|  |  |
| --- | --- |
| 교육 제목 | AI 프로젝트 기반 S/W 전문가 양성과정 |
| 교육 일시 | 22.1.25. |
| 교육 장소 | 집 |
| **교육 내용** | |
| 오전 | 1. one hot encoding 실습 2. 데이터 만들기 3. 라벨링 4. 중복 단어 제거 5. 각 번호별 단어 6. text to id 만들고 7. onehot\_matrix = np.eye(len(word\_to\_id) # 비정방단위행렬 만들기 🡪 one hot encoding은 한줄에 1이 하나씩 있어야 하기 때문, 차원수 = dict key 개수 그 외 np.identity() 🡪 정방단위행렬 8. Boost Camp Embedding 시청각 수업 9. WORD2VEC from gensim.models import Word2Vec # 단어들을 벡터(norm)연산 가능하게 model = Word2Vec(sentences = corpus, size = 100, window = 5, min\_count = 5, workers = 4, sg = 0) # sg = cbow or skipgram select  model.wv.most\_similar("str") # str과 유사한 값을 가진 단어와 정도를 가져옴 model.wv.save\_word2vec\_format("./w2v.word2vec") # model 저장 |
| 오후 | 1. 한국어 word2vec 만들기 2. 영화평점 데이터셋 불러오기 3. 전처리 4. 한글만 따오기 train\_data["document"] = train\_data["document"].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣]", " ") 5. 불용어 정의 stopwords = ['의','가','이','은','들','는','좀','잘','걍','과','도','를','으로','자','에','와','한','하다'] 6. for문 안에서 tokenizer, 불용어 제거 7. maxlen, meanlen 확인 🡪 sum(map(len, tokenized\_data)) / len(tokenized\_data)) 8. 맷플롯립으로 그려보기 9. 유사도 확인해보기 from genism.models import Word2Vec model = word2Vec(sentences = tokenized\_data, size = 100, window = 5, min\_count = 5, workers = 4, sg = 0) model.wv.most\_similar(“str”) 🡪 결과 나옴 10. 영어 word2vec 만들기 import genism genism.models.KeyedVectors.load\_word2vec\_format(“구글 드라이브 경로”) "/content/drive/MyDrive/GoogleNews-vectors-negative300.bin.gz", binary = True # 바이너리로 읽어와야 함 model.vectors.shape 🡪 3000000, 300차원 model.similarity(“str1”, “str2”) # str1, str2 유사정도를 보여줌 model[“book”] # 300차원 결과가 나옴 model.most\_similar(“memory”) 11. 위키피디아 Data Word2Vec 12. !pip install wikiextractor 🡪 위키피디아 데이터 가져오기 13. Mecab 설치 14. !wget <https://dumps.wikimedia.org/kowiki/latest/kowiki-latest-pages-articles.xml.bz2>, !python -m wikiextractor.WikiExtractor kowiki-latest-pages-articles.xml.bz2 🡪 완료되면 repository에 폴더, 파일이 쌓여있음 15. 파일 목록 리스트 만들고 16. 파일을 전부 하나로 병합하고 17. 불러와서 온전한지 확인 18. 형태소분석 🡪 from tqdm import tqdm로 하려고 했는데 코랩 터져서 mecab으로 했음 19. mecab.morphs() 20. Word2Vec 학습 from genism.models import Word2Vec model = Word2Vec(result, size=100, window=5, min\_count=5, workers=4, sg=0) # CBOW 21. model.wv.most\_similar(“str”) 🡪 str과 유사한 값을 가진 단어와 정도를 가져옴 |