BIG_PY

버트를 활용한 키워드 추출

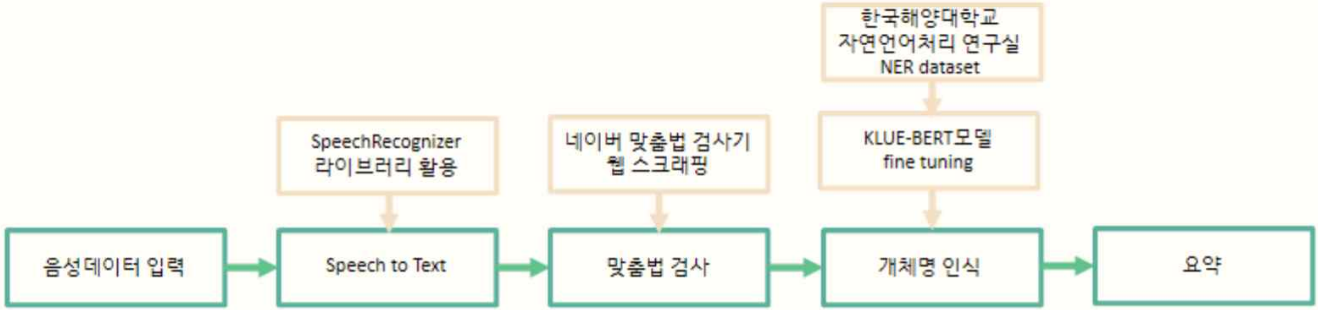
빅데이터전공 황현수
빅데이터전공 김강현
글로벌경영 신보비

목차페이지입니다

CHAPTER. 1

모델 도식화

모델 도식화

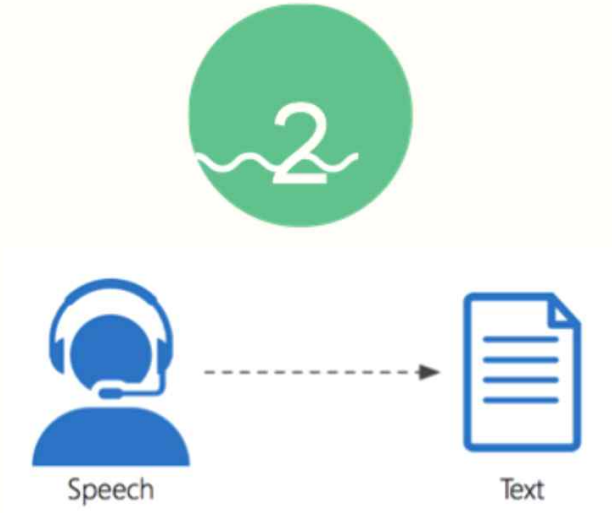

한국해양대학교
자연언어처리 연구실
NER dataset
SpeechRecognizer
라이브러리 활용
네이버 맞춤법 검사기
웹 스크래핑
KLUE-BERT모델
fine tuning
음성데이터 입력
Speech to Text
맞춤법 검사
개체명 인식
요약

CHAPTER. 2

SST

(speech to text)

사람이 말하는 음성 언어를 컴퓨터가
해석해 그 내용을 무자 데이터로 전환하는 기술



라이브러리 사용

speechrecognizer 라이브러리를 사용.

```
import speech_recognition as sr
```

인스턴스 생성

음성을 텍스트로 변경하기 위해
speechrecognizer 를 인스턴스화

이 때 객체화 된 라이브러리를 이용해 wav
소스에서 텍스트를 추출

```
r = sr.Recognizer()
```

오디오 소스에서 음성 기록

오디오 파일에서 초단위로 음성을 기록
이 기록된 오디오 정보에서 텍스트를 추출

```
with sr.AudioFile(sound_file) as source:
    audio = r.record(source)
text_1 = r.recognize_google(audio, language='ko-KR')
```

SST

speech to text

```
!pip install speechrecognition
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.de
Collecting speechrecognition
  Downloading SpeechRecognition-3.9.0-py2.py3-none-any.whl (32.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 32.8/32.8 MB 29.1 MB/s eta 0:00:00
Collecting requests>=2.26.0
  Downloading requests-2.28.2-py3-none-any.whl (62 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.8/62.8 kB 8.4 MB/s eta 0:00:00
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/li
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/p
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local
Installing collected packages: requests, speechrecognition
  Attempting uninstall: requests
    Found existing installation: requests 2.25.1
    Uninstalling requests-2.25.1:
      Successfully uninstalled requests-2.25.1
Successfully installed requests-2.28.2 speechrecognition-3.9.0
```

```
[ ] %cd /content/drive/MyDrive

/content/drive/MyDrive
```

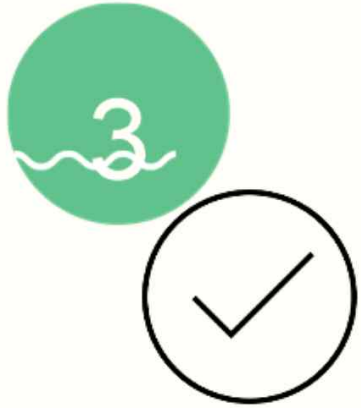
구글 드라이브에 음성파일을 업로드하고 speech_recognition을 진행

```
import speech_recognition as sr

sound_file = 'voice.wav'
def sst(sound_file):
  r = sr.Recognizer()
  with sr.AudioFile(sound_file) as source:
      audio = r.record(source)
  text_1 = r.recognize_google(audio, language='ko-KR')
  return text_1


sst(sound_file)
```

```
result2:
{   'alternative': [   {   'confidence': 0.87675762,
                           'transcript': '고려대학교 김광현은 11시 30분 점심을 먹었다'},
                       {'transcript': '고려대학교 김강현은 11시 30분 점심을 먹었다'},
                       {'transcript': '고려대학교 김광현은 11시 30분 먹었다'},
                       {'transcript': '고려대학교 김광현은 11시 30분'},
                       {'transcript': '고려대학교 김강현은 11시 30분 먹었다'}],
    'final': True}
'고려대학교 김광현은 11시 30분 점심을 먹었다'
```

CHAPTER. 3

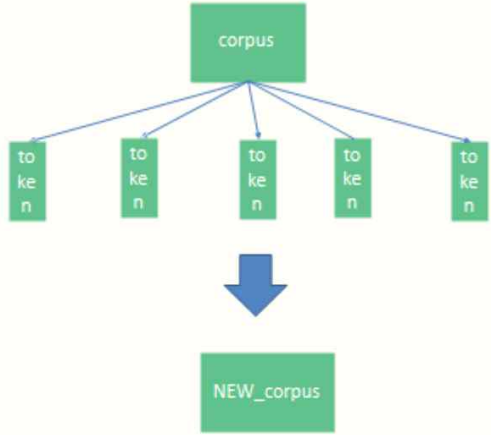
맞춤법 자동 검사



음성데이터를 텍스트로 변환하는 결과에서 맞춤법이 이상한 단어나 어색한 문장이 있으면 맞춤법 검사를 통해 올바르고 정확한 문장으로 수정

웹 스크래핑

웹 스크래핑을 통해 네이버의 맞춤법 검사기를 파싱하여 결과 도출



최대 500자까지 제공하기에 corpus에서 500자 단위로 토큰화

토큰 별 결과를 합쳐서 NEW_corpus 생성

맞춤법 검사

맞춤법 크롤링

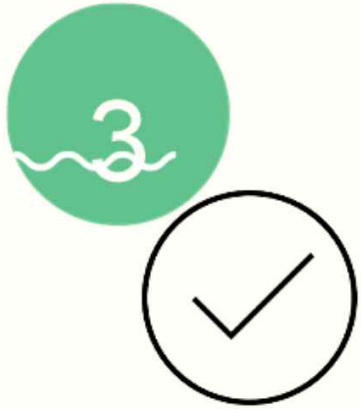
```
[44] def text_checker(error_text):
       if len(error_text) >= 500:
         ready_list = []
         check_list=[]
         while (len(error_text) >= 500):
           temp_str = error_text[:500]
           last_space = temp_str.rfind(' ')
           temp_str = error_text[0:last_space]
           ready_list.append(temp_str)

           error_text = error_text[last_space:]
         ready_list.append(error_text)

         for i in range(len(ready_list)):
           check_list.append(check(ready_list[i]).checked)

         sum_check_list = ' '.join(check_list)
         return sum_check_list

       else:
         return check(error_text).checked
```

네이버 맞춤법 크롤링을 통해 맞춤법 교정

```
[71] error_text = "여기 안암역앞에 사람이 쓰러져있어요 고려대 학생같은데빨리 와주세요"

[72] sent1=text_checker(error_text)
     sent1

     '여기 안암역 앞에 사람이 쓰러져있어요 고려대 학생 같은데 빨리 와주세요'
```

CHAPTER. 4

개체명 인식

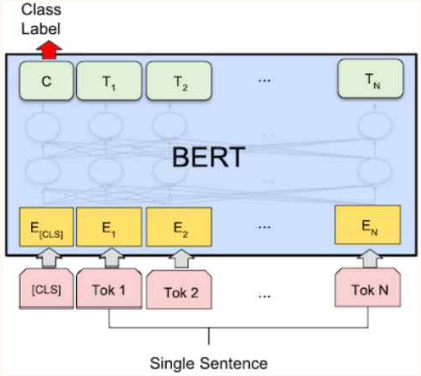


텍스트를 받아 토큰별로 개체명 Tag를 붙여 반환

개체명 인식

사용 모델

KLUE - BERT



KLUE-BERT는 한국어 자연어처리 벤치마크 데이터인 KLUE를 구글에서 발표한 언어 모델 BERT에 적용시킨 모델 ,
도두의 말뭉치, CC-100-Kor, 나무위키, 뉴스, 청원 등 문서에서 추출한 63GB의 데이터로 학습되었습니다.

NER

Tag(개체명)	설명
PER	사람이름
LOC	장소
ORG	기관명
POH	기타
DAT	날짜
TIM	시간
DUR	기간
MNY	통화
NOH	기타 수량표현

한국 해양대학교 개체명 코퍼스
- train data로 23033개의 문장을 사용
- test data로 930개의 문장을 사용

```
print(len(train_tagged_sentences))
print(len(test_tagged_sentences))
```

```
23033
930
```

개체명 인식

하이퍼 파라미터 설정

```
model = TFBertForTokenClassification("klue/bert-base", num_labels=tag_size)
optimizer = tf.keras.optimizers.Adam(learning_rate=5e-5)
model.compile(optimizer=optimizer, loss=compute_loss)
```

```
model.fit(
    X_train, y_train, epochs=3, batch_size=32,
    callbacks = [f1_score_report]
)
```

```
Epoch 3/3
30/30 [==============================] - 8s 275ms/step
 - f1: 88.05
              precision    recall  f1-score   support

         DAT       0.96      0.96      0.96       182
         DUR       0.69      0.72      0.71        50
         LOC       0.82      0.77      0.80       206
         MNY       0.79      0.95      0.86        20
         NOH       0.88      0.91      0.89      1007
         ORG       0.83      0.93      0.88       795
         PER       0.94      0.95      0.94       853
         PNT       0.82      0.77      0.79        60
         POH       0.65      0.64      0.65       214
         TIM       0.78      0.95      0.86        19

   micro avg       0.86      0.90      0.88      3406
   macro avg       0.82      0.86      0.83      3406
weighted avg       0.86      0.90      0.88      3406

720/720 [==============================] - 585s 813ms/step - loss: 0.0576
<keras.callbacks.History at 0x7fd1de25a0d0>
```

Parameter	설명	값
Optimizer	최적화 알고리즘	Adam
Learning Rate	학습률	5e-5
Epochs	학습 횟수	3
batch_size	학습 1step당 입력데이터 크기	32

paramerter 설정

Tag(개체명)	설명	f1_score
DAT	날짜	0.96
LOC	장소	0.80
ORG	기관명	0.88
PER	사람이름	0.94
TIM	시간	0.86

f1_score를 통해 요약 할 개체명 선별

개체명 인식

Tagging

```
[ ] error_text = "사장님 2월 9일 고려대학교의 김강현과 11시 30분에 하나스퀘어에서 회의를 진행 할 예정입니다."

[ ] sent1=text_checker(error_text)
    sent1

    '사장님 2월 9일 고려대학교의 김강현과 11시 30분에 하나스퀘어에서 회의를 진행할 예정입니다.'
```

```
[ ] def ner_def(sent1):
      test_samples = [sent1]
      result_list = ner_prediction(test_samples, max_seq_len=128, tokenizer=tokenizer, lang='ko')
      return result_list

    result_list = ner_def(sent1)
    result_list
```

텍스트를 모델에 적용하면 토큰과 Tag이 리스트로 반환된다.

```
100%|██████████| 1/1 [00:00<00:00, 697.19it/s]1

[[('사장', 'O'),
  ('님', 'O'),
  ('2', 'B-DAT'),
  ('월', 'I-DAT'),
  ('9', 'I-DAT'),
  ('일', 'I-DAT'),
  ('고려', 'B-ORG'),
  ('대학교', 'I-ORG'),
  ('의', 'O'),
  ('김강현', 'B-PER'),
  ('과', 'O'),
  ('11', 'B-TIM'),
  ('시', 'I-TIM'),
  ('30', 'I-TIM'),
  ('분', 'I-TIM'),
  ('에', 'O'),
  ('하나', 'B-LOC'),
  ('스퀘어', 'I-LOC'),
  ('에서', 'O'),
  ('회의', 'O'),
  ('를', 'O'),
  ('진행할', 'O'),
  ('예정', 'O'),
  ('입니다', 'O'),
  ('.', 'O')]]
```

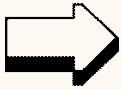
개체명 인식

키워드 추출

```
100%|██████████| 1/1 [00:00<00:00, 697.19it/s]1

[[('사장', 'O'),
  ('님', 'O'),
  ('2', 'B-DAT'),
  ('월', 'I-DAT'),
  ('9', 'I-DAT'),
  ('일', 'I-DAT'),
  ('고려', 'B-ORG'),
  ('대학교', 'I-ORG'),
  ('의', 'O'),
  ('김강현', 'B-PER'),
  ('과', 'O'),
  ('11', 'B-TIM'),
  ('시', 'I-TIM'),
  ('30', 'I-TIM'),
  ('분', 'I-TIM'),
  ('에', 'O'),
  ('하나', 'B-LOC'),
  ('스퀘어', 'I-LOC'),
  ('에서', 'O'),
  ('회의', 'O'),
  ('를', 'O'),
  ('진행할', 'O'),
  ('예정', 'O'),
  ('입니다', 'O'),
  ('.', 'O')]]
```

```
def keyword(ner_list):

    dat_list = []
    per_list = []
    org_list = []
    tim_list = []
    loc_list = []

    for i in range(len(ner_list[0])):
        word = ner_list[0][i][0]
        tag = ner_list[0][i][1]

        if len(tag) == 5:
            if tag[2:] == 'DAT':
                dat_list.append(word)

        if len(tag) == 5:
            if tag[2:] == 'ORG':
                org_list.append(word)
```

```
    tim = ''
    for i in tim_list:
        tim += i

    loc = ''
    for i in loc_list:
        loc += i

    return print(f"기관 : {org} ₩n장소 :
```



```
keyword(result_list)
```

```
기관 : 고려대학교
장소 : 하나스퀘어
날짜 : 2월9일
시간 : 11시30분
이름 : 김강현
```

Tagging 된 리스트에 키워드 추출 함수를 적용하여 요약

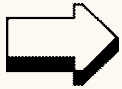
개체명 인식

최종 모델

▾ 최종

인풋에 보이스를 넣으면 맞춤법 검사 후 ner 이후 요약까지

```
#input
sound_file = 'voice.wav'


def main(sound_file):
  voice2text = sst(sound_file)#음성을 텍스트로
  sent1=text_checker(voice2text)#맞춤법 수정
  result_list = ner_def(sent1)#한문장 넣으면 ner 결과
  keyword(result_list)#결과 프린트

main(sound_file)
```



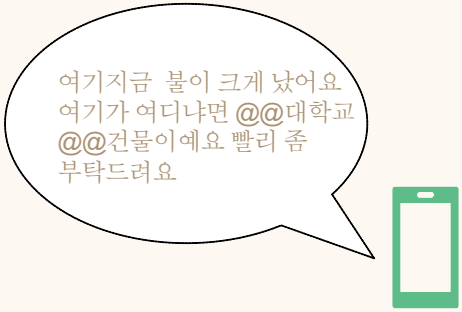
```
result2:
{   'alternative': [   {   'confidence': 0.73669213,
                           'transcript': '고려대학교 김광현은 부산에서 11시 30분 회의를 진행했다'},
                       {'transcript': '고려대학교 김광현은 11시 30분 회의를 진행했다'},
                       {'transcript': '고려대학교 김강현은 부산에서 11시 30분 회의를 진행했다'},
                       {'transcript': '고려대학교 김광현은 11시 30분에'},
                       {'transcript': '고려대학교 김강현은 11시 30분 회의를 진행했다'}],
    'final': True}
100%|██████████| 1/1 [00:00<00:00, 926.30it/s]1/1 [==============================] - 0s 57ms/step
기관 : 고려대학교
장소 : 부산
날짜 :
시간 : 11시30분
이름 : 김광현
```

지금까지 만든 함수들을 한번에 적용하는 main 함수

CHAPTER. 5

활용방안

긴급 신고



빠른 요약

기관 : @@대학교
장소 : @@건물
장소 : @@전자
시간 : 13:30분

가독성 높은 요약으로 빠른 대응 가능

일정요약

녹음

빠른 요약

회의 내용을 이용하여 자동으로 일정 요약