|  |  |
| --- | --- |
| 교육제목 | 데이터 기반 인공지능 시스템 엔지니어 양성 과정 |
| 교육일시 | 2021-12-02 |
| 교육장소 | YGL 학과장 및 자택(디스코드 사용 온라인 학습) |
| **교육내용** | |
| 1. 워드넷   import pandas as pd  import numpy as np  import urllib.request  from sklearn.feature\_extraction.text import CountVectorizer  from sklearn.feature\_extraction.text import TfidfVectorizer  import nltk  from nltk.corpus import stopwords  from nltk.stem import WordNetLemmatizer # 접사 (affix) ; 단어의 추가적의 의미를 주는 부분  # 접사 (affix) ; 단어의 추가적의 의미를 주는 부분  # 어간 (stem) ; 단어의 의미를 담고 있는 단어의 핵심 부분  # cats ; cat(어간) -s(접사)  # fox ;  nltk.download('wordnet')  lemmatizer = WordNetLemmatizer()  words = ['policy', 'doing', 'organization', 'have', 'going', 'love', 'lives', 'fly', 'dies', 'watched', 'has', 'starting']  print([lemmatizer.lemmatize(w) for w in words])  lemmatizer.lemmatize('dies', 'v')  lemmatizer.lemmatize('watched', 'v')  nltk.download('punkt')  nltk.download('wordnet')  nltk.download('stopwords')  urllib.request.urlretrieve("https://raw.githubusercontent.com/franciscadias/data/master/abcnews-date-text.csv",  filename="/content/abcnews-data-text.csv")  data = pd.read\_csv('/content/abcnews-data-text.csv', error\_bad\_lines=False)  data  text = data[['headline\_text']]  text.nunique()  text.drop\_duplicates(inplace=True) #중복된 요소를 제거  text = text.reset\_index(drop=True)  print(len(text))   * 데이터 정제 및 정규화   text['headline\_text'] = text.apply(lambda row:nltk.word\_tokenize(row['headline\_text']), axis=1)  stop\_words = stopwords.words('english')  text['headline\_text'] = text['headline\_text'].apply(lambda x : [word for word in x if word not in (stop\_words)])  text['headline\_text'] = text['headline\_text'].apply(lambda x: [WordNetLemmatizer().lemmatize(word, pos='v') for word in x])  text = text['headline\_text'].apply(lambda x: [word for word in x if len(word)>2])  detokenized\_doc = []  for i in range(len(text)):  t = ' '.join(text[i])  detokenized\_doc.append(t)  train\_data = detokenized\_doc  # DTM  c\_vectorizer = CountVectorizer(stop\_words='english', max\_features = 5000)  document\_term\_matrix = c\_vectorizer.fit\_transform(train\_data)  print('행렬의 크기 : ', document\_term\_matrix.shape)  tfidf\_vectorizer = TfidfVectorizer(stop\_words='english', max\_features= 5000)  tf\_idf\_matrix = tfidf\_vectorizer.fit\_transform(train\_data)  print('행렬의 크기 :', tf\_idf\_matrix.shape)   1. Abc 뉴스데이터로 word2vec   from nltk.corpus import abc  import nltk  nltk.download('abc')  nltk.download('punkt')  corpus = abc.sents()  print(corpus[:3])  print('코퍼스의 크기 :', len(corpus))  from gensim.models import Word2Vec  model = Word2Vec(sentences= corpus, size = 100, window=5, min\_count=5, workers=4, sg=0) # cbow  model\_result = model.wv.most\_similar("man")  from gensim.models import KeyedVectors  model.wv.save\_word2vec\_format('./w2v')  loaded\_model = KeyedVectors.load\_word2vec\_format("./w2v")  print("모델 load완료!")  model\_result = loaded\_model.wv.most\_similar("man")  print(model\_result)  loaded\_model.most\_similar('overacting')  loaded\_model.most\_similar('memory')  ‘   1. 한국어 word2vec 만들기   !pip install konlpy  import pandas as pd  import matplotlib.pyplot as plt  import urllib.request  from gensim.models.word2vec import Word2Vec  from konlpy.tag import Okt  urllib.request.urlretrieve("https://raw.githubusercontent.com/e9t/nsmc/master/ratings.txt", filename="ratings.txt")  train\_data = pd.read\_table('ratings.txt')  print(len(train\_data)) #리뷰 갯수 출력  # Null값 존재 유무  print(train\_data.isnull().values.any())  train\_data = train\_data.dropna(how='any') #null값이 존재하는 행 제거  print(train\_data.isnull().values.any()) #Null값이 존재하는지 확인!  # 정규 표현식을 통한 한글 외 문자 제거  train\_data['document'] = train\_data['document'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣 ]", "")  # 불용어 정의  stopwords = ['의','가','이','은','들','는','좀','잘','걍','과','도','를','으로','자','에','와','한','하다']  okt = Okt()  tokenized\_data = []  for sentence in train\_data['document']:  temp\_x = okt.morphs(sentence, stem=True) # 토큰화  temp\_x = [word for word in temp\_x if not word in stopwords] #불용어 제거  tokenized\_data.append(temp\_x)  # 리뷰 길이 분포 확인  print('리뷰의 최대 길이 :', max(len(l) for l in tokenized\_data))  print('리뷰의 평균 길이 :', sum(map(len, tokenized\_data))/len(tokenized\_data))  plt.hist([len(s) for s in tokenized\_data], bins=50)  plt.xlabel('length of samples')  plt.ylabel('number of samples')  plt.show()  from gensim.models import Word2Vec  model = Word2Vec(sentences=tokenized\_data, size=100, window=5, min\_count=5, workers=4, sg=0) # CBOW  model.wv.vectors.shape  print(model.wv.most\_similar("최민식"))   1. 사전 훈련된 워드 임베딩 (한국어)   import gensim  model = gensim.models.Word2Vec.load('/content/drive/MyDrive/Colab Notebooks/영우4기\_자연어 (10일완성)/dataset/ko.bin')  result = model.wv.most\_similar("강아지")  print(result)   1. 사전 훈련된 워드 임베딩 (영어)   import gensim  model = gensim.models.KeyedVectors.load\_word2vec\_format('/content/drive/MyDrive/Colab Notebooks/영우4기\_자연어 (10일완성)/dataset/GoogleNews-vectors-negative300.bin.gz', binary=True)  print(model.vectors.shape) # 3백만개의 단어와 각단어차원이 300  print(model.similarity('this', 'is'))  print(model.similarity('post','book'))  print(model['book'])   1. Wikipedia word2Vec   !pip install wikiextractor  # Colab에 Mecab 설치  !git clone https://github.com/SOMJANG/Mecab-ko-for-Google-Colab.git  %cd Mecab-ko-for-Google-Colab  !bash install\_mecab-ko\_on\_colab190912.sh  !wget <https://dumps.wikimedia.org/kowiki/latest/kowiki-latest-pages-articles.xml.bz2>  !python -m wikiextractor.WikiExtractor kowiki-latest-pages-articles.xml.bz2  # 위키익스트랙터를 사용하여 위키피디아 덤프를 파싱한다.  import os  import re  os.listdir('text')  def list\_wiki(dirname):  filepaths = []  filenames = os.listdir(dirname)  for filename in filenames:  filepath = os.path.join(dirname, filename)  if os.path.isdir(filepath):  # 재귀 함수  filepaths.extend(list\_wiki(filepath))  else:  find = re.findall(r"wiki\_[0-9][0-9]", filepath)  if 0 < len(find):  filepaths.append(filepath)  return sorted(filepaths)  filepaths = list\_wiki('text')  with open("output\_file.txt", "w") as outfile:  for filename in filepaths:  with open(filename) as infile:  contents = infile.read()  outfile.write(contents)  f = open('output\_file.txt', encoding="utf8")  i=0  while True:  line = f.readline()  if line != '\n':  i = i+1  print("%d번째 줄 :" %i+line)  if i==10:  break  f.close() | |