**\*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

-모든 소설 텍스트 파일

-소설 텍스트 파일 이름은 순차적인 정수여야 함

-예시> 1.txt, 2.txt, 3.txt, ....

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

**1. 소설 txt 파일들 인코딩 필요 (ANSI -> UTF-8)**

-실험에서는 Masoris’ Software Pack의 Btec이라는 프로그램을 이용함

하지만 다른 텍스트 인코딩 프로그램을 이용하여도 무관할 것으로 보임

입력 : 모든 소설 텍스트 파일

1.txt, 2.txt, ...

출력 : 인코딩된 소설 텍스트 파일

/text/1.txt, 2.txt, ...

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

**2. txt\_pre.py : 텍스트 파일 속 불필요한 개행 제거, 형태소 분석기 구동을 위한 전처리**

입력 : 모든 소설 텍스트 파일, 모든 소설 텍스트 파일의 갯수

/text/1.txt, 2.txt, ...

출력 : 전처리된 모든 소설 텍스트 파일

/Txt/1.txt, 2.txt, ...

실행 예시 : python txt\_pre.py 500 (리눅스 커널 명령어, 소설 갯수가 500개일때)

(의미 : text 폴더의 1.txt 부터 500.txt 까지 순차적으로 처리하여 Txt 폴더에 출력파일을 생성함)

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

**3. noun\_extracter.py : 형태소 분석/ 텍스트 내 명사들 추출하여 하나의 파일로 생성**

입력 : 전처리된 모든 소설 텍스트 파일

/Txt/1.txt, 2.txt ...

출력 : 추출된 명사들이 모두 통합된 하나의 텍스트 파일

LimaFull\_nouns.txt

실행 예시 : python noun\_extracter.py 500(리눅스 커널 명령어)

(1.txt부터 500.txt 까지 순차적으로 형태소 분석을 통하여 명사를 추출한 뒤 하나의 텍스트 파일에 기록)

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

**4. morph.py : 형태소 분석/ 텍스트 내 문장들 의미 단위로 분석하여 하나의 텍스트 파일로 생성**

입력 : 전처리된 모든 소설 텍스트 파일

/Txt/1.txt, 2.txt, ...

출력 : 의미 단위로 처리된 단어들을 하나의 텍스트 파일로 통합

LimaFull.txt

실행 예시 : python morph.py 500

(1.txt부터 500.txt 까지 순차적으로 형태소 분석을 통하여 의미 단위를 추출한 뒤 하나의 텍스트 파일에 기록)

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

**5. word2vec : word2vec 라이브러리 이용 (c언어 기반), makefile을 통해 컴파일**

<사용1> **LimaFull.txt 학습**

사용 커맨드 : ./word2vec -train LimaFull.txt -output LimaFull\_w2v.txt -size 200

-cbow 0 -iter 20

(train : 학습 파일

output : 출력 파일

size : word2vec 벡터의 크기

cbow : word2vec 기법인 skip-gram과 cbow중 무엇을 사용할 것인가 ( 0 또는 1 )

iter : 학습 반복 횟수는 몇 회로 할 것인가

)

용도 : 각 단어들의 vector representation을 word2vec 기법을 통하여 학습. 단어 간의 유사도를 각 단어의 vector representation을 기반으로 계산하기 위한 과정

입력 : 같은 폴더에 LimaFull.txt 있어야 함

출력 : Lima\_Full\_w2v.txt

<사용2> **LimaFull\_nouns.txt 중복 제거**

사용 커맨드 : ./word2vec -train LimaFull\_nouns.txt -output nouns.txt -size 0 -iter 1 -min-count 0 -negative 0

용도 : word2vec 학습 시에 전체 단어들 중 중복을 체크하여 제거하는 기능이 있음. 이를 이용하여 중복 제거

입력 : 같은 폴더에 LimaFull\_nouns.txt 있어야 함

출력 : nouns.txt

<참고>

\*word2vec 학습 후 인코딩 문제로 인해 결과 텍스트 파일에 한글 깨짐 현상이 발생할 수 있음.

추후 유사도 계산 과정에서 결과가 이상할 경우 이를 확인해야함.

한글 깨짐 현상이 발생하였을 때 vi로 텍스트 파일을 열어

:e ++enc=utf-8

을 입력하고

:wq!

를 입력하여 덮어쓰기 하면 됨