TweetRank

TweetRank is an attempt to apply the PageRank algorithm on Twitter statuses (tweets).

It uses a different rank calculation which considers attributes such as number of replies/retweets and hash tags.

|  |  |
| --- | --- |
| Victor Hallberg | [victorha@kth.se](mailto:victorha@kth.se) |
| Johan Stjernberg | [stjer@kth.se](mailto:stjer@kth.se) |
| Joan Puigcerver I Perez | [joanpip@kth.se](mailto:joanpip@kth.se) |
| Alexander Hjalmarsson | [alehja@kth.se](mailto:hjalle@sgh.se) |
| Christoffer Rydberg | [chrryd@kth.se](mailto:chrryd@kth.se) |

# Components



**Ranker:** computes TweetRank from crawler data

**Solr:** indexes and provides query interface for end user

**Crawler:** collects tweets via the Twitter API

# Crawler

* Uses the Twitter *HTTP REST* API
  + Twitter limits the number of queries to 150 per hour
  + Crawler gathers as much data as possible from each query
  + Use multiple proxies to bypass the query limit
  + Runs on multiple threads in multiple machines
* How does it work?
  + Query the tweets and friends from the first user in a queue of users

# Ranker

* Uses the complete path *Monte Carlo* algorithm, stopping at dangling nodes
* Starts a randomized walk from each node at least 100 times (at most total tweets / 100)
* Random path length - 20% chance at each node that the surfer stops
* Ranker runs on multiple threads, where each thread computes one walk at a time
* Final rank is calculated as the normalized sum of visits for each node from every walk:

|  |  |  |  |
| --- | --- | --- | --- |
| Thread 1 | Thread 2 | Thread 3 | Thread 4 |
| https://github.com/mogelbrod/TweetRank/blob/master/doc/graphs/mc-threads-0.png?raw=true | https://github.com/mogelbrod/TweetRank/blob/master/doc/graphs/mc-threads-1.png?raw=true | https://github.com/mogelbrod/TweetRank/blob/master/doc/graphs/mc-threads-2.png?raw=true | https://github.com/mogelbrod/TweetRank/blob/master/doc/graphs/mc-threads-3.png?raw=true |
| {2, 1, 0, 1} | {1, 1, 1, 1} | {2, 1, 1, 0} | {1, 1, 1, 1} |

**Divide by the maximum number of visits for a node  
and then multiply by 10**

{6, 4, 3, 3}

{10, 20/3, 5, 5}

# Solr / Lucene

* Handles indexing and searching.
* Crawler sends tweets to be indexed by Solr through *HTTP POST* requests (in XML format)
* Current TweetRank data is fetched from a text file on the server
  + Enables rank updates without having to replace (re-index) existing documents
  + Utilizes the *ExternalFileField* format in Solr
* Scores for individual statuses are calculated as a product of:
  + TweetTrank
  + *tf-idf* for terms, where matches against hash tags are boosted

# Results