170925

DISCUSSIONS (TF-IDF)

REVIEW

Term Frequency (TF)

- "ATF can be very misleading"

- Normalize term frequency to take into account document size, DB size/characteristics
  - $n(d)$ = number of (all) terms in the document d
  - $n(d, t)$ = number of occurrences of term t in the document d.(ATF)
  - $TF(d, t)$ = normalized term frequency of a *term* t in a document d

- Candidates for $TF(d, t)$
  - $n(d, t) / n(d)$
  - $n(d, t) / max_{k \in d}(n(d, k))$
  - ***the one used in textbook*** $TF(d, t) = \log\left(1 + \frac{n(d, t)}{n(d)}\right)$

TF-IDF

- **IDF** (Inverse Document frequency)
  - *n(t)* = number of documents (in DB) containing term *t*
  - Importance of a doc *d* to a *term t* is inverse proportional to *n(t)*
  - *Candidates for IDF:*

    IDF(t) = **1 / *n(t)*** **(the one used in textbook)**

    1 / [*(n(t)/N)* + 1] (*N*: num. of docs in DB)

    [*logN – log n(t)* + 1]

- **TF-IDF** (Term Frequency-Inverse Document frequency)

$$TF\text{-}IDF\,(d, t) = \quad TF(d, t)\ \text{X}\ IDF\,(d, t) = \frac{TF\,(d, t)}{n(t)}$$

Inverted Index (File)

- Most commonly used index structure for IR
- Mapping from each keyword K_i to the set of documents S_i that contain the keyword
  - documents identified by identifiers

- Inverted index may record
  - keyword locations within document
  - number of occurrences of keyword (or TF)

- ***and*** operation: Finds documents that contain all of K_1, K_2, ..., K_n.
  - Intersection $S_1 \cap S_2 \cap ..... \cap S_n$
- ***or*** operation: documents that contain at least one of K_1, K_2, …, K_n
  - union, $S_1 \cup S_2 \cup ..... \cup S_n$,.

170925

EXERCISE: TF-IDF WITH WIKIPEDIA

Exercise 1. Wiki table 생성

- Table SQL은 과목 게시판에서 download!
  - Wikipedia 문서 dump 중 500개의 문서 선정

컬럼명	데이터 타입	Primary Key 여부	설명
id	int(11)	O	Wikipedia 문서 고유 id
title	mediumtext	X	Wikipedia 문서 제목
text	mediumtext	X	Wikipedia 문서 내용

id	title	text
16583123	Albert_Brunies	Albert Brunies.Albert Abbie Brunies (January 19, 19...
16585069	Michel_Creton	Michel Creton.Michel Creton (17 August 1942 in Wa...
16629152	Habranthus	Habranthus.Habranthus (copperlily) is a genus of t...
16732148	Domestic_violence_in_Peru	Domestic violence in Peru.Domestic violence in Peru...

Exercise2: Inverted index table 생성

- Wiki table을 이용하여 Inverted index 테이블 생성
  - 문서에 존재하는 모든 단어에 대하여 생성하는 것이 원칙
    - 하지만 문제의 간소화를 위해 정해진 단어에 대해서만 생성!
    - 대상 단어: **debut, two, language, also**
  - 문제에서의 단어의 기준:
    - Wiki table의 text column data를 Python String으로 간주 했을 때, 이를 ' '**(space)로 split하여** 나오는 각 토큰들을 단어로 간주
    - Ex) "Computer is going to be…
      - **"Computer, is, going, to, be…**

term	id
finally	1739797
financed	416890
financial	416890
financial	1019791
financial	2266706
find	2266706

컬럼명	데이터 타입	Primary Key 여부	설명
term	varchar(1000)	X	개별 단어
id	int(11)	X	해당 단어가 포함된 문서 id

Exercise3: 단어의TF-IDF 계산

- Wiki table을 이용하여 문서 내 각 단어의TF-IDF 계산
  - TF-IDF 공식은 강의자료의 **“the one used in the text book”** 이용
  - 문제의 간소화를 위해 주어진 Query에 대해서만 TF-IDF 값을 계산하여 출력
    - Query 1: ID=**41631770**인 text의 단어 **also** 에 대한 TF-IDF
    - Query 2: ID=**6688599**인 text의 단어 **debut**에 대한 TF-IDF
    - Query 3: ID=**13794826**인 text의 단어 **language**에 대한 TF-IDF

Submission

- 제출: lecture@europa.snu.ac.kr
  - 제목: [bde2_IR] <이름>
    - ex) [bde2_IR] 김태욱
  - 개인 별 제출 (단, 실습 동안 discussion 권장)
- 제출물: 아래를 압축하여 제출 (파일 이름: <이름>.zip)
  - 실습 수행 Python 코드
  - Query 1,2,3에 대한 수행 결과 값에 대한 text 파일 (results.txt)