소스 코드

```
#-*- coding:utf-8 -*-
import sys
from konlpy.tag import Kkma
import re

BI = 2
TRI = 3

class NGram:
    def __init__(self, args):
        self.syllable_split_var = []
        self.ngram = []

        with open(args, "r", encoding='UTF8') as f:
            self.data = f.read()
            pattern = '[^\t\n\r\f\v\w]'
            repl = ''
            self.data = re.sub(pattern=pattern, repl=repl, string=self.data)

    def syllable_split(self, num_gram):
        if 0 < num_gram < 5:
            text = tuple(self.data)
            self.ngram = [text[x:x + num_gram] for x in range(0, len(text) - num_gram + 1)]
            return self.ngram
        elif num_gram == 5:
            text = tuple(self.data)
            self.ngram = []
            for i in range(len(text) - 1):
                self.ngram.extend([text[i:i + 2], text[i:i + 3]])
            return self.ngram
        else:
            print("num_gram error")
            return []

class KonLPy:
    def __init__(self, args):
        kkma = Kkma()
        with open(args, "r", encoding='UTF8') as f:
            self.data = f.read()
            self.konlpy_split = kkma.nouns(self.data)
        print(self.konlpy_split)
        self.set_nouns = set(self.konlpy_split)
```

```python
def check_similarity(sen_01, sen_02, is_konlpy=False):
    if not is_konlpy:
        same_count = 0
        for (i, tuples) in enumerate(sen_01):
            if sen_01[i] == sen_02[i]:
                same_count += 1
        return (same_count/len(sen_01)) * 100
    else:
        same_count = len(sen_01 & sen_02)
        return (same_count/len(sen_01)) * 100

def n_gram_model(n_gram_num):

    n_gram_1 = NGram(args_1)
    n_gram_1_syllable_split = n_gram_1.syllable_split(n_gram_num)
    print(n_gram_1_syllable_split)
    n_gram_2 = NGram(args_2)
    n_gram_2_syllable_split = n_gram_2.syllable_split(n_gram_num)
    print(n_gram_2_syllable_split)

    n_gram_similarity_percentage = check_similarity(n_gram_1_syllable_split, n_gram_2_syllable_split)
    if n_gram_num == BI:
        print(f'bi_gram similarity = {n_gram_similarity_percentage}%')
    elif n_gram_num == TRI:
        print(f'tri_gram similarity = {n_gram_similarity_percentage}%')
    elif n_gram_num == 5:
        print(f'bi+tri_gram similarity = {n_gram_similarity_percentage}%')


if __name__ == '__main__':
    args_1, args_2 = sys.argv[1], sys.argv[2]

    n_gram_model(BI)
    n_gram_model(TRI)
    n_gram_model(5)

    kon_01 = KonLPy(args_1)
    kon_02 = KonLPy(args_2)
    kon_similarity_percentage = check_similarity(kon_01.set_nouns, kon_02.set_nouns, True)

    print(f'konlpy similarity = {kon_similarity_percentage}%')
```

코드 분석

```
#-*- coding:utf-8 -*-
import sys
from konlpy.tag import Kkma
import re

BI = 2
TRI = 3
```

한글 주석을 위해 utf-8 형식을 지원하도록 하였다. (한글 주석이 없긴하다.)

형태소 분석은 konlpy 라이브러리를 사용한다.

re는 정규식을 사용하기 위해 import 하였다.

BI, TRI 상수는 n gram 모델 시 사용하는 상수이다.

```
class NGram:
    def __init__(self, args):
        self.syllable_split_var = []
        self.ngram = []

        with open(args, "r", encoding='UTF8') as f:
            self.data = f.read()
            pattern = '[^\t\n\r\f\v\w]'
            repl = ''
            self.data = re.sub(pattern=pattern, repl=repl, string=self.data)

    def syllable_split(self, num_gram):
        if 0 < num_gram < 5:
            text = tuple(self.data)
            self.ngram = [text[x:x + num_gram] for x in range(0, len(text) - num_gram + 1)]
            return self.ngram
        elif num_gram == 5:
            text = tuple(self.data)
            self.ngram = []
            for i in range(len(text) - 1):
                self.ngram.extend([text[i:i + 2], text[i:i + 3]])
            return self.ngram
        else:
            print("num_gram error")
            return []
```

음절 n gram을 하기 위한 class

__init__ 초기화 함수에서 한글 문장을 입력 받으면 정규식 표현으로 특수문자들을 모두 제거하고 self.data 변수에 초기화를 해준다.

syllable_split 함수에서 num_gram(n gram의 n 숫자)를 인자로 받으면 0< n_gram <5인 경우 n_gram으로 분석하고, 5인 경우 bi+tri gram으로 분석한다. 그 외의 경우에는 num_gram error 문자열을 출력하고 []를 반환한다.

```
class KonLPy:
    def __init__(self, args):
        kkma = Kkma()
        with open(args, "r", encoding='UTF8') as f:
            self.data = f.read()
            self.konlpy_split = kkma.nouns(self.data)
        print(self.konlpy_split)
        self.set_nouns = set(self.konlpy_split)
```

konlpy를 위한 class

konlpy 라이브러리에 있는 방식 중 명사만 추출하도록 하여 유사도를 검사한다.

__init__ 함수에서 한글 문자열을 입력받으면 konlpy의 nouns함수에 인자로 한글 문장을 넘기고 self.konlpy_split 변수에 담아준다. 그리고 set을 사용하여 집합으로 만들어서 self.set_nouns 변수에 담아준다.

```
def check_similarity(sen_01, sen_02, is_konlpy=False):
    if not is_konlpy:
        same_count = 0
        for (i, tuples) in enumerate(sen_01):
            if sen_01[i] == sen_02[i]:
                same_count += 1
        return (same_count/len(sen_01)) * 100
    else:
        same_count = len(sen_01 & sen_02)
        return (same_count/len(sen_01)) * 100
```

유사도를 검사하는 함수다.

파라미터로 is_konlpy는 default로 False이고, False인 경우 n_gram끼리 유사도를 검사한다.

n_gram끼리 검사할 때 음절단위로 검사를 하고 같은 요소가 있다면 same_count를 1 증가시키고 같은 횟수를 첫 문장의 n_gram 개수로 나눠서 유사도를 검사한다.

is_konlpy가 True인 경우 집합에서 같은 원소를 갖고 있는 개수만큼 same_count 값을 초기화해주고 첫 문장의 원소의 개수로 나눠서 유사도를 검사한다.

```python
def n_gram_model(n_gram_num):

    n_gram_1 = NGram(args_1)
    n_gram_1_syllable_split = n_gram_1.syllable_split(n_gram_num)
    print(n_gram_1_syllable_split)
    n_gram_2 = NGram(args_2)
    n_gram_2_syllable_split = n_gram_2.syllable_split(n_gram_num)
    print(n_gram_2_syllable_split)

    n_gram_similarity_percentage = check_similarity(n_gram_1_syllable_split, n_gram_2_syllable_split)
    if n_gram_num == BI:
        print(f'bi_gram similarity = {n_gram_similarity_percentage}%')
    elif n_gram_num == TRI:
        print(f'tri_gram similarity = {n_gram_similarity_percentage}%')
    elif n_gram_num == 5:
        print(f'bi+tri_gram similarity = {n_gram_similarity_percentage}%')
```

NGram class를 사용하여 첫 문장과 두 번째 문장을 인자로 넘겨주는 함수다.

n_gram_num을 파라미터로 받아서 bi, tri, bi+tri n gram 방식을 구분한다.

n_gram_syllable_split 변수들에 음절단위로 n gram한 값을 넣어주고 check_similarity 함수를 사용하여 이 값들의 유사도를 n_gram_similarity_percentage 변수에 담아준다.

그러면 bi인지 tri인지 bi+tri로 구분지어서 출력문장을 바꿔서 n_gram_similarity_percentage변수를 출력한다.

```python
if __name__ == '__main__':
    args_1, args_2 = sys.argv[1], sys.argv[2]

    n_gram_model(BI)
    n_gram_model(TRI)
    n_gram_model(5)

    kon_01 = KonLPy(args_1)
    kon_02 = KonLPy(args_2)
    kon_similarity_percentage = check_similarity(kon_01.set_nouns, kon_02.set_nouns, True)

    print(f'konlpy similarity = {kon_similarity_percentage}%')
```

main함수다.

터미널에서 인자를 2개를 받는다. 그 값들을 args_1, args_2에 담아준다.

konlpy로 구한 유사도는 main함수에서 바로 check_similarity 함수를 사용하여 구한다. 이때 3번째 인자로 True를 넘겨줘서 konlpy의 유사도를 구한다.

case01의 두 문장

대부분 유사하지만 음절이 조금 다른 두 문장의 유사도를 비교해보았다. 뉴스 기사에서 발췌한 문장이 아니라 유명한 두 문장을 직접 가져와봤다. 일부러 문장 부호를 섞어주었다.

첫 번째 문장: 내가 그린 기린 그림은 "잘"그린 기린 그림이고,

두 번째 문장: 네가 그린 기린 그림은 "못"그린 기린 그림이다.

출력

```
[('내', '가'), ('가', '그'), ('그', '린'), ('린', '기'), ('기', '린'), ('린', '그'), ('그', '림'), ('림', '은'), ('은', '잘'), ('잘', '그'),
('그', '린'), ('린', '기'), ('기', '린'), ('린', '그'), ('그', '림'), ('림', '이'), ('이', '고')]
[('네', '가'), ('가', '그'), ('그', '린'), ('린', '기'), ('기', '린'), ('린', '그'), ('그', '림'), ('림', '은'), ('은', '못'), ('못', '그'),
('그', '린'), ('린', '기'), ('기', '린'), ('린', '그'), ('그', '림'), ('림', '이'), ('이', '다')]
bi_gram similarity = 76.47058823529412%
[('내', '가', '그'), ('가', '그', '린'), ('그', '린', '기'), ('린', '기', '린'), ('기', '린', '그'), ('린', '그', '림'), ('그', '림', '은'),
('림', '은', '잘'), ('은', '잘', '그'), ('잘', '그', '린'), ('그', '린', '기'), ('린', '기', '린'), ('기', '린', '그'), ('린', '그', '림'),
('그', '림', '이'), ('림', '이', '고')]
[('네', '가', '그'), ('가', '그', '린'), ('그', '린', '기'), ('린', '기', '린'), ('기', '린', '그'), ('린', '그', '림'), ('그', '림', '은'),
('림', '은', '못'), ('은', '못', '그'), ('못', '그', '린'), ('그', '린', '기'), ('린', '기', '린'), ('기', '린', '그'), ('린', '그', '림'),
('그', '림', '이'), ('림', '이', '다')]
tri_gram similarity = 68.75%
[('내', '가'), ('내', '가', '그'), ('가', '그'), ('가', '그', '린'), ('그', '린'), ('그', '린', '기'), ('린', '기'), ('린', '기', '린'), ('기',
'린'), ('기', '린', '그'), ('린', '그'), ('린', '그', '림'), ('그', '림'), ('그', '림', '은'), ('림', '은'), ('림', '은', '잘'), ('은', '잘'),
('은', '잘', '그'), ('잘', '그'), ('잘', '그', '린'), ('그', '린'), ('그', '린', '기'), ('린', '기'), ('린', '기', '린'), ('기', '린'), ('기',
'린', '그'), ('린', '그'), ('린', '그', '림'), ('그', '림'), ('그', '림', '이'), ('림', '이'), ('림', '이', '고'), ('이', '고'), ('이', '고')]
[('네', '가'), ('네', '가', '그'), ('가', '그'), ('가', '그', '린'), ('그', '린'), ('그', '린', '기'), ('린', '기'), ('린', '기', '린'), ('기',
'린'), ('기', '린', '그'), ('린', '그'), ('린', '그', '림'), ('그', '림'), ('그', '림', '은'), ('림', '은'), ('림', '은', '못'), ('은', '못'),
('은', '못', '그'), ('못', '그'), ('못', '그', '린'), ('그', '린'), ('그', '린', '기'), ('린', '기'), ('린', '기', '린'), ('기', '린'), ('기',
'린', '그'), ('린', '그'), ('린', '그', '림'), ('그', '림'), ('그', '림', '이'), ('림', '이'), ('림', '이', '다'), ('이', '다'), ('이', '다')]
bi+tri_gram similarity = 70.58823529411765%
['내가', '그린', '기린', '그림']
['네가', '그린', '기린', '그림']
konlpy similarity = 75.0%
```

분석

음절 bi gram 에서는 약 76.47%의 유사도를 보였고, tri gram 에서는 68.75%의 유사도를 보였고, bi+tri gram 에서는 약 70.59%의 유사도를 보였다. 또한 konlpy 라이브러리에서 명사만 추출하면 위와 같이 추출하고 75%의 유사도를 보인다.

case02 의 두 문장

네이버 신문 기사에서 발췌한 문장이다. case02_1.txt 는 기사 원문이고 case02_2.txt 는 기사 원문의 어절 순서를 문맥상 흐름에 맞게 직접 바꾸어 보았고, 설명이라는 단어와 유사한 전달이라는 단어로 한 단어만 직접 바꾸어서 유사도를 검사했다.

첫 번째 문장: 끝으로 연구진은 “이런 데이터는 텔로미어 길이를 복원해 재프로그래밍하는 데 극단적 나이가 절대적인 장벽은 아니라는 점을 보여준다”고 설명했다.

두 번째 문장: 연구진은 끝으로 “이런 데이터는 텔로미어 길이를 복원해 재프로그래밍하는 데 극단적 나이가 절대적인 장벽은 아니라는 점을 보여준다”고 전달했다.

출력

```
[('끝', '으'), ('으', '로'), ('로', '연'), ('연', '구'), ('구', '진'), ('진', '은'), ('은', '이'), ('이', '런'), ('런', '데'), ('데', '이'),
('이', '터'), ('터', '는'), ('는', '텔'), ('텔', '로'), ('로', '미'), ('미', '어'), ('어', '길'), ('길', '이'), ('이', '를'), ('를', '복'),
('복', '원'), ('원', '해'), ('해', '재'), ('재', '프'), ('프', '로'), ('로', '그'), ('그', '래'), ('래', '밍'), ('밍', '하'), ('하', '는'),
('는', '데'), ('데', '극'), ('극', '단'), ('단', '적'), ('적', '나'), ('나', '이'), ('이', '가'), ('가', '절'), ('절', '대'), ('대', '적'),
('적', '인'), ('인', '장'), ('장', '벽'), ('벽', '은'), ('은', '아'), ('아', '니'), ('니', '라'), ('라', '는'), ('는', '점'), ('점', '을'),
('을', '보'), ('보', '여'), ('여', '준'), ('준', '다'), ('다', '고'), ('고', '설'), ('설', '명'), ('명', '했'), ('했', '다')]
[('연', '구'), ('구', '진'), ('진', '은'), ('은', '끝'), ('끝', '으'), ('으', '로'), ('로', '이'), ('이', '런'), ('런', '데'), ('데', '이'),
('이', '터'), ('터', '는'), ('는', '텔'), ('텔', '로'), ('로', '미'), ('미', '어'), ('어', '길'), ('길', '이'), ('이', '를'), ('를', '복'),
('복', '원'), ('원', '해'), ('해', '재'), ('재', '프'), ('프', '로'), ('로', '그'), ('그', '래'), ('래', '밍'), ('밍', '하'), ('하', '는'),
('는', '데'), ('데', '극'), ('극', '단'), ('단', '적'), ('적', '나'), ('나', '이'), ('이', '가'), ('가', '절'), ('절', '대'), ('대', '적'),
('적', '인'), ('인', '장'), ('장', '벽'), ('벽', '은'), ('은', '아'), ('아', '니'), ('니', '라'), ('라', '는'), ('는', '점'), ('점', '을'),
('을', '보'), ('보', '여'), ('여', '준'), ('준', '다'), ('다', '고'), ('고', '전'), ('전', '달'), ('달', '했'), ('했', '다')]
bi_gram similarity = 83.05084745762711%
[('끝', '으', '로'), ('으', '로', '연'), ('로', '연', '구'), ('연', '구', '진'), ('구', '진', '은'), ('진', '은', '이'), ('은', '이', '런'),
('이', '런', '데'), ('런', '데', '이'), ('데', '이', '터'), ('이', '터', '는'), ('터', '는', '텔'), ('는', '텔', '로'), ('텔', '로', '미'),
('로', '미', '어'), ('미', '어', '길'), ('어', '길', '이'), ('길', '이', '를'), ('이', '를', '복'), ('를', '복', '원'), ('복', '원', '해'),
('원', '해', '재'), ('해', '재', '프'), ('재', '프', '로'), ('프', '로', '그'), ('로', '그', '래'), ('그', '래', '밍'), ('래', '밍', '하'),
('밍', '하', '는'), ('하', '는', '데'), ('는', '데', '극'), ('데', '극', '단'), ('극', '단', '적'), ('단', '적', '나'), ('적', '나', '이'),
('나', '이', '가'), ('이', '가', '절'), ('가', '절', '대'), ('절', '대', '적'), ('대', '적', '인'), ('적', '인', '장'), ('인', '장', '벽'),
('장', '벽', '은'), ('벽', '은', '아'), ('은', '아', '니'), ('아', '니', '라'), ('니', '라', '는'), ('라', '는', '점'), ('는', '점', '을'),
('점', '을', '보'), ('을', '보', '여'), ('보', '여', '준'), ('여', '준', '다'), ('준', '다', '고'), ('다', '고', '설'), ('고', '설', '명'),
('설', '명', '했'), ('명', '했', '다')]
[('연', '구', '진'), ('구', '진', '은'), ('진', '은', '끝'), ('은', '끝', '으'), ('끝', '으', '로'), ('으', '로', '이'), ('로', '이', '런'),
('이', '런', '데'), ('런', '데', '이'), ('데', '이', '터'), ('이', '터', '는'), ('터', '는', '텔'), ('는', '텔', '로'), ('텔', '로', '미'),
('로', '미', '어'), ('미', '어', '길'), ('어', '길', '이'), ('길', '이', '를'), ('이', '를', '복'), ('를', '복', '원'), ('복', '원', '해'),
('원', '해', '재'), ('해', '재', '프'), ('재', '프', '로'), ('프', '로', '그'), ('로', '그', '래'), ('그', '래', '밍'), ('래', '밍', '하'),
('밍', '하', '는'), ('하', '는', '데'), ('는', '데', '극'), ('데', '극', '단'), ('극', '단', '적'), ('단', '적', '나'), ('적', '나', '이'),
('나', '이', '가'), ('이', '가', '절'), ('가', '절', '대'), ('절', '대', '적'), ('대', '적', '인'), ('적', '인', '장'), ('인', '장', '벽'),
```

```
('장', '벽', '은'), ('벽', '은', '아'), ('은', '아', '니'), ('아', '니', '라'), ('니', '라', '는'), ('라', '는', '점'), ('는', '점', '을'),
('점', '을', '보'), ('을', '보', '여'), ('보', '여', '준'), ('여', '준', '다'), ('준', '다', '고'), ('다', '고', '전'), ('고', '전', '달'),
('전', '달', '했'), ('달', '했', '다')]
tri_gram similarity = 81.03448275862068%
[('끝', '으'), ('끝', '으', '로'), ('으', '로'), ('으', '로', '연'), ('로', '연'), ('로', '연', '구'), ('연', '구'), ('연', '구', '진'), ('구',
'진'), ('구', '진', '은'), ('진', '은'), ('진', '은', '이'), ('은', '이'), ('은', '이', '런'), ('이', '런'), ('이', '런', '데'), ('런', '데'),
('런', '데', '이'), ('데', '이'), ('데', '이', '터'), ('이', '터'), ('이', '터', '는'), ('터', '는'), ('터', '는', '텔'), ('는', '텔'), ('는',
'텔', '로'), ('텔', '로'), ('텔', '로', '미'), ('로', '미'), ('로', '미', '어'), ('미', '어'), ('미', '어', '길'), ('어', '길'), ('어', '길',
'이'), ('길', '이'), ('길', '이', '를'), ('이', '를'), ('이', '를', '복'), ('를', '복'), ('를', '복', '원'), ('복', '원'), ('복', '원', '해'),
('원', '해'), ('원', '해', '재'), ('해', '재'), ('해', '재', '프'), ('재', '프'), ('재', '프', '로'), ('프', '로'), ('프', '로', '그'), ('로',
'그'), ('로', '그', '래'), ('그', '래'), ('그', '래', '밍'), ('래', '밍'), ('래', '밍', '하'), ('밍', '하'), ('밍', '하', '는'), ('하', '는'),
('하', '는', '데'), ('는', '데'), ('는', '데', '극'), ('데', '극'), ('데', '극', '단'), ('극', '단'), ('극', '단', '적'), ('단', '적'), ('단',
'적', '나'), ('적', '나'), ('적', '나', '이'), ('나', '이'), ('나', '이', '가'), ('이', '가'), ('이', '가', '절'), ('가', '절'), ('가', '절',
'대'), ('절', '대'), ('절', '대', '적'), ('대', '적'), ('대', '적', '인'), ('적', '인'), ('적', '인', '장'), ('인', '장'), ('인', '장', '벽'),
('장', '벽'), ('장', '벽', '은'), ('벽', '은'), ('벽', '은', '아'), ('은', '아'), ('은', '아', '니'), ('아', '니'), ('아', '니', '라'), ('니',
'라'), ('니', '라', '는'), ('라', '는'), ('라', '는', '점'), ('는', '점'), ('는', '점', '을'), ('점', '을'), ('점', '을', '보'), ('을', '보'),
('을', '보', '여'), ('보', '여'), ('보', '여', '준'), ('여', '준'), ('여', '준', '다'), ('준', '다'), ('준', '다', '고'), ('다', '고'), ('다',
'고', '설'), ('고', '설'), ('고', '설', '명'), ('설', '명'), ('설', '명', '했'), ('명', '했'), ('명', '했', '다'), ('했', '다'), ('했', '다')]
[('연', '구'), ('연', '구', '진'), ('구', '진'), ('구', '진', '은'), ('진', '은'), ('진', '은', '끝'), ('은', '끝'), ('은', '끝', '으'), ('끝',
'으'), ('끝', '으', '로'), ('으', '로'), ('으', '로', '이'), ('로', '이'), ('로', '이', '런'), ('이', '런'), ('이', '런', '데'), ('런', '데'),
('런', '데', '이'), ('데', '이'), ('데', '이', '터'), ('이', '터'), ('이', '터', '는'), ('터', '는'), ('터', '는', '텔'), ('는', '텔'), ('는',
'텔', '로'), ('텔', '로'), ('텔', '로', '미'), ('로', '미'), ('로', '미', '어'), ('미', '어'), ('미', '어', '길'), ('어', '길'), ('어', '길',
'이'), ('길', '이'), ('길', '이', '를'), ('이', '를'), ('이', '를', '복'), ('를', '복'), ('를', '복', '원'), ('복', '원'), ('복', '원', '해'),
('원', '해'), ('원', '해', '재'), ('해', '재'), ('해', '재', '프'), ('재', '프'), ('재', '프', '로'), ('프', '로'), ('프', '로', '그'), ('로',
'그'), ('로', '그', '래'), ('그', '래'), ('그', '래', '밍'), ('래', '밍'), ('래', '밍', '하'), ('밍', '하'), ('밍', '하', '는'), ('하', '는'),
('하', '는', '데'), ('는', '데'), ('는', '데', '극'), ('데', '극'), ('데', '극', '단'), ('극', '단'), ('극', '단', '적'), ('단', '적'), ('단',
'적', '나'), ('적', '나'), ('적', '나', '이'), ('나', '이'), ('나', '이', '가'), ('이', '가'), ('이', '가', '절'), ('가', '절'), ('가', '절',
'대'), ('절', '대'), ('절', '대', '적'), ('대', '적'), ('대', '적', '인'), ('적', '인'), ('적', '인', '장'), ('인', '장'), ('인', '장', '벽'),
('장', '벽'), ('장', '벽', '은'), ('벽', '은'), ('벽', '은', '아'), ('은', '아'), ('은', '아', '니'), ('아', '니'), ('아', '니', '라'), ('니',
'라'), ('니', '라', '는'), ('라', '는'), ('라', '는', '점'), ('는', '점'), ('는', '점', '을'), ('점', '을'), ('점', '을', '보'), ('을', '보'),
('을', '보', '여'), ('보', '여'), ('보', '여', '준'), ('여', '준'), ('여', '준', '다'), ('준', '다'), ('준', '다', '고'), ('다', '고'), ('다',
'고', '전'), ('고', '전'), ('고', '전', '달'), ('전', '달'), ('전', '달', '했'), ('달', '했'), ('달', '했', '다'), ('했', '다'), ('했', '다')]
bi+tri_gram similarity = 82.20338983050848%
['끝', '연구진', '데이터', '텔로미어', '길이', '복원', '재', '재프로그래밍', '프로그래밍', '데', '극단적', '나이', '절대적', '장벽', '점', '고',
'설명']
['연구진', '끝', '데이터', '텔로미어', '길이', '복원', '재', '재프로그래밍', '프로그래밍', '데', '극단적', '나이', '절대적', '장벽', '점', '고',
'전달']
konlpy similarity = 94.11764705882352%
```

분석

음절 bi gram 에서는 약 83.05%의 유사도를 보였고, tri gram 에서는 약 81.03%의 유사도를 보였고, bi+tri gram 에서는 약 82.20%의 유사도를 보였다. 또한 konlpy 라이브러리에서 명사만 추출하면 위와 같이 추출하고 약 94.12%의 유사도를 보인다.

case03 의 두 문장

전혀 유사하지 않은 두 문장을 네이버 신문에서 발췌하여 유사도를 검사하였다. 일부러 영어도 섞어보았다.

첫 번째 문장: LG 전자가 'G 시리즈' 명칭을 버리고 새로운 브랜드 도입을 통해 스마트폰 사업 재건에 나선다. 지난해 말 LG 전자 MC 사업본부 수장에 올라선 이연모 부사장이 스마트폰 사업 위기를 극복할 소방수가 될 수 있을지 이목이 쏠린다.

두 번째 문장: 이번 연구는 그런 초백세인에게서 채취한 세포를 재프로그래밍하려고 시도한 것이다.

출력

```
[('L', 'G'), ('G', '전'), ('전', '자'), ('자', '가'), ('가', 'G'), ('G', '시'), ('시', '리'), ('리', '즈'), ('즈', '명'), ('명', '칭'), ('칭',
'을'), ('을', '버'), ('버', '리'), ('리', '고'), ('고', '새'), ('새', '로'), ('로', '운'), ('운', '브'), ('브', '랜'), ('랜', '드'), ('드',
'도'), ('도', '입'), ('입', '을'), ('을', '통'), ('통', '해'), ('해', '스'), ('스', '마'), ('마', '트'), ('트', '폰'), ('폰', '사'), ('사',
'업'), ('업', '재'), ('재', '건'), ('건', '에'), ('에', '나'), ('나', '선'), ('선', '다'), ('다', '지'), ('지', '난'), ('난', '해'), ('해',
'말'), ('말', 'L'), ('L', 'G'), ('G', '전'), ('전', '자'), ('자', 'M'), ('M', 'C'), ('C', '사'), ('사', '업'), ('업', '본'), ('본', '부'),
('부', '수'), ('수', '장'), ('장', '에'), ('에', '올'), ('올', '라'), ('라', '선'), ('선', '이'), ('이', '연'), ('연', '모'), ('모', '부'),
('부', '사'), ('사', '장'), ('장', '이'), ('이', '스'), ('스', '마'), ('마', '트'), ('트', '폰'), ('폰', '사'), ('사', '업'), ('업', '위'),
('위', '기'), ('기', '를'), ('를', '극'), ('극', '복'), ('복', '할'), ('할', '소'), ('소', '방'), ('방', '수'), ('수', '가'), ('가', '될'),
('될', '수'), ('수', '있'), ('있', '을'), ('을', '지'), ('지', '이'), ('이', '목'), ('목', '이'), ('이', '쏠'), ('쏠', '린'), ('린', '다')]
[('이', '번'), ('번', '연'), ('연', '구'), ('구', '는'), ('는', '그'), ('그', '런'), ('런', '초'), ('초', '백'), ('백', '세'), ('세', '인'),
('인', '에'), ('에', '게'), ('게', '서'), ('서', '채'), ('채', '취'), ('취', '한'), ('한', '세'), ('세', '포'), ('포', '를'), ('를', '재'),
('재', '프'), ('프', '로'), ('로', '그'), ('그', '래'), ('래', '밍'), ('밍', '하'), ('하', '려'), ('려', '고'), ('고', '시'), ('시', '도'),
('도', '한'), ('한', '것'), ('것', '이'), ('이', '다')]
bi_gram similarity = 0.0%
[('L', 'G', '전'), ('G', '전', '자'), ('전', '자', '가'), ('자', '가', 'G'), ('가', 'G', '시'), ('G', '시', '리'), ('시', '리', '즈'), ('리',
'즈', '명'), ('즈', '명', '칭'), ('명', '칭', '을'), ('칭', '을', '버'), ('을', '버', '리'), ('버', '리', '고'), ('리', '고', '새'), ('고',
'새', '로'), ('새', '로', '운'), ('로', '운', '브'), ('운', '브', '랜'), ('브', '랜', '드'), ('랜', '드', '도'), ('드', '도', '입'), ('도',
'입', '을'), ('입', '을', '통'), ('을', '통', '해'), ('통', '해', '스'), ('해', '스', '마'), ('스', '마', '트'), ('마', '트', '폰'), ('트',
'폰', '사'), ('폰', '사', '업'), ('사', '업', '재'), ('업', '재', '건'), ('재', '건', '에'), ('건', '에', '나'), ('에', '나', '선'), ('나',
'선', '다'), ('선', '다', '지'), ('다', '지', '난'), ('지', '난', '해'), ('난', '해', '말'), ('해', '말', 'L'), ('말', 'L', 'G'), ('L', 'G',
'전'), ('G', '전', '자'), ('전', '자', 'M'), ('자', 'M', 'C'), ('M', 'C', '사'), ('C', '사', '업'), ('사', '업', '본'), ('업', '본', '부'),
('본', '부', '수'), ('부', '수', '장'), ('수', '장', '에'), ('장', '에', '올'), ('에', '올', '라'), ('올', '라', '선'), ('라', '선', '이'),
('선', '이', '연'), ('이', '연', '모'), ('연', '모', '부'), ('모', '부', '사'), ('부', '사', '장'), ('사', '장', '이'), ('장', '이', '스'),
('이', '스', '마'), ('스', '마', '트'), ('마', '트', '폰'), ('트', '폰', '사'), ('폰', '사', '업'), ('사', '업', '위'), ('업', '위', '기'),
('위', '기', '를'), ('기', '를', '극'), ('를', '극', '복'), ('극', '복', '할'), ('복', '할', '소'), ('할', '소', '방'), ('소', '방', '수'),
('방', '수', '가'), ('수', '가', '될'), ('가', '될', '수'), ('될', '수', '있'), ('수', '있', '을'), ('있', '을', '지'), ('을', '지', '이'),
('지', '이', '목'), ('이', '목', '이'), ('목', '이', '쏠'), ('이', '쏠', '린'), ('쏠', '린', '다')]
[('이', '번', '연'), ('번', '연', '구'), ('연', '구', '는'), ('구', '는', '그'), ('는', '그', '런'), ('그', '런', '초'), ('런', '초', '백'),
```

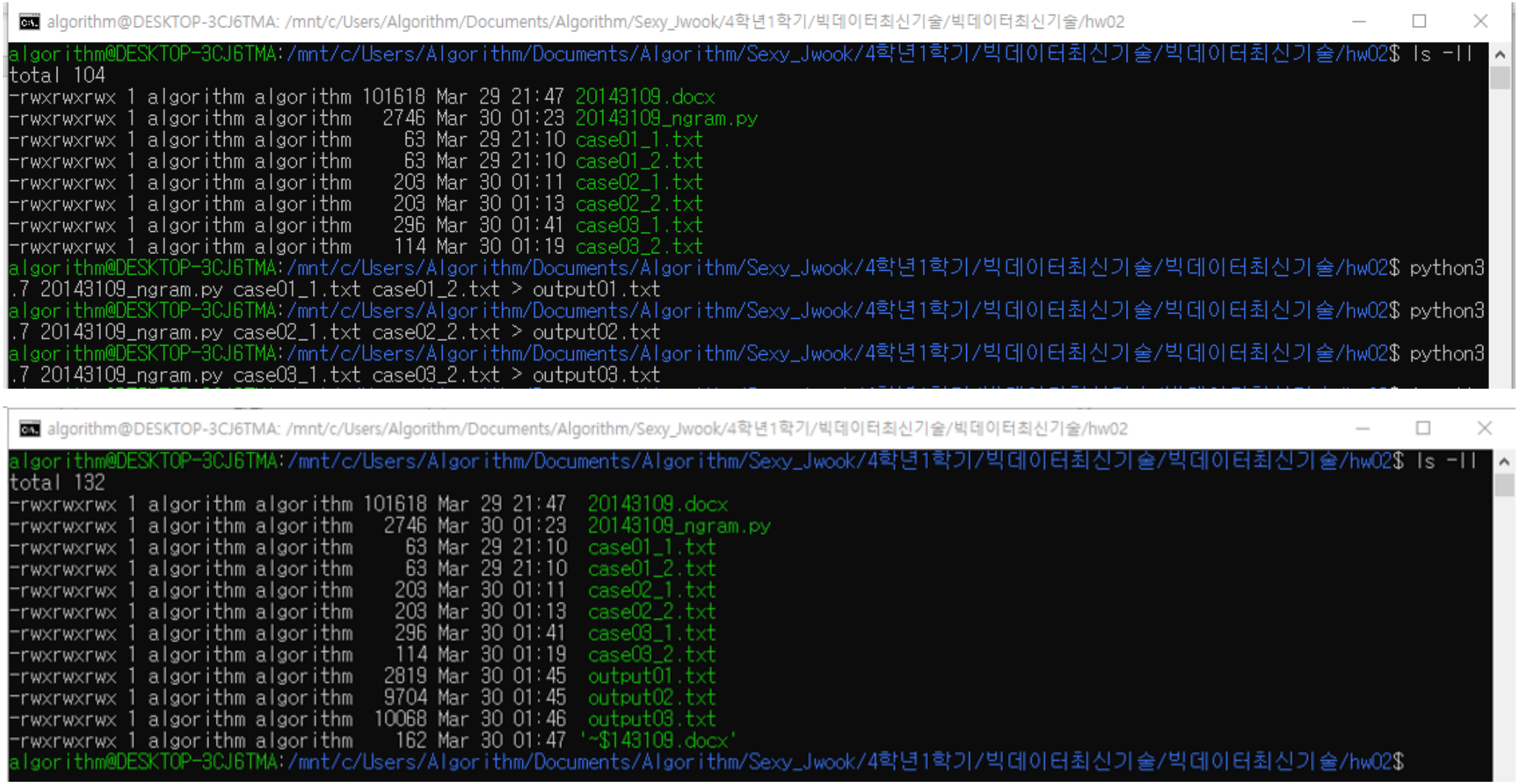
```
('초', '백', '세'), ('백', '세', '인'), ('세', '인', '에'), ('인', '에', '게'), ('에', '게', '서'), ('게', '서', '채'), ('서', '채', '춰'),
('채', '춰', '한'), ('춰', '한', '세'), ('한', '세', '포'), ('세', '포', '를'), ('포', '를', '재'), ('를', '재', '프'), ('재', '프', '로'),
('프', '로', '그'), ('로', '그', '래'), ('그', '래', '밍'), ('래', '밍', '하'), ('밍', '하', '려'), ('하', '려', '고'), ('려', '고', '시'),
('고', '시', '도'), ('시', '도', '한'), ('도', '한', '것'), ('한', '것', '이'), ('것', '이', '다')]
tri_gram similarity = 0.0%
[('L', 'G'), ('L', 'G', '전'), ('G', '전'), ('G', '전', '자'), ('전', '자'), ('전', '자', '가'), ('자', '가'), ('자', '가', 'G'), ('가', 'G'),
('가', 'G', '시'), ('G', '시'), ('G', '시', '리'), ('시', '리'), ('시', '리', '즈'), ('리', '즈'), ('리', '즈', '명'), ('즈', '명'), ('즈',
'명', '칭'), ('명', '칭'), ('명', '칭', '을'), ('칭', '을'), ('칭', '을', '버'), ('을', '버'), ('을', '버', '리'), ('버', '리'), ('버', '리',
'고'), ('리', '고'), ('리', '고', '새'), ('고', '새'), ('고', '새', '로'), ('새', '로'), ('새', '로', '운'), ('로', '운'), ('로', '운', '브'),
('운', '브'), ('운', '브', '랜'), ('브', '랜'), ('브', '랜', '드'), ('랜', '드'), ('랜', '드', '도'), ('드', '도'), ('드', '도', '입'), ('도',
'입'), ('도', '입', '을'), ('입', '을'), ('입', '을', '통'), ('을', '통'), ('을', '통', '해'), ('통', '해'), ('통', '해', '스'), ('해', '스'),
('해', '스', '마'), ('스', '마'), ('스', '마', '트'), ('마', '트'), ('마', '트', '폰'), ('트', '폰'), ('트', '폰', '사'), ('폰', '사'), ('폰',
'사', '업'), ('사', '업'), ('사', '업', '재'), ('업', '재'), ('업', '재', '건'), ('재', '건'), ('재', '건', '에'), ('건', '에'), ('건', '에',
'나'), ('에', '나'), ('에', '나', '선'), ('나', '선'), ('나', '선', '다'), ('선', '다'), ('선', '다', '지'), ('다', '지'), ('다', '지', '난'),
('지', '난'), ('지', '난', '해'), ('난', '해'), ('난', '해', '말'), ('해', '말'), ('해', '말', 'L'), ('말', 'L'), ('말', 'L', 'G'), ('L',
'G'), ('L', 'G', '전'), ('G', '전'), ('G', '전', '자'), ('전', '자'), ('전', '자', 'M'), ('자', 'M'), ('자', 'M', 'C'), ('M', 'C'), ('M',
'C', '사'), ('C', '사'), ('C', '사', '업'), ('사', '업'), ('사', '업', '본'), ('업', '본'), ('업', '본', '부'), ('본', '부'), ('본', '부',
'수'), ('부', '수'), ('부', '수', '장'), ('수', '장'), ('수', '장', '에'), ('장', '에'), ('장', '에', '올'), ('에', '올'), ('에', '올', '라'),
('올', '라'), ('올', '라', '선'), ('라', '선'), ('라', '선', '이'), ('선', '이'), ('선', '이', '연'), ('이', '연'), ('이', '연', '모'), ('연',
'모'), ('연', '모', '부'), ('모', '부'), ('모', '부', '사'), ('부', '사'), ('부', '사', '장'), ('사', '장'), ('사', '장', '이'), ('장', '이'),
('장', '이', '스'), ('이', '스'), ('이', '스', '마'), ('스', '마'), ('스', '마', '트'), ('마', '트'), ('마', '트', '폰'), ('트', '폰'), ('트',
'폰', '사'), ('폰', '사'), ('폰', '사', '업'), ('사', '업'), ('사', '업', '위'), ('업', '위'), ('업', '위', '기'), ('위', '기'), ('위', '기',
'를'), ('기', '를'), ('기', '를', '극'), ('를', '극'), ('를', '극', '복'), ('극', '복'), ('극', '복', '할'), ('복', '할'), ('복', '할', '소'),
('할', '소'), ('할', '소', '방'), ('소', '방'), ('소', '방', '수'), ('방', '수'), ('방', '수', '가'), ('수', '가'), ('수', '가', '될'), ('가',
'될'), ('가', '될', '수'), ('될', '수'), ('될', '수', '있'), ('수', '있'), ('수', '있', '을'), ('있', '을'), ('있', '을', '지'), ('을', '지'),
('을', '지', '이'), ('지', '이'), ('지', '이', '목'), ('이', '목'), ('이', '목', '이'), ('목', '이'), ('목', '이', '쏠'), ('이', '쏠'), ('이',
'쏠', '린'), ('쏠', '린'), ('쏠', '린', '다'), ('린', '다'), ('린', '다')]
[('이', '번'), ('이', '번', '연'), ('번', '연'), ('번', '연', '구'), ('연', '구'), ('연', '구', '는'), ('구', '는'), ('구', '는', '그'), ('는',
'그'), ('는', '그', '런'), ('그', '런'), ('그', '런', '초'), ('런', '초'), ('런', '초', '백'), ('초', '백'), ('초', '백', '세'), ('백', '세'),
('백', '세', '인'), ('세', '인'), ('세', '인', '에'), ('인', '에'), ('인', '에', '게'), ('에', '게'), ('에', '게', '서'), ('게', '서'), ('게',
'서', '채'), ('서', '채'), ('서', '채', '춰'), ('채', '춰'), ('채', '춰', '한'), ('춰', '한'), ('춰', '한', '세'), ('한', '세'), ('한', '세',
'포'), ('세', '포'), ('세', '포', '를'), ('포', '를'), ('포', '를', '재'), ('를', '재'), ('를', '재', '프'), ('재', '프'), ('재', '프', '로'),
('프', '로'), ('프', '로', '그'), ('로', '그'), ('로', '그', '래'), ('그', '래'), ('그', '래', '밍'), ('래', '밍'), ('래', '밍', '하'), ('밍',
'하'), ('밍', '하', '려'), ('하', '려'), ('하', '려', '고'), ('려', '고'), ('려', '고', '시'), ('고', '시'), ('고', '시', '도'), ('시', '도'),
('시', '도', '한'), ('도', '한'), ('도', '한', '것'), ('한', '것'), ('한', '것', '이'), ('것', '이'), ('것', '이', '다'), ('이', '다'), ('이',
'다')]
bi+tri_gram similarity = 0.0%
['전자', '시리즈', '명칭', '브랜드', '도입', '스마트', '스마트폰', '폰', '사업', '재건', '지난해', '말', '사업본부', '본부', '수장', '이',
'이연모', '연모', '부사', '위기', '극복', '소방수', '수', '이목']
```

```
['이번', '연구', '초', '초백세', '백세', '채취', '세포', '재', '재프로그래밍', '프로그래밍', '시도']
konlpy similarity = 0.0%
```

분석

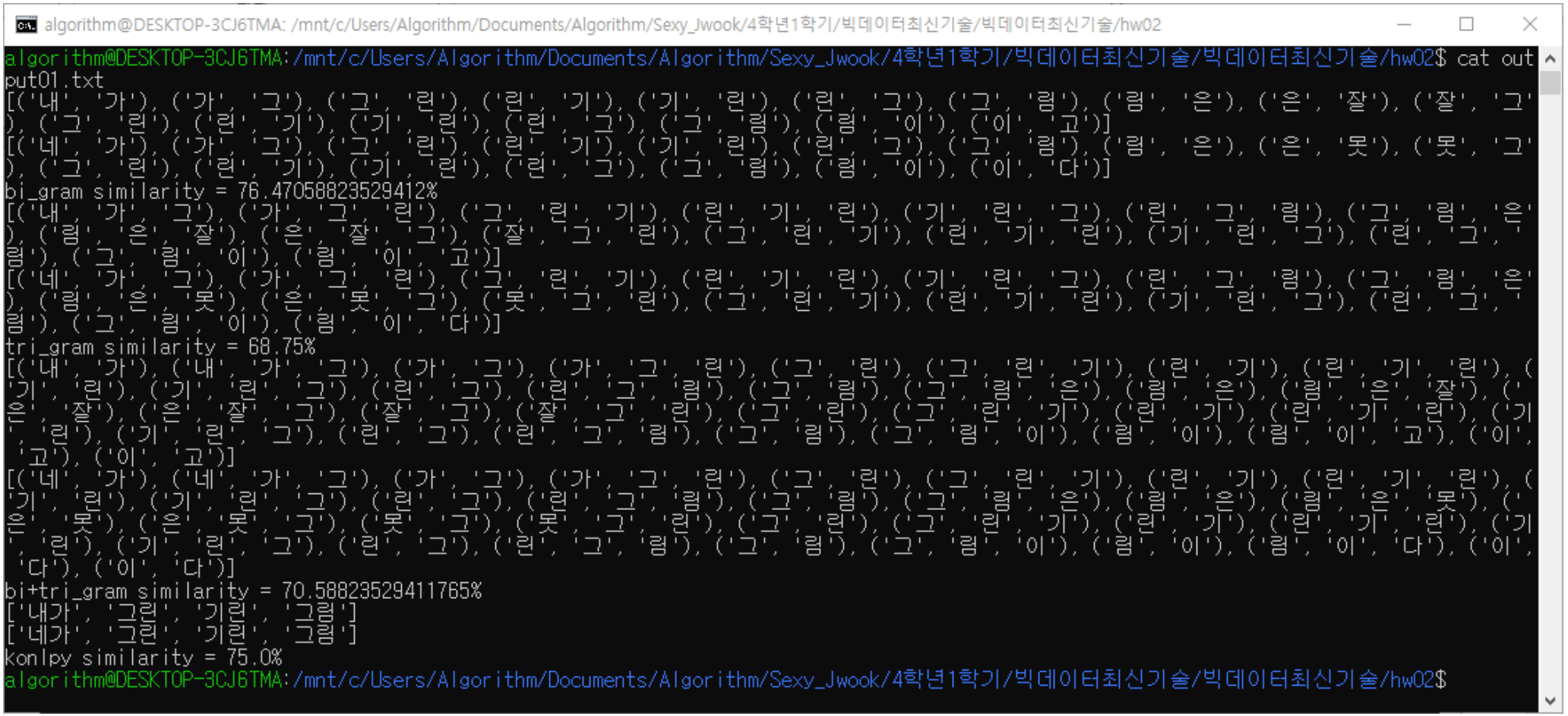
음절 bi gram 에서는 0%의 유사도를 보였고, tri gram 과 bi+tri gram 에서도 0%의 유사도를 보였다. 또한 konlpy 라이브러리에서 명사만 추출하면 위와 같이 추출하고 0%의 유사도를 보인다.

실행 화면

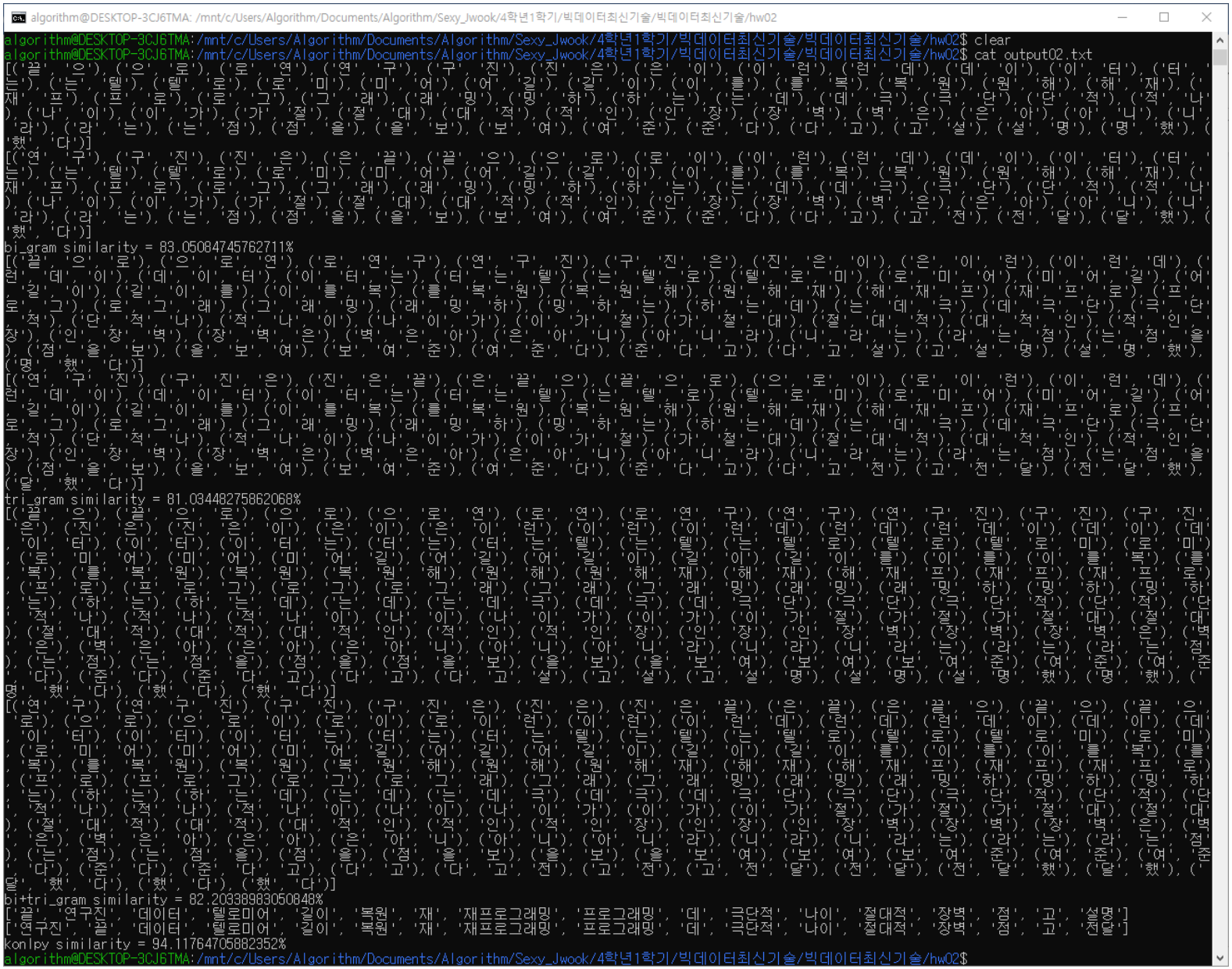

algorithm@DESKTOP-3CJ6TMA: /mnt/c/Users/Algorithm/Documents/Algorithm/Sexy_Jwook/4학년1학기/빅데이터최신기술/빅데이터최신기술/hw02
algorithm@DESKTOP-3CJ6TMA:/mnt/c/Users/Algorithm/Documents/Algorithm/Sexy_Jwook/4학년1학기/빅데이터최신기술/빅데이터최신기술/hw02$ ls -ll
total 104
-rwxrwxrwx 1 algorithm algorithm 101618 Mar 29 21:47 20143109.docx
-rwxrwxrwx 1 algorithm algorithm 2746 Mar 30 01:23 20143109_ngram.py
-rwxrwxrwx 1 algorithm algorithm 63 Mar 29 21:10 case01_1.txt
-rwxrwxrwx 1 algorithm algorithm 63 Mar 29 21:10 case01_2.txt
-rwxrwxrwx 1 algorithm algorithm 203 Mar 30 01:11 case02_1.txt
-rwxrwxrwx 1 algorithm algorithm 203 Mar 30 01:13 case02_2.txt
-rwxrwxrwx 1 algorithm algorithm 296 Mar 30 01:41 case03_1.txt
-rwxrwxrwx 1 algorithm algorithm 114 Mar 30 01:19 case03_2.txt
algorithm@DESKTOP-3CJ6TMA:/mnt/c/Users/Algorithm/Documents/Algorithm/Sexy_Jwook/4학년1학기/빅데이터최신기술/빅데이터최신기술/hw02$ python3
.7 20143109_ngram.py case01_1.txt case01_2.txt > output01.txt
algorithm@DESKTOP-3CJ6TMA:/mnt/c/Users/Algorithm/Documents/Algorithm/Sexy_Jwook/4학년1학기/빅데이터최신기술/빅데이터최신기술/hw02$ python3
.7 20143109_ngram.py case02_1.txt case02_2.txt > output02.txt
algorithm@DESKTOP-3CJ6TMA:/mnt/c/Users/Algorithm/Documents/Algorithm/Sexy_Jwook/4학년1학기/빅데이터최신기술/빅데이터최신기술/hw02$ python3
.7 20143109_ngram.py case03_1.txt case03_2.txt > output03.txt

algorithm@DESKTOP-3CJ6TMA: /mnt/c/Users/Algorithm/Documents/Algorithm/Sexy_Jwook/4학년1학기/빅데이터최신기술/빅데이터최신기술/hw02
algorithm@DESKTOP-3CJ6TMA:/mnt/c/Users/Algorithm/Documents/Algorithm/Sexy_Jwook/4학년1학기/빅데이터최신기술/빅데이터최신기술/hw02$ ls -ll
total 132
-rwxrwxrwx 1 algorithm algorithm 101618 Mar 29 21:47 20143109.docx
-rwxrwxrwx 1 algorithm algorithm 2746 Mar 30 01:23 20143109_ngram.py
-rwxrwxrwx 1 algorithm algorithm 63 Mar 29 21:10 case01_1.txt
-rwxrwxrwx 1 algorithm algorithm 63 Mar 29 21:10 case01_2.txt
-rwxrwxrwx 1 algorithm algorithm 203 Mar 30 01:11 case02_1.txt
-rwxrwxrwx 1 algorithm algorithm 203 Mar 30 01:13 case02_2.txt
-rwxrwxrwx 1 algorithm algorithm 296 Mar 30 01:41 case03_1.txt
-rwxrwxrwx 1 algorithm algorithm 114 Mar 30 01:19 case03_2.txt
-rwxrwxrwx 1 algorithm algorithm 2819 Mar 30 01:45 output01.txt
-rwxrwxrwx 1 algorithm algorithm 9704 Mar 30 01:45 output02.txt
-rwxrwxrwx 1 algorithm algorithm 10068 Mar 30 01:46 output03.txt
-rwxrwxrwx 1 algorithm algorithm 162 Mar 30 01:47 '~$143109.docx'
algorithm@DESKTOP-3CJ6TMA:/mnt/c/Users/Algorithm/Documents/Algorithm/Sexy_Jwook/4학년1학기/빅데이터최신기술/빅데이터최신기술/hw02$

output01.txt 출력화면



output02.txt 출력 화면



output03.txt 출력 화면