2023 [Backend Project] Searching Tweets

Problem

Engineers at Noovi have built their own search engine, and now it's time to add tweets! After building an initial version of their tweet search system, they believe it can be better. Right now, it returns the most recent tweet (highest timestamp) whose tweet text contains all of the words in the query. The Noovi team wants your to help improve it in 3 specific ways:

1. They want searching to be faster on a large set of tweets
2. They want to return the top 5 most recent tweets instead of a single tweet
3. They want to be able to handle search operators like logical and `&`, logical or `|`, and logical not `!`, as well as grouping by parentheses `(` and `)`. Please see the specification below for examples.

They have given you their current prototype and a subset of tweet data and tasked you with improving it in the above ways.

Running the starter code:

- `python starter_code.py` (requires python 3.7 or greater)
- `go test index_test.go` (requires golang)

Query language specification

The current Noovi prototype retrieves a tweet that contains all the given words in the query. The spaces between the words in the query act as implied logical ANDs, where each word separated by spaces must be in the returned tweet.

To extend the query language, we will be ***explicit*** about logical ANDs in queries, and we task you to add functionality to your search function to parse and evaluate the following operators:

- `&` (ampersand) means logical AND (both the word/expression to the left and right must exist in the resulting tweet

- `|` = (pipe) means logical OR (either the word/expression to the left or the right must be in the resulting tweet, or both)
- `!` (exclamation point) means logical NOT (the returned tweets should NOT contain the following word/expression following this operator)
- Spaces in the query will just exist to separate out words and operators from one another

You can assume the OR operator `|` is between a single word/expression:

- Valid:
  - `Noovi & is & (fast | (very & quick))`
  - `Noovi & is & (fast | quick)`
  - `Noovi | Neeva`
- Invalid:
  - `Noovi & is & fast | very & quick` (this would be ambiguous without parentheses or prioritizing symbols over one another)

The logical NOT operator only applies to the word or expression immediately following it:

- `Noovi & fast & !quick & fun` should return tweets with words “Noovi” AND “fast” AND “fun” AND without “quick”
- `Noovi & search & fast & !(slow | sluggish)` should return tweets with “Noovi” AND “search” AND “fast” but neither slow nor sluggish.

Deliverables (submit as zip file):

- A README that describes:
  - Your approach to the problem.
  - How to run your program (especially important if you use a language other than go or python).
  - Any design decisions, tradeoffs, or assumptions you made.
  - Complexity analysis of your solution vs the starter code.
- Code, updated tests, and time benchmarks you write.

Considerations

1. You are welcome to implement your library in either of the two provided languages or any other language of your choice as long as you follow the pattern of having `ProcessTweets` and `Search` functions like in the starter code. You do not have to implement your solution in both languages.

2. Please avoid the use of third-party libraries. While we enjoy learning about new open source projects, we are most interested in your original technical work. "Standard" libraries and libraries which implement commonly used data structures/functionalities (numpy, matplotlib,...etc.) are fine. Please use your best judgement.
3. You may assume that tweet timestamps are globally unique.
4. Please list any assumptions about tweet data or structure of queries in your readme.
5. You are welcome to create additional tweet CSV files if it helps you test or debug issues.
6. You should treat all queries as CASE INSENSITIVE (“hello” should match “HeLLo”)
7. Please **DO NOT** post your solution publicly on the internet. We sincerely appreciate the time you spend on this, but please time-box yourself and don’t spend more than a few hours on your solution! Hopefully you’ll get to think and learn a bit more about search engines through solving it 🙂

More Examples:

Query: `Noovi & rocks`

- Could return tweets like:
  - “Noovi rocks”
  - “Noovi discovered rare rocks on mars”
- Wouldn’t return tweets like:
  - “Noovi is a rocketship”
  - “Noovi was founded by rockstars”

Query: `Noovi & is & (interesting | exciting) & !boring`

- Could return tweets like:
  - Fixing Noovi tweet search is an interesting problem
- Wouldn’t return tweets like:
  - The work at Noovi is exciting and never boring

Query: `Noovi & search & ((works & great) | (needs & improvement))`

- Could return tweets like:
  - Noovi has great search that actually works
  - I just tried Noovi tweet search and it needs a lot of improvement
- Wouldn’t return tweets like:

- Noovi search is great
- Noovi search needs more great people to work on it

neeva