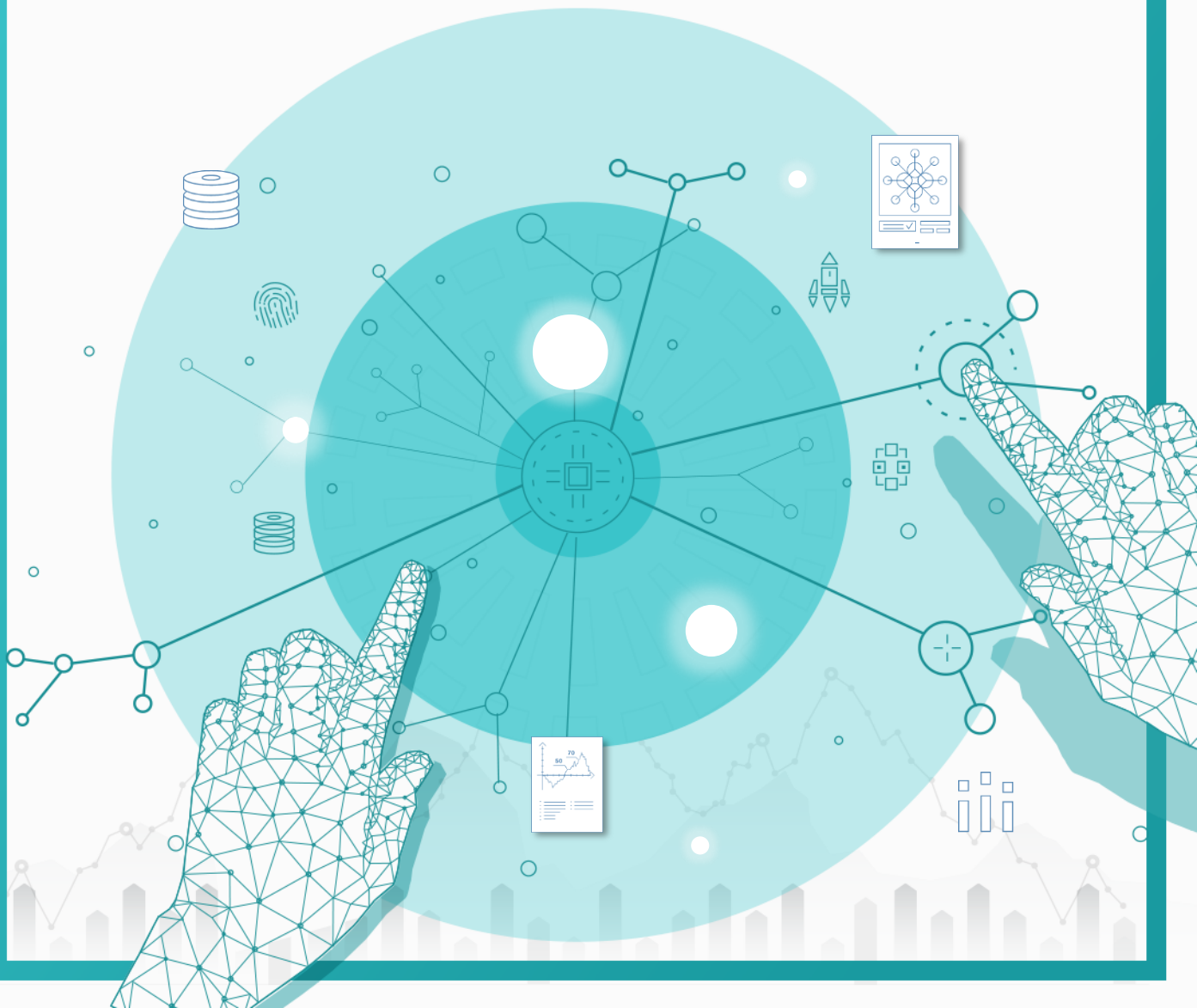
한국기술교육대학교
온라인평생교육원

파이썬을 활용한 인공지능 자연어 처리(실습)

자연어 처리를 위한 Word2Vec



자연어 처리를 위한 Word2Vec

학습 목표

1. Word2Vec 모델을 적용할 수 있다.
2. Pre-trained된 Word2Vec 모델을 활용할 수 있다.

학습 내용

1. Word2Vec 모델 적용
2. Pre-trained Word2Vec 모델 활용

1. Word2Vec 모델 적용

1) Word2Vec 모델

(1) Word2Vec 모델 적용을 위한 데이터 세트
| 뉴스 데이터 세트

- Google Colab의 파일 업로드 코드를 실행하여 데이터 세트 업로드

```
from google.colab import files
import os

data_dir = 'data'

if not os.path.exists(data_dir):
    os.mkdir(data_dir)
os.chdir(data_dir)
files.upload()
os.chdir('..')
```

파일 선택 선택된 파일 없음 Cancel upload

- news.csv 파일 선택 및 업로드 실행

파일 선택 news.csv

- **news.csv**(application/vnd.ms-excel) - 4037045 bytes, last modified:

Saving news.csv to news.csv

- Pandas 패키지를 이용하여 CSV 파일 읽기 및 '스포츠' 카테고리 뉴스 텍스트로 병합

```
df_news = pd.read_csv('./data/news.csv')
text = ' '.join(df_news[df_news.category == 7].news.values)
```

1. Word2Vec 모델 적용

1) Word2Vec 모델

(1) Word2Vec 모델 적용을 위한 데이터 세트 | 뉴스 데이터 세트 소개

- 뉴스 카테고리(정치, 경제, 스포츠)와 일련번호, 뉴스 데이터로 구성

	category	num	news
1248	7	7100	남자축구 뉴질랜드 대항전에서 패배한 뒤 상대편 선수의 악수를 거부한 이동경(울산) ...
1249	7	7101	올림픽에서 정치적 의사 표현은 금지되어 있지만 도쿄올림픽에서 스스로 경기를 포기함으...
1250	7	7102	대한민국 축구 대표팀 선수 이동경이 2020 도쿄올림픽 남자축구 조별리그 뉴질랜드전...
1251	7	7103	도쿄올림픽 개막식이 열린 23일 도쿄 시부야 스카이 건물에서 바라본 올림픽스타디움에...
1252	7	7104	수영 황선우, 체조 여서정, 탁구 신유빈, 양궁 김재덕... 왼쪽 위부터 1996년...
...	...	...	...
1341	7	7193	한국 여자 양궁 대표팀의 막내 안산(20·광주여대)이 올림픽 양궁 사상 첫 3관왕 ...
1342	7	7194	- ..'도쿄 전초전' 에비앙 첫날 .. ..세계 2위 고진영 1오버 중...
1343	7	7195	2승1패 돼도 골득실 따져야 .. ..역대 최상의 조에서 만난 최약체.....
1344	7	7196	신한은행은 2020 도쿄올림픽 야구 국가대표팀 선전을 기원하며 메타버스 구장 ..'...
1345	7	7197	도쿄올림픽 개막을 앞두고 스포츠 종목 협회장을 맡고 있는 재계 총수들이 선수들을 격...

※출처: 공공데이터포털, 한국언론진흥재단_뉴스빅데이터_메타데이터_올림픽, 2021, https://www.data.go.kr

1. Word2Vec 모델 적용

1) Word2Vec 모델

(2) 전처리, 형태소 분석 및 명사 추출
| 텍스트 전처리

- 문자만 추출, '\n' 문자 제거

```
import re

cleaned_text = re.sub('[^\w\s]','',text)
cleaned_text = re.sub('\n',' ',cleaned_text)
```

- 형태소 분석 및 명사 추출(2글자 이상)

```
from konlpy.tag import Okt
tagger = Okt()

noun = tagger.nouns(cleaned_text)
noun2more = [ele for ele in noun if len(ele) > 1]
```

1. Word2Vec 모델 적용

1) Word2Vec 모델

(3) gensim 패키지를 이용한 Word2Vec 모델 생성

- genism 패키지 import 및 Word2Vec 모델 생성

```
from gensim.models import Word2Vec

model=Word2Vec([noun2more], sg=1, size=100, window=3, min_count=3)
```

- Word2Vec 모델 생성 생성에 필요한 주요 매개변수

```
model = Word2Vec(data,
                 sg=1,
                 size=100,
                 window=3,
                 min_count=3)
```

1. Word2Vec 모델 적용

2) Word2Vec 모델을 활용한 유사도 분석

(1) 유사어 검색

- 모델 객체의 model.wv.most_similar( ) 함수를 이용하여 유사도 높은 단어 검색

```
sim = model.wv.most_similar('대회')
sim
```

```
[('출전', 0.9996535181999207),
 ('이번', 0.9996469020843506),
 ('인천', 0.9996358156204224),
 ('지난', 0.9996283054351807),
 ('결승', 0.9996280670166016),
 ('지난해', 0.999619722366333),
 ('시작', 0.9996119737625122),
 ('기록', 0.9996067881584167),
 ('서울', 0.999600887298584),
 ('개최', 0.9995979070663452)]
```

1. Word2Vec 모델 적용

2) Word2Vec 모델을 활용한 유사도 분석

(2) 단어 간 유사도 산출

- 모델 객체의 model.wv.similarity( ) 함수를 이용하여 두 단어 간 유사도 산출

```
sim = model.wv.similarity('대회', '올림픽')
print(sim)
```

```
0.9995143
```

- 유사도가 높게 산출되는 이유는?

전체 단어 사전(Vocabulary)의 크기가 작기 때문에
단어 간 상대적 유사도가 전체적으로 **높게** 산출됨

1. Word2Vec 모델 적용

3) Word2Vec 모델의 문제점

(1) OOV(Out Of Vocabulary)

- 단어 사전(Vocabulary)에 존재하지 않는 단어 처리 불가

```
sim = model.wv.most_similar('아이폰')
print(sim)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-15-d6f7ce747f9d> in <module>()
----> 1 sim = model.wv.most_similar('아이폰')
      2 print(sim)

                                  1 frames
/usr/local/lib/python3.7/dist-packages/gensim/models/keyedvectors.py in word_ve
    450             return result
    451         else:
--> 452             raise KeyError("word '%s' not in vocabulary" % word)
    453
    454     def get_vector(self, word):

KeyError: "word '아이폰' not in vocabulary"
```

- 단어 사전(Vocabulary)에 존재하지 않는 단어 처리 방법

facebook

단어를 벡터로 만드는 방법
facebook에서 개발한 FastText 활용 고려

2. Pre-trained Word2Vec 모델 활용

1) 한글 모델

(1) Pre-trained Word2Vec 모델 다운로드
| 한글 Wiki 데이터 세트

- Google Colab의 파일 업로드 코드를 실행하여 데이터 세트 업로드

```
from google.colab import files
import os

data_dir = 'data'

if not os.path.exists(data_dir):
    os.mkdir(data_dir)
os.chdir(data_dir)
files.upload()
os.chdir('..')
```

파일 선택 선택된 파일 없음 Cancel upload

- ko.bin 파일 선택 및 업로드 실행

파일 선택 ko.bin

- **ko.bin**(application/octet-stream) - 50697568 bytes, last modified: 2016. 12. 21. -

Saving ko.bin to ko.bin

2. Pre-trained Word2Vec 모델 활용

1) 한글 모델

(2) Pre-trained Word2Vec 모델 메모리 로드

- gensim.models.Word2Vec.load( ) 함수 이용

```
model = gensim.models.Word2Vec.load('./data/ko.bin')
```

2. Pre-trained Word2Vec 모델 활용

2) 한글 모델을 활용한 유사도 분석

(1) 유사어 검색

- 모델 객체의 model.wv.similarity( ) 함수를 이용하여 유사도 높은 단어 검색

```
sim = model.wv.most_similar('대회')
sim
```

```
[('체전', 0.7017310857772827),
 ('개인전', 0.6350301504135132),
 ('콘테스트', 0.6301697492599487),
 ('그랑프리', 0.6146349906921387),
 ('콩쿠르', 0.6100002527236938),
 ('대항전', 0.6005889177322388),
 ('단체전', 0.5925721526145935),
 ('선수권', 0.5885537266731262),
 ('박람회', 0.5802826881408691),
 ('선발전', 0.5797377824783325)]
```

(2) 단어 간 유사도 산출

- 모델 객체의 model.wv.similarity( ) 함수를 이용하여 두 단어 간 유사도 산출

```
sim = model.wv.similarity('대회', '올림픽')
print(sim)
```

```
0.5271696
```

2. Pre-trained Word2Vec 모델 활용

3) 영문 모델

(1) Pre-trained Word2Vec 모델 다운로드
| 영문 GoogleNews 데이터 세트

- GoogleNews-vectors-negative300 다운로드 및 압축 해제
  - GoogleNews-vectors-negative300.bin 파일 사용

- 주의할 점

파일의 용량이 크기 때문에 구글 드라이브에 업로드 후
사용하거나 로컬 PC에서 실습 진행 권장

```
from google.colab import drive
drive.mount('/content/drive')
```

(2) Pre-trained Word2Vec 모델 메모리 로드

- gensim.models.KeyedVectors.load_word2vec_format( ) 함수 이용

```
model_file = '/content/drive/MyDrive/GoogleNews-vectors-negative30
# model_file = './data/GoogleNews-vectors-negative300.bin'

model = gensim.models.KeyedVectors.load_word2vec_format(model_file
```

4) 영문 모델을 활용한 유사도 분석

(1) 유사어 검색

- 모델 객체의 model.most_similar( ) 함수를 이용하여 유사도 높은 단어 검색

```
sim = model.most_similar('championship')
sim
```

```
[('championships', 0.807525634765625),
 ('title', 0.7559840083122253),
 ('champs', 0.7184150218963623),
 ('Championship', 0.7039351463317871),
 ('finals', 0.6765252947807312),
 ('semifinals', 0.6745949983596802),
 ('tournament', 0.6655316352844238),
 ('champi_onship', 0.6635136604309082),
 ('champions', 0.6614832878112793),
 ('championsip', 0.6548882722854614)]
```

(2) 단어 간 유사도 산출

- 모델 객체의 model.similarity( ) 함수를 이용하여 두 단어 간 유사도 산출

```
sim = model.similarity('championship', 'olympic')
print(sim)
```

```
0.15580767
```