Homework 04

202401833 신해솔

0. 실행 결과

```
단어를 입력하세요(Quit: q): 좋아요
[('좋네요', 0.9285770654678345), ('좋으네요', 0.8782296776771545), ('좋습니다', 0.8771317005157471), ('좋아용', 0.8196434974670041), ('편해요', 0.8104217648506165)]
단어를 입력하세요(Quit: q): 리뷰
[('후기', 0.9199358224868774), ('댓글', 0.8034829497337341), ('평', 0.6939096450805664), ('블로그', 0.6902716159820557), ('유투브', 0.6513345241546631)]
단어를 입력하세요(Quit: q): 반품
[('환불', 0.8858242034912109), ('교환', 0.8435035943984985), ('반송', 0.8337404131889343), ('대충', 0.6563467383384705), ('패쓰', 0.6483377814292908)]
단어를 입력하세요(Quit: q): 흠집
[('스크래치', 0.7944432497024536), ('스크레치', 0.7890100479125977), ('물질', 0.7844354510307312), ('얼룩', 0.7742737531661987), ('오염', 0.7595584988594055)]
단어를 입력하세요(Quit: q): 박스
[('상자', 0.9023770689964294), ('봉투', 0.7849109768867493), ('아이스박스', 0.7527062892913818), ('봉지', 0.7371666431427002), ('비닐', 0.7285019755363464)]
단어를 입력하세요(Quit: q): q
```

0-1. 전체 코드

```python
from konlpy.tag import Okt
from gensim.models import Word2Vec


with open("naver_shopping_corpus.txt", "r", encoding="utf-8") as file:
    corpus = [line[1:].strip() for line in file.readlines()]

okt = Okt()
tokens = [okt.morphs(sentence) for sentence in corpus]

model = Word2Vec(tokens, vector_size=100, min_count=5, sg=0)

while True:
    target_word = input("단어를 입력하세요(Quit: q): ")
    if target_word == "q" or target_word == "Q":
        break

    similar_words = model.wv.similar_by_word(word=target_word, topn=5)
    print(similar_words)
```

1. 필요한 라이브러리 import

```
from konlpy.tag import Okt
from gensim.models import Word2Vec
```

- konlpy.tag 모듈에서 Okt 클래스 import

- genism.models 모듈에서 Word2Vec 클래스 import

2. 파일 읽기 및 문장별 저장

```
with open("naver_shopping_corpus.txt", "r", encoding="utf-8") as file:
    corpus = [line[1:].strip() for line in file.readlines()]
```

- with 구문과 open 함수를 통해 텍스트 파일을 읽기 모드로 엶

- readlines 함수로 줄별로 나누어 저장

- 각 줄의 시작에 있는 평점을 제외하고(슬라이싱) escape 문자도 제거(strip()) 후 리스트에 저장

- 문장별로 구분된 말뭉치 생성 (corpus)

3. 형태소 단위 추출

```
okt = Okt()
tokens = [okt.morphs(sentence) for sentence in corpus]
```

- Okt 클래스를 이용해 각 문장을 형태소 단위로 분리

4. 워드 벡터 생성

```
model = Word2Vec(tokens, vector_size=100, min_count=5, sg=0)
```

- Word2Vec 클래스를 이용해 워드 벡터 모델 생성

- vector_siz는 100, min_count는 5, sg는 0으로 설정

5. 사용자 입력 및 유사 단어 출력

```
while True:
    target_word = input("단어를 입력하세요(Quit: q): ")
    if target_word == "q" or target_word == "Q":
        break

    similar_words = model.wv.similar_by_word(word=target_word, topn=5)
    print(similar_words)
```

- 사용자의 입력을 받아 저장

- q 또는 Q를 입력하면 반복문 종료

- model의 내부 객체 wv의 similar_by_word 함수를 통해 유사도가 높은 5개 단어 추출

- 리스트[단어, 정확도] 형태로 출력