 팀원분들이 각각 맡은 부분에 대해 자세히 공유하고 설명해주신 덕분에 제가 맡은 부분 이외의 부분들도 이해할 수 있어서 좋았습니다.

나창대 (데이터 수집 및 전처리, 학습모델 구축 및 조정)

자소서를 입력 받고 그에 대한 질문을 생성해주는 일을 쉽게 생각하고 처음에 시작 하였으나, 생각보다 고려사항이 많이 존재 하였고 글의 문맥, 흐름을 컴퓨터에 학습 시키는것에 대해서 아직도 고민중이며, 방법을 찾아 내지 못한것에 대해 아쉬움을 느꼈습니다.

이현준 (Client HTML/CSS 기능 구현, 학습모델 구축 및 조정)

AI분야 처음 접해보아 코드를 분석하고 구현하는 것에 많은 어려움이 있었습니다. 하지만 팀원들과 함께 공부하고 정보를 공유하며 프로젝트를 진행한 덕분에 딥러닝과 자연어처리를 많이 이해 할 수 있었습니다.

조윤재 (데이터 수집 및 전처리, 학습모델 분석, Server 배포 작업)

처음으로 맡은 프로젝트 팀장이었는데, 부족한 부분이 많았음에도 끝까지 도와준 나머지 5명의 팀원들에게 너무 감사하다는 말을 하고 싶습니다. '챗봇'이라는 개념에 대해 막연히만 알고있었는데 이번 프로젝트를 통해 어느정도 챗봇에 대해 알 수 있는 시간을 가졌습니다.

탁성대 (Server 기능 구현, Client-Server 연결 구현, 학습모델 조정)

모든 팀원들이 이렇게 열심히 해주었던 팀이 앞으로도 있을까? 생각이 들 정도로 너무 열심히 해주시고 어떤 문제가 발생하였을때 모두가 다 관심을 가지고 의견을 나누고 해결해 나아가는 과정이 너무 즐거웠습니다. 데이터를 수집 하면서 막혔던 부분을 해결하며 이걸 완성할 수 있을지에 대해서도 의문이었는데 모두가 다 너무 열심히 해주셔서 예상보다 훨씬 더 훌륭한 결과를 맺을수 있어 저희 팀원들에게 너무 감사드립니다.

감사합니다!

