2014130211 고동현

사용언어: Python 3.5.1

<Data structure>

제가 이 project에 사용한 데이터 구조는 다음과 같습니다.

\*User: id, Screen name 을 쌍으로 정렬합니다.(key-value?) 이 프로젝트의 경우 screen name을 활용할 일은 없어 Id기반으로 sorting합니다.

\*Tweet: 이 프로젝트의 경우 한 user의 일부 tweet을 발췌해야 하는 경우는 없으므로 user부터 시작되는 circular link list를 사용했습니다,

word 검색에는 원본을 사용합니다.

\*Friend: linked list를 이용한 directed graph 구조를 지니고 있습니다.

<Expected Performance>

\*tweet, friend 수 관련 통계

Data 파일을 읽을 때 미리 count를 합니다. 단순 탐색 시 기존 유저 list에서 대조를 해야 하므로 O(n)의 시간이 걸립니다. Search 방법을 바꾼다면 O(log n)만큼으로 축소 시킬 수 있을 것으로 보입니다.

\*Tweet세기

길이, 앞 글자로 2개의 index를 사용하여 count를 수행합니다.

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

ex)연세대, yo, 연세

2 - 연 : "연세"

- y : "yo"

3 – 연 : “연세대”

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

1. Index에 동일한 길이가 없을 때 =>새 word로 추가

2. Index에 동일한 길이가 있으나 동일한 첫 글자가 없을 때 => 새 word로 추가

3. Index에 동일한 길이와 첫 글자가 있으나 같은 단어가 없을 때 => 새 word로 추가

4. Index에 동일한 길이와 첫 글자가 있고 같은 단어가 있을 때 =>+1 count

직관적으로 볼 때 길이와 첫 글자가 같지만 다른 단어가 많이 있을 것 같진 않습니다. 하지만 이론적으로는 index 2개와 (리스트)를 합쳐 각각 O(n)이므로 O(n^3)이기에 굉장히 비효율적 입니다.

<Future improvement>

\*Project 요구사항 이외의 것

항상 원본 데이터가 하나여야 한다는 점이 가장 큰 한계로 보입니다. 이 프로그램이 실제로 쓰이기 위해서는 원본데이터 추가 및 배제 기능이 필요할 것 같습니다. 간단하게는 데이터를 추가로 투입하는 것에서(물론 기존에 투입된 데이터를 활용하여 O( ) 감소 기대) 일부 데이터를 배제하는 경우도 생각해 볼 수 있겠습니다.(ex-신규 user들만의 tweet만 수집)