과목 명: 시스템프로그래밍

담당 교수 명: 박 운 상

<<Assignment 5>>

**서강대학교 컴퓨터공학과**

**20141578**

**정민교**

목 차

1. 프로그램 개요 3

2. 프로그램 설명 3

2.1 프로그램 흐름도 3

3. 변수 정의 4

3.1 mapper.py 4

3.1.1 str1 4

3.1.2 prev 4

3.2 reducer.py 4

3.2.1 words 4

3.2.2 str1 4

3.2.3 splited 5

3.2.4 key 5

3.2.5 word 5

3.2.6 count 5

4. 코드 5

# 프로그램 개요

대용량 corpus로부터 Bi-Gram 언어모델을 python과 AWS Elastic MapReduce 시스템을 이용한 분산처리를 통해 빠르게 생성한다.

요구사항은 다음과 같다.

1. N-Gram 언어모델을 구현한다.
2. 각 줄에 단어 조합과 그 단어 조합이 나온 횟수를 Tab으로 구분하여 저장한다.
3. 단어 조합은 알파벳 순으로 정렬한다.

# 프로그램 설명

## 프로그램 흐름도

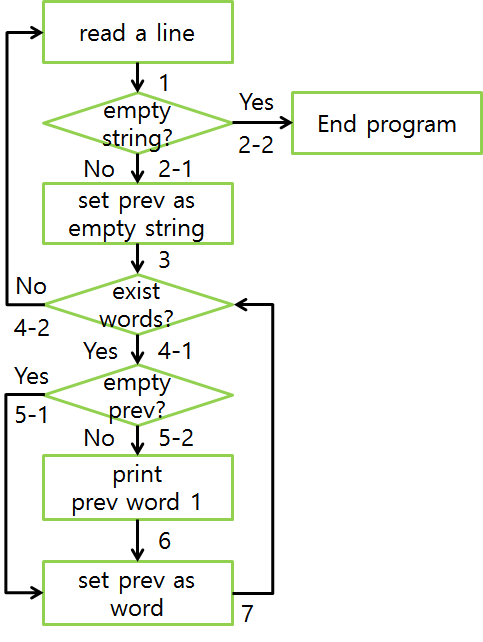


그림 1> mapper.py 흐름도

string에 더 이상 문자열이 들어오지 않을 때까지 읽으면서, 각 줄을 읽어 수행할 때 마다 이전 단어가 있는지 확인하고. 이전 단어가 있다면 이전 단어와 현재 단어와 1을 출력해주고 이전 단어를 갱신한다.

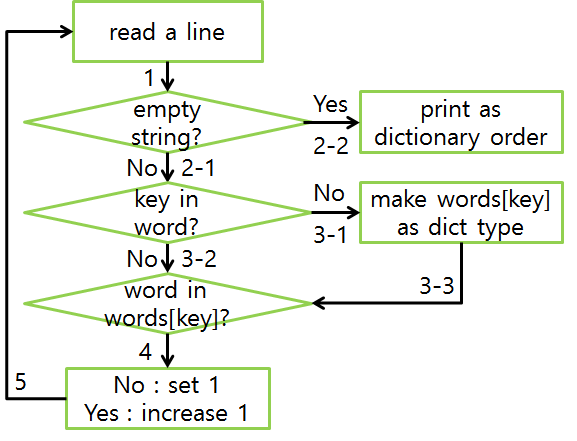


그림 2> reducer.py 흐름도

mapper.py로부터 생성된 standard output으로부터 더 이상 문자열이 들어오지 않을 때까지 읽으면서 {key : {key : value, …}, …}로 이루어진 중첩된 dictionary type의 변수인 words에 첫 번째 읽은 단어를 바깥의 key로, 두 번째 읽은 단어를 안쪽의 key로 지정하고 이 조합이 있었다면 value를 1증가, 없었다면 value를 증가시키지 않고 1을 할당한다. 다 읽은 후, 바깥쪽의 key순으로 정렬하고 그 key에 해당하는 dictionary type의 key순으로 정렬해 알파벳 순으로 출력한다. 출력형식은 다음과 같다 : (word1 word2\tvalue)

# 변수 정의

## mapper.py

### str1

corpus를 standard input으로 받은 문자열(한 줄)

### prev

해당 줄에서 이전 단어를 나타내는 변수(맨 처음 word를 읽을 때는 prev는 empty string이다.)

## reducer.py

### words

결과값을 dictionary타입으로 저장하기 위한 변수. 한 element는 {key : {key : value, … }}로 이루어져있다.(중첩된 dictionary type)

### str1

mapper.py의standard output을 standard input으로 받는 문자열

### splited

str1이 받은 문자열을 공백문자 단위로 쪼개놓은 list 자료구조

### key

previous word에 해당하는 문자열

### word

current word에 해당하는 문자열

### count

해당 pair가 몇 번 나왔는지 알 수 있는 숫자(항상 1)

# 코드

mapper.py

#!/usr/bin/python

from sys import stdin, stdout

while True:

str1 = stdin.readline().lower()

if not str1:

break

prev = ""

for word in str1.split():

if prev:

stdout.write("%s %s\t1\n" % (prev,word))

prev = word

reducer.py

#!/usr/bin/python

from sys import stdin, stdout

words = {}

while True:

str1 = stdin.readline()

if not str1:

break

splited = str1.split()

key, word, count = splited[0], splited[1], splited[2]

if key not in words:

words[key] = {}

if word in words[key]:

words[key][word] += 1

else:

words[key][word] = 1

for keys in sorted(words):

for word in sorted(words[keys]):

stdout.write("%s %s\t%d\n" % (keys, word, words[keys][word]))