Text Mining(텍스트 마이닝)

㈜와이즈넛

DM팀 토이 프로젝트

1. 목표

비정형 텍스트 데이터에 대해 다양한 처리 및 텍스트 마이닝을 경험 해 본다.

2. 요구 기능

2-1 비정형 텍스트 읽기/쓰기 기능
2-2 분석 설정 기능
2-3 텍스트 전처리 기능
2-4 단어 중요도 구하기
2-5 단어 분석 하기
2-6 분석 속도 프로파일링

2-1. 비정형 텍스트 읽기/쓰기 기능

- *.json 파일을 읽기
- *.json 필드 파싱하기(설정 참고)
- 문서 키는 doc_id 필드를 활용

2-2. 분석 설정 기능

- config.yaml 타입의 설정 파일 만들기
- 설정 파일 내용
  - 파일 읽는 경로 및 파일 명
  - 파일 출력 경로 및 파일 명
  - 분석할 필드 명 기재
  - 전처리 옵션 값 기재(1: 특수문자 제거, 2: 한글 이외에 모두 제거)
  - 단어 출력 개수: 0 ~ N, 0으로 설정할 시 모든 단어 출력
  - 단어 출력 방법(1: 단어 점수 기준 내림차순, 2: 단어 글자 기준 오름차순)

2-3 텍스트 전처리

1. 문서 내용을 문장 단위로 분리할 것
- 문장 분리 조건: [!], [?], [.] 기준으로 분리
2. 설정 옵션에 따라 텍스트를 전처리 할 것
- 전처리 옵션: 설정 참고
3. 공백단위로 단어 분리하기

2-4 단어 중요도 구하기

TF-IDF 알고리즘을 이용하여 각 문서 별로 단어의 스코어를 계산하기
가장 높은 단어의 스코어가 100이 되도록 정규화 하기(정규화 방법 설계 필요)

문서 별로 분석 결과 출력하기
출력 포맷
{“DOCID”: “문서번호”, “KEYWORD”, “단어^점수 단어^점수 단어^점수…”}

단어 출력 방법은 설정에 따라 출력하기

2-5 단어 분석하기

1. 전체 문서에서 단어 개수 구하여 정렬하여 출력하기 (상위 100개)
Ex) tf.txt
가 10
나 9
다 8
….

2. 각 문서에 가장 많이 등장한 단어 정렬하기 출력하기(상위 100개)
Ex) df.txt
가 10
나 9
다 8
…

3. 단어 스코어 상위 10개 기준으로 스코어 편차가 가장 많이 나는 문서 구하기
EX) diff.txt
1 50 가 100 나 50
2 35 가 100 나 65
3 20 가 100 나 80

2-5 단어 분석하기

4. 스코어 100점이 가능 많은 문서대로 나열하기
Ex) doc.txt
1 10
2 9
3 8
…

5. 전체문서에서 단어 길이가 긴 순서대로 정렬하기(상위10개)
Ex) long.txt
가나다라마바사
가나다라마바
가나다라
가나다
…

6. 전체문서에서 한글자 단어 출력하기
Ex) one.txt
가
나
다
…

2-6 분석 속도 프로파일링

- 파일 읽기
- 기능 수행(Function 별)
- 파일 출력

고려사항

- 프로그래밍은 재사용성/확장성을 고려하여 설계/개발 되어야 한다
- 데이터 크기가 커지는 것에 대해 처리시간이 Linear하게 증가하는것을 염두해 두어야 한다
- 기능의 삭제/수정/추가가 용이하도록 구조를 정의해야 한다

선택사항

- 형태소 분석기를 이용하여 단어를 공백 단위가 아닌 형태소 단위로 분리
- 키워드 추출 알고리즘을 TF-IDF 외 다른 알고리즘을 적용