**\*개발 환경\***

**-Python 2.7.12 / Anaconda custom (64-bit) on linux**

**-Visual studio 2010 (window)**

**\*주의사항\***

**-텍스트 파일을**

**python 프로그램의 입력으로 사용할 때는 UTF-8 인코딩**

**cpp 프로그램의 입력으로 사용할 때는 ANSI 인코딩을 준수하여야 올바른 결과를 얻을 수 있음.**

요구되는 파일

**1. category\_small.txt**

-제공해주신 광고 키워드 분류 엑셀표에서 category\_small 행을 그대로 복사한 것

/key/category\_small.txt

**2. label.txt**

-제공해주신 광고 키워드 분류 엑셀표에서 category\_small, keyword 행을 그대로 복사한 것

/key/label.txt

**3. nouns.txt**

-<코드매뉴얼-소설가공>에서 5번 word2vec 매뉴얼 <사용 2>참고

* **엑셀 기준으로 카테고리의 경우 1166개의 분류를 기준으로 코드를 작성하였음**

-----------------------------------------------------------------------------------------------

**1. cat\_redd.py : 카테고리 텍스트 파일의 중복을 제거하고 카테고리 번호를 붙여줌**

입력 : 카테고리 텍스트 파일

/key/category\_small.txt

출력 : 중복 제거된 카테고리 텍스트 파일

/key/category\_small\_red.txt

<출력형식>

1

치과

2

성형외과

3

신경과

....

실행 예시 : python cat\_redd.py

(카테고리 텍스트 파일의 중복을 제거)

-----------------------------------------------------------------------------------------------

**2. labeling.cpp : 레이블 텍스트 파일 개행**

입력 : 레이블 텍스트 파일

label.txt

출력 : 개행된 레이블 텍스트 파일

label1.txt

<출력형식> 홀수행 : 카테고리명, 짝수행 : 해당 카테고리의 키워드

치과

치아교정

....

성형외과

모발이식

...

실행 예시 : 윈도우 환경 visual studio에서 컴파일 및 실행 ( 리눅스 gcc 컴파일러 가능)

(사용자 입력 없음, 단순 실행)

(레이블 텍스트 파일 개행 수행)

**3. key\_sep.py : 카테고리 번호와 속한 키워드를 매칭시켜 저장**

-입력 1

/key/category\_small\_red.txt

-입력 2

/key/label1.txt

-출력

/key\_sep/cat\_total.txt

<형식> 홀수행 : 카테고리 번호, 짝수행 : 해당 카테고리의 키워드

1

치아

1

치아교정

...

2

모발이식

2

성형

...

실행 예시 : python key\_sep.py 1077

(1077은 카테고리의 개수)

(각 키워드에 맞는 카테고리 번호를 레이블링 해줌)

**4. keyword.cpp : 소설 속에서 등장하는 키워드만 추출, 한글자 짜리 키워드 제거, 카테고리 별 키워드 개수 계산**

입력 : nouns.txt, cat\_total.txt

출력 : cat\_appear.txt / 소설 속에 등장하는 키워드 추출, 한글자 짜리 키워드 제거

<형식>

1 치아

1 교정

...

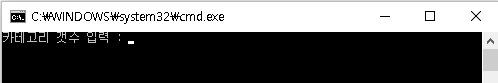
2 모발

2 이식

2 성형외과

...

실행 예시 : window visual studio 실행 ( 리눅스 환경에서 gcc 컴파일러도 무관)



(카테고리 개수 입력하면 (정수형태) 프로그램 시작)

(소설 내 등장하는 키워드와 전체 키워드를 이용하여 처리)

-----------------------------------------------------------------------------------------------

**5. numbering.cpp : 전체 의미 단위 (형용사, 부사 등 포함)의 행렬에 대하여**

1. 각 소설 내 명사에 대하여 몇번째 의미 단위에 속하는지 저장 (몇번째 인덱스인지..)

2. 각 소설 내 등장 키워드에 대하여 몇번째 의미 단위에 속하는지 저장

입력 : nouns.txt, cat\_appear.txt, Lima\_Full\_w2v.txt

출력 1 : noun\_numbering.txt

<형식> 각 행은 nouns.txt의 각 행의 명사가 전체 의미 단위 행렬에서 몇번째 인덱스인지 나타냄

0

23

20

22

28

...

출력 2 : cat\_appear(numbering).txt

<형식> 각 행은 cat\_appear.txt.의 각 행의 키워드가 전체 의미단위 행렬에서 몇번째 인덱스인지 나타냄

(카테고리 번호, 인덱스)

1 10115

1 9750

1 14804

출력 3: num\_of\_nouns.txt

소설 속 명사의 개수를 기록

<형식>

81041

...

출력 4: cat\_appear\_num.txt / 각 카테고리별 키워드 개수를 카운팅

<형식> : 각 n 번째 행은 카테고리 번호 n의 키워드 갯수를 나타냄

19

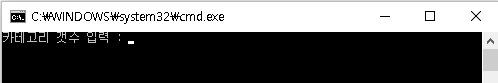
20

15

...

실행 예시 : window visual studio 이용 ( linux gcc 무관 )

(사용자 입력 : 카테고리 개수 입력)



-----------------------------------------------------------------------------------------------