Twitter Keyword Search

Junyou Chi & Hao Zuo

Task

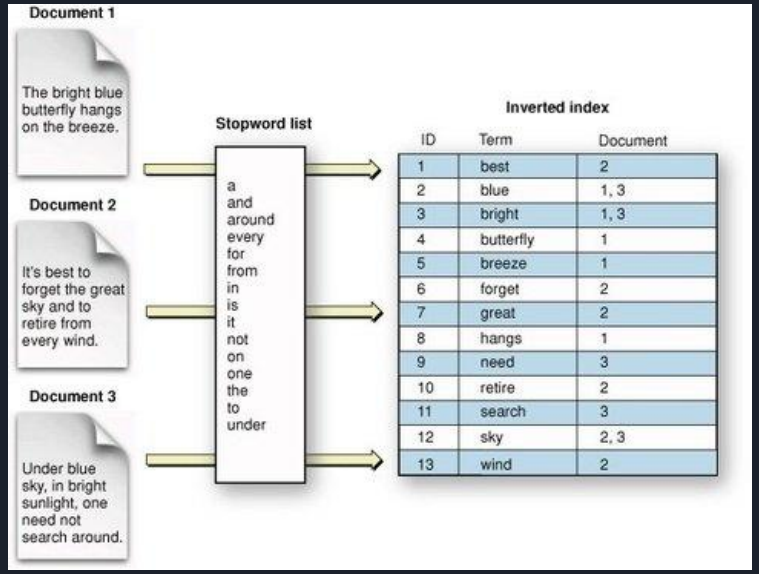
1. Create a keyword search engine that accepts a text file containing a number of tweets as input.

2. Provides a simple user interface for querying the list of tweets against keywords.

Steps

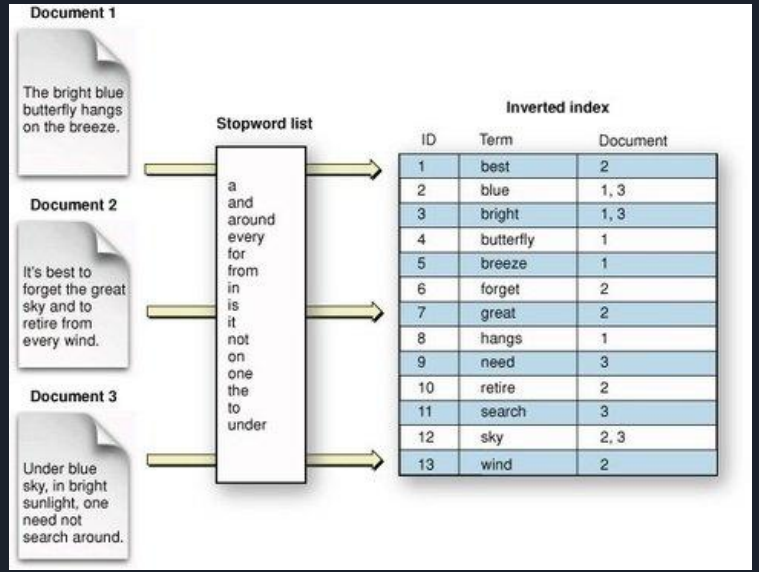
1. Get tweeters from internet(240k twitter dataset)
2. Handle the raw data
   a. Twitter segmentation
   b. Word segmentation
3. Database
   a. Using inverted index to store twitter feeds
   b. Using hashmap to store key words for inverted index
4. Searching function to find and sort twitter feeds
5. GUI

About Inverted Index



An inverted index (also referred to as a postings file or inverted file) is a database index storing a mapping from content, such as words or numbers, to its locations in a table, or in a document or a set of documents (named in contrast to a forward index, which maps from documents to content). The purpose of an inverted index is to allow fast full-text searches, at a cost of increased processing when a document is added to the database.

Data Structure in program



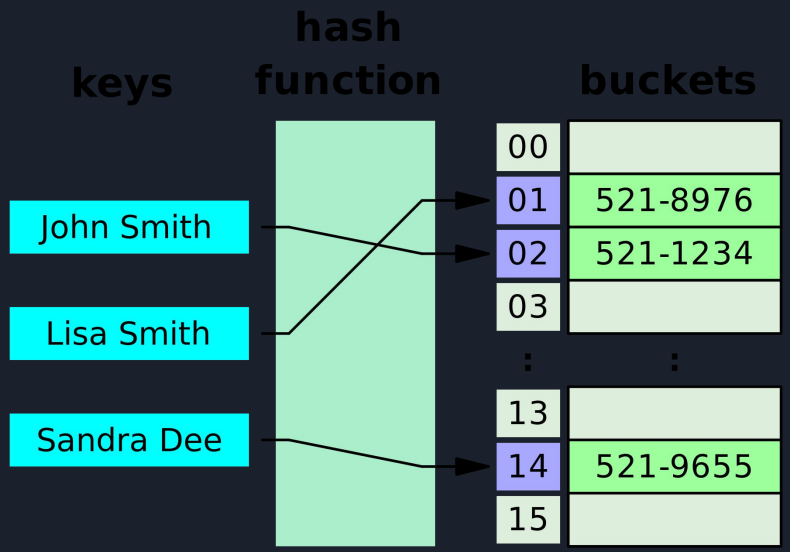
Twi1 : the bright blue butterfly hangs on the breeze

Twi2: the dark red butterfly

the ---> <1,2> <2,1>
bright ---> <1,1>
blue ---> <1,1>
butterfly ---> <1,1> <2,1>
.
.
.

Because the number of keyword is important, we need to save it for twitter feeds.

About Hash Map



In computing, a hash table (hash map) is a data structure that implements an associative array abstract data type, a structure that can map keys to values. A hash table uses a hash function to compute an *index*, also called a *hash code*, into an array of *buckets* or *slots*, from which the desired value can be found. During lookup, the key is hashed and the resulting hash indicates where the corresponding value is stored.

Searching Keyword

1. Finding candidate
    a. Count keywords each twitter has
2. Doing a search to find top 10 related twitter.
    a. Much like finding the max value. Using a array to save top 10 and comparing next candidate with this 10 twitter.

Future Tasks

1. Do more test and compare our program with other keyword searching way.
2. Make a better UI
3. Moving UI to CPP

Demo