구글 드라이브 관련해서 몇자 남겨놓습니다.

제 구글 드라이브는 **100GB까지 사용가능**해서 용량 걱정 안하셔도 됩니다. (현재 30GB)

**~TRAINING\_STATUS\_날짜.ipynb 로 끝나는 코드는 현재 실행하고 있는 코드**이니,

혹시 수정하셨으면 알려주세요!

**성능이 안좋고 오래된 코드는 old\_code 폴더**로 넣었습니다.

-> 제가 Pytorch를 사용한지 얼마 안되서 AutoGluon 튜닝이 어렵네요..

-> Tensorflow로 가능한 autokeras 한번 돌려볼게요!

- autokeras의 경우 앙상블은 수동으로 코드 작성 필요

-> 사용해봤는데 버그랑 제약이 너무 많고 업데이트가 느려서… ㅠㅠ **AutoGluon**으로 쓰는게 좋을 것 같아요!

추가로 해볼 것(예정)

1. 전이 학습

* 전이학습 대상 모델(Huggingface val acc: 0.9114)

<https://huggingface.co/shahidul034/chatGPT_classifier?text=I+like+you.+I+love+you>

* AutoGluod 전이학습 사용법

<https://auto.gluon.ai/stable/tutorials/multimodal/advanced_topics/customization.html?highlight=epoch>

1. 데이터셋 추가

* 추가 데이터셋

<https://huggingface.co/datasets/Hello-SimpleAI/HC3/viewer/open_qa/train>

⇒ 저작권을 살펴보니 완전 무료로 사용할 수 있는 데이터는 wiki-csai 로 생각됩니다 …!

캐글에 어떤 데이터를 추가로 사용할 때 어떤 라이선스까지 가능한지 안내사항에 안나와있어 잘 모르겠지만, 맘 편히 쓸 수 있는 건 일단 wiki-csai 로 생각됩니다!

* 3월 5일

1. **ChatGPT - 문법 -**
2. **전처리 코드 변경**

**import re**

**import emoji**

**from unidecode import unidecode**

**#remove html tags**

**def removeHTML(x):**

**html=re.compile(r'<.\*?>')**

**return html.sub(r'',x)**

**#remove emoji and symbols**

**def dataPreprocessing(x):**

**x = x.lower()**

**x = removeHTML(x)**

**x = emoji.demojize(x, delimiters=(" ", " "))**

**x = re.sub("'re", ' are',x) # 're n't --> are not**

**x = re.sub("n't", ' not',x) # 're n't --> are not**

**x = re.sub(r'\([^)]\*\)', '' ,x)**

**x = re.sub("/n",' ',x) # index 3번 \r, \n 삭제**

**x = re.sub("/r",' ',x) # index 3번 \r, \n 삭제**

**x = re.sub(r'[^A-Za-z0-9!.?, ]', '', x)**

**# x = re.sub(r'[!.?]','/n',x) # removed to try sentence structure analysis**

**x = re.sub(r'[,]',' ',x)**

**# x = re.sub("@\w+", '',x) # removing mentions (@)**

**# x = re.sub("'\d+", '',x)**

**# x = re.sub("\d+", '',x)**

**# x = re.sub(r"[^\w\s]", '',x) # to remove symbols**

**# x = re.sub("http\w+", '',x)**

**# x = re.sub("\s[a-z]\s", '',x)**

**x = re.sub("[ ]+",' ',x) # space\*n --> space**

**x = re.sub("\_",'',x) # \_people\_ --> people**

**x = x.strip()**

**return unidecode(x) #emily bronté --> emily bronte, mi² --> mi2, πr --> pr**

* **기호 관련**
* **기호를 세는 것**
* **사람 -> ?(물음표), !(느낌표) O / ChatGPT -> ? X**
  + **찾아보기**

1. **토큰화**

* **PoS 분석을 통해 문장의 structure 를 분석, Noun - Verb - ... 이런식으로 구성된 변수를 생성해 활용. (AI의 문장이 훨씬 문법적으로 옳으며, 정형화 되어있을 것으로 예상)**
* **텍스트+정형 데이터 가이드**
  + **https://auto.gluon.ai/stable/tutorials/multimodal/multimodal\_prediction/multimodal\_text\_tabular.html**

1. **문법 체크**

* **라이브러리 사용 가능한지**
* **nltk**

1. **k-fold**

* **옵션 or 구현**

-> 감정 - 모호

* Detect\_AutoGluon\_QA\_signcheck\_8hour\_TRAINING\_STATUS\_230305
  + 모델저장