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

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-\( k \) microblogs \( d \) with a score function in the following form:
  \[
  f(d,q) = w_1 \cdot \text{sig}(d) + w_2 \cdot \text{sim}(d,q) + w_3 \cdot \text{fresh}(d,q)
  \]
  such that
  - \( w_1 + w_2 + w_3 = 1 \) and \( w_1, w_2, w_3 > 0 \),
  - \( \text{sig}(d) \) quantifies the significance of \( d \),
  - \( \text{sim}(d,q) \) measures the relevance of \( d \) to \( q \),
  - \( \text{fresh}(d,q) \) 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 \( q \).

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.
- **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.
  - \( I_{i-1} \ldots I_0 \) are the same with TPL, except that the posting lists are implemented as sorted arrays.

5. Handling Score Update
- **Problem:** The significance scores of microblogs may get updated.
- Naive 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 merges.
- Query processing takes both the posting lists and their associated buffers into account.

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
  - 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**
  - \( r_0 \): the size of \( I_0 \)
  - \( r_n \): the size of queries.
  - \( r_s \): the size of answer set.
  - \( r_u \): the size of users in a personalized query.