[실습] 워드 임베딩

- 실습 내용
  - Gensim을 이용한 word2vec 학습
  - Cosine similarity 계산

- nltk, genism 및 KoNLTK 설치

  C> pip install nltk

  C> pip install genism

  C> pip install konlp # http://konltk.github.io/

  # 또는 “pip install konlpy” 설치

[실습] 워드 임베딩: gensim 및 cosine similarity 계산

실습 1. nltk의 'movie_review' 데이터

```
C> pip install gensim
C> pip install nltk
C> python
import nltk
nltk.download('movie_reviews')

from nltk.corpus import movie_reviews
sentences = [list(s) for s in movie_reviews.sents()]

from gensim.models.word2vec import Word2Vec
model = Word2Vec(sentences)
model.save('mytest.model')

model.init_sims(replace=True)

model.wv.similarity('actor', 'actress')
model.wv.most_similar("accident")
model.wv.most_similar(positive=['she', 'actor'], negative='actress', topn=5)

model.wv.get_vector('actor')          # a vector  for ‘actor’
model.wv.get_vector('actress')        # a vector  for ‘actress’
```

실습 2. 벡터 유사도 계산

```
from sklearn.metrics.pairwise import cosine_similarity
import math

cosine_similarity([[1, 0, -1]], [[-1,-1, 0]])

# Compute cosine similarity of v1 to v2: (v1 dot v2)/{||v1||*||v2||)
def cosine_similarity(v1,v2):
    sumxx, sumxy, sumyy = 0, 0, 0
    for i in range(len(v1)):
        x = v1[i]; y = v2[i]
        sumxx += x*x
        sumyy += y*y
        sumxy += x*y
    return sumxy/math.sqrt(sumxx*sumyy)

v1,v2 = [3, 45, 7, 2], [2, 54, 13, 15]
print(v1, v2, cosine_similarity([v1], [v2]))

va = model.wv.get_vector('actor')            # a vector  for 'actor'
vb = model.wv.get_vector('actress')          # a vector  for 'actress'
cosine_similarity([va], [vb])
```

실습 3. 네이버 영화평 데이터

```
# !wget - nc https://raw.githubusercontent.com/e9t/nsmc/master/ratings_train.txt
import codecs
def read_data(filename):
    with open(filename, mode='r', encoding='utf-8') as f:
        data = [line.split('\t') for line in f.read().splitlines()]
    data = data[1:] # header 제외
    return data
train_data = read_data('ratings_train.txt')          # you may use 'ratings_test.txt' for small dataset

from konlpy.tag import Okt
from konlp.kma.klt2023 import klt2023
tagger = klt2023()    # Okt()

def tokenize(doc):                                   # 토큰화 -- KLT().pos 형태소 분석기 사용
    #return ['/'.join(t) for t in tagger.pos(doc)]   # Okt
    return tagger.pos(doc)

print('KLT2000 -- morph analysis for 200K Naver movie reviews. Waiting several minutes...')
train_docs = [row[1] for row in train_data]
sentences = [tokenize(d) for d in train_docs]        # 형태소 분석 시간이 매우 오래 걸리는 문제...

from gensim.models import word2vec
model = word2vec.Word2Vec(sentences)
model.save('mytest.model')              // model = word2vec.Word2Vec.load('mytest.model')

model.init_sims(replace=True)
print(model.wv.similarity('배우', '여배우'))
print(model.wv.similarity('배우', '남자'))

print(model.wv.most_similar(positive=tokenize(['남자', '여배우']), negative=tokenize(u'배우'), topn=5))
```

토큰화: 한국어 형태소 분석 시간 문제

- 문제점

  sentences = [tokenize(d) for d in train_docs] # 형태소 분석 시간이 매우 오래 걸림!

  # "[tokenize(d) for d in train_docs]" → sentences를 토큰화된 파일에서 읽어옴!

  model = Word2Vec(sentences)

<참고> 형태소 분석기에 따라 실행시간 차이가 매우 큼!

- 해결 방안
  - 영어의 경우 형태소 분석 안함!
  - 한국어도 미리 형태소 분석된(토큰화된) 데이터셋 사용 가능

미리 형태소 분석된(토큰화된) 데이터셋 생성

- KLT2000 형태소 분석기를 이용한 토큰화된 파일 생성 방법
  - https://cafe.naver.com/nlpkang/3
  - https://cafe.naver.com/nlpkang/45

```
C> iconv -c -f utf-8 -t cp949 kowiki.txt > kowiki-cp949.txt     // utf8 → cp949 (ks완성형)
C> index2018.exe kowiki-cp949.txt tokens.txt                    // 입출력: ks완성형
C> iconv -c -f cp949 -t utf-8 tokens.txt > kowiki-tokens.txt    // cp949 → utf-8
```

- 미리 토큰화된 파일(한글 위키 텍스트)

  http://nlp.kookmin.ac.kr/kcc/word2vec/ko_wiki_text.zip

<참고> KCC 원시 말뭉치에 대한 미리 토큰화된 파일 (7억3천만 어절)

http://nlp.kookmin.ac.kr/kcc

미리 형태소 분석된(토큰화된) 텍스트 예

ko_wiki_text-KMA-euckr.txt

```
5
제임스 얼 지미 카터 2세 1924년 10월 1일 민주당 출신 미국 제39대 대통령 1977년 1981년
생애 어린 시절 그 조지아 주 섬터 카운티 플레인스 마을 출생
조지아 공과대학교 졸업 1946년 메릴랜드 주 에 있는 아나폴리스 해군사관학교 United States Naval Academy 졸업
그 후 해군 들어가 전함 원자력 잠수함 승무원 일하였다
1953년 해군 대위 제대 땅콩 면화 등 가꿔 많은 돈 벌었다
그 별명 땅콩 농부 Peanut Farmer 알려졌다
정계 입문 1962년 조지아 주 상원 의원 선거 낙선 그 선거 부정선거 였음 입증 되어 당선 1966년 조지아 주 지사 선거
대통령 되기 전 조지아 주 상원 원 두 번 연임 1971년 1975년 조지아 주 주지사 근무
조지아 주지사 지내면서 미국 사는 흑인 등용법 내세웠다
대통령 재임 1976년 대통령 선거 민주당 후보 출마 도덕주의 정책 내세워 포드 누르고 당선
카터 대통령 에너지 개발 촉구 공화당 의 반대 무산
외교 정책 카터 이집트 와 이스라엘 을 조정 캠프 데이비드 안와르 사다트 대통령 메나헴 베긴 수상 함께 중동 평화 위한
그러나 이것 공화당 유대 단체 반발 일으켰다
1979년 세 정상 백악관 에서 평화 조약 맺었다
또한 소련 과 제2차 전략 무기 제한 협상 에 조인
카터 1970년
그러나 이란 의
또한 임기 말
대한민국 관계
```

KCC150-KMA_euckr.txt

```
통합보건 교육 이 대학 특화 프로그램
이에 따라 전달 후원금 저소득층 사회복지시설 2억원 상당 중구 푸드뱅크 설립 지원 6000만원 저소득 긴급 지원 시설 등 지원 5000만원
오리콤 이에 모든 조직 프로세스 바꾸기로 했다
방사청 이번 원가관리 안내서 희망하는 기업 무료 배포 예정
그러나 이 건물 겉 보는 것 달리 지어진 지가 꽤 오래 방 약간 낡았다는 느낌 주었다
그런데 황희찬 다시 1부 올라갈 수도 있다
응답자 중 가장 많은 의견
p씨 23일 오전 8시30분 벤츠 승용차 몰고 서울 마장동 내부순환로 달리다 커브길 좌우 방호벽 차례 들 받았다
니트 당초 수능 영어 2015년 대체 계획 2019년 미뤄지는 등 난항 겪어왔다
세계 금융시장 극심한 공포 잠시 벗어났다
반면 글로벌 금융위기 주범 선진국 살림살이 크게 악화
경기도 체육회 주최 경기도 및 가평군 태권도 협회 주관 이번 대회 도내 31개 시군 생활체육 동호인 2600 명 참가 겨루기 품세 격파 종목
사실 국내 업체 확장성 좋은 평가 받고 있다
군 송기섭 군수 지시 작년 말 2017년 주요업무계획 수립 기존 연도 바꿔 소위 표지갈이 식 연례 반복 작성 업무계획 탈피 수차례 성과 회 개최
웹접근성 인증 한국정보화진흥원 장애인 고령자 편리 이용 수 있도록 제작 관리 홈페이지 부여 합니다
연예인 방송인 김지민 대표령 표창 방송인 최현정씨과 프로야구 선수 권혁 국무총리 표창 배우 김서형 배우 김정은 각각 금융위원장 표창
사실 실물 경제 붕괴 걱정 정부 속내 구조조정 약간 미흡 끝나더라도 경기부양 기울고 있는 것 같다
소방당국 진화 완료 즉시 목격자 등 상대 정확 화재 원인 조사 할 방침
18거래일 동안 매섭게 매도 물량 내놓았던 외국인 현재 91억 순매도 있다
정부 조속히 해당 부품 교체 한편 사건 책임자 문책 및 재발방지 만전 기한다는 방침
```

Word2vec 학습: 입력파일 '토큰 리스트' (형태소 분석되어 있는 파일)

```
# C> wv_train_tokens.py "tokenized_text_file“      // 형태소 분석되어 있는 '토큰 리스트' 파일
from gensim.models import word2vec
import sys

def wv_train_tokens(filename):
        print(f"Training word embedding vectors for <{filename}>.")
        f = open(filename, "r", encoding='utf-8')
        text = f.readlines()
        f.close()

        tokens = []
        for sent in text:
                tokens.append(sent.split())

        model = word2vec.Word2Vec(sentences=tokens, vector_size=300, window=5, min_count=2, workers=4)
        return model

if __name__ == "__main__":
        if len(sys.argv) < 2:
                print("C> wv_train_tokens.py token-list.txt")
                exit()
        tokenized_file = sys.argv[1]
        model_file = tokenized_file[:-4] + '.model'

        model = wv_train_tokens(tokenized_file)          # 'KMA tokenized text file'
        model.save(model_file)
        print(f"--> Model file <{model_file}> was created!\n")
```

```
from gensim.models.fasttext import FastText
model = FastText(vector_size=300)
model.train(sentences, total_examples=len(sentences), epochs=100)
```

Word2vec 테스트: 모델파일 load

```
# C> wv_test.py “model file name“     // 학습으로 생성된 ‘모델 파일명’
# Load and test embedding model
# model_name = 'word2vec-kowiki.model'            # Word2Vec model
# C> wv_test.py 'word2vec-kowiki.model'

from gensim.models import Word2Vec
import sys

if __name__ == "__main__":
        if len(sys.argv) < 2:
                print("C> wv_test.py word2vec.model")
                exit()
        print("Loading Korean word embedding vectors for 'KMA tokenized text file'.\n")
        model_name = sys.argv[1]                # Word2Vec model -- 'word2vec-kowiki.model'
        model = Word2Vec.load(model_name)

        print(model.wv.get_vector(u'배우'))
        print(model.wv.get_vector(u'여배우'))

        print(model.wv.similarity(u'배우', u'여배우'))
        print(model.wv.similarity(u'배우', u'남자'))
        print(model.wv.similarity(u'남자', u'여배우'))

        print(model.wv.most_similar(positive=[u'남자'], topn=5))
        print(model.wv.most_similar(positive=[u'남자', u'여배우'], negative=[u'배우'], topn=5
```