빅데이터최신기술 과제 2 - 문장생성확률 계산(음절 단위)

20171701 정지현

전체 레포지토리 링크: https://github.com/ghyeon0/BigData_Homework/tree/master/HW2

혹시 보고서 형식이 많이 깨진다면 https://github.com/ghyeon0/BigData_Homework/blob/master/HW2/hw2.md 를 참조 부탁드립니다.

1. 알고리즘

```
def if_utf8(text):
    first = ""
    second = ""
    third = ""
    idx = 0
    while idx < len(text):
        first = text[idx]
        idx += 1

        if first & 0x80:
            second = text[idx]
            idx += 1
            third = text[idx]
            idx += 1

        if first >= 0xE0 and first <= 0xEF and second >= 0x80 and second <=
0xBF and third >= 0x80 and third <= 0xBF:
            count_idx = ((first & 0x0f) << 12) | ((second & 0x3f) << 6) |
(third & 0x3f)
            count_idx -= 0xAC00

            try:
                unifreq[count_idx] += 1
            except:
                pass
            global total_hangul
            total_hangul += 1
```

주어지는 글을 분석해서 한글인 경우 각 글자의 갯수와 전체 한글 음절의 갯수를 모두 카운팅한다.

```
def utf8_calc():
    for i in range(11172):
        if unifreq[i]:
            uni_probability[i] = unifreq[i] / total_hangul
```

각 음절이 존재하는 경우 카운팅 된 횟수 / 전체 한글 음절의 갯수를 통해 확률을 계산해서 저장한다.

```
def calc_probability(sentence):
    first = ""
    second = ""
    third = ""
    idx = 0
    probability = 1
    while idx < len(sentence):
        first = sentence[idx]
        idx += 1

        if first & 0x80:
            second = sentence[idx]
            idx += 1
            third = sentence[idx]
            idx += 1

        if first >= 0xE0 and first <= 0xEF:
            prob_idx = ((first & 0x0f) << 12) | ((second & 0x3f) << 6) |
(third & 0x3f)
            prob_idx -= 0xAC00
            probability *= uni_probability[prob_idx]

    return probability
```

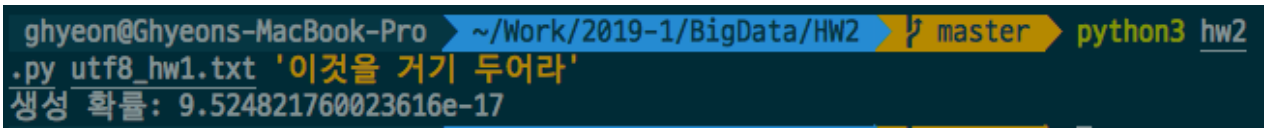
계산한 확률을 기반으로 문장 발생 확률을 계산한다. 입력한 문장를 글자별로 분리하여 그 확률을 가져오고, 확률을 모두 곱한 결과를 return 한다.

2. 사용법

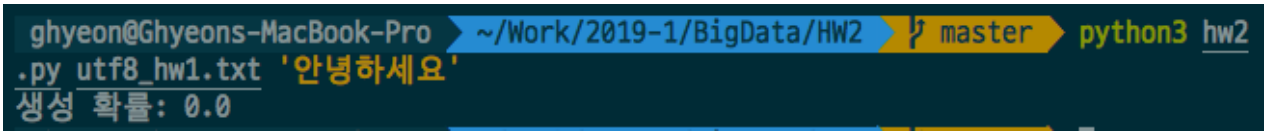
```
python3 hw2.py {음절발생확률 계산할 데이터 파일 이름} {확률 계산할 문장}
```

3. 실행결과

```
python3 hw2.py utf8_hw1.txt '이것을 거기 두어라'
```



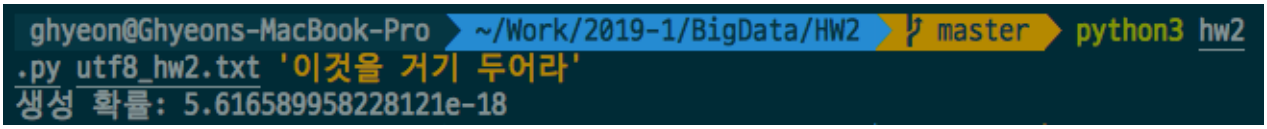

```
python3 hw2.py utf8_hw1.txt '안녕하세요'
```



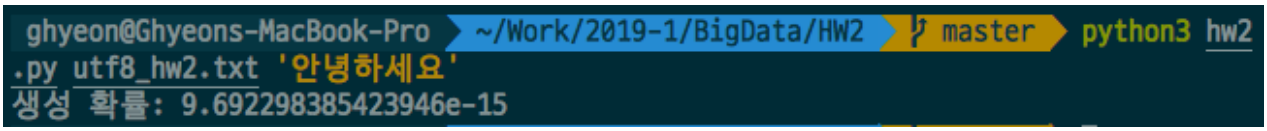
데이터의 문제로 '안', '녕' 등이 없어서 자주 쓰이는 문장임에도 확률이 0.0으로 나오는 문제가 발생하였다. 아무래도 데이터 자체도 짧고, 한글입숨을 통해 생성한 것이다 보니 비슷한 말이 계속하여 반복되어 이런 문제가 발생한 것으로 추측된다.

그래서 데이터를 중앙선거관리위원회에서 나온 [한국형 선거빅데이터 구축방안 최종보고서.hwp] [http://www.nec.go.kr:8088/files/B0000235/201410/BBS_201410160558297010.pdf] 로 대체하여 다시 진행하였다.

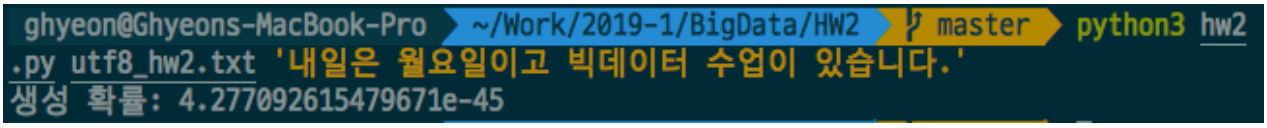
```
python3 hw2.py utf8_hw2.txt '이것을 거기 두어라'
```



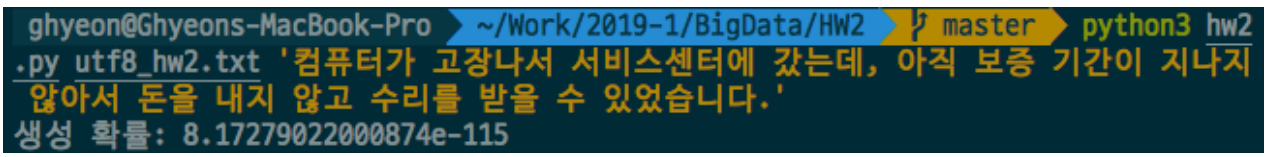

```
python3 hw2.py utf8_hw2.txt '안녕하세요'
```




```
python3 hw2.py utf8_hw2.txt '내일은 월요일이고 빅데이터 수업이 있습니다.'
```




```
python3 hw2.py utf8_hw2.txt '컴퓨터가 고장나서 서비스센터에 갔는데, 아직 보증 기간이 지나지 않아서 돈을 내지 않고 수리를 받을 수 있었습니다.'
```



4. 전체 소스코드

```
import sys
```

```
# 유니코드 한글 카운트
unifreq = [0 for i in range(11172)]
# 유니코드 한글 확률
uni_probability = [0.0 for i in range(11172)]
# 전체 한글 갯수
total_hangul = 0

def if_utf8(text):
    first = ""
    second = ""
    third = ""
    idx = 0
    while idx < len(text):
        first = text[idx]
        idx += 1

        if first & 0x80:
            second = text[idx]
            idx += 1
            third = text[idx]
            idx += 1

        if first >= 0xE0 and first <= 0xEF and second >= 0x80 and second <=
0xBF and third >= 0x80 and third <= 0xBF:
            count_idx = ((first & 0x0f) << 12) | ((second & 0x3f) << 6) |
(third & 0x3f)
            count_idx -= 0xAC00

            try:
                unifreq[count_idx] += 1
            except:
                pass
            global total_hangul
            total_hangul += 1


def utf8_calc():
    for i in range(11172):
        if unifreq[i]:
            uni_probability[i] = unifreq[i] / total_hangul


def calc_probability(sentence):
    first = ""
    second = ""
    third = ""
    idx = 0
    probability = 1
```

```
    while idx < len(sentence):
        first = sentence[idx]
        idx += 1

        if first & 0x80:
            second = sentence[idx]
            idx += 1
            third = sentence[idx]
            idx += 1

        if first >= 0xE0 and first <= 0xEF:
            prob_idx = ((first & 0x0f) << 12) | ((second & 0x3f) << 6) |
(third & 0x3f)
            prob_idx -= 0xAC00
            probability *= uni_probability[prob_idx]

    return probability


def main(file_name, sentence):
    f = open(file_name, "rb")
    text = f.read()
    f.close()
    if_utf8(text)
    utf8_calc()
    sentence_probability = calc_probability(sentence.encode())
    print("생성 확률:", sentence_probability)


if __name__ == "__main__":
    # 실행할 때 매개변수로 파일명 지정
    if len(sys.argv) == 2:
        file_name = sys.argv[1]
        sentence = input("문장 입력: ")
    elif len(sys.argv) == 3:
        file_name = sys.argv[1]
        sentence = sys.argv[2]
    # 아니면 입력 받음
    else:
        file_name = input("Input File Name: ")
        sentence = input("문장 입력: ")
    main(file_name, sentence)
```