

The background image is a high-action scene from a Transformers movie. It depicts Optimus Prime, the leader of the Autobots, in his iconic blue and red armor, standing amidst a chaotic battle. He is surrounded by other Autobots and is engaged in combat with Megatron, the leader of the Decepticons, who is shown in his dark, menacing form. The scene is set against a backdrop of a city being destroyed, with smoke, fire, and debris filling the air. The lighting is dramatic, with bright flashes from explosions and the sun low on the horizon, creating a sense of urgency and scale.

Into the GPT world !

TransFormer

김민지 이화림 국동훈 김해민

목차

01

아이디어
개발환경

02

주제 선정
데이터수집

03

개념설명

04

코드설명

05

최종결과물

01.아이디어



트랜스포머 기반 언어모델

GPT-2 한국어 버전인

ko-GTP2 사용



x개의 문장을 대입했을 때

논리적인 문장 도출하는 모델



Q -> A 서비스를 만들어보자 !

너무 글 잘쓰는 AI... 깜짝 놀란 개발자들, 비공개 결정

📖 A16면 1단 | 기사입력 2019.02.20. 오전 3:09 기사원문 스크랩 🌐 본문듣기 · 설정

😊👍 45 💬 38

요약본^{Beta} 가 📄

소설·뉴스·학교 숙제 척척

논리적으로 문장 구성하고 새로운 문장 탁월하게 작성

미국의 비영리 인공지능(AI) 연구기관 '오픈 AI(Open AI)'가 새로 개발한 '글짓기 인공지능'의 글쓰기 실력이 너무 뛰어나 연구자들이 '악용이 우려된다'는 이유로 원천 기술을 비공개하기로 결정했다. 오픈 AI는 테슬라 CEO 일론 머스크 등 미국 IT 기업 대표들이 2015년 '인공지능을 통해 인류에 기여하자'며 공동으로 설립한 비영리 기관이다. 그동안 모든 연구 결과와 특허를 무료로 공유해왔지만 처음으로 비공개를 결정한 것이다.

01. 개발환경



GPT2



KoGPT2

01. 개발환경



Markup Language
Content



Style sheet Language
Presentation



P Y T O R C H



02.주제설명

nate 판

NAVER 지식iN



연애, 결혼, 생활 고민상담

청소년 고민상담

가족 고민상담

고민상담

자격증

레시피

NAVER 지식iN

컴퓨터, 전기 분야 자격증

요리 레시피

02. 데이터수집

← → ↻ kin.naver.com/userinfo/answerList.nhn?u=%2BtGfCq9V7ttCQMaerGI3Uki53svVwaY1w%2BB6%2BDsu6A%3D&isSearch=true&query=자격증&sd=answer&y=0§ion=qna&isWorry=false&x=0&page=2

Q&A	60
61	
디렉토리별 보기 ▾ 연도별 보기 ▾	62
63	
제목	64
65	
전기산업기사책질문	66
67	
전기 산업기사 책 인강질문	68
69	
데브닌하고 노턴의 법칙 문제 알려주세...	45
46	
전기기능사	47
48	
전기기사 푸는식좀 알려주세요 🖼	48
49	
전기산업기사 필기 기출질문	50
51	
회로이론 문제 전기/전자산업기사 질문...	51

```
driver.get(url)
time.sleep(1)

search_res = driver.find_element_by_xpath(xpath)

search_res.click()
time.sleep(1)

driver.switch_to_window(driver.window_handles[1])
time.sleep(1)

url = f'https://kin.naver.com/userinfo/answerList.nhn?u=w6lLUADsTiE2\
WDOOrNVtf1Qxgc3ft9bDXpkXY1Mua2f4%3D&isSearch=true&query\
=%EC%9E%90%EA%B2%A9%EC%A6%9D&sd=answer&y=0&section=qna&isWorry=\
false&x=0&page={page}'

# xml path language // local path or absolute path?
xpath = f'//*[@id="au_board_list"]/tr[{num}]/td[1]/a'
```


02. 데이터수집



100

전기 자격증

제가 현재 전기 관련고등학교를 다니고 있는데 자격증을 어떤걸 따는게 나중에 취업할때 도움이 될까요? 전기기능사와 전자기기 기능사를 이미 따 놓은 상태예요

≡ 전기, 전자 공학

```
53 | question_selector = '#content > div.question-content > div > \
54 |   div.c-heading._questionContentsArea.c-heading--default-old > \
55 |   div.c-heading__title > div > div.title'
56 | question_detail_selector = '#content > div.question-content > div\
57 |   > div.c-heading._questionContentsArea.c-heading--default-old >\
58 |   div.c-heading__content'
```

```
71 | try :
72 |     question = driver.find_element_by_css_selector(question_selector).text
73 | except :
74 |     pass
```


02. 데이터수집



무꿈 님 답변

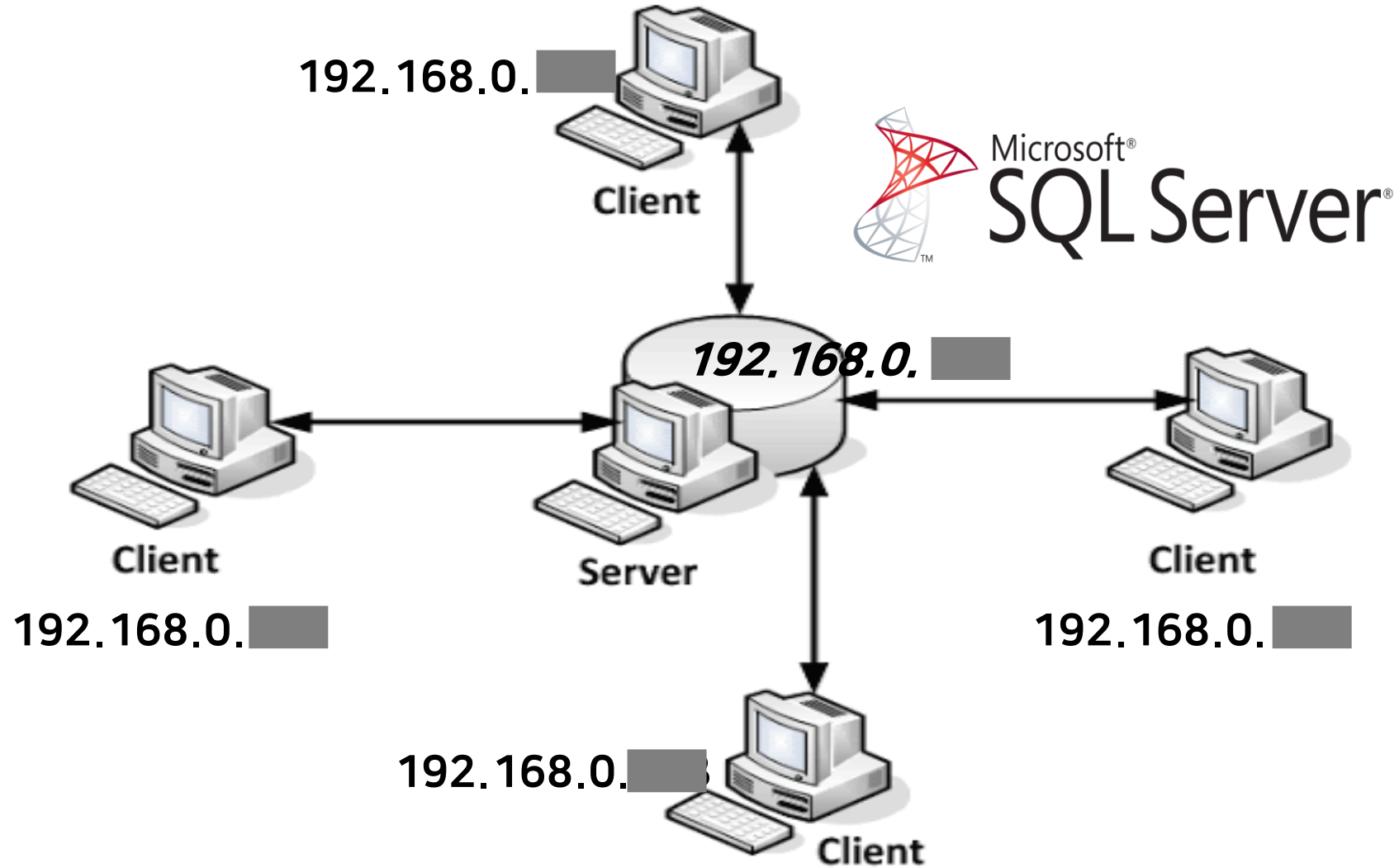
수호신 · 전기공사기사 · 채택답변수 32,129 · 받은감사수 29 · 전기, 전자 공학3위, 컴퓨터 자격증, 학원2위, ...

제가 현재 전기 관련고등학교를 다니고 있는데 자격증을 어떤걸 따는게 나중에 취업할때 도움이 될까요? 전기기능사와 전자기기 기능사를 이미 따 놓은 상태예요

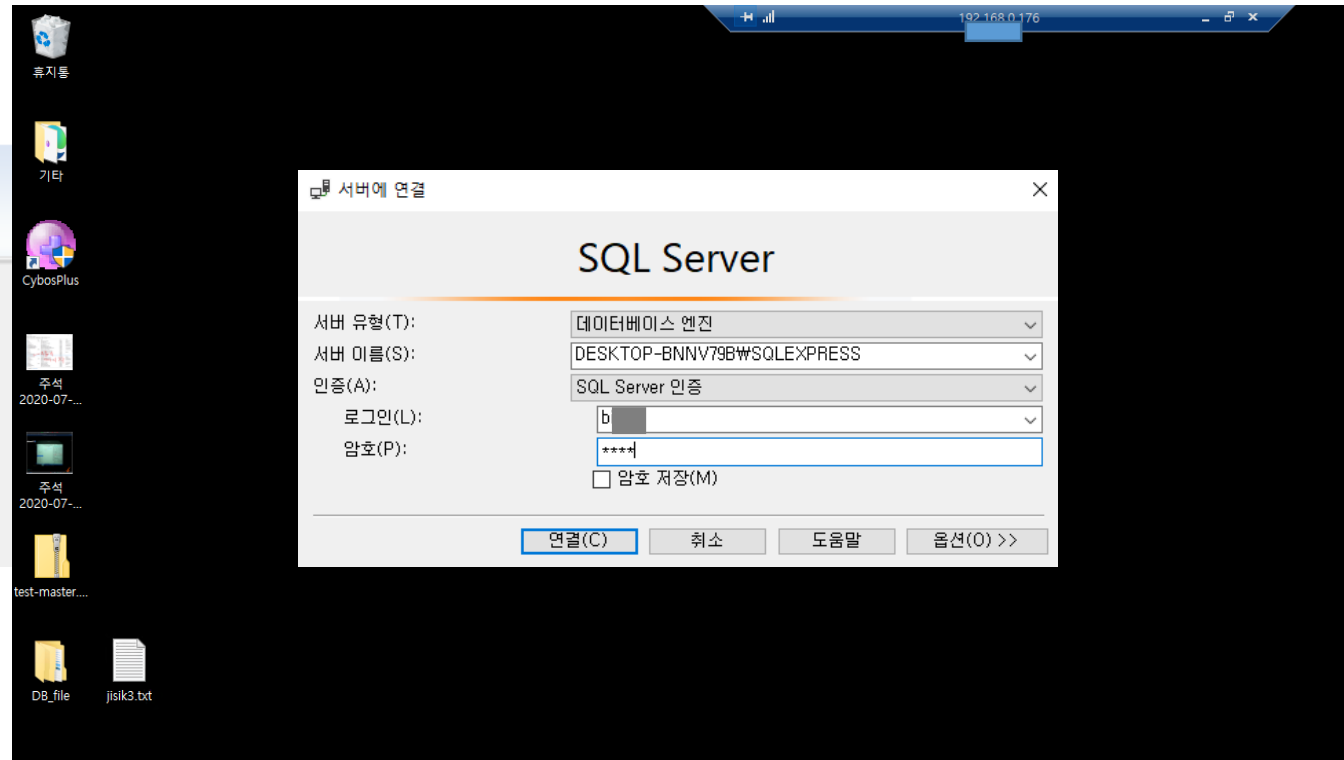
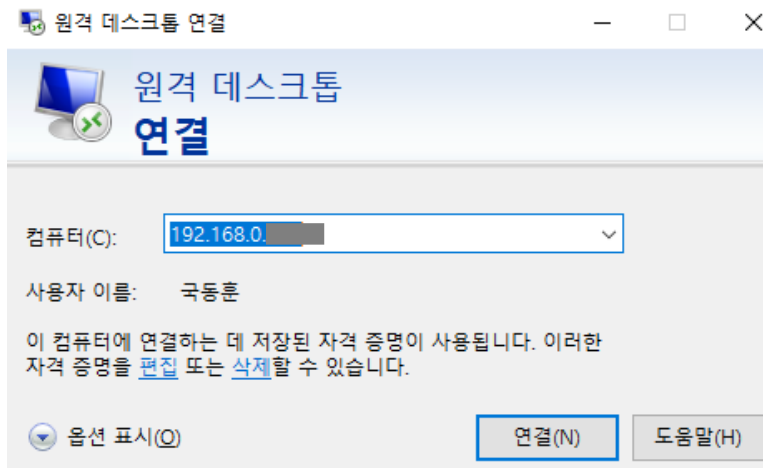
답변 - 전기기능사와 같이 취득하시면 좋은 자격증은 승강기 기능사도 취업에 유리합니다.
승강기기능사 취득을 권합니다.

```
99      while True:
100          answer_writer_selector = f'#answer_{answer_num} > div.c-heading-answer \
101          > div.c-heading-answer__body > div.c-heading-answer__title > p'
102          answer_detail_selector = f'#answer_{answer_num} > \
103          div._endContents.c-heading-answer__content'
```

03. 데이터수집



03. 데이터수집



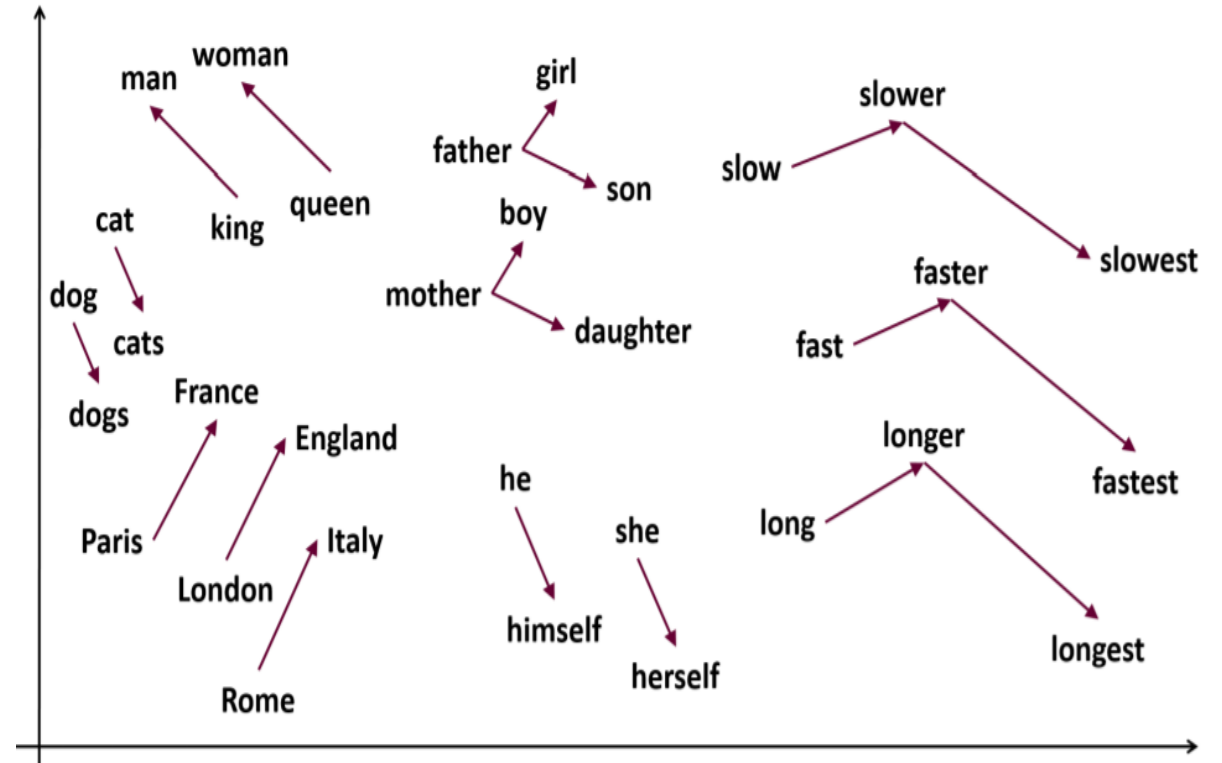
02. 데이터수집

```
14 def create_table(tablename):
15     # conn = ms.connect(server='192.168.0.███', user='███', password='███', database='███')
16     conn = ms.connect(server='127.0.0.1', user='███', password='███', database='███')
17     cursor = conn.cursor()
18     sql = f"IF NOT EXISTS (SELECT * FROM sysobjects WHERE name='{tablename}' AND xtype='U')\
19         CREATE TABLE {tablename} (id int identity, que text null, que_detail text null,\
20         ans_writer text null, ans_detail text null)"
21     cursor.execute(sql)
22     conn.commit()
23     conn.close()

5 def dbconn(tablename, insert_data):
6     # conn = ms.connect(server='192.168.0.███', user='███', password='███', database='███')
7     conn = ms.connect(server='127.0.0.1', user='███', password='███', database='███')
8     cursor = conn.cursor()
9     sql = f'INSERT INTO {tablename} values(%s ,%s ,%s ,%s)'
10    cursor.execute(sql, (insert_data[0], insert_data[1], insert_data[2], insert_data[3]))
11    conn.commit()
12    conn.close()
```

03.개념설명

2008년 nnlm
2013년 word2vec
2018년 11월 BERT
2020년 5월 GPT3



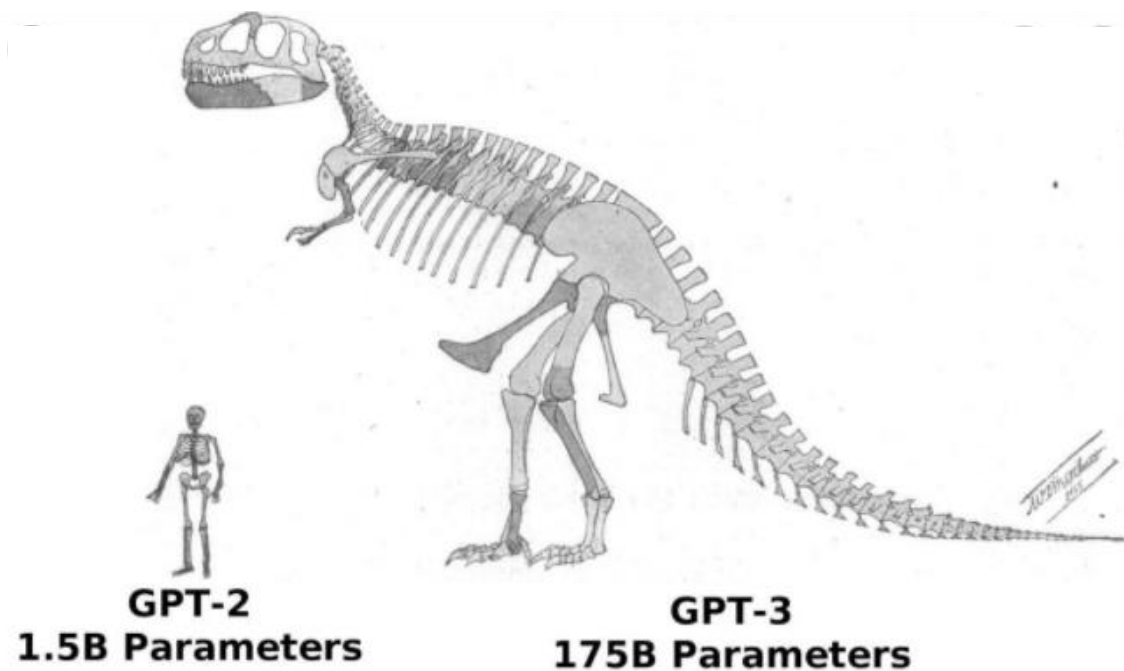
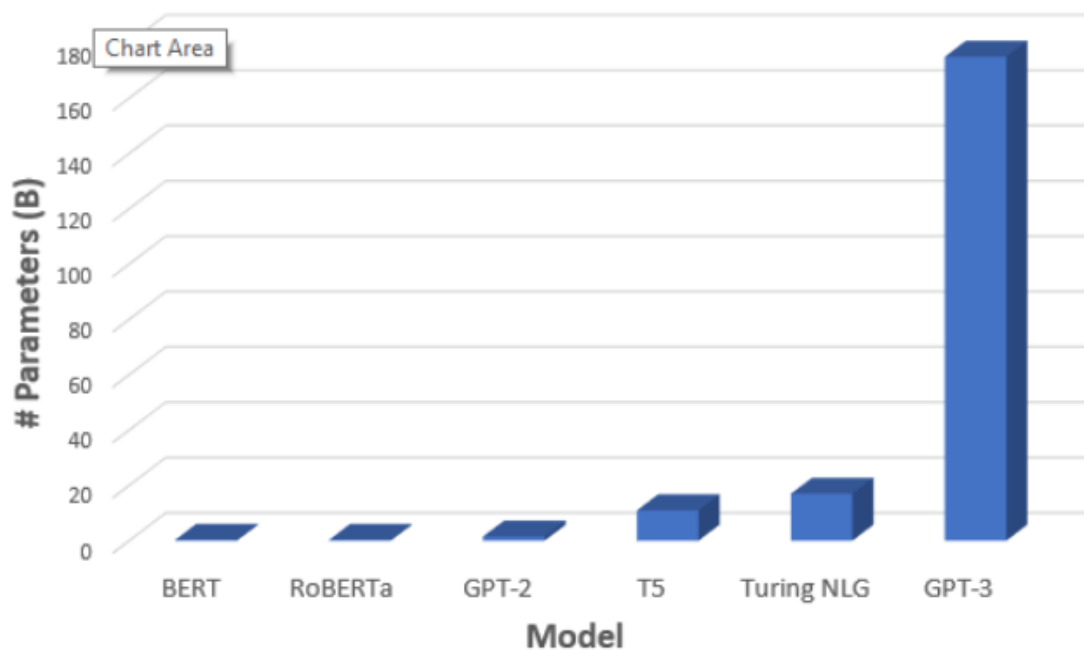
03.개념설명

BERT 3 억개

GPT2 15억개

GPT3 1750 억개

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

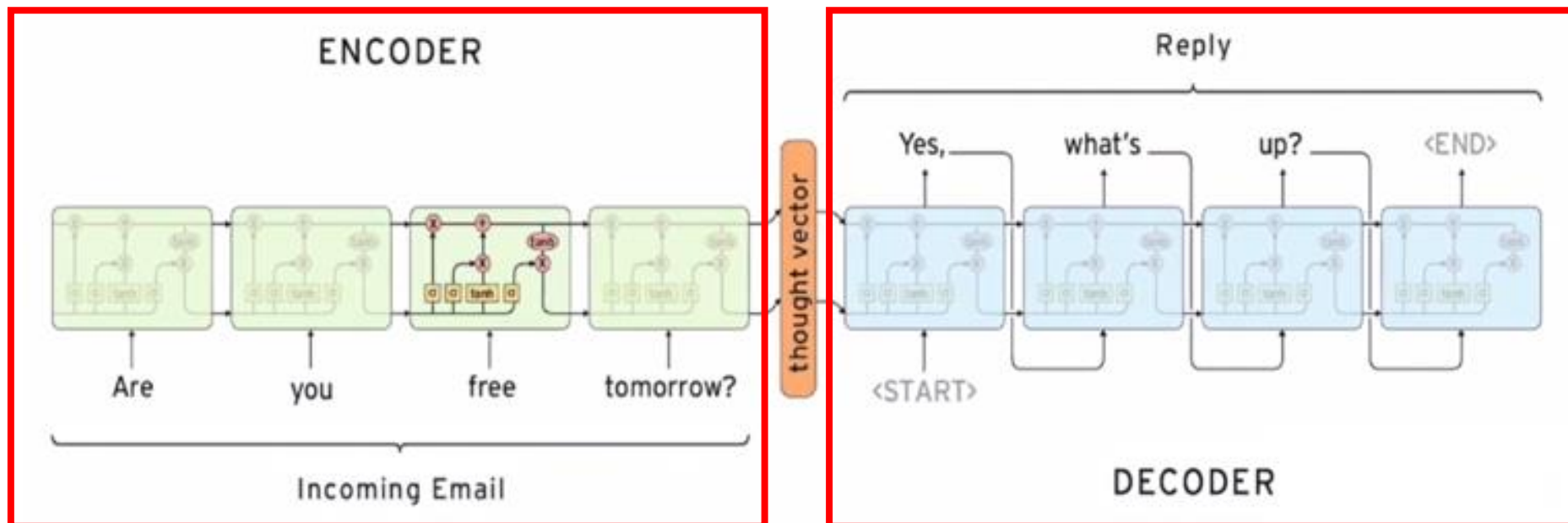


03.개념설명



03.개념설명

<seq2seq>



03.개념설명

encoder

```
▶ 1 # encoder_inputs = layers.Input(shape=(None,dic_len))
2 encoder_inputs = layers.Input(shape=(None,))
3 encoder_outputs = layers.Embedding(len(words), 100)(encoder_inputs)
4 encoder_outputs, state_h, state_c = layers.LSTM(64, return_state=True)(encoder_outputs)
5 encoder_states = [state_h, state_c]
```

decoder

```
1 # 디코더는 인코더의 최종 hidden state와 cell state 값을 받아
2 초기 상태를 초기화하고 *teacher forcing 과정을 진행하며 학습한다.
```

```
▶ 1 decoder_inputs = layers.Input(shape=(None,))
2 decoder_layers, _, _ = layers.LSTM(64, return_state=True, return_sequences=True)\
3 (decoder_inputs, initial_state=encoder_states)
```

```
▶ 1 decoder_outputs = layers.Dense(dic_len, activation='softmax')(decoder_layers)
```

```
▶ 1 model = models.Model(inputs=[encoder_inputs, decoder_inputs], outputs=decoder_outputs)
2 model.summary()
```

03.코드설명

```
# 문장을 인덱스로 변환
```

```
input_seq = make_predict_input('3과목 시험봐서 다 A등급을 맞았는데 그냥 자격증 발급할 필요없이 #  
그냥 가만히 내버려둬도 자격증 취득한 걸로 인정되나요? 그리고 oa master자격증도 신청하려고 했는데 #  
이런거 신청안해도 oa master자격증으로 인정되나요?')
```

```
input_seq
```

```
array([[1099, 2540, 4035, 3564, 3720, 1714, 3855, 5569, 5310, 4651, 5443,  
       2855, 5623, 4578, 4651, 6353, 4998, 2504, 5443, 5624, 806, 2626,  
       477, 381, 353, 3920, 5051, 3277, 5443, 2326, 182, 1110, 1507,  
       2461, 182, 6083, 2326, 5051, 3277, 5443, 1747, 381, 353, 0,  
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
       0, 0]))
```

03.코드설명

user > 3과목 시험봐서 다 A등급을 맞았는데 그냥 자격증 발급할 필요없이 그냥 가만히 내버려둬도 자격증 취득한 걸로 인정되나요? 그리고 oa master자격증도 신청하려고 했는데 이런거 신청안해도 oa master자격증으로 인정되나요?

GPT

Simsimi > 그냥 가만히 내버려두어도 자격증 취득해도 되요 어차피 a등급으로 합격한 기록은 계속 남기 때문에 언제든지 질문자님께서 카드형 자격증이나 oa master(인증서) 발급 신청을 해서 아무때나 받으시면 됩니다

'그냥 가만히 내버려 두 어도 자격증 취득 해도 되요 어차피 a 등급 으로 합격 한 기록 은 계속 남기 때문 에 언제 든지 질문 자 님 께서 카드 형 자격증 이나 oa master 인증 서 발급 신청 을 해서 아무 때 나 받으시면 됩니다 '

Seq2Seq

03.코드설명

user > 컴활이나 워드 그런 기초 자격증 말고요 경영 쪽과 관련된 자격증 좋은 거는 뭐가 있을까요?

GPT

Simsimi > 경영 관련 자격증이면 다음의 자격증들을 공부를 해서 취득하시는 것이 좋습니다 - 매경 테셋 - 유통관리사2급 - 회계관리(1급, 2급) - 물류관리사 - 가맹거래사 - 경영관리사 - SMAT

'경영 관련 자격증 이면 다음 의 자격증 들 을 공부 를 해서 취득 하시는 것 이 좋습니다 - 매경 테셋 - 유통 관리사 2 급 - 회계 관리 1 급 2 급 - 물류 관리사 - 가맹 거래사 - 경영 관리사 - SMAT '

Seq2Seq

03.코드설명

user > itq 자격증을 취득하고 싶습니다!!

GPT

Simsimi > ITQ 시험의 경우 2020년 3월 31일 이전에 취득한 사람은 보수교육 의무가 없어서 갱신 안해도 되구요 (평생 유효) 2020년 4월 1일 이후에 취득한 사람은 보수교육 의무가 있고 5년마다 갱신을 해야 합니다

'itq 한글 자격증 의 경우 도 같이 묶어져 있는 걸 로 구입 을 해서 공부 를 해서 언제 든지 질문 으로 다 합격 을 하 거든요 '

Seq2Seq

03.코드설명

user > 제가 고등학교 3학년인데요 컴퓨터 자격증 뭐를 따면 좋을 까요?

GPT

Simsimi > 컴퓨터 자격증 하나도 없으시면 기본적인 것부터 먼저 공부해서 취득하세요 워드1급, 컴활2급, itq oa master(한글, 엑셀, 파포) 요렇게 3개 준비해서 취득하세요

'2013년 10월 1일 이후 부터 보수 교육 제도 폐지 하였고 폐지 가 됨에 따라 국가 공인 자격 효력 으로 평생 유효한 걸 로 바뀌었습니다 따라서 갱신 안해 도 됩니다 '

Seq2Seq

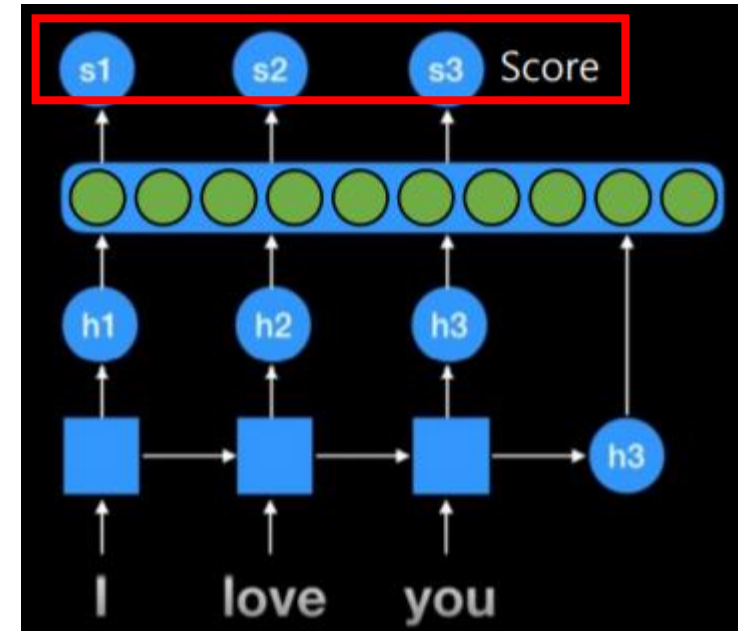
03.개념설명

RNN의 구조적 문제점

- 입력의 길이가 매우 긴 경우, 초반의 정보가 희석
- 고정된 문맥벡터 사이즈로 인해 긴 입력에 대한 정보를 함축하기 어려움
- 모든 token이 동일한 중요도로 학습되며 비효율적



Attention 모델



- 인간의 언어 정보처리 방식에 착안하여, 중요한 token을 더욱 중요하게 학습할 수 있도록

score값 계산

03.개념설명

Attention layer

도입으로 인해

- Attention 가중치에 따른 동적인 문맥벡터
- 기존 seq2seq의 성능 비약적 향상
- 하지만, 여전히 속도는 느림

LSTM을

없애는건

어떨까?

Self-attention 모델

- Attention is all you need!
- RNN을 encoder와 decoder에서 제거

Transformer 모델

multi-head attention으로 이루어진

encoder를 여러 층 쌓아서 encoding을 수행

03.개념설명

Attention layer

도입으로 인해

- Attention 가중치에 따른
- 기존 seq2seq의 성능 비
- 하지만, 여전히 속도는 느림

SQuAD v1.1 dataset leaderboard

Rank	Model	EM	F1
	Human Performance Stanford University (Rajpurkar et al. '16)	82.304	91.221
1 Oct 05, 2018	BERT (ensemble) Google AI Language https://arxiv.org/abs/1810.04805	87.433	93.160
2 Oct 05, 2018	BERT (single model) Google AI Language https://arxiv.org/abs/1810.04805	85.083	91.835

on 모델

you need!
와 decoder에서 제거

er 모델

tion으로 이루어진
층 쌓아서 encoding을 수행

04. KoGPT2

KoGPT2 (한국어 GPT-2)

Why'?'

- OpenAI GPT-2 모델의 한국어 성능 한계

Model

- GPT-2 base 모델

```
GPT2Model(units=768,  
           max_length=1024,  
           num_heads=12,  
           num_layers=12,  
           dropout=0.1,  
           vocab_size=50000)
```

Tokenizer

- 2천 5백만 이상의 문장으로 학습(wiki + news)
- BPE(Byte Pair Encoding)
- 50,000 토큰

04. KoGPT2

In [5]:



1

model

```
Out[5]: GPT2LMHeadModel(  
  (transformer): GPT2Model(  
    (wte): Embedding(50000, 768)  
    (wpe): Embedding(1024, 768)  
    (drop): Dropout(p=0.1, inplace=False)  
    (h): ModuleList(  
      (0): Block(  
        (ln_1): LayerNorm((768,), eps=1e-05, elementwise_affine=True)  
        (attn): Attention(  
          (c_attn): Conv1D()  
          (c_proj): Conv1D()  
          (attn_dropout): Dropout(p=0.1, inplace=False)  
          (resid_dropout): Dropout(p=0.1, inplace=False)  
        )  
        (ln_2): LayerNorm((768,), eps=1e-05, elementwise_affine=True)  
        (mlp): MLP(  
          (c_fc): Conv1D()  
          (c_proj): Conv1D()  
          (dropout): Dropout(p=0.1, inplace=False)  
        )  
      )  
    )  
  )  
)
```

04. KoGPT2

```
1 while 1:
2     input_idx = torch.tensor([vocab[vocab.bos_token]] + vocab[toked_test]).unsqueeze(0)
3     pred = model(input_idx)[0]
4     new_token = vocab.to_tokens(torch.argmax(pred, axis=-1).squeeze().tolist())[-1]
5     if new_token == '</s>':
6         break
7     test += new_token.replace('_', ' ')
8     print(test)
9     print('=====')
10    toked_test = tok(test)
11
```

2019년 한해를 보내며, 새해에는 더

=====

2019년 한해를 보내며, 새해에는 더 많은

=====

2019년 한해를 보내며, 새해에는 더 많은 사람들이

=====

2019년 한해를 보내며, 새해에는 더 많은 사람들이 새해

04. KoGPT2-chatbot

소소한 변화도 삶의 원동력이예요.</s>

KoGPT2

<usr>시계줄 바꿔야지</s>

<sent>1</s>

<sys>소소한 변화도 삶의 원동력이예요

04. KoGPT2-Chatbot

```
while 1:
    q = input('user > ').strip()
    if q == 'quit':
        break
    q_tok = tok(q)
    a = ''
    a_tok = []
    while 1:
        input_ids = torch.LongTensor([
            self.vocab[U_TKN]] + self.vocab[q_tok] +
            self.vocab[EOS, SENT] + self.vocab[sent_tokens] +
            self.vocab[EOS, S_TKN] +
            self.vocab[a_tok]).unsqueeze(dim=0)
        pred = self(input_ids)
        gen = self.vocab.to_tokens(
            torch.argmax(
                pred,
                dim=-1).squeeze().numpy().tolist())[-1]
        if gen == EOS:
            break
        a += gen.replace('_', ' ')
        a_tok = tok(a)
    print("Simsimi > {}".format(a.strip()))
```

05. 최종결과물

```
from flask import Flask, request, render_template
# -*- coding: utf-8 -*-
import argparse
import logging
import time
import gluonmlp as mlp
import numpy as np
import pandas as pd
import torch
from gluonmlp.data import SentencepieceTokenizer
from kogpt2.pytorch_kogpt2 import get_pytorch_kogpt2_model
from kogpt2.utils import get_tokenizer
from pytorch_lightning import Trainer
from pytorch_lightning.callbacks import ModelCheckpoint
from pytorch_lightning.core.lightning import LightningModule
from torch.utils.data import DataLoader, Dataset
from transformers.optimization import AdamW, get_cosine_schedule_with_warmup
import kss
ques = None
ans = None
```

05. 최종결과물

```
parser = argparse.ArgumentParser(description='Simsimi based on KoGPT-2')

parser.add_argument('--chat',
                    action='store_true',
                    default=False,
                    help='response generation on given user input')

parser.add_argument('--sentiment',
                    type=str,
                    default='0',
                    help='sentiment for system. 0 is neutral, 1 is negative, 2 is positive.')

parser.add_argument('--model_params',
                    type=str,
                    default='model_chp/model_last.ckpt',
                    help='model binary for starting chat')

parser.add_argument('--train',
                    action='store_true',
                    default=False,
                    help='for training')
```

05. 최종결과물

```
@staticmethod
def add_model_specific_args(parent_parser):
    # add model specific args
    parser = argparse.ArgumentParser(parents=[parent_parser], add_help=False)
    parser.add_argument('--max-len',
                        type=int,
                        default = 700,
                        help='max sentence length on input (default: 400)')

    parser.add_argument('--batch-size',
                        type=int,
                        default = 1,
                        help='batch size for training (default: 4)')
    parser.add_argument('--lr',
                        type=float,
                        default=5e-5,
                        help='The initial learning rate')
    parser.add_argument('--warmup_ratio',
                        type=float,
                        default=0.1,
                        help='warmup ratio')

    return parser
```

05. 최종결과물

```
@staticmethod
def add_model_specific_args(parent_parser):
    # add model specific args
    parser = argparse.ArgumentParser(parents=[parent_parser], add_help=False)
    parser.add_argument('--max_len')
```

```
>python train_torch.py --gpus 1 --train --max_epochs 2
```

```
parser.add_argument('--batch-size',
                    type=int,
                    default = 1,
                    help='batch size for training (default: 4)')
parser.add_argument('--lr',
                    type=float,
                    default=5e-5,
                    help='The initial learning rate')
parser.add_argument('--warmup_ratio',
                    type=float,
                    default=0.1,
                    help='warmup ratio')
return parser
```

05. 최종결과물

```
while 1:
    time2 = time.time()
    input_ids = torch.LongTensor([
        self.vocab[U_TKN]] + self.vocab[q_tok] +
        self.vocab[EOS, SENT] + self.vocab[sent_tokens] +
        self.vocab[EOS, S_TKN] +
        self.vocab[a_tok]).unsqueeze(dim=0)
    pred = self(input_ids)
    filtered_logits = self.top_k_top_p_filtering(pred[0, -1, :]/0.3, top_k=5, top_p=0.9)
    probabilities = F.softmax(filtered_logits, dim=-1)
    gen = self.vocab.to_tokens(torch.multinomial(probabilities, 1).numpy().tolist())[-1]
    # gen = self.vocab.to_tokens(next_token, dim=-1).numpy().tolist()[-1]
    # gen = self.vocab.to_tokens(
    #     torch.argmax(
    #         pred,
    #         dim=-1).squeeze().numpy().tolist())[-1]
    if gen == EOS:
        break
    a += gen.replace('_', ' ')
    a_tok = tok(a)
    print(a)
    if time2-time1>60:
        break
    try:
        if a[-1]==a[-2]==a[-3]==a[-4] : break
    except: continue
```

05. 최종결과물

헬프봇이 이야기를 들어드릴게요.
저와 대화해요

하고싶은 말을 적어주세요

대화하기

질문: 코로나는 언제 끝날까요?

답변을 드려요 !

코로나19바이러스가 완전히 끝날 때가 되면 코로나19의 증상이 조금씩 나아질거예요. 시간이 지나면서 코로나19의 증상이 점점 열어지고 점점 좋아지다가 어느 순간 사라질거예요. 그러니 너무 걱정하지 말고 예방차원에서 집에 돌아오면 좋겠어요. 답변이 도움이 되었기를 바랍니다.

헬프봇이 이야기를 들어드릴게요.
저와 대화해요

하고싶은 말을 적어주세요

대화하기

질문: 공부가 하기 싫어요

답변을 드려요 !

자신이 공부하는 이유를 찾는 게 가장 중요할 것 같아요. 자신이 좋아하는 게 무엇인지, 장래에 꼭 되고 싶은 게 무엇인지 계속 찾아보고 찾아보면 좋겠지요. 또, 수학이나 과학 같은 과목을 좋아하면 공부 외에 다른 것을 할 때 도움이 될 수도 있을 거예요. 자신이 좋아하는 게 무엇인지, 흥미 있는 게 무엇인지 계속 찾아보면 좋겠네요.

05. 최종결과물

헬프봇이 이야기를 들어드릴게요.
저와 대화해요

하고싶은 말을 적어주세요

대화하기

질문: 희망이 없는것 같아요

답변을 드려요 !

거듭된 좌절에 뭐라 위로의 말씀을 드리기 어렵습니다. 어렵게 취직이 되었는데 그마저 일주일 또 늦어지는군요. 신용회복위원, 금융위원회, 구청이며 심지어 주민센터까지 찾아다녔는데... 뭐가 안 풀리는 것 같아 좌절감이 더 크리라 짐작됩니다. 하지만 다시 생각해보면, 어려움을 탈출하는 턱밑까지 오지 않았습니까? 직장만 나가면 돌파구가 생기지 않습니까? 지금까지 잘 견디어 왔는데 조금 더 힘을 내면 어떨까요? 산에 오른 적이 있습니다. 정상에 가까울수록 숨이 턱에 차올라 숨쉬기도 힘들었습니다. 그만 돌아설까 하는데 이런 소리를 들었습니다. “죽을 만큼 숨이 차냐? 수고했다. 이제 내려갈 일만 남았다.” 우리가 많이 실수하는 것 중의 하나는 마지막 단계에서 포기하는 일입니다.님은 잘못된 것이 없으며, 죄도 없습니다. 그동안 처절할 정도로 열심히 살아왔고, 재정과 싸워왔을 뿐입니다.

헬프봇이 이야기를 들어드릴게요.
저와 대화해요

대화하기

질문: 남자친구랑 헤어졌어요

답변을 드려요 !

이별의 아픔 속에서만 사랑의 깊이를 안다는데, 지금 느끼는 아픔과 허망함. 이 모든 것이 아마 그만큼 그 분을 사랑하는 마음이 컸기에 더 크게 다가오는 것이겠지요. 그래도 남자친구와 헤어진 지 한 달이 다 되어 가는데, 아무 것도 하지 못한 채 그냥 아무 것도 할 수 없는 나날이라고 하니 남자친구가 얼마나 지친 것인가 하는 생각도 해 보게 됩니다. 사실 남자친구도 나름대로 삶의 방편이 없어 움츠러드는 것이겠지요. 하지만, 어쩔 수 없이 지금이 아닌 내일, 혹은 가족이나 사랑하는 사람 등을 생각하며 참고 견디는 일이기도 합니다. 우선은 이 상황을 누구와도 이야기 하지 못한 채, 홀로 견디며 살아간다는 것이 얼마나 외롭고 힘든 것인지도 모르겠습니다. 그동안 남자친구와 잘 이별했기에, 더 이상 그 인연을 이어나가지 못함을 부모님이나 남자친구에게 이야기 해보기도 하고, 주변에도 이야기를 해보았으나, 이루어 지지 않은 채 타지에서 홀로 힘들어하고 있으신 것은 아닌가 싶습니다.

05. 최종결과물(시연)

이 페이지를 표시할 수 없습니다.

- 웹 주소 이(가) 올바르지 확인하세요.
- 검색 엔진으로 페이지를 찾아보세요.
- 몇 분 후에 페이지를 새로 고치세요.

연결 문제 진단

05. 최종결과물

한계 및 의의

아직 여러 문장을 도출할 정도로 정교한 모델은 아님

특히 Ko-GPT2는 GPT2의 가장 작은 모델로 한문장 ~ 두문장 정도의 출력이 적절
비슷한 말투, 추상적인 내용의 데이터의 학습이 확연히 잘됨 -> 주제선정의 아쉬움

자연어처리 원리 공부와

자연어처리의 다양한 모델을 실습해 볼 수 있었음

감사합니다

https://github.com/Mean518/KoGPT2_chatbot

학습파일 : train_torch.py

실행파일 : run.py