Project #1

M1505.001600 정보모델링기법과 응용

2019년도 봄학기

제출기한: **2019년 4월 15일 23:55까지**

1. (40점) 팀별로 주어진 url의 arxiv를 crawling하여 다음과 같이 분석하시오.

   1-1. (10점) BeautifulSoup를 사용해 논문의 제목, 저자(여러 명일 경우 모두), 제출 날짜, Abstract, Subjects 내용을 출력하시오.

   1-2. (5점) 1-1의 결과 얻어낸 Abstract 내용을 단어 단위로 tokenize를 진행 한 후, 단어들을 POS_Tagging한 결과를 출력하시오

   1-3. (5점) 1-2의 결과 tokenize된 단어들의 등장 빈도를 count하여 빈도가 높은 순으로 정렬한 결과를 출력하시오.

   1-4. (20점) 1-3의 결과를 이용하여 주어진 코드를 실행하여 wordcloud를 생성해보고, 주어진 wordcloud보다 abstract의 내용을 잘 표현할 수 있는 wordcloud를 생성하고 그 근거 또는 처리방식을 서술하시오.

조	배정 url
1	https://arxiv.org/abs/1809.02121
2	https://arxiv.org/abs/1711.00937
3	https://arxiv.org/abs/1811.06128
4	https://arxiv.org/abs/1810.06773
5	https://arxiv.org/abs/1806.10474
6	https://arxiv.org/abs/1810.12894
7	https://arxiv.org/abs/1806.00250
8	https://arxiv.org/abs/1512.02325
9	https://arxiv.org/abs/1805.08318
10	https://arxiv.org/abs/1703.08840

2. (20점) 아래 3가지 조건들을 만족하도록 word_tokenizer 함수를 작성하시오.

   2-1. (10점) 따옴표로 시작해서 따옴표로 끝나는 단어는 따옴표만 없애시오. 그리고 단어 도중에 따옴표가 나오는 경우 따옴표를 포함한 뒤의 글자들을 모두 삭제하시오.

   - 예시: 'hello' --> hello, imlab's --> imlab, 'hello'world' --> hello

   2-2. (3점) ".com"으로 끝나는 단어는 토큰화되지 않도록 하시오.

   - 예시: naver.com --> naver.com

   2-3. (7점) 마침표(.)로 연결된 단어에서, 마침표 앞, 뒤, 및 사이에 있는 글자가 모두 1개일 경우 마침표를 삭제하고, 0개 혹은 2개 이상일 경우 토큰화되지 않도록 하시오.

   - 예시: i.b.m --> ibm, ieee.803.99 --> ieee.803.99, 127.0.0.1 --> 127.0.0.1

   (* 2-3 조건을 만족하면 2-2도 자동적으로 만족하지만, .com 으로 끝나는지 확인하는 코드를 2-3과는 별도로 코드에 명시)

3. (40점) Zipf의 법칙을 확인할 수 있는 python 코드를 작성하시오.

   3-1. (20점) 주어진 텍스트(bible.txt)로부터 각 단어의 등장 빈도를 count하여 Zipf의 법칙이 성립하는지 확인할 수 있는 python 코드를 작성하시오. 결과로 Zipf의 법칙을 보여줄 수 있는 결과를 출력하시오.

   3-2. (20점) NLTK에서 제공하는 Bigram, Trigram을 이용하여 Zipf의 법칙이 성립하는지 확인해보고 3-1의 결과(Unigram), Bigram, Trigram 의 결과를 차트를 통해 비교하시오. (print 결과 출력 X)

eTL에 업로드된 예시 코드를 응용하시오. 발표자료는 PPT나 PDF의 파일 형태로, 보고서는 Word나 PDF로, python 코드 결과물과 함께 압축하여 eTL 사이트에 제출하시오.