|  |  |
| --- | --- |
| 교육제목 | 데이터 기반 인공지능 시스템 엔지니어 양성 과정 |
| 교육일시 | 211126 |
| 교육장소 | YGL 학과장 및 자택(디스코드 사용 온라인 학습) |
| **교육내용** | |
| 1. 공백 기반 토큰화   corpus = \  """  in the days that followed i learned to spell in this uncomprehending way a great many words , among them pin , hat , cup and a few verbs like sit , stand and walk .  but my teacher had been with me several weeks before i understood that everything has a name .  one day , we walked down the path to the well house , attracted by the fragrance of the honeysuckle with which it was covered .  some one was drawing water and my teacher placed my hand under the spout .  as the cool stream gushed over one hand she spelled into the other the word water , first slowly , then rapidly .  i stood still , my whole attention fixed upon the motions of her fingers .  suddenly i felt a misty consciousness as of something forgotten a thrill of returning thought and somehow the mystery of language was revealed to me .  i knew then that w a t e r meant the wonderful cool something that was flowing over my hand .  that living word awakened my soul , gave it light , hope , joy , set it free !  there were barriers still , it is true , but barriers that could in time be swept away .  """  # HINT : split()을 사용하여 공백토큰화를 수행하세요.  tokens = corpus.split()  print("문장이 포함하는 Tokens:", tokens)   * 문장이 포함하는 Tokens: ['in', 'the', 'days', 'that', 'followed', 'i', 'learned', 'to', 'spell', 'in', 'this', 'uncomprehending', 'way', 'a', 'great', 'many', 'words', ',', 'among', 'them', 'pin', ',', 'hat', ',', 'cup', 'and', 'a', 'few', 'verbs', 'like', 'sit', ',', 'stand', 'and', 'walk', '.', 'but', 'my', 'teacher', 'had', 'been', 'with', 'me', 'several', 'weeks', 'before', 'i', 'understood', 'that', 'everything', 'has', 'a', 'name', '.', 'one', 'day', ',', 'we', 'walked', 'down', 'the', 'path', 'to', 'the', 'well', 'house', ',', 'attracted', 'by', 'the', 'fragrance', 'of', 'the', 'honeysuckle', 'with', 'which', 'it', 'was', 'covered', '.', 'some', 'one', 'was', 'drawing', 'water', 'and', 'my', 'teacher', 'placed', 'my', 'hand', 'under', 'the', 'spout', '.', 'as', 'the', 'cool', 'stream', 'gushed', 'over', 'one', 'hand', 'she', 'spelled', 'into', 'the', 'other', 'the', 'word', 'water', ',', 'first', 'slowly', ',', 'then', 'rapidly', '.', 'i', 'stood', 'still', ',', 'my', 'whole', 'attention', 'fixed', 'upon', 'the', 'motions', 'of', 'her', 'fingers', '.', 'suddenly', 'i', 'felt', 'a', 'misty', 'consciousness', 'as', 'of', 'something', 'forgotten', 'a', 'thrill', 'of', 'returning', 'thought', 'and', 'somehow', 'the', 'mystery', 'of', 'language', 'was', 'revealed', 'to', 'me', '.', 'i', 'knew', 'then', 'that', 'w', 'a', 't', 'e', 'r', 'meant', 'the', 'wonderful', 'cool', 'something', 'that', 'was', 'flowing', 'over', 'my', 'hand', '.', 'that', 'living', 'word', 'awakened', 'my', 'soul', ',', 'gave', 'it', 'light', ',', 'hope', ',', 'joy', ',', 'set', 'it', 'free', '!', 'there', 'were', 'barriers', 'still', ',', 'it', 'is', 'true', ',', 'but', 'barriers', 'that', 'could', 'in', 'time', 'be', 'swept', 'away', '.']  1. 형태소 기반 토큰화   형태소 : 뜻을 가진 가장 작은 말의 단위  한국어 형태소 분석기는 대표적으로 아래 두 가지가 사용 됨.   1. KoNLPy 파이썬 한국어 패키지 2. Kakao/khaiii   KoNLPy는 내부적으로 5가지의 형태소 분석 Class를 포함하고 있음. Khaiii까지 총 6개나 되는 형태소 분석기들은 특수한 문장(띄어쓰기 X, 오탈자) 처리 성능, 속도 측면에서 차이를 보임.   1. 형태소 기반 – mecab   from konlpy.tag import Mecab  mecab = Mecab()  # None자리에 문장을 넣어보고 토큰화 결과를 출력해보세요.  # 예시문장 : 자연어처리가너무재밌어서밥먹는것도가끔까먹어요  print(mecab.morphs('자연어처리가너무재밌어서밥먹는것도가끔까먹어요'))   * ['자연어', '처리', '가', '너무', '재밌', '어서', '밥', '먹', '는', '것', '도', '가끔', '까먹', '어요']   from konlpy.tag import Hannanum,Kkma,Komoran,Mecab,Okt  tokenizer\_list = [Hannanum(),Kkma(),Komoran(),Mecab(),Okt()]  kor\_text = '코로나바이러스는 2019년 12월 중국 우한에서 처음 발생한 뒤 전 세계로 확산된, 새로운 유형의 호흡기 감염 질환입니다.'  for tokenizer in tokenizer\_list:  print('[{}] \n{}'.format(tokenizer.\_\_class\_\_.\_\_name\_\_, tokenizer.pos(kor\_text)))   * [Hannanum] * [('코로나바이러스', 'N'), ('는', 'J'), ('2019년', 'N'), ('12월', 'N'), ('중국', 'N'), ('우한', 'N'), ('에서', 'J'), ('처음', 'M'), ('발생', 'N'), ('하', 'X'), ('ㄴ', 'E'), ('뒤', 'N'), ('전', 'N'), ('세계', 'N'), ('로', 'J'), ('확산', 'N'), ('되', 'X'), ('ㄴ', 'E'), (',', 'S'), ('새롭', 'P'), ('은', 'E'), ('유형', 'N'), ('의', 'J'), ('호흡기', 'N'), ('감염', 'N'), ('질환', 'N'), ('이', 'J'), ('ㅂ니다', 'E'), ('.', 'S')] * [Kkma] * [('코로나', 'NNG'), ('바', 'NNG'), ('이러', 'MAG'), ('슬', 'VV'), ('는', 'ETD'), ('2019', 'NR'), ('년', 'NNM'), ('12', 'NR'), ('월', 'NNM'), ('중국', 'NNG'), ('우', 'NNG'), ('하', 'XSV'), ('ㄴ', 'ETD'), ('에', 'VV'), ('서', 'ECD'), ('처음', 'NNG'), ('발생', 'NNG'), ('하', 'XSV'), ('ㄴ', 'ETD'), ('뒤', 'NNG'), ('전', 'NNG'), ('세계', 'NNG'), ('로', 'JKM'), ('확산', 'NNG'), ('되', 'XSV'), ('ㄴ', 'ETD'), (',', 'SP'), ('새', 'NNG'), ('롭', 'XSA'), ('ㄴ', 'ETD'), ('유형', 'NNG'), ('의', 'JKG'), ('호흡기', 'NNG'), ('감염', 'NNG'), ('질환', 'NNG'), ('이', 'VCP'), ('ㅂ니다', 'EFN'), ('.', 'SF')] * [Komoran] * [('코로나바이러스', 'NNP'), ('는', 'JX'), ('2019', 'SN'), ('년', 'NNB'), ('12월', 'NNP'), ('중국', 'NNP'), ('우', 'NNP'), ('한', 'NNP'), ('에서', 'JKB'), ('처음', 'NNG'), ('발생', 'NNG'), ('하', 'XSV'), ('ㄴ', 'ETM'), ('뒤', 'NNG'), ('전', 'MM'), ('세계로', 'NNP'), ('확산', 'NNG'), ('되', 'XSV'), ('ㄴ', 'ETM'), (',', 'SP'), ('새롭', 'VA'), ('ㄴ', 'ETM'), ('유형', 'NNP'), ('의', 'JKG'), ('호흡기', 'NNG'), ('감염', 'NNP'), ('질환', 'NNG'), ('이', 'VCP'), ('ㅂ니다', 'EF'), ('.', 'SF')] * [Mecab] * [('코로나', 'NNP'), ('바이러스', 'NNG'), ('는', 'JX'), ('2019', 'SN'), ('년', 'NNBC'), ('12', 'SN'), ('월', 'NNBC'), ('중국', 'NNP'), ('우한', 'NNP'), ('에서', 'JKB'), ('처음', 'NNG'), ('발생', 'NNG'), ('한', 'XSV+ETM'), ('뒤', 'NNG'), ('전', 'NNG'), ('세계', 'NNG'), ('로', 'JKB'), ('확산', 'NNG'), ('된', 'XSV+ETM'), (',', 'SC'), ('새로운', 'VA+ETM'), ('유형', 'NNG'), ('의', 'JKG'), ('호흡기', 'NNG'), ('감염', 'NNG'), ('질환', 'NNG'), ('입니다', 'VCP+EF'), ('.', 'SF')] * [Okt] * [('코로나바이러스', 'Noun'), ('는', 'Josa'), ('2019년', 'Number'), ('12월', 'Number'), ('중국', 'Noun'), ('우한', 'Noun'), ('에서', 'Josa'), ('처음', 'Noun'), ('발생', 'Noun'), ('한', 'Josa'), ('뒤', 'Noun'), ('전', 'Noun'), ('세계', 'Noun'), ('로', 'Josa'), ('확산', 'Noun'), ('된', 'Verb'), (',', 'Punctuation'), ('새로운', 'Adjective'), ('유형', 'Noun'), ('의', 'Josa'), ('호흡기', 'Noun'), ('감염', 'Noun'), ('질환', 'Noun'), ('입니다', 'Adjective'), ('.', 'Punctuation')]  1. 한글-영어 코퍼스(실제 데이터로)로 토큰화 수행하기   import os  import matplotlib.pyplot as plt  import tensorflow as tf  import numpy as np  path\_to\_file = '/content/drive/MyDrive/Colab Notebooks/영우4기\_자연어/dataset/korean-english-park.train.ko'  with open(path\_to\_file, "r", encoding='utf-8') as f:  raw = f.read().splitlines()  print("Data Size", len(raw))  print("Example:")  for sen in raw[0:100][::20]: print(">>", sen)  min\_len = 999  max\_len = 0  sum\_len = 0  for sen in raw:  length = len(sen)  if min\_len > length : min\_len = length  if max\_len < length : max\_len = length  sum\_len += length  print("문장의 최단 길이 :",min\_len)  print("문장의 최장 길이 :", max\_len)  print("문장의 평균 길이 :", sum\_len//len(raw))  sentence\_length = np.zeros((max\_len), dtype=np.int)  for sen in raw:  sentence\_length[len(sen)-1] += 1  plt.bar(range(max\_len), sentence\_length, width =1.0)  plt.title("Sentence Length Distribution")  plt.show()  def check\_sentence\_with\_length(raw, length):  count = 0  for sen in raw:  if len(sen) == length:  print(sen)  count += 1  if count > 100: return  check\_sentence\_with\_length(raw, 1)  for idx, \_sum in enumerate(sentence\_length):  if \_sum > 1500:  print("Outlier Index :", idx+1)  check\_sentence\_with\_length(raw, 11)  min\_len = 999  max\_len = 0  sum\_len = 0  cleaned\_corpus = list(set(raw)) # set 사용해서 중복 제거!!  print("Data Size :", len(cleaned\_corpus))  for sen in cleaned\_corpus:  length = len(sen)  if min\_len > length : min\_len = length  if max\_len < length : max\_len = length  sum\_len += length  print("문장의 최단 길이 :",min\_len)  print("문장의 최장 길이 :", max\_len)  print("문장의 평균 길이 :", sum\_len//len(cleaned\_corpus))  sentence\_length = np.zeros((max\_len), dtype=np.int)  for sen in cleaned\_corpus: # 중복이 제거된 코퍼스  sentence\_length[len(sen)-1] += 1  plt.bar(range(max\_len), sentence\_length, width =1.0)  plt.title("Sentence Length Distribution")  plt.show()  max\_len = 150  min\_len = 10  filtered\_corpus = [s for s in cleaned\_corpus if (len(s) < max\_len) & (len(s) >= min\_len)]  sentence\_length = np.zeros((max\_len), dtype=np.int)  for sen in filtered\_corpus: # 필터가 적용된 코퍼스  sentence\_length[len(sen)-1] += 1  plt.bar(range(max\_len), sentence\_length, width =1.0)  plt.title("Sentence Length Distribution")  plt.show()   1. 공백 기반 토큰화   # Quiz : 정제된 데이터를 공백 기반으로 토큰화하여 list에 저장한 후, tokenize()함수를 사용해 단어 사전과 Tensor데이터를 얻으세요!  # 그리고 단어 사전의 크기를 확인하세요.  def tokenize(corpus):  tokenizer = tf.keras.preprocessing.text.Tokenizer(filters='')  tokenizer.fit\_on\_texts(corpus) # 문자 -> 리스트  tensor = tokenizer.texts\_to\_sequences(corpus) # 텍스트 -> 시퀀스  tensor = tf.keras.preprocessing.sequence.pad\_sequences(tensor, padding='post') # 패딩처리 , padding='post'  return tensor, tokenizer  # 정제된 데이터를 공백 기반으로 토큰화하여 저장하는 코드를 직접 작성해보세요.  split\_corpus = []  for kor in filtered\_corpus:  split\_corpus.append(kor.split())  # 코드작성해주세요.  split\_tensor, split\_tokenizer = tokenize(split\_corpus)  print("Split Vocab Size :", len(split\_tokenizer.index\_word))  for idx, word in enumerate(split\_tokenizer.word\_index):  print(idx, ":", word)  if idx > 10: break   1. 형태소 기반 토큰화   # 위에서 사용한 코드를 활용해 Mecab단어 사전을 만들어주세요.  # Hint mecab.morphs() --> 형태소 분석 수행  def mecab\_split(sentence): # 형태소 분석역할  return mecab.morphs(sentence)  mecab\_corpus = []  # mecab 단어장 생성  for kor in filtered\_corpus:  mecab\_corpus.append(mecab\_split(kor)) # 형태소 분석 '텍스트'형태  mecab\_tensor, mecab\_tokenizer = tokenize(mecab\_corpus) # 텍스트 ---> 숫자  print("Mecab Vocab size :", len(mecab\_tokenizer.index\_word))   1. 퀴즈   1) tokenizer.sequences\_to\_texts() 함수를 사용하여 Decoding<br> 2) tokenizer.index\_word 를 사용하여 Decoding  두 가지 방법으로 mecab\_tensor[100] 을 원문으로 되돌려 보세요! (여기서 띄어쓰기는 고려하지 않습니다!)  # Case 1 : mecab\_tokenizer.sequences\_to\_texts()  # Case 1  texts = mecab\_tokenizer.sequences\_to\_texts([mecab\_tensor[100]]) # 코드 작성  print(texts[0])  # Case 2 : mecab\_tokenizer.index\_word[]  sentence = ""  for w in mecab\_tensor[100]:  if w == 0: continue  sentence += mecab\_tokenizer.index\_word[w] + " "# 코드 작성  print(sentence) | |