LSII: An Indexing Structure for Exact Real-Time Search on Microblogs

§,†Lingkun Wu, §Wenqing Lin, §Xiaokui Xiao, #Yabo Xu

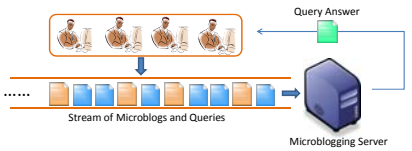
§ NANYANG TECHNOLOGICAL UNIVERSITY

† Agency for Science, Technology and Research SINGAPORE

SUN YAT-SEN UNIVERSITY

1. Introduction

- Microblogging (e.g., Twitter) is an increasingly popular form of online communications.
  - Short text (up to 140 characters).
  - Users can reply/forward others' microblogs.



- Twitter gets **200 million tweets** and **1.6 billion queries** per day in 2011.
- Existing approaches reduce update overhead at the cost of query accuracy.

2. Problem Formulation

- **Objective**: Retrieve *exact* results for *real-time* search on microblogs.
- Each query q asks for the top-k_q microblogs d with a score function in the following form:

$$f(d,q) = w_1 \bullet sig(d) + w_2 \bullet sim(d,q) + w_3 \bullet fresh(d,q)$$

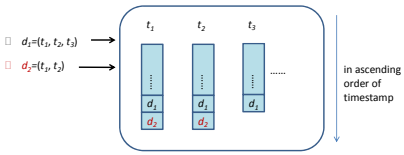
such that
  - $w_1 + w_2 + w_3 = 1$ and $w_1, w_2, w_3 > 0$,
  - $sig(d)$ quantifies the significance of d,
  - $sim(d,q)$ measures the relevance of d to q,
  - $fresh(d,q)$ gauges the freshness of d w.r.t. q based on timestamp.
- i.e., $f(d, q)$ is a linear function of d's scores in terms of its significance, freshness, and relevance to q.

5. Handling Score Update

- Problem: The significance scores of microblogs may get updated.
- Naïve solution: Re-arrange the sorted arrays whenever the significance score of a microblog is updated, which incurs considerable overhead.
- Our solution: Add a buffer to each posting list to cache updates in significance scores; flush the buffer during index mergers.
- Query processing takes both the posting lists and their associated buffers into account.

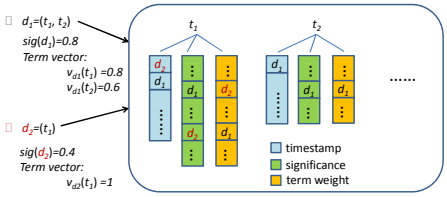
3. Baseline Approaches

- Append-Only Approach
  - Append new microblogs to the ends of posting lists.
  - Each query is processed by scanning all the entries of posting lists relevant to the query.
  - Fast insertion but significant query overhead.



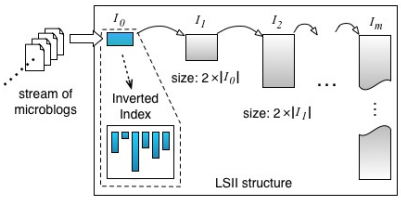
- d1 and d2 are new microblogs.
- t1, t2, and t3 are terms in the dataset.

- Triple-Posting-List (TPL) Approach
  - For each term t, maintain three sorted posting lists, each of which is indexed by B-trees.
  - Process each query with the *Threshold Algorithm* (TA).
  - Efficient query processing but significant update overhead.



4. Log-Structure Inverted Indices (LSII)

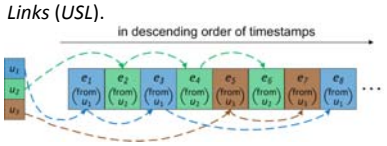
- Basic Idea
  - Maintain a sequence of inverted indices with exponentially increasing sizes.
  - Insert new microblogs into the smallest index.
  - When an index is full, merge it with its immediate successor in the sequence
- Index Structure
  - The structure of I_0 is identical to the one in Append-only Approach.
  - $I_1, \ldots, I_m$ are the same with TPL, except that the posting lists are implemented as *sorted arrays*.



- Query Processing
  - Scan I_0 to obtain the top-k_q microblogs I_0, and put them into a candidate pool C.
  - Apply TA to traverse the entries in $I_1, \ldots, I_m$; update C accordingly.
- Handling Updates
  - First, append the new microblog d to the end of I_0.
  - If I_0 reaches its maximum size, sort I_0 and merge it with I_1.
  - If I_p ($p > 0$) reaches its maximum size, we merge the posting lists in I_p to I_{p+1} in linear time.
- Comparison with TPL
  - Better query efficiency: $I_1, \ldots, I_m$ are implemented as sorted arrays, which makes it more efficient to perform TA; search on I_0 incurs a negligible cost since I_0 is small in size
  - Better update efficiency: insertions are handled by I_0 in a append-only manner; occasional mergers of indices incur small amortized cost.
- Comparison with Append-Only
  - Slightly higher update cost
  - Significantly better query efficiency

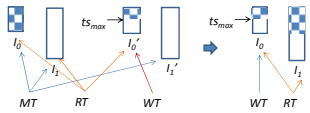
6. Personalized Search

- To avoid visiting irrelevant microblogs, we augment the posting lists with *User-Specific Links* (*USL*).



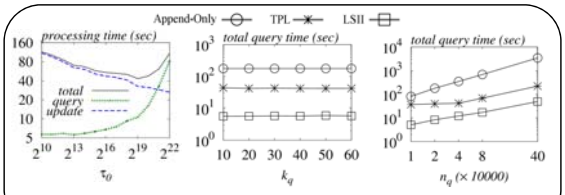
7. Multi-threaded Approach

- The multi-threading approach has three types of threads: Reader Thread (RT), Writer Thread (WT), and Merger Thread (MT).

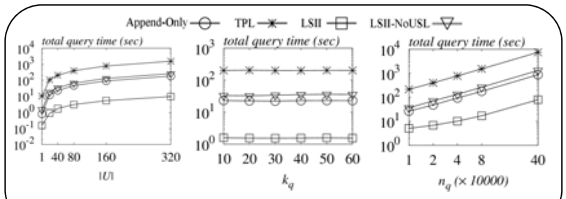


8. Experiments

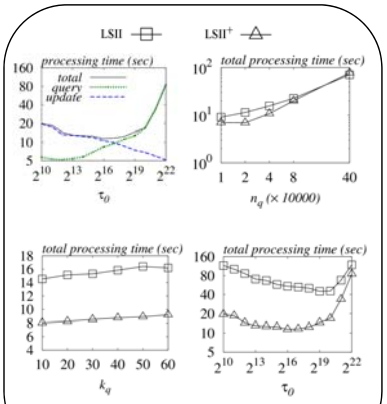
- Dataset: The Microblogs of Twitter
  - 10.4 million microblogs
  - Posted by 260k users
  - 2.6 million distinct terms (excluding stop words)
- Mixed query and update stream
  - 10 million microblogs are first inserted into LSII.
  - The rest 0.4 million microblogs are mixed with queries with random positions.
- Queries
  - Each query contains 1 to 5 terms.
  - Each personalized query associates with a set of users by randomly selected with non-empty answer guarantee.
- Parameters in Experiments
  - τ_0: the size of I_0.
  - n_q: the size of queries.
  - k_q: the size of answer set.
  - U: the size of users in a personalized query.



Experiment 1: Handle Non-Personalized Queries and Updates



Experiment 2: Handle Personalized Queries and Updates



Experiment 3: Study the Performance of the Multi-threaded Approach (LSII+)