

# 빅데이터 HW3 (문장생성확률 계산 - 어절단위)

컴퓨터공학부

20163084 권보경

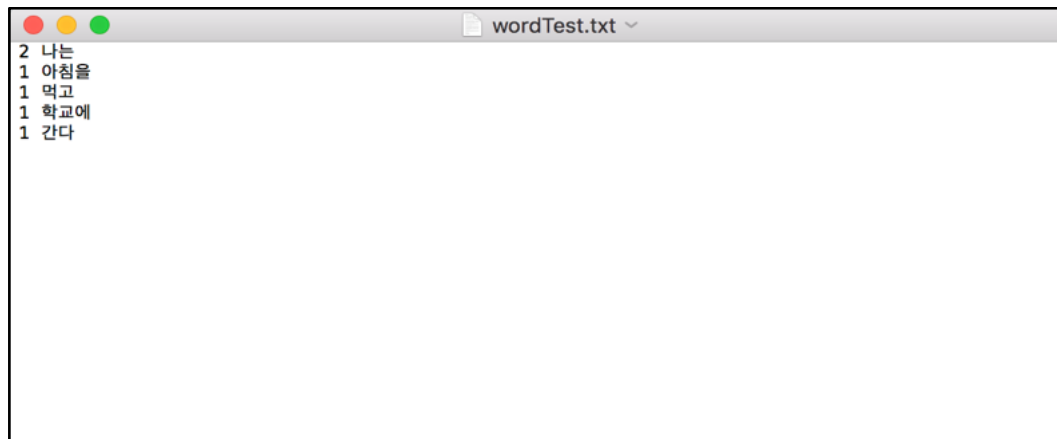
## \* 알고리즘

- 1) 어절단위로 빈도 수를 세어 <freq, word> 형태로 txt 에 저장한다.
- 2) 각 word 별 출현 확률을 계산한다. (해당 단어의 빈도수 / 전체 단어 빈도수)
- 3) 각 단어의 출현 확률을 모두 곱하여 문장 생성 확률을 계산한다.

## \* input text

나는 나는 아침을 먹고 학교에 간다

## \* wordcount.exe 실행 결과



## \* 문장 생성 확률 계산 결과

```
gwonbogyeong-ui-MacBook-Pro:빅 데이터 bokyeong$ ./a.out
count : 6.000000
2 나는 freq = 0.333333
1 아침을 freq = 0.166667
1 먹고 freq = 0.166667
1 학교에 freq = 0.166667
1 간다 freq = 0.166667
probability : 0.000257
```

## \* 전체 코드

```
#include <stdio.h>
#include <string.h>

float count = 0;
float freq;
float prob = 1.0;

void sum(FILE* fp)
{
    int i1;
    char c1[100];
    fp = fopen("wordTest.txt", "r");

    while(!feof(fp))
    {
        fscanf(fp,"%d %s", &i1, &c1);
        if(feof(fp)) break;
        count += i1;
    }
}

int main()
{
    FILE *fp = NULL;

    fp = fopen( "wordTest.txt", "r" );

    sum(fp);
    printf("count : %f\n", count);

    while(!feof(fp))
    {
        char c[100];
        int i;

        fscanf(fp,"%d %s", &i, &c);
```

```
    if(!feof(fp)) break;

    printf("%d %s", i, c);
    freq = i/count;
    printf("freq = %f\n", freq);
    prob *= freq;

}
printf("probability : %f\n", prob);

    fclose(fp);

    return 0;

}
```