ETRI KorBERT 모델로 KorQuAD 1.0 수행

- 한국어 데이터로 기계독해 수행 방법 두 가지
  - 구글 다국어 BERT 모델
    - 형태소 분석 없이 원문 텍스트 사용
    - 장점: 언어모델 학습 및 언어분석기 없이 빠르게 실행 가능
      - 공개 된 모델 사용
  - ETRI 형태소 BERT 모델(KorBERT)
    - 원문에 형태소분석 결과를 포함하여 사용
    - 장점: 구글 다국어 BERT모델 보다 높은 정확률

구분	의미역인식	기계독해	단락순위화	문장유사도추론	문서주제분류
평가데이터 및 규격	Korean Propbank, 학습: 19,302 문장 평가: 3,773 문장	KorQuAD 데이터, 학습: 60,406건 평가: 5,773건 (dev셋)	학습: 45,521 질문 평가: 1,000 질문 (질문당 평균 8.7개 단락)	학습: 10,874문장쌍 평가: 1,209문장쌍 (이진 분류체계: 유사, 무관)	학습: 9,301건 평가: 1,035건 (54개 분류체계)
평가 방법	F1[2]	Exact Match[3] / F1	Precision@Top1	Accuracy	Accuracy
(Google) Word Piece[4]기반 한국어 언어모델	81.85%	80.82% / 90.68% (정답 경계 구분을 위해 후처리 수행)	66.3%	79.4%	91.1%
(엑소브레인) Word Piece 기반 한국어 언어모델	85.10%	80.70% / 91.94% (정답 경계 구분을 위해 후처리 수행)	70.5%	82.7%	93.4%
(엑소브레인) 형태소 기반 한국어 언어모델	85.77%	86.40% / 94.18%	73.7%	83.4%	93.7%

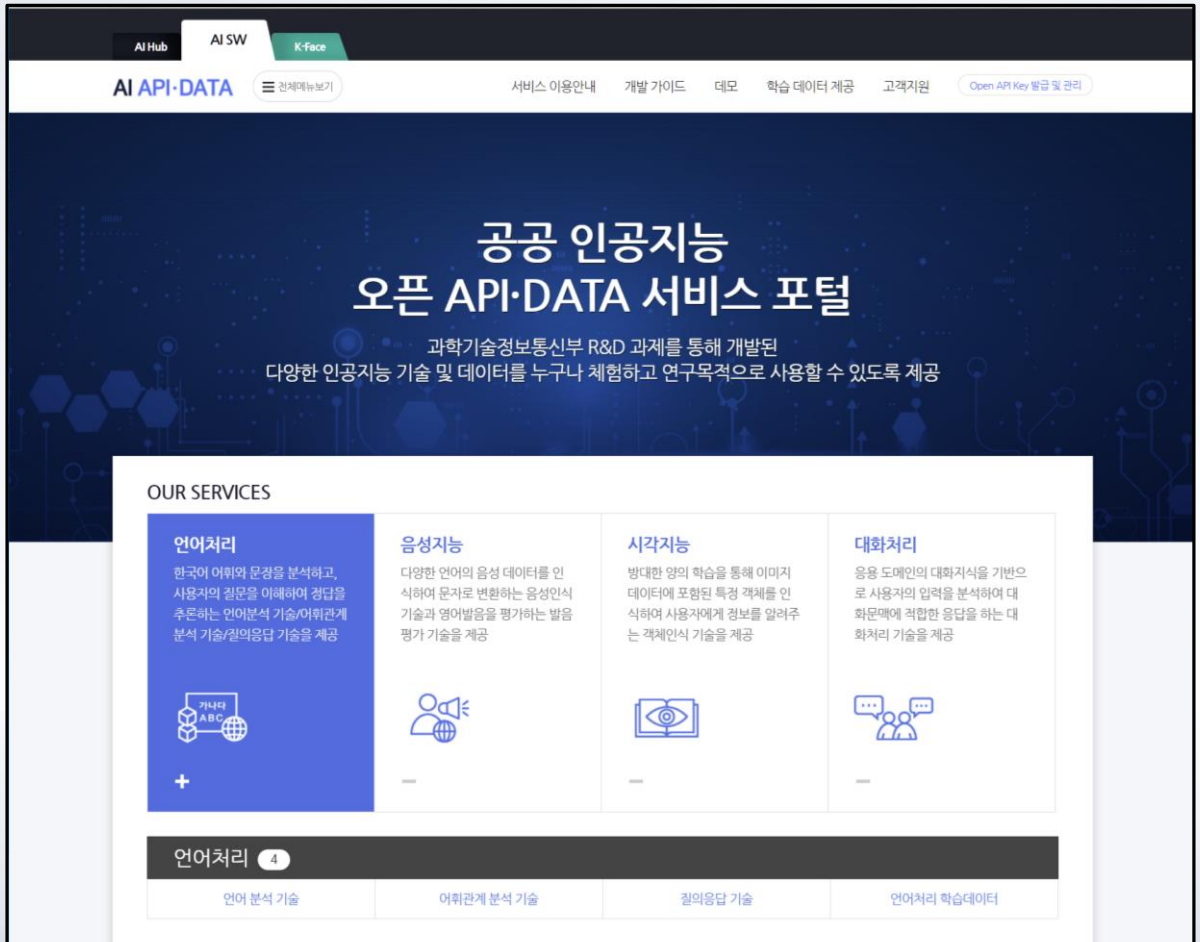
ETRI KorBERT 모델로 KorQuAD 1.0 수행

- 데이터 셋 포멧 변경
  - 원본 데이터
    - (원문) 질문, 정답, 단락, 정답에 대한 포지션 정보
  - 변환 데이터
    - (원문)질문, 정답, 단락
    - (형태소분석 결과)질문, 단락의 형태소 정보
    - (형태소분석 결과)질문, 단락의 형태소 위치 정보
    - (원문+형태소분석 결과)정답의 형태소 번호

- 사전 준비 내용
  - 형태소 분석기
  - KorBERT 모델 다운
  - 한국어 형태소 기반 기계독해 데이터 변환

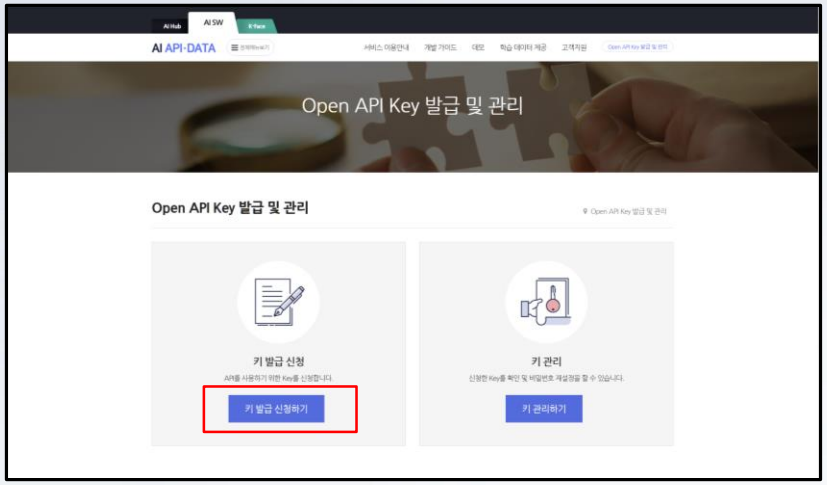
OpenAPI 형태소 분석기 사용 & KorBERT 다운로드

- OpenAPI 사이트 주소
  - https://aiopen.etri.re.kr/
  - Key 발급

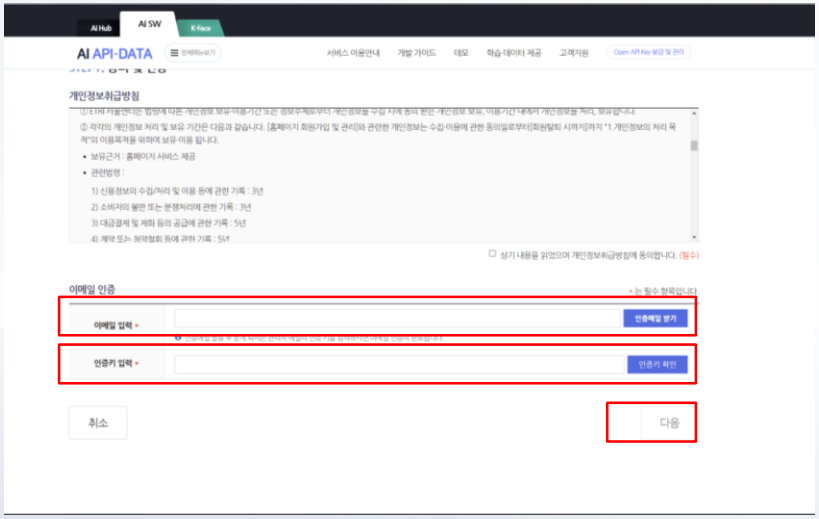


OpenAPI 형태소 분석기 사용 & KoBERT 다운

- 키 발급 신청

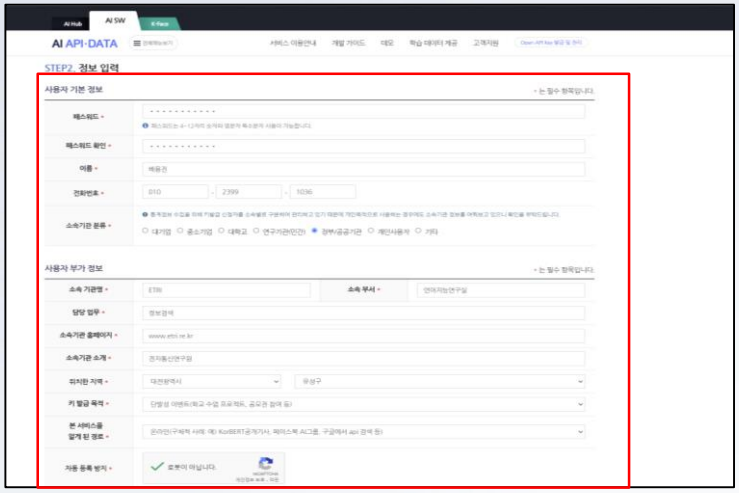


- 이메일 인증

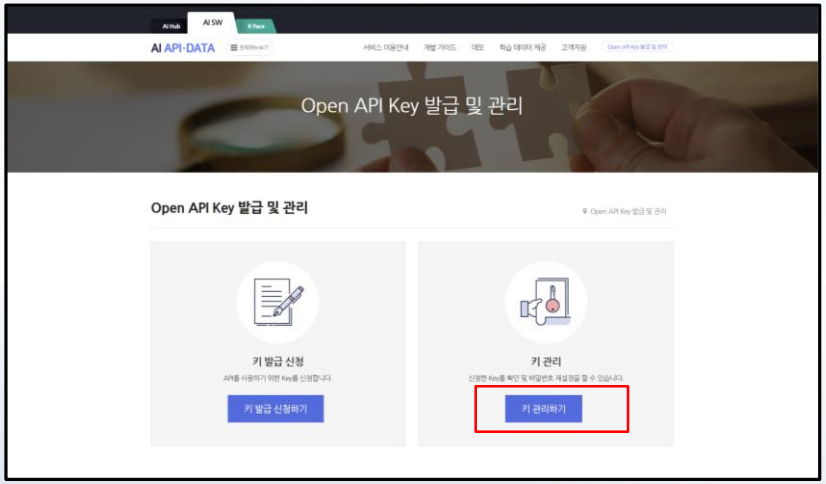


OpenAPI 형태소 분석기 사용 & KoBERT 다운

- 정보입력

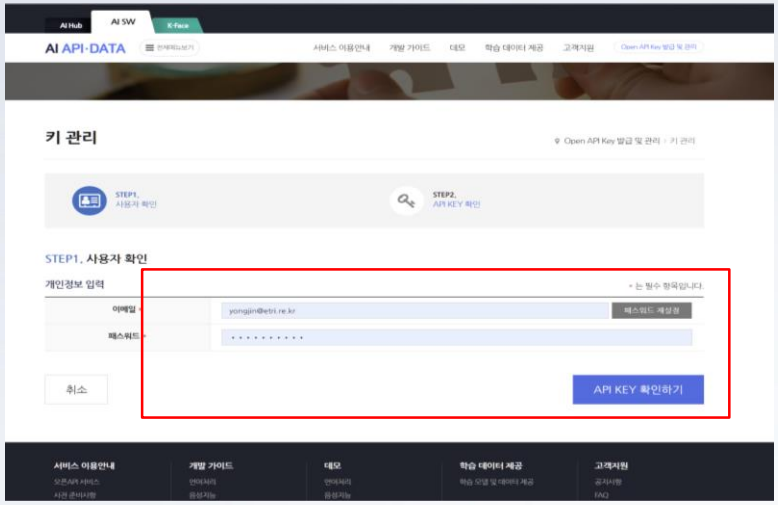


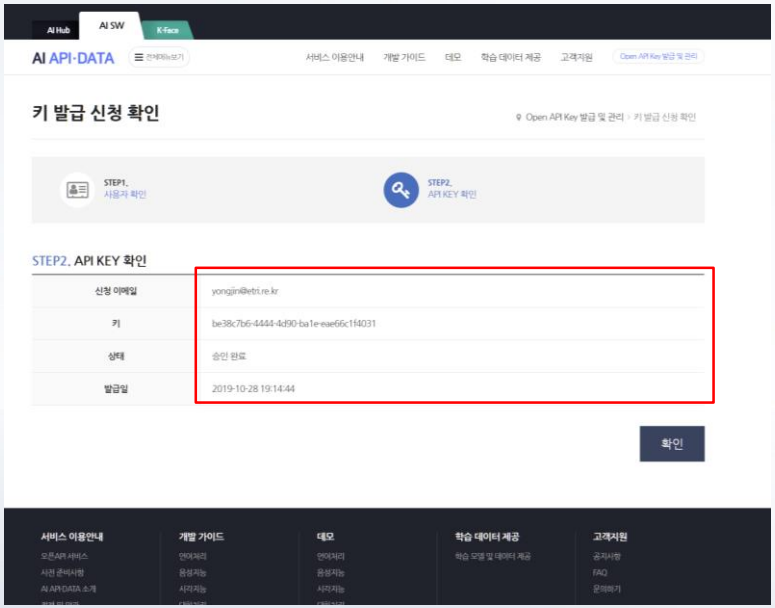
- OpenAPI 키 확인



OpenAPI 형태소 분석기 사용 & KoBERT 다운

- OpenAPI 키 확인

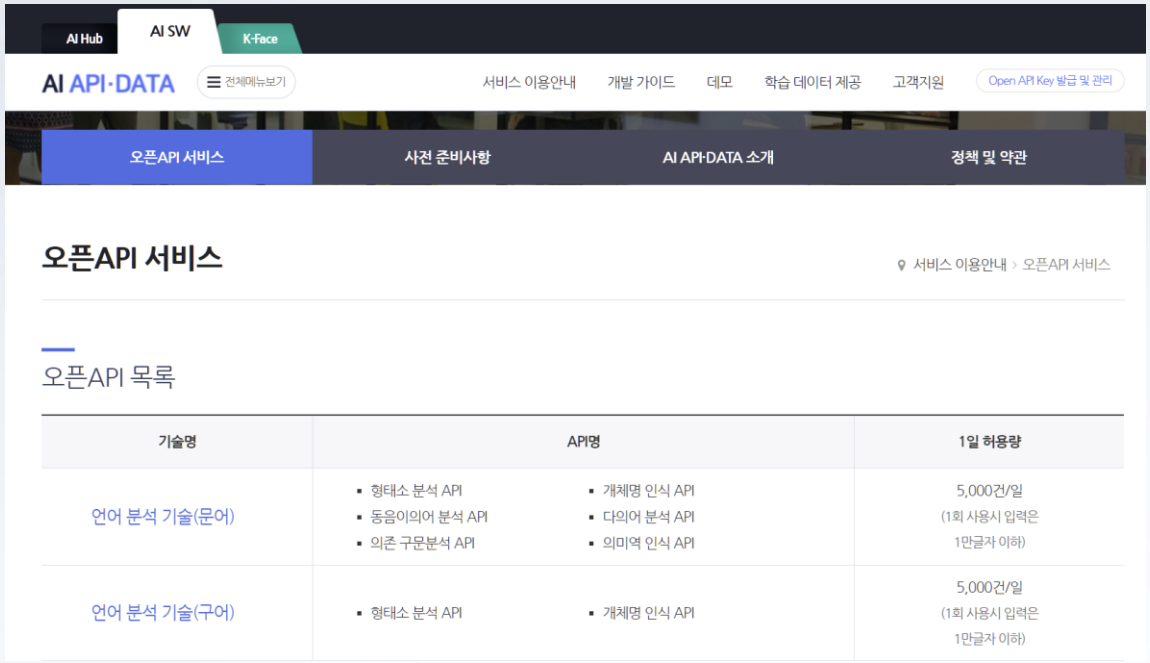




* 키 신청 후 관리자의 승인이 되어야 사용 가능합니다.

OpenAPI 형태소 분석기 사용

- **형태소 분석기**
  - 문어, 구어 구분하여 사용



*주의
- 형태소 분석 API의 1일 허용량은 5,000건으로 제한되어 있습니다.
- 1회 사용시 1만 글자 이하만 사용 가능합니다.

OpenAPI 형태소 분석기 사용 예제

- (Python) 형태소 분석기 사용 예제
  - python morp_openapi.py

```
#-*- coding:utf-8 -*-
import urllib3
import json
#문어 형태소 분석기
openApiURL = "http://aiopen.etri.re.kr:8000/WiseNLU"
#구어 형태소 분석기
#openApiURL = "http://aiopen.etri.re.kr:8000/WiseNLU_spoken"

accessKey = "be38c7b6-4444-4d90-ba1e-eae66c1f4031"
analysisCode = "morp"

//형태소 분석 대상 문장
text = "윤동주(尹東柱, 1917년 12월 30일 ~ 1945년 2월 16일)는 한국의 독립운동가, 시인, 작가이다."

requestJson = {
    "access_key": accessKey,
        "argument": {
                "text": text,
                 "analysis_code": analysisCode
        }
}

http = urllib3.PoolManager()
response = http.request("POST", openApiURL, headers={"Content-Type": "application/json; charset=UTF-8"},
body=json.dumps(requestJson))

print("[responseCode] " + str(response.status))
print("[responBody]")
print(str(response.data,"utf-8"))
```

OpenAPI 형태소 분석기 사용 예제

- (Python) 형태소 분석기 예제 결과

{"result":0,"return_object":{"doc_id":"","DCT":"","category":"","category_weight":0.0,"title":{"text":"","NE":""},"metaInfo":{},"paragraphInfo":[],"sentence":[{"id":0.0,"reserve_str":"","text":"윤동주(尹東柱, 1917년 12월 30일 ~ 1945년 2월 16일)는 한국의 독립운동가, 시인, 작가이다.","morp":[{"id":0.0,"lemma":"윤동주","type":"NNP","position":0.0,"weight":0.0566805},{"id":1.0,"lemma":"(","type":"SS","position":9.0,"weight":1.0},{"id":2.0,"lemma":"尹東柱","type":"SH","position":10.0,"weight":1.0},{"id":3.0,"lemma":",","type":"SP","position":19.0,"weight":1.0},{"id":4.0,"lemma":"1917","type":"SN","position":21.0,"weight":1.0},{"id":5.0,"lemma":"년","type":"NNB","position":25.0,"weight":0.0597013},{"id":6.0,"lemma":"12","type":"SN","position":29.0,"weight":1.0},{"id":7.0,"lemma":"월","type":"NNB","position":31.0,"weight":0.0583588},{"id":8.0,"lemma":"30","type":"SN","position":35.0,"weight":1.0},{"id":9.0,"lemma":"일","type":"NNB","position":37.0,"weight":0.0487802},{"id":10.0,"lemma":"~","type":"SO","position":41.0,"weight":1.0},{"id":11.0,"lemma":"1945","type":"SN","position":43.0,"weight":1.0},{"id":12.0,"lemma":"년","type":"NNB","position":47.0,"weight":0.0518608},{"id":13.0,"lemma":"2","type":"SN","position":51.0,"weight":1.0},{"id":14.0,"lemma":"월","type":"NNB","position":52.0,"weight":0.0649133},{"id":15.0,"lemma":"16","type":"SN","position":56.0,"weight":1.0},{"id":16.0,"lemma":"일","type":"NNB","position":58.0,"weight":0.0426752},{"id":17.0,"lemma":")","type":"SS","position":61.0,"weight":1.0},{"id":18.0,"lemma":"는","type":"JX","position":62.0,"weight":0.0897092},{"id":19.0,"lemma":"한국","type":"NNP","position":66.0,"weight":0.156358},{"id":20.0,"lemma":"의","type":"JKG","position":72.0,"weight":0.100211},{"id":21.0,"lemma":"독립","type":"NNG","position":76.0,"weight":0.160432},{"id":22.0,"lemma":"운동","type":"NNG","position":82.0,"weight":0.160432},{"id":23.0,"lemma":"가","type":"XSN","position":88.0,"weight":0.160432},{"id":24.0,"lemma":",","type":"SP","position":91.0,"weight":1.0},{"id":25.0,"lemma":"시인","type":"NNG","position":93.0,"weight":0.0583063},{"id":26.0,"lemma":",","type":"SP","position":99.0,"weight":1.0},{"id":27.0,"lemma":"작가","type":"NNG","position":101.0,"weight":0.0498619},{"id":28.0,"lemma":"이","type":"VCP","position":107.0,"weight":0.0484025},{"id":29.0,"lemma":"다","type":"EF","position":110.0,"weight":0.0749575},{"id":30.0,"lemma":".","type":"SF","position":113.0,"weight":1.0}],"WSD":[],"word":[{"id":0.0,"text":"윤동주(尹東柱,","type":"","begin":0.0,"end":3.0},{"id":1.0,"text":"1917년","type":"","begin":4.0,"end":5.0},{"id":2.0,"text":"12월","type":"","begin":6.0,"end":7.0},{"id":3.0,"text":"30일","type":"","begin":8.0,"end":9.0},{"id":4.0,"text":"~","type":"","begin":10.0,"end":10.0},{"id":5.0,"text":"1945년","type":"","begin":11.0,"end":12.0},{"id":6.0,"text":"2월","type":"","begin":13.0,"end":14.0},{"id":7.0,"text":"16일)는","type":"","begin":15.0,"end":18.0},{"id":8.0,"text":"한국의","type":"","begin":19.0,"end":20.0},{"id":9.0,"text":"독립운동가,","type":"","begin":21.0,"end":24.0},{"id":10.0,"text":"시인,","type":"","begin":25.0,"end":26.0},{"id":11.0,"text":"작가이다.","type":"","begin":27.0,"end":30.0}],"NE":[],"NE_Link":[],"dependency":[],"SRL":[]}],"entity":[]}}

OpenAPI 형태소 분석기 사용 예제

- (Python) 형태소 분석기 예제 결과


```
{
  "doc_id": "",   "DCT": "",   "category": "",   "category_weight": 0.0,  "title": {    "text": "",    "NE": ""  },  "metaInfo": {},  "paragraphInfo": [],
  "sentence": [    { "id": 0.0,
  "reserve_str": "",
  "text": "윤동주(尹東柱, 1917년 12월 30일 ~ 1945년 2월 16일)는 한국의 독립운동가, 시인, 작가이다.",
  "morp": [
     { "id": 0.0, "lemma": "윤동주", "type": "NNP", "position": 0.0,  "weight": 0.0566805 },
     { "id": 1.0, "lemma": "(", "type": "SS", "position": 9.0, "weight": 1.0 },
     { "id": 2.0, "lemma": "尹東柱", "type": "SH", "position": 10.0, "weight": 1.0   },
     { "id": 3.0, "lemma": ",", "type": "SP", "position": 19.0, "weight": 1.0   },
     { "id": 4.0, "lemma": "1917", "type": "SN", "position": 21.0, "weight": 1.0   },
     { "id": 5.0, "lemma": "년", "type": "NNB", "position": 25.0, "weight": 0.0597013   },
     { "id": 6.0, "lemma": "12", "type": "SN", "position": 29.0, "weight": 1.0   },
        .. (생략) ...
     { "id": 27.0, "lemma": "작가", "type": "NNG", "position": 101.0, "weight": 0.0498619   },
     { "id": 28.0, "lemma": "이", "type": "VCP", "position": 107.0, "weight": 0.0484025   },
     { "id": 29.0, "lemma": "다", "type": "EF", "position": 110.0, "weight": 0.0749575   },
     { "id": 30.0, "lemma": ".", "type": "SF", "position": 113.0, "weight": 1.0   }
 ],
 "WSD": [],
 "word": [
     { "id": 0.0, "text": "윤동주(尹東柱,", "type": "", "begin": 0.0, "end": 3.0   },
     { "id": 1.0, "text": "1917년", "type": "", "begin": 4.0, "end": 5.0   },
     { "id": 2.0, "text": "12월", "type": "", "begin": 6.0, "end": 7.0   },
     { "id": 3.0, "text": "30일", "type": "", "begin": 8.0, "end": 9.0   },
        .. (생략) ...
     { "id": 10.0, "text": "시인,", "type": "", "begin": 25.0, "end": 26.0   },
     { "id": 11.0, "text": "작가이다.", "type": "", "begin": 27.0, "end": 30.0   }
 ],
 ...
    }
  ],
  "entity": []
}
```

OpenAPI 형태소 분석기 사용 예제

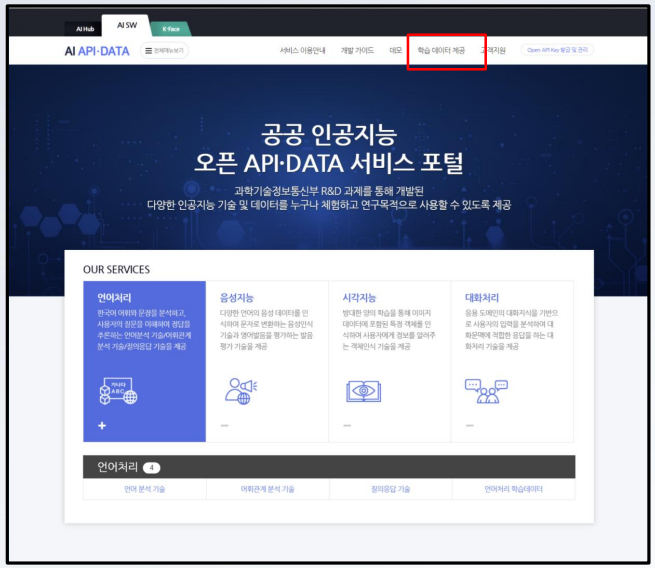
- 형태소 분석기의 품사 태그 정의
  - TTA 표준 형태소 태그 셋

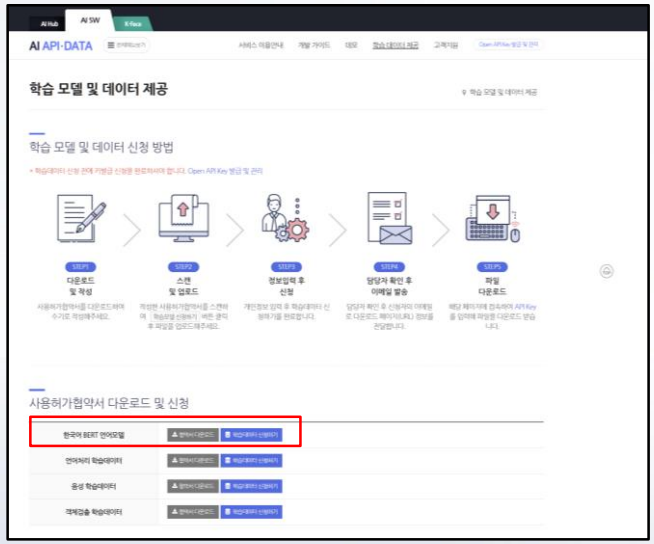
대분류	소분류	세분류
(1) 체언	명사(NN)	일반명사(NNG)
		고유명사(NNP)
		의존명사(NNB)
	대명사(NP)	대명사(NP)
	수사(NR)	수사(NR)
(2) 용언	동사(VV)	동사(VV)
	형용사(VA)	형용사(VA)
	보조용언(VX)	보조용언(VX)
	지정사(VC)	긍정지정사(VCP)
		부정지정사(VCN)
(3) 수식언	관형사(MM)	성상 관형사(MMA)
		지시 관형사(MMD)
		수 관형사(MMN)
	부사(MA)	일반부사(MAG)
		접속부사(MAJ)
(4) 독립언	감탄사(IC)	감탄사(IC)
(5) 관계언	격조사(JK)	주격조사(JKS)
		보격조사(JKC)
		관형격조사(JKG)
		목적격조사(JKO)
		부사격조사(JKB)
		호격조사(JKV)
		인용격조사(JKQ)
	보조사(JX)	보조사(JX)
	접속조사(JC)	접속조사(JC)
(6) 의존형태	어미(EM)	선어말어미(EP)
		종결어미(EF)
		연결어미(EC)
		명사형전성어미(ETN)
		관형형전성어미(ETM)
	접두사(XP)	체언접두사(XPN)
	접미사(XS)	명사파생접미사(XSN)
		동사파생접미사(XSV)
		형용사파생접미사(XSA)
	어근(XR)	어근(XR)
(7) 기호	일반기호(ST)	마침표, 물음표, 느낌표(SF)
		쉼표, 가운뎃점, 콜론, 빗금(SP)
		따옴표, 괄호표, 줄표(SS)
		줄임표(SE)
		붙임표(물결)(SO)
		기타 기호(SW)
	외국어(SL)	외국어(SL)
	한자(SH)	한자(SH)
	숫자(SN)	숫자(SN)
	분석불능범주(NA)	분석불능범주(NA)

TTA표준 형태소 태그 셋:
https://aiopen.etri.re.kr/data/001.%ED%98%95%ED%83%9C%EC%86%8C%EB%B6%84%EC%84%9D_%EA%B0%80%EC%9D%B4%EB%93%9C%EB%9D%BC%EC%9D%B8.pdf

KorBERT 언어모델 다운로드

- KorBERT 모델 다운로드



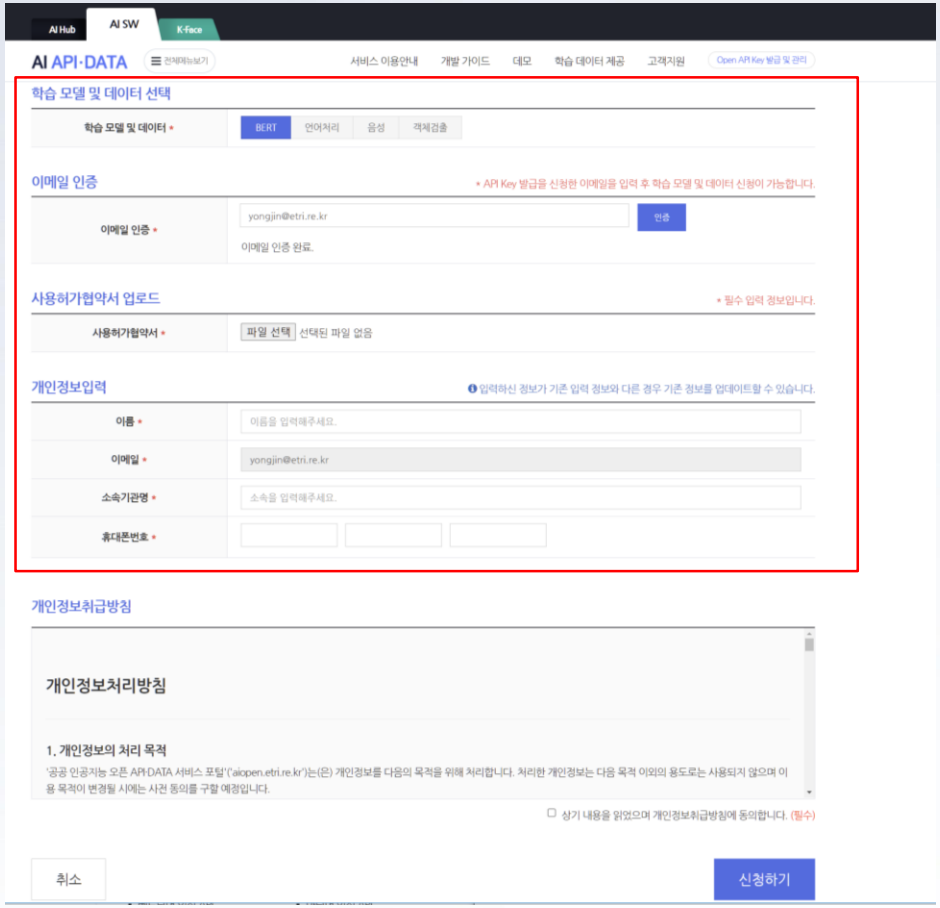


http://aiopen.etri.re.kr/aidata_download.php

accessKey = “be38c7b6-4444-4d90-ba1e-eae66c1f4031"

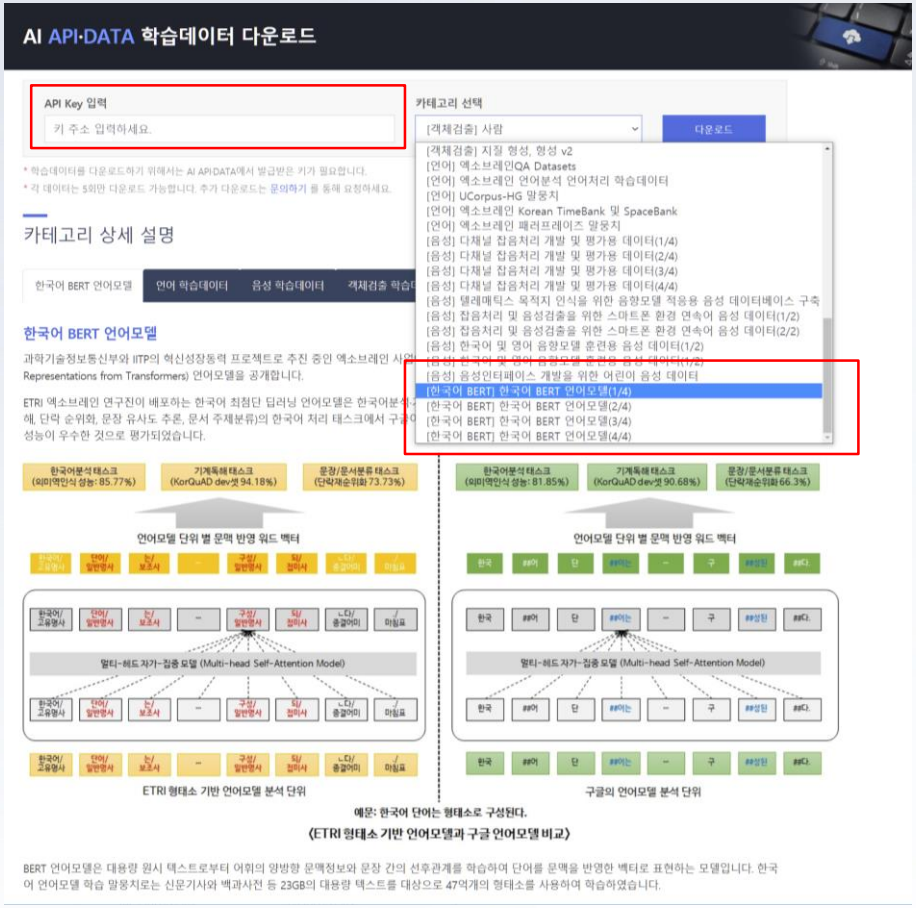
KorBERT 언어모델 다운로드

- KorBERT 언어모델 다운로드 신청서 작성



KorBERT 언어모델 다운로드

- KorBERT 언어모델 4종 다운로드



*실습시 사용 모델
[한국어 BERT] 한국어 BERT 언어모델(1/4)

*모델 설명
[한국어 BERT] 한국어 BERT 언어모델(1/4)
→ pytorch, 형태소 사전 & 토크나이져

[한국어 BERT] 한국어 BERT 언어모델(2/4)
→ tensorflow, 형태소 사전 & 토크나이져

[한국어 BERT] 한국어 BERT 언어모델(3/4)
→ pytorch, word piece 사전 & 토크나이져

[한국어 BERT] 한국어 BERT 언어모델(4/4)
→ tensorflow, word piece 사전 & 토크나이져

KorQuAD 1.0을 위한 데이터 포멧 변경

- 포멧 변경 대상
  - 원본 데이터
    - (원문) 질문, 정답, 단락, 정답에 대한 포지션(길이 기반) 정보
  - 변환 데이터
    - (원문)질문, 정답, 단락
    - (형태소분석 결과)질문, 단락의 형태소 정보
    - (형태소분석 결과)질문, 단락의 형태소 위치 정보
    - (원문+형태소분석 결과)정답의 형태소(바이트 기반) 번호
  - 데이터 변환 순서
    - 형태소 분석 결과 추가
    - 학습 데이터 포멧으로 변경
    - 형태소 정답 찾기
    - 검증 및 오류 디버깅

KorQuAD 1.0 데이터 변경

- **초기 원본 데이터**
  - **기계독해 KorQuAD 1.0 데이터 예제**


```
{
  "version": "KorQuAD_v1.0_dev",
  "data": [
    {
      "paragraphs": [
        {
          "qas": [  // 1개의 단락으로 복수개의 질문-정답 쌍 생성
            {
              "answers": [
                { "text": "신상철",  "answer_start": 26   }     // 정답, 정답 위치
              ],
              "id": "6559048-14-0",
              "question": "조사단의 구성원 문제나 은폐 가능성을 제기한 서프라이즈 대표는?"   // 질문
            }
          ],
          "context": "조사단의 구성원 문제나 은폐 가능성을 제기했던 신상철(서프라이즈 대표)에 대한 비판도 존재한다. 민주당 안규백 의원은 12일 동아일보와의 통화에서 "외부 모 인사에게서 '신 씨가 가장 적합하다'는 얘기를 들었다"며 신상철을 추천한 경위를 밝혔다. 신상철은 조사단 회의에 1회에 한해 2시간밖에 안 있는 등 조사활동에 참여하지 않은 채 군사기밀 공개를 요청하고 진보성향 언론들을 통해 ₩"미군 함선과 충돌했다₩" 혹은 ₩"주한미군 사령관이 한주호 준위 분향소를 방문한 것이 미군이 연루된 증거다₩" 등의 주장을 내세웠다. 이에 국방부는 ₩"전문성이 없는 인사가 조사위원으로 활동하기에 적절하지 않으며 이로 인해 공식결론에 반하는 내용을 조사위원 자격을 내세워 주장하는 등 대외적으로 불신 여론을 조장하여 공신력을 실추시키고 있다₩"고 밝히고 민주당에 교체를 요청했고, 신상철을 추천한 것에 대한 민주당의 책임론도 제기되었다. 민주당은 조사단 활동이 일주일 정도밖에 남지 않았기 때문에 교체는 어렵지만 문제가 되는 활동에 대해서 앞으로 공명정대하게 할 수 있도록 감독하겠다고 밝혔다."
        }
      ]
    }
  ]
}
```

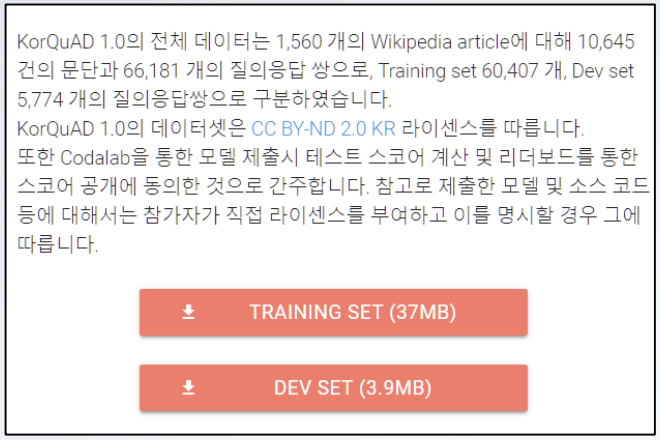
KorQuAD 1.0 데이터 변환

- **최종 변환 데이터**
  - **기계독해 KorQuAD 1.0 데이터 변환 후 예제**

```
{
   "version": "KorQuAD_v1.0_dev",
   "data": [
      {
         "id": "6559027-14-0",
         "question.text": "천안함 사건 조사단의 구성원 문제나 은폐 가능성을 제기한 것은 누구인가?",   // 질문 텍스트
         "question.morp_list": ["천안함/NNG", "사건/NNG", "조사/NNG", "단/XSN", "의/JKG", ..(생략).. , "은/JX", "누구/NP", "이/VCP", "ㄴ가/EF", "?/SF" ], // 질문 형태소 분석 결과
         "question.position_list": [0, 10, 17, 23, 26, ..(생략).. ,83, 87, 93, 93, 99], // 질문 형태소 byte position
         "answer.text": "신상철",   // 정답 텍스트
         "answer.begin_morp": 15, // 정답의 시작 형태소 번호
         "answer.end_morp": 15, // 정답의 마지막 형태소 번호
         "passage.text": "조사단의 구성원 문제나 ..(생략) .. 되는 활동에 대해서 앞으로 공명정대하게 할 수 있도록 감독하겠다고 밝혔다.",   // 단락 텍스트
         "passage.morp_list": ["조사/NNG", "단/XSN", "의/JKG", "구성/NNG", ..(생략)..  , "다고/EC", "밝히/VV", "었/EP", "다/EF", "./SF"],
        // 단락 형태소 분석 결과
         "passage.position_list": [0, 6, 9, 13, ..(생략).. , 1334, 1337, 1340, 1343]      // 단락 형태소의 byte position
         "answers.answer_start": 26       // 원본 데이터의 정답 및 위치
      }
   ]
}
```

KorQuAD 1.0 데이터 변환

- KorQuAD 1.0 다운로드
  - https://korquad.github.io/KorQuad%201.0/



*학습데이터: KorQuAD_v1.0_train.json
*개발데이터: KorQuAD_v1.0_dev.json

- KorQuAD 1.0 데이터에 형태소 분석 결과 추가
  - OpenAPI 형태소 분석기 사용

KorQuAD 1.0 데이터에 형태소 분석 결과 추가

- **원본데이터에 형태소 분석 결과 추가**
  - **개발 데이터 예제(KorQuAD_v1.0_dev.json)**

```
{
   "version": "KorQuAD_v1.0_dev",
   "data": [
      {
         "paragraphs": [
            {
               "qas": [  // 1개의 단락으로 복수개의 질문-정답 쌍 생성
                  {
                     "answers": [
                        {  "text": "신상철",   "answer_start": 26    }     // 정답, 정답 위치
                     ],
                     "id": "6559048-14-0",
                     "question": "조사단의 구성원 문제나 은폐 가능성을 제기한 서프라이즈 대표는?"   // 질문
                     "question_morp": { }    // 형태소 분석 결과 JSONObject 저장
                  }
                   ....(생략).....
               ],
               "context": "조사단의 구성원 문제나 은폐 가능성을 제기했던 신상철(서프라이즈 대표)에 대한 비판도 존재한다. 민주당 안규백 의원은 12일 동아일보와의 통화에서 "외부 모 인사에게서 '신 씨가 가장 적합하다'는 얘기를 들었다"며 신상철을 추천한 경위를 밝혔다. 신상철은 조사단 회의에 1회에 한해 2시간밖에 안 있는 등 조사활동에 참여하지 않은 채 군사기밀 공개를 요청하고 진보성향 언론들을 통해 ₩"미군 함선과 충돌했다₩" 혹은 ₩"주한미군 사령관이 한주호 준위 분향소를 방문한 것이 미군이 연루된 증거다₩" 등의 주장을 내세웠다. 이에 국방부는 ₩"전문성이 없는 인사가 조사위원으로 활동하기에 적절하지 않으며 이로 인해 공식결론에 반하는 내용을 조사위원 자격을 내세워 주장하는 등 대외적으로 불신 여론을 조장하여 공신력을 실추시키고 있다₩"고 밝히고 민주당에 교체를 요청했고, 신상철을 추천한 것에 대한 민주당의 책임론도 제기되었다. 민주당은 조사단 활동이 일주일 정도밖에 남지 않았기 때문에 교체는 어렵지만 문제가 되는 활동에 대해서 앞으로 공명정대하게 할 수 있도록 감독하겠다고 밝혔다."
                  "context_morp": { }    // 형태소 분석 결과 JSONObject 저장
            }
         ]
      }
   ]
}
```

KorQuAD 1.0 데이터에 형태소 분석 결과 추가

- (python)원본데이터에 형태소 분석 결과 추가(계속)
  - build_data_step1_ai4001.py
    - 실행방법 python build_data_step1_ai4001.py [입력파일] [출력파일]
      - 입력파일: KorQuAD 1.0 다운로드 받은 학습/개발 데이터

```
#-*- coding:utf-8 -*-
import urllib3
import json
import sys

openApiURL = "http://aiopen.etri.re.kr:8000/WiseNLU"
accessKey = "be38c7b6-4444-4d90-ba1e-eae66c1f4031"
analysisCode = "morp"

http = urllib3.PoolManager()

def morp_openapi(text):
    requestJson = {
        "access_key": accessKey,
        "argument": {
            "text": text,
            "analysis_code": analysisCode
        }
    }
    response = http.request("POST", openApiURL, headers={"Content-Type": "application/json; charset=UTF-8"},
body=json.dumps(requestJson))
    morp_result = str(response.data,"utf-8")
    morp_json = json.loads(morp_result)['return_object']
    return morp_json
```

OpenAPI 형태소 분석 URL 및 사용자 키 설정

형태소 분석 수행 모듈

KorQuAD 1.0 데이터에 형태소 분석 결과 추가(ai4001_quiz)

아래 명령어로 실행되게 수정하기
python build_data_step1_ai4001_quiz.py

- **(python)원본데이터에 형태소 분석 결과 추가**

```
#input
input_file='./step1_input.txt'
with open(input_file, 'r') as f:
    jObj = json.loads(''.join(f.readlines()))

#output
output_file='step1_output.txt'
wf = open(output_file, 'w', encoding='UTF-8')

datas = jObj['data']
for di, paragraphs in enumerate(datas):

    paragraphs = paragraphs['quiz']
    for pi, paragraph in enumerate(paragraphs):
        context = paragraph['quiz']
        context_lang = morp_openapi(context) ← 형태소 분석 수행
        paragraph['context_morp'] = context_lang

        qas = paragraph['quiz']
        for qi, q in enumerate(qas):
            question = q['quiz']
            question_lang = morp_openapi(question) ← 형태소 분석 수행
            q['question_morp']=question_lang
            qas[qi] = q

        paragraph['quiz'] = qas
        paragraphs[pi] = paragraph

    datas[di]['quiz'] =  paragraphs
jObj['data'] = datas

wf.write(json.dumps(jObj ,ensure_ascii=False))
wf.close()
```

입력: KorQuAD 1.0 사이트로부터 다운 받은 학습/개발 파일
. KorQuAD_v1.0_train.json
. KorQuAD_v1.0_dev.json

출력: 출력 파일 이름

학습/개발 데이터 내의 'question'과 'context'를 형태소 분석하여 기존 데이터(JSONObject)에 추가

KorQuAD 1.0 데이터에 형태소 분석 결과 추가(정답1)

- (python)원본데이터에 형태소 분석 결과 추가

```
#input
input_file=''
with open(input_file, 'r') as f:
    jObj = json.loads(''.join(f.readlines()))

#output
output_file=''
wf = open(output_file, 'w', encoding='UTF-8')

datas = jObj['data']
for di, paragraphs in enumerate(datas):

    paragraphs = paragraphs['paragraphs']
    for pi, paragraph in enumerate(paragraphs):
        context = paragraph['context']
        context_lang = morp_openapi(context) ← 형태소 분석 수행
        paragraph['context_morp'] = context_lang

        qas = paragraph['qas']
        for qi, q in enumerate(qas):
            question = q['question']
            question_lang = morp_openapi(question) ← 형태소 분석 수행
            q['question_morp']=question_lang
            qas[qi] = q

        paragraph['qas'] = qas
        paragraphs[pi] = paragraph

    datas[di]['paragraphs'] =  paragraphs
jObj['data'] = datas

wf.write(json.dumps(jObj ,ensure_ascii=False))
wf.close()
```

입력: KorQuAD 1.0 사이트로부터 다운 받은 학습/개발 파일
. KorQuAD_v1.0_train.json
. KorQuAD_v1.0_dev.json

출력: 출력 파일 이름

학습/개발 데이터 내의 'question'과 'context'를 언어분석 하여 기존 데이터에 추가

KorQuAD 1.0 학습데이터 포멧으로 변경

- **학습 데이터 포멧 변경 예제**

{
 "version": "KorQuAD_v1.0_dev",
"data": [
 {
  "question.text": "미국 군대 내 두번째로 높은 직위는 무엇인가?",

  "question.morp_list": [미국/NNP, 군대/NNG, 내/NNB, 두/MM, 번째/NNB, 로/JKB, 높/VA, 은/ETM, 직위/NNG, 는/JX, 무엇/NP, 이/VCP, ㄴ가/EF, ?/SF],
  "question.position_list":[0, 7, 14, 18, 21, 27, 31, 34, 38, 44, 48, 54, 54, 60,

  "passage.text": "알렉산더 메이그스 헤이그 2세(영어: Alexander Meigs Haig, Jr., 1924년 12월 2일 ~ 2010년 2월 20일)는 미국의 국무 장관을 지낸 미국의 군인, 관료 및 정치인이다. 로널드 레이건 대통령 밑에서 국무장관을 지냈으며, 리처드 닉슨과 제럴드 포드 대통령 밑에서 백악관 비서실장을 지냈다. 또한 그는 미국 군대에서 2번째로 높은 직위인 미국 육군 부참모 총장과 나토 및 미국 군대의 유럽연합군 최고사령관이었다.",
   "passage.morp_list": [알렉산더/NNP, 메이그스/NNP, 헤이그/NNP, 2/SN, 세/NNB, (/SS, 영어/NNP, :/SP, Alexander/SL, Meigs/SL, Haig/SL, ,/SP, Jr/SL, ./SF, ,/SP, 1924/SN, 년/NNB, 12/SN, 월/NNB, 2/SN, 일/NNB, ~/SO, 2010/SN, 년/NNB, 2/SN, 월/NNB, 20/SN, 일/NNB,)/SS, 는/JX, 미국/NNP, 의/JKG, 국무/NNG, 장관/NNG, 을/JKO, 지내/VV, ㄴ/ETM, 미국/NNP, 의/JKG, 군인/NNG, ,/SP, 관료/NNG, 및/MAJ, 정치/NNG, 인/XSN, 이/VCP, 다/EF, ./SF, 로널드/NNP, 레이건/NNP, 대통령/NNG, 밑/NNG, 에서/JKB, 국무/NNG, 장관/NNG, 을/JKO, 지내/VV, 었/EP, 으며/EC, ,/SP, 리처드/NNP, 닉슨/NNP, 과/JC, 제럴드/NNP, 포드/NNP, 대통령/NNG, 밑/NNG, 에서/JKB, 백악관/NNP, 비서/NNG, 실/XSN, 장/XSN, 을/JKO, 지내/VV, 었/EP, 다/EF, ./SF, 또한/MAG, 그/NP, 는/JX, 미국/NNP, 군대/NNG, 에서/JKB, 2/SN, 번째/NNB, 로/JKB, 높/VA, 은/ETM, 직위/NNG, 이/VCP, ㄴ/ETM, 미국/NNP, 육군/NNG, 부참모/NNG, 총장/NNG, 과/JC, 나토/NNP, 및/MAJ, 미국/NNP, 군대/NNG, 의/JKG, 유럽연/NNP, 합군/NNG, 최고/NNG, 사령/NNG, 관/XSN, 이/VCP, 었/EP, 다/EF, ./SF],
   "passage.position_list: [0, 13, 26, 36, 37, 40, 41, 47, 49, 59, 65, 69, 71, 73, 74, 76, 80, 84, 86, 90, 91, 95, 97, 101, 105, 106, 110, 112, 115, 116, 120, 126, 130, 137, 143, 147, 150, 154, 160, 164, 170, 172, 179, 183, 189, 192, 195, 198],

  "answers": [
     {'text': '미국 육군 부참모 총장', 'answer_start': 204}
  ]
 }
]
}

KorQuAD 1.0 학습데이터 포멧으로 변경(quiz2)

- **(python)학습데이터 포멧으로 변경(계속)**
  - **형태소 분석 결과 활용하기**

아래 명령어로 실행되게 수정하기
python morp_rep_quiz.py

```
import sys
import json
def morp_representation(morp_json):
    morp_list=[]
    morp_position_list=[]
    for sentence in morp_json['quiz']:
        for morp in sentence['quiz']:
            morp_list.append(morp['quiz']+'/'+morp['quiz'])
            morp_position_list.append(morp['quiz'])
    return morp_list, morp_position_list

with open('ndoc_input.txt', 'r') as f:
    jObj = json.loads(' '.join(f.readlines()))
    morp_list, position_list = morp_representation(jObj)
    print('morp_list: ', morp_list)
    print('position_list: ', position_list)
```

1. 형태소 분석 결과 lemma/type 형태로 변경
2. 형태소 byte 포지션 정보 저장

def morp_representation()
- 입력: 형태소 분석 결과 json object
- 출력
  morp_list = ['미국/NNP', '군대/NNG', '내/NNB', '두/MM', '번째/NNB' .....]
  morp_position_list =[0, 7, 14, 18, 21, 27, 31, ...]

KorQuAD 1.0 학습데이터 포멧으로 변경(quiz2)

- python morp_rep_quiz.py

입력: ndoc_input.txt


```
{
  "doc_id": "",
  "DCT": "",
  "category": "",
  "category_weight": 0.0,
  "title": {
                "text": "",
                "NE": ""
  },
  "metaInfo": {},
  "paragraphInfo": [],
  "sentence": [
  {
      "id": 0.0,
      "reserve_str": "",
      "text": "윤동주(尹東柱, 1917년 12월 30일 ~ 1945년 2월 16일)는 한국의 독립운동가, 시인, 작가이다.",
      "morp": [
        {  "id": 0.0,  "lemma": "윤동주",  "type": "NNP",  "position": 0.0,  "weight": 0.0566805  },
        {  "id": 1.0,  "lemma": "(",  "type": "SS",  "position": 9.0,  "weight": 1.0  },
        {  "id": 2.0,  "lemma": "尹東柱",  "type": "SH",  "position": 10.0,  "weight": 1.0  },
        {  "id": 3.0,  "lemma": ",",  "type": "SP",  "position": 19.0,  "weight": 1.0  },
        {  "id": 4.0,  "lemma": "1917",  "type": "SN",  "position": 21.0,  "weight": 1.0  },
        {  "id": 5.0,  "lemma": "년",  "type": "NNB",  "position": 25.0,  "weight": 0.0597013  },
        {  "id": 6.0,  "lemma": "12",  "type": "SN",  "position": 29.0,  "weight": 1.0  },
                    ... (생략)..
        {  "id": 17.0,  "lemma": ")",  "type": "SS",  "position": 61.0,  "weight": 1.0  },
        {  "id": 18.0,  "lemma": "는",  "type": "JX",  "position": 62.0,  "weight": 0.0897092  },
        {  "id": 19.0,  "lemma": "한국",  "type": "NNP",  "position": 66.0,  "weight": 0.156358  },
        {  "id": 20.0,  "lemma": "의",  "type": "JKG",  "position": 72.0,  "weight": 0.100211  },
        {  "id": 21.0,  "lemma": "독립",  "type": "NNG",  "position": 76.0,  "weight": 0.160432  },
        {  "id": 22.0,  "lemma": "운동",  "type": "NNG",  "position": 82.0,  "weight": 0.160432  },
        {  "id": 23.0,  "lemma": "가",  "type": "XSN",  "position": 88.0,  "weight": 0.160432  },
        {  "id": 24.0,  "lemma": ",",  "type": "SP",  "position": 91.0,  "weight": 1.0  },
        {  "id": 25.0,  "lemma": "시인",  "type": "NNG",  "position": 93.0,  "weight": 0.0583063  },
        {  "id": 26.0,  "lemma": ",",  "type": "SP",  "position": 99.0,  "weight": 1.0  },
        {  "id": 27.0,  "lemma": "작가",  "type": "NNG",  "position": 101.0,  "weight": 0.0498619  },
        {  "id": 28.0,  "lemma": "이",  "type": "VCP",  "position": 107.0,  "weight": 0.0484025  },
        {  "id": 29.0,  "lemma": "다",  "type": "EF",  "position": 110.0,  "weight": 0.0749575  },
        {  "id": 30.0,  "lemma": ".",  "type": "SF",  "position": 113.0,  "weight": 1.0  }
      ]
    }
  ],
  "entity": []
}
```


출력: ndoc_output.txt

morp_list: ['윤동주/NNP', '(/SS', '尹東柱/SH', ',/SP', '1917/SN', '년/NNB', '12/SN', '월/NNB', '30/SN', '일/NNB', '~/SO', '1945/SN', '년/NNB', '2/SN', '월/NNB', '16/SN', '일/NNB', ')/SS', '는/JX', '한국/NNP', '의/JKG', '독립/NNG', '운동/NNG', '가/XSN', ',/SP', '시인/NNG', ',/SP', '작가/NNG', '이/VCP', '다/EF', './SF']

position_list: [0.0, 9.0, 10.0, 19.0, 21.0, 25.0, 29.0, 31.0, 35.0, 37.0, 41.0, 43.0, 47.0, 51.0, 52.0, 56.0, 58.0, 61.0, 62.0, 66.0, 72.0, 76.0, 82.0, 88.0, 91.0, 93.0, 99.0, 101.0, 107.0, 110.0, 113.0]

KorQuAD 1.0 학습데이터 포멧으로 변경(정답2)

- (python)학습데이터 포멧으로 변경(계속)
  - build_data_step2_ai4001.py
    - 실행방법 python build_data_step2_ai4001.py [입력파일] [출력파일]
      - 입력파일: build_data_step1_ai4001.py의 결과파일

```
import sys
import json

def morp_representation(morp_json):
    morp_list=[]
    morp_position_list=[]

    for sentence in morp_json['sentence']:
        for morp in sentence['morp']:
            morp_list.append(morp['lemma']+'/'+morp['type'])
            morp_position_list.append(morp['position'])

    return morp_list, morp_position_list
```

1. 형태소 분석 결과 lemma/type 형태로 변경
2. 형태소 포지션 정보 저장

def morp_representation()
- 입력: 형태소 분석 결과 json object
- 출력
  morp_list = [미국/NNP 군대/NNG 내/NNB 두/MM 번째/NNB .....]
  morp_position_list =[0 7 14 18 21 27 31 .....]

KorQuAD 1.0 학습데이터 포멧으로 변경

- (python)학습데이터 포멧으로 변경
  - build_data_step2_ai4001.py
    - 실행방법 python build_data_step2_ai4001.py [입력파일] [출력파일]
      - 입력파일: build_data_step1_ai4001.py의 결과파일

KorQuAD 1.0 학습데이터 포멧으로 변경

```
converted_data={}
converted_data_list=[]

with open(sys.argv[1], 'r') as f:
    for data in json.loads(' '.join(f.readlines()))['data']:
        for paragraph in data['paragraphs']:
            e_dic={}
            context = paragraph['context']
            context_morp = paragraph['context_morp']
            text_list, context_morp_list, context_morp_position_list = morp_representation(context_morp)
            e_dic['passage.text'] = context
            e_dic['passage.morp_list'] = context_morp_list
            e_dic['passage.position_list'] = context_morp_position_list
            qas = paragraph['qas']
            for qa in qas:
                _id = qa['id']
                question = qa['question']
                question_morp = qa['question_morp']
                question_text, question_morp_list, question_morp_position_list = morp_representation(question_morp)
                e_dic['id'] = _id
                e_dic['question.text'] = question
                e_dic['question.morp_list'] = question_morp_list
                e_dic['question.position_list'] = question_morp_position_list
                e_dic['answers'] = qa['answers']
                e_dic['answer.text'] = qa['answers'][0]['text']
                e_dic['answer.begin_morp'] = 0
                e_dic['answer.end_morp'] = 0
            converted_data_list.append(e_dic)
with open(sys.argv[2], 'w') as wf:
    converted_data['data']=converted_data_list
    wf.write(json.dumps(converted_data, indent=3, ensure_ascii=False))
```

파일 읽기

단락의 형태소 정보, 번호, 위치 정보 추출

질문의 형태소 정보, 번호, 위치 정보 추출

결과저장

KorQuAD 1.0 형태소 정답 찾기

- **형태소 정답 찾기 예제**

{
"version": "KorQuAD_v1.0_dev",
"data": [
{
"question.text": "미국 군대 내 두번째로 높은 직위는 무엇인가?",
"question.morp_list": [미국/NNP, 군대/NNG, 내/NNB, 두/MM, 번째/NNB, 로/JKB, 높/VA, 은/ETM, 직위/NNG, 는/JX, 무엇/NP, 이/VCP, ㄴ가/EF, ?/SF],
"question.position_list":[0, 7, 14, 18, 21, 27, 31, 34, 38, 44, 48, 54, 54, 60],
"passage.text": "알렉산더 메이그스 헤이그 2세(영어: Alexander Meigs Haig, Jr., 1924년 12월 2일 ~ 2010년 2월 20일)는 미국의 국무 장관을 지낸 미국의 군인, 관료 및 정치인이다. 로널드 레이건 대통령 밑에서 국무장관을 지냈으며, 리처드 닉슨과 제럴드 포드 대통령 밑에서 백악관 비서실장을 지냈다. 또한 그는 미국 군대에서 2번째로 높은 직위인 미국 육군 부참모 총장과 나토 및 미국 군대의 유럽연합군 최고사령관이었다.",
"passage.morp_list": [알렉산더/NNP, 메이그스/NNP, 헤이그/NNP, 2/SN, 세/NNB, (/SS, 영어/NNP, :/SP, Alexander/SL, Meigs/SL, Haig/SL, ,/SP, Jr/SL, ./SF, ,/SP, 1924/SN, 년/NNB, 12/SN, 월/NNB, 2/SN, 일/NNB, ~/SO, 2010/SN, 년/NNB, 2/SN, 월/NNB, 20/SN, 일/NNB,)/SS, 는/JX, 미국/NNP, 의/JKG, 국무/NNG, 장관/NNG, 을/JKO, 지내/VV, ㄴ/ETM, 미국/NNP, 의/JKG, 군인/NNG, ,/SP, 관료/NNG, 및/MAJ, 정치/NNG, 인/XSN, 이/VCP, 다/EF, ./SF, 로널드/NNP, 레이건/NNP, 대통령/NNG, 밑/NNG, 에서/JKB, 국무/NNG, 장관/NNG, 을/JKO, 지내/VV, 었/EP, 으며/EC, ,/SP, 리처드/NNP, 닉슨/NNP, 과/JC, 제럴드/NNP, 포드/NNP, 대통령/NNG, 밑/NNG, 에서/JKB, 백악관/NNP, 비서/NNG, 실/XSN, 장/XSN, 을/JKO, 지내/VV, 었/EP, 다/EF, ./SF, 또한/MAG, 그/NP, 는/JX, 미국/NNP, 군대/NNG, 에서/JKB, 2/SN, 번째/NNB, 로/JKB, 높/VA, 은/ETM, 직위/NNG, 이/VCP, ㄴ/ETM, 미국/NNP, 육군/NNG, 부참모/NNG, 총장/NNG, 과/JC, 나토/NNP, 및/MAJ, 미국/NNP, 군대/NNG, 의/JKG, 유럽연/NNP, 합군/NNG, 최고/NNG, 사령/NNG, 관/XSN, 이/VCP, 었/EP, 다/EF, ./SF, ...],
"passage.position_list: [0, 13, 26, 36, 37, 40, 41, 47, 49, 59, 65, 69, 71, 73, 74, 76, 80, 84, 86, 90, 91, 95, 97, 101, 105, 106, 110, 112, 115, 116, 120, 126, 130, 137, 143, 147, 150, 154, 160, 164, 170, 172, 179, 183, 189, 192, 195, 198, 200, 210, 220, 230, 233, 240, 246, 252, 256, 259, 262, 268, 270, 280, 286, 290, 300, 307, 317, 320, 327, 337, 343, 346, 349, 353, 356, 359, 362, 364, 371, 374, 378, 385, 391, 398, 399, 405, 409, 412, 416, 422, 422, 426, 433, 440, 450, 456, 460, 467, 471, 478, 484, 488, 494, 500, 504, 510, 516, 519, 522, 525, 528, 530, 537, 544, 551, 564, 574, 580, 584, 590, 593, 597, 603, 607, 614, 620, 623, 626, 632, 634, 644, 654, 661, 667, 674, 681, 687, 693, 696, 700, 703, 707, 711, 718, 730, 734, 740, 744, 750, 754, 760, 760, 764, 764, 767, 770, 772, 778, 782, 788, 792, 795, 801, 802, 808, 814, 816, 825, 829, 836, 842, 845, 846, 850, 854, 860, 861, 865, 868, 871]
"answer.text": "미국 육군 부참모 총장",
"answer.begin_morp": 91,
"answer.end_morp": 94
"answers.answer_start': 204
] }] }

텍스트와 인코딩

text='이순신은 조선 중기의 무신이었다.'

print(text) ➔ 이순신은 조선 중기의 무신이었다.

print(len(text)) ➔ 18

print(text[5:10]) ➔ 조선 중기

text_bytes = text.encode()

print(text_bytes) ➔
b'₩xec₩x9d₩xb4₩xec₩x88₩x9c₩xec₩x8b₩xa0₩xec₩x9d₩x80
₩xec₩xa1₩xb0₩xec₩x84₩xa0 ₩xec₩xa4₩x91₩xea₩xb8₩xb0₩xec₩x9d₩x98
₩xeb₩xac₩xb4₩xec₩x8b₩xa0₩xec₩x9d₩xb4₩xec₩x97₩x88₩xeb₩x8b₩xa4.'

print(len(text_bytes)) ➔ 46

print(text_bytes[13:26]) ➔ b'₩xec₩xa1₩xb0₩xec₩x84₩xa0
₩xec₩xa4₩x91₩xea₩xb8₩xb0'

print(text_bytes[13:26].decode()) ➔ 조선 중기

텍스트와 인코딩

- **형태소 분석 결과**
  - **text: 이순신은 조선 중기의 무신이었다.**


```
{
   "doc_id": "",    "DCT": "",    "category": "", "category_weight": 0.0,
    "title": {       "text": "",       "NE": ""    },
   "metaInfo": {},
   "paragraphInfo": [],
   "sentence": [
      {
         "id": 0.0,
         "reserve_str": "",
         "text": "이순신은 조선 중기의 무신이었다.",
         "morp": [
            { "id": 0,   "lemma": "이순신",   "type": "NNP",   "position": 0,   "weight": 0.0656567    },
            { "id": 1,   "lemma": "은",   "type": "JX",   "position": 9,   "weight": 0.106171    },
            { "id": 2,   "lemma": "조선",   "type": "NNP",   "position": 13,   "weight": 0.0909715    },
            { "id": 3,   "lemma": "중기",   "type": "NNG",   "position": 20,   "weight": 0.0802036    },
            { "id": 4,   "lemma": "의",   "type": "JKG",   "position": 26,   "weight": 0.078636    },
            { "id": 5,   "lemma": "무신",   "type": "NNG",   "position": 30,   "weight": 0.0342741    },
            { "id": 6,   "lemma": "이",   "type": "VCP",   "position": 36,   "weight": 0.0592733    },
            { "id": 7,   "lemma": "었",   "type": "EP",   "position": 39,   "weight": 0.05747    },
            { "id": 8,   "lemma": "다",   "type": "EF",   "position": 42,   "weight": 0.0842117    },
            { "id": 9,   "lemma": ".",   "type": "SF",   "position": 45,   "weight": 1.0    }
         ]
      }
   ],
   "entity": []
}
```


position = byte position

텍스트와 인코딩

- 형태소 분석 결과내 바이트 포지션과 형태소 인덱스
  - text: 이순신은 조선 중기의 무신이었다.
  - position_list: [0, 9, 13, 20, 26, 30, 36, 39, 42, 45]
    - len(position_list): 10
    - index 범위: 2~4
  - morp_list: ['이순신/NNP', '은/JX', '조선/NNP', '중기/NNG', '의/JKG', '무신/NNG', '이/VCP', '었/EP', '다/EF', './SF']
    - Len(morp_list): 10
    - Index 범위: 2~4
  - print(morp_list[2:4]) ➔ ['조선/NNP', '중기/NNG']

KorQuAD 1.0 형태소 정답 찾기

- 형태소 정답 찾기 예제
  - “answers” :[{'text': '미국 육군 부참모 총장', 'answer_start': 204}]
  - “passage.text”: “알렉산더 메이그스 헤이그 2세(영어: Alexander Meigs Haig, Jr., 1924년 12월 2일 ~ 2010년 2월 20일)는 미국의 국무 장관을 지낸 미국의 군인, 관료 및 정치인이다. 로널드 레이건 대통령 밑에서 국무장관을 지냈으며, 리처드 닉슨과 제럴드 포드 대통령 밑에서 백악관 비서실장을 지냈다. 또한 그는 미국 군대에서 2번째로 높은 직위인 미국 육군 부참모 총장과 나토 및 미국 군대의 유럽연합군 최고사령관이었다.”,

length=204

  - byte 포지션 찾기
    - 노란색 문자열의 바이트 길이: 426
    - 정답의 바이트 길이: 30
      - 단락에서 정답의 끝 위치: 426+30

KorQuAD 1.0 형태소 정답 찾기

- **형태소 정답 찾기 예제**

passage.position_list: [0, 13, 26, 36, 37, 40, 41, 47, 49, 59, 65, 69, 71, 73, 74, 76, 80, 84, 86, 90, 91, 95, 97, 101, 105, 106, 110, 112, 115, 116, 120, 126, 130, 137, 143, 147, 150, 154, 160, 164, 170, 172, 179, 183, 189, 192, 195, 198, 200, 210, 220, 230, 233, 240, 246, 252, 256, 259, 262, 268, 270, 280, 286, 290, 300, 307, 317, 320, 327, 337, 343, 346, 349, 353, 356, 359, 362, 364, 371, 374, 378, 385, 391, 398, 399, 405, 409, 412, 416, 422, 422, 426, 433, 440, 450, 456, 460, 467, 471, 478, 484, 488, 494, 500, 504, 510, 516, 519, 522, 525, 528, 530, 537, 544, 551, 564, 574, 580, 584, 590, 593, 597, 603, 607, 614, 620, 623, 626, 632, 634, 644, 654, 661, 667, 674, 681, 687, 693, 696, 700, 703, 707, 711, 718, 730, 734, 740, 744, 750, 754, 760, 760, 764, 764, 767, 770, 772, 778, 782, 788, 792, 795, 801, 802, 808, 814, 816, 825, 829, 836, 842, 845, 846, 850, 854, 860, 861, 865, 868, 871]

"passage.morp_list": [알렉산더/NNP, 메이그스/NNP, 헤이그/NNP, 2/SN, 세/NNB, (/SS, 영어/NNP, :/SP, Alexander/SL, Meigs/SL, Haig/SL, ,/SP, Jr/SL, ./SF, ,/SP, 1924/SN, 년/NNB, 12/SN, 월/NNB, 2/SN, 일/NNB, ~/SO, 2010/SN, 년/NNB, 2/SN, 월/NNB, 20/SN, 일/NNB,)/SS, 는/JX, 미국/NNP, 의/JKG, 국무/NNG, 장관/NNG, 을/JKO, 지내/VV, ㄴ/ETM, 미국/NNP, 의/JKG, 군인/NNG, ,/SP, 관료/NNG, 및/MAJ, 정치/NNG, 인/XSN, 이/VCP, 다/EF, ./SF, 로널드/NNP, 레이건/NNP, 대통령/NNG, 밑/NNG, 에서/JKB, 국무/NNG, 장관/NNG, 을/JKO, 지내/VV, 었/EP, 으며/EC, ,/SP, 리처드/NNP, 닉슨/NNP, 과/JC, 제럴드/NNP, 포드/NNP, 대통령/NNG, 밑/NNG, 에서/JKB, 백악관/NNP, 비서/NNG, 실/XSN, 장/XSN, 을/JKO, 지내/VV, 었/EP, 다/EF, ./SF, 또한/MAG, 그/NP, 는/JX, 미국/NNP, 군대/NNG, 에서/JKB, 2/SN, 번째/NNB, 로/JKB, 높/VA, 은/ETM, 직위/NNG, 이/VCP, ㄴ/ETM, 미국/NNP, 육군/NNG, 부참모/NNG, 총장/NNG, 과/JC, 나토/NNP, 및/MAJ, 미국/NNP, 군대/NNG, 의/JKG, 유럽연/NNP, 합군/NNG, 최고/NNG, 사령/NNG, 관/XSN, 이/VCP, 었/EP, 다/EF, ./SF, ...],

. begin_morp = 91
. end_morp=95 – 1 <-- array 인덱스 맞추기 위해 -1 필요

KorQuAD 1.0 형태소 정답 찾기(quiz3)

아래 명령어로 실행되게 수정하기
python build_data_step3_ai4001_quiz.py

- (python)형태소 정답 찾기(계속)
  - build_data_step3_ai4001_quiz.py
    - 실행방법 python build_data_step3_ai4001_quiz.py [입력파일] [출력파일]
      - 입력파일: build_data_step2_ai4001.py의 결과파일

```
import sys
import json

converted_data={}
converted_data_list=[]
with open(sys.argv[1], 'r') as f:

    for data in json.loads(' '.join(f.readlines()))['data']:

        passage_text = data['passage.text']
        passage_morp_list = data['passage.morp_list']
        passage_position_list = data['passage.position_list']

        answer = data['answers'][0]['text']
        answer_enc = answer.encode()
        len_answer_enc = len(answer_enc)

        answer_start = data['answers'][0]['answer_start']

        before_answer_text = quiz              ← 정답 위치 전까지 텍스트 추출
        before_answer_text_enc = quiz          ← 'before_answer_text' 바이트 인코딩
        len_before_answer_text_enc = quiz      ← 'before_answer_text' 바이트 인코딩 길이

        begin_morp=0
        end_morp=0

        end_byte_answer_position = len_before_answer_text_enc + quiz
```

파일 읽기

단락 형태소 정보

정답 및 정답의 byte 정보

단락에서 바이트로 정답 이전 범위 찾기(노란색 블록)

단락에서 바이트로 정답 마지막 범위 찾기

KorQuAD 1.0 형태소 정답 찾기(정답3)

- (python)형태소 정답 찾기(계속)
  - build_data_step3_ai4001.py
    - 실행방법 python build_data_step3_ai4001.py [입력파일] [출력파일]
      - 입력파일: build_data_step2_ai4001.py의 결과파일

```
import sys
import json

converted_data={}
converted_data_list=[]
with open(sys.argv[1], 'r') as f:

    for data in json.loads(' '.join(f.readlines()))['data']:

        passage_text = data['passage.text']
        passage_morp_list = data['passage.morp_list']
        passage_position_list = data['passage.position_list']

        answer = data['answers'][0]['text']
        answer_enc = answer.encode()
        len_answer_enc = len(answer_enc)

        answer_start = data['answers'][0]['answer_start']

        before_answer_text = passage_text[0:answer_start]
        before_answer_text_enc = before_answer_text.encode()
        len_before_answer_text_enc = len(before_answer_text_enc)

        begin_morp=0
        end_morp=0

        end_byte_answer_position = len_before_answer_text_enc + len_answer_enc
```

파일 읽기

단락 형태소 정보

정답 및 정답의 byte 정보

단락에서 바이트로 정답 이전 범위 찾기(노란색 블록)

단락에서 바이트로 정답 마지막 범위 찾기

KorQuAD 1.0 형태소 정답 찾기

- (python)형태소 정답 찾기

```
        if len_before_answer_text_enc in passage_position_list:
            begin_morp = passage_position_list.index(len_before_answer_text_enc)
        else:
            for idx, position in enumerate(passage_position_list):
                if len_before_answer_text_enc > position:
                    begin_morp = idx

        if end_byte_answer_position in passage_position_list:
            end_morp = passage_position_list.index(end_byte_answer_position)-1
        else:
            for idx, position in enumerate(passage_position_list):
                if end_byte_answer_position > position:
                    end_morp = idx

        data['answer.begin_morp'] = begin_morp
        data['answer.end_morp'] = end_morp
        data['answers.answer_start'] = answer_start
        data.pop('answers')

        converted_data_list.append(data)

with open(sys.argv[2], 'w') as wf:
    converted_data['data']=converted_data_list
    wf.write(json.dumps(converted_data, ensure_ascii=False))
```

정답 시작 형태소 인덱스

정답 종료 형태소 인덱스

KorQuAD 1.0 생성 데이터 디버깅

- 올바른 데이터 생성 확인을 위한 디버깅
  - 목적
    - 형태소 분석의 결과가 기계가 처리 하는 것이라 일부 오류가 포함됨
  - debug_data.py
    - 실행방법 python debug_data.py [입력파일]
      - 입력파일: build_data_step3_ai4001.py의 결과파일

KorQuAD 1.0 생성 데이터 디버깅

- **올바른 데이터 생성 확인을 위한 디버깅**

```
import sys
import json

converted_data={}
converted_data_list=[]
with open(sys.argv[1], 'r') as f:

    for data in json.loads(' '.join(f.readlines()))['data']:

        passage_text = data['passage.text']
        passage_position_list = data['passage.position_list']

        answer_text = data['answer.text']

        answer_morp_begin = data['answer.begin_morp']
        answer_morp_end = data['answer.end_morp']+1
        if len(passage_position_list) <= answer_morp_end:
            answer_morp_end -= 1

        begin_byte_position = passage_position_list[answer_morp_begin]
        end_byte_position = passage_position_list[answer_morp_end]
        passage_text_enc = passage_text.encode()
        gen_ans = passage_text_enc[begin_byte_position : end_byte_position].decode('utf-8','ignore').strip()

        if answer_text != gen_ans:
            print('['+answer_text+'] != ['+gen_ans+']')
```

파일 읽기

gold 정답

형태소 정답 인덱스 추출

단락에서 바이트로 정답 범위 찾기

gold 정답과 형태소 정답 비교

KorQuAD 1.0 생성 데이터 디버깅

- **debug_data.py 수행 결과**

gold 정답 != 형태소 기반 정답

[이슬람교 경전이나 길가메쉬 서사시] != [이슬람교 경전이나 길가메쉬 서사시등]
[1.25배에서 1.3배] != [1.25배에서 1.3배정도]
[미네르바] != [미네르바라]
[후지와라] != [후지와라는]
[14살] != [14살때]
[새정치국민회] != [새정치국민회의]
[뉴욕타임스] != [뉴욕타임스지]
[이려] != [이려는]
[치하포] != [치하포에서]
[3살] != [3살때]
[27살] != [27살때]

KorQuAD 1.0 생성 데이터 디버깅

{"id": "6534713-0-0", "question.text": "다음의 아고라 경제방에서 활동하던 대한민국의 유명 인터넷 논객의 별칭은 무엇인가?", "question.morp_list": ["다음/NNG", "의/JKG", "아고라/NNG", "경>제방/NNG", "에서/JKB", "활동/NNG", "하/XSV", "던/ETM", "대한민국/NNP", "의/JKG", "유명/NNG", "인터넷/NNG", "논객/NNG", "의/JKG", "별칭/NNG", "은/JX", "무엇/NP", "이/VCP", "ㄴ가/EF", "?/SF"], "question.position_list": [0, 6, 10, 20, 29, 36, 42, 45, 49, 61, 65, 72, 82, 88, 92, 98, 102, 108, 108, 114], "answer.text": "미네르바", "answer.begin_morp": 9, "answer.end_morp": 0, "passage.text": "박대성(1978년 8월 ~)은 미네르바라는 별명으로 포털 사이트 다음의 아고라 경제방에서 활동하던 대한민국의 유명 인터넷 논객이다. 2008년 하반기 리먼 브러더스의 부실과 환율 폭등 등, 대한민국 경제의 변동 추이를 정확히 예견하여 주목을 받았다. 11월에 절필을 선언하기도 했으나, 이후에도 글쓰기를 계속하다가 2009년 1월 초 허위사실을 유포한 혐의로 검찰에 체포 및 구속되었다. 미네르바의 변호인단(박찬종 변호사 등)은 1월 13일 구속적부심 심사를 청구했으나 기각되었다. 구속상태에서 수사를 받다가 2009년 4월 20일 1심 판결에서 무혐의로 무죄를 선고받고 풀려났다. 판결 이후 그는 온라인에 글을 올리는 것을 그만두겠다고 밝힌 적이 있다. 2009년 7월 2일부터 《일간스포츠》에 그의 칼럼이 주 2회 연재됐다.", "passage.morp_list": ["박대성/NNP", "(/SS", "1978/SN", "년/NNB", "8/SN", "월/NNB", "~/SO", ")/SS", "은/JX", "미네르바라/NNP", "는/JX", "별명/NNG", "으로/JKB", "포털/NNG", "사이트/NNG", "다음/NNG", "의/JKG", "아고라/NNG", "경제방/NNG", "에서/JKB", "활동/NNG", "하/XSV", "던/ETM", "대한민국/NNP", "의/JKG", "유명/NNG", "인터넷/NNG", "논객/NNG", "이/VCP", "다/EF", "./SF", "2008/SN", "년/NNB", "하/NNG", "반기/NNG", ">리먼/NNP", "브러더스/NNP", "의/JKG", "부실/NNG", "과/JC", "환율/NNG", "폭등/NNG", "등/NNB", ",/SP", "대한민국/NNP", "경제/NNG", "의/JKG", "변동/NNG", "추이/NNG", "를/JKO", "정확히/MAG", "예견/NNG", "하/XSV", "어/EC", "주목/NNG", "을/JKO", "받/VV", "았/EP", "다/EF", "./SF", "11/SN", "월/NNB", "에/JKB", "절필/NNG", "을/JKO", "선언/NNG", "하/XSV", "기/ETN", "도/JX", "하/VX", "었/EP", "으나/EC", ",/SP", "이후/NNG", "에/JKB", "도/JX", "글쓰/NNG", "기/NNG", "를/JKO", "계속/NNG", "하/XSV", "다가/EC", "2009/SN", "년/NNB", "1/SN", "월/NNB", "초/NNB", "허위/NNG", "사실/NNG", "을/JKO", "유포/NNG", "하/XSV", "ㄴ/ETM", "혐의/NNG", "로/JKB", "검찰/NNG", "에/JKB", "체포/NNG", "및/MAJ", "구속/NNG", "되/XSV", "었/EP", "다/EF", "./SF", "미네르바/NNP", "의/JKG", "변호/NNG", "인/XSN", "단/XSN", "(/SS", "박찬종/NNP", "변호/NNG", "사/XSN", "등/NNB", ")/SS", "은/JX", "1/SN", "월/NNB", "13/SN", "일/NNB", "구속/NNG", "적부/NNG", "심/NNG", ">심사/NNG", "를/JKO", "청구/NNG", "하/XSV", "었/EP", "으나/EC", "기각/NNG", "되/XSV", "었/EP", "다/EF", "./SF", "구속/NNG", "상태/NNG", "에서/JKB", "수사/NNG", "를/JKO", "받/VV", "다가/EC", "2009/SN", "년/NNB", "4/SN", "월/NNB", "20/SN", "일/NNB", "1/SN", "심/NNG", "판결/NNG", "에서/JKB", "무/NNG", "혐의/NNG", "로/JKB", "무죄/NNG", "를/JKO", "선고/NNG", "받/VV", "고/EC", "풀려나/VV", "았/EP", "다/EF", "./SF", "판결/NNG", "이후/NNG", "그/NP", "는/JX", "온라인/NNG", "에/JKB", "글/NNG", "을/JKO", "올리/VV", "는/ETM", "것/NNB", "을/JKO", "그만두/VV", "겠/EP", "다고/EC", "밝히/VV", "ㄴ/ETM", "적/NNB", "이/JKS", "있/VA", "다/EF", "./SF", "2009/SN", "년/NNB", "7/SN", "월/NNB", "2/SN", "일/NNB", "부터/JX", "《/SS", "일간스포츠/NNP", "》/SS", "에/JKB", "그/NP", "의/JKG", "칼럼/NNG", "이/JKS", "주/NNG", "2/SN", "회/NNB", "연재/NNG", "되/XSV", "었/EP", "다/EF", "./SF"], "passage.position_list": [0, 9, 10, 14, 18, 19, 23, 25, 26, 30, 45, 49, 55, 62, 69, 79, 85, 89, 99, 108, 115, 121, 124, 128, 140, 144, 151, 161, 167, 170, 173, 175, 179, 183, 186, 193, 200, 212, 216, 222, 226, 233, 240, 243, 245, 258, 264, 268, 275, 281, 285, 295, 301, 304, 308, 314, 318, 321, 324, 327, 329, 331, 334, 338, 344, 348, 354, 357, 360, 364, 364, 367, 373, 375, 381, 384, 388, 394, 397, 401, 407, 410, 417, 421, 425, 426, 430, 434, 440, 446, 450, 456, 456, 460, 466, 470, 476, 480, 487, 491, 497, 500, 503, 506, 508, 520, 524, 530, 533, 536, 537, 547, 553, 557, 560, 561, 565, 566, 570, 572, 576, 582, 588, 592, 598, 602, 608, 608, 611, 618, 624, 627, 630, 633, 635, 641, 647, 654, 660, 664, 667, 674, 678, 682, 683, 687, 689, 693, 694, 698, 704, 711, 714, 720, 724, 730, 734, 740, 743, 747, 753, 756, 759, 761, 768, 775, 778, 782, 791, 795, 798, 802, 808, 812, 815, 819, 828, 831, 838, 841, 845, 848, 852, 855, 858, 860, 864, 868, 869, 873, 874, 877, 884, 887, 902, 905, 909, 912, 916, 922, 926, 930, 931, 935, 941, 941, 944, 947], "answers.answer_start": 18, "answers.byte_begin": 30, "answers.byte_end": 42}