프로젝트 #3: Document search engine & Classification and Clustering

본 프로젝트는 텍스트 데이터에 대해 검색 엔진 모듈, 분류 및 군집화 모델 구현을 목적으로 한다. 본 프로젝트는 크게 두 부분으로 나뉜다.

PART Ⅰ. 문서 검색 엔진

PART Ⅱ. 문서 분류 및 군집화

채점 기준

- **PART Ⅰ** (35%) : 성능평가(20%), 성능 개선 방법에 대한 근거와 정당화 내용(15%),
- **PART Ⅱ**(55%) : 분류 모델 성능평가(15%, 15%), 군집화 모델 성능평가(15%), 군집화 분석 (10%)
- 보고서 품질(5%), 발표(5%)

PART Ⅰ. 문서 검색 엔진

PART Ⅰ에서는 주어진 질의어와 관련이 높은 순서대로 문서들을 나열하는 검색 엔진 모듈을 python의 whoosh 라이브러리를 사용하여 구현하여야 한다.

- 사용 데이터: NPL 데이터셋
  - document.txt: 11,429개 문서 파일
  - query.txt: 93개 질의어 파일
  - relevance.txt: 각 질의어의 실제 연관 문서가 명시된 정답 파일
- 작성 모듈
  - [선택] make_index.py: 문서의 ID와 contents를 index에 저장하는 함수. 기본으로 주어진 index를 사용하지 않을 시에 해당 모듈 작성
  - QueryResult.py: 텍스트 형태의 질의어를 입력 받아 whoosh 질의어 객체로 변환 후 검색 결과 반환
  - CustomScoring.py: 문서들을 질의어와 관련 높은 순서대로 나열할 때 사용하는 문서 채점 함수. 사용 가능한 기본 정보로는
    - ✓ 문서 내 단어 빈도(TF)
    - ✓ 역문서 빈도(IDF)
    - ✓ 전체 데이터셋 내 단어 빈도
    - ✓ 문서 개수
    - ✓ 문서 길이 (단어 개수)
    - ✓ 전체 데이터셋 내 단어 개수
    - ✓ 문서 당 평균 단어 개수

    등이 있으며, 제공되는 정보 이외의 정보를 추출 가능하다면 추가적으로 사용 가능

- 평가 방법
  - 93개의 질의어 중 임의로 정해진 15개의 test 질의어에 대한 검색 성능 평가
  - 평가 지표로는 BPREF 사용 [evaluate.py에서 자동으로 계산]

$$\text{BPREF} = \frac{1}{R}\sum_{d_r}\left(1 - \frac{N_{d_r}}{R}\right)$$

    · d_r은 연관 문서

    · N_{d_r}은 d_r보다 높게 랭크된 비연관 문서의 수

  - 점수 산정: 총 35점
    - ✓ 성능 평가 점수 20점: 15개 test 질의어의 평균 BPREF을 [0.5,1] scale한 후 * 20
    - ✓ 성능 개선 방법에 대한 근거와 정당화 내용 15점: 절대평가

- 주의 사항
  - 제공되는 질의어에 test 질의어가 포함된 프로젝트 상황 특성을 활용한 **질의어–문서 매칭 금지**
    - ✓ 성능 개선 방법에 대한 근거 점수. 성능 평가 점수 모두 0점
  - 문서 분석은 허용, 질의어 분석은 금지

PART Ⅱ. 문서 분류 및 군집화

PART Ⅱ에서는 New York Post의 영어 신문 기사를 분류, 군집화하는 모델을 python의 sklearn 라이브러리를 사용하여 구현하여야 한다.

(R2-1) 영어 신문 기사 분류

- Business, Entertainment, Living, Metro, Shopping, Sports, Tech 총 7개의 카테고리
- 각 카테고리마다 약 200개의 기사 데이터 제공
- text 폴더 내부에 train / test 폴더 구분, 각 폴더 내에 7개의 카테고리를 이름으로 하는 폴더 내에 텍스트 파일로 존재
- 제공 시에 test 폴더에는 각 카테고리별 마지막 날짜의 기사 약 5개 정도만 존재
- 주어진 데이터 외에 추가로 크롤링 등의 방법을 이용한 **데이터 추가, 파일 수정, 일부만 사용 불가능**
- 2020.12.09(프로젝트 마감일) 이후 작성된 기사 30개에 대해 올바르게 분류한 개수로 성능 평가
- 각 모델에 대해 학습시킨 모델을 pickle 파일로 저장하여 코드와 함께 제출

- 2-1-1. Naïve Bayes Classifier
  - 점수 15점: 30개 신문기사 중 올바르게 분류한 개수 * 0.5
- 2-1-2. SVM
  - 점수 15점: 30개 신문기사 중 올바르게 분류한 개수 * 0.5

(R2-2) 영어 신문 기사 군집화

- K-means Clustering
- 분류와 같은 데이터 사용. train / test 구분이 필요 없기에 합쳐 놓은 text_all 폴더 사용
- 보고서에 군집화 결과에 대한 분석 포함 필수
- 평가 지표로는 V-measure 사용 = homogeneity와 completeness의 조화 평균

$$\text{Homogeneity: } \mathrm{h} = 1 - \frac{H[C|K]}{H[C]}$$

$$\text{Completeness: } \mathrm{c} = 1 - \frac{H[K|C]}{H[K]}$$

$$\mathrm{V-measure: } \mathrm{v} = \frac{2hc}{h+c}$$

- $H[C]$: 클래스 엔트로피, 여러 클래스에 분산되어 있을수록 큰 값
- $H[C|K]$: 군집화 종료 후 클래스 엔트로피

- $H[K]$: 군집 엔트로피, 여러 군집에 분산되어 있을수록 큰 값
- $H[K|C]$: 클래스 별로 분류한 후의 군집 엔트로피

- 점수 산정: 총 25점
  - ✓ 성능 평가 점수 15점: 분류에서 test 데이터로 사용한 데이터까지 모두 포함하여 측정한 V-measure * 15
  - ✓ 군집화 결과 분석 10점: 절대평가

결과물들을 'DMA_project3_team##.zip'파일로 압축하여 발표일 전날인 12월 9일 23:59까지 ETL에 업로드해야 한다. ETL 상에 문제가 생겼을 경우 changjin9653@snu.ac.kr 로 오류 증명 파일과 함께 해당 일시까지 보내야 한다. 제출해야 할 결과물과 파일명, 파일 확장자는 다음과 같다.

- 보고서 (20페이지 이내)
  - 파일명: DMA_project3_team##_보고서.pdf
- 발표 자료 및 발표 동영상 (5분 이내)
  - 발표자료 파일명: DMA_project3_team##_발표자료.pdf
  - 발표동영상 파일명: DMA_project3_team##_발표동영상.mp4
  - 발표동영상은 팀당 5분 이내로 제작되어야 하며 powerpoint의 녹화기능을 사용한다.
- Python 프로그램 코드 및 결과물
  - 사용 라이브러리로는 nltk, whoosh, sklearn, numpy, pickle를 기반으로 한다. 필요하다면 다른 라이브러리를 사용할 수 있으나 이에 대해 changjin9653@snu.ac.kr로 사전에 메일을 보내야 한다.
  - 문서 검색 엔진 구현에서 사용하려는 방법이 허용되는지 애매할 경우 역시 메일
  - SE 폴더 (PART Ⅰ)
    - ✓ index 폴더: 주어진 index 혹은 신규 생성한 index
    - ✓ make_index.py: 주어진 index를 사용하지 않았을 경우 첨부
    - ✓ CustomScoring.py
    - ✓ QueryResult.py
  - CC 폴더 (PART Ⅱ)
    - ✓ classification.py
    - ✓ clustering.py
    - ✓ DMA_project3_team##_nb.pkl: Naïve Bayes Classifier 이용한 모델
    - ✓ DMA_project3_team##_svm.pkl: SVM 이용한 모델
  - 추가 데이터 및 데이터 파일 가공은 허용하지 않았기에 첨부할 필요 없음.
  - 해당 코드로 re-train한 결과와 제출한 모델의 결과가 같아야 함.