HomeLessNet

SocialCops Twitter-Elasticsearch Challenge

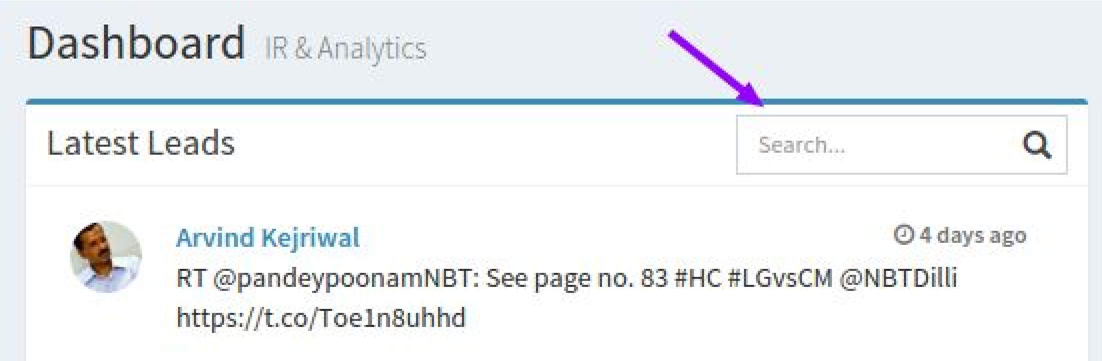
Introduction

HomelessNet (HLN) provides a powerful platform for Mr. Holmes to keep track of leads from reporters of his network. It would enable him to analyse the leads data and reporter activity.

The platform makes the leads searchable and filterable. It provides Graphical analysis tools to visualize the leads data and reporter charts.

Features

- Search



RT @pandeypoonamNBT: See page no. 83 #HC #LGvsCM @NBTDilli https://t.co/Toe1n8uhhd

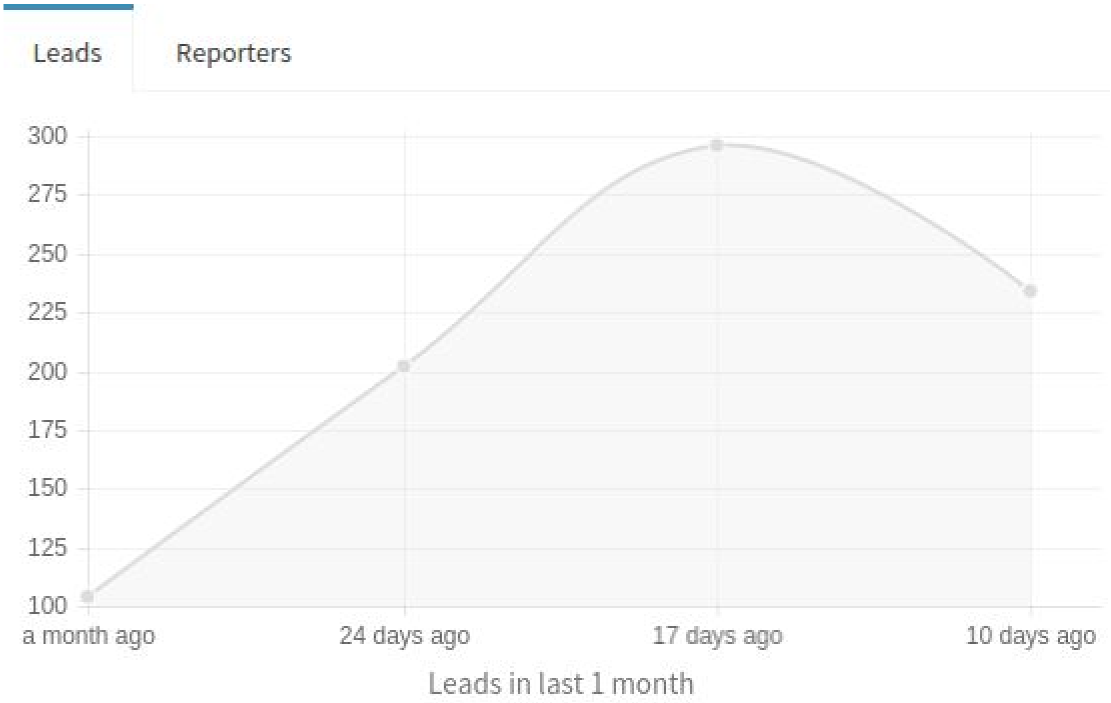
- ➢ Search functionality is powered by Elasticsearch DSL queries.
- ➢ The search query is processed first for extracting hashtags(categories) and @mentions(entities) input by user as a means to filter docs with the use of nltk library
- ➢ The search terms are sent to ES query handler which produces output based on which ES documents matches atleast one of these fields with higher relevance score for those matching most terms.
- ➢ Search is enabled with filter function. For example, a user input *“delhi connect #arvindkejriwal @kejriwal”* would retrieve all the documents

containing terms ['delhi', 'connect', 'arvindkejriwal', 'kejriwal'] and filter them with documents with **categories field** matching 'arvindkejriwal' and **entities field** matching 'kejriwal'. This is done with the use of 'filter' query in 'filtered' ES query. (source : hln.es_handler, hln.es_queries modules)

- Inside the 'filtered' ES query, the search query contains a nested boolean query with 'should' clause to match either or both of following two queries. First being, search tokens match with 'data.text' field containing the lead text. The second one, contains nested query to match 'entities'. If the second query is matched the document's relevance score is boosted by a factor of 2.0 The factor has to be decided based on user xp since, there is no definite way to find most apt boost factor.

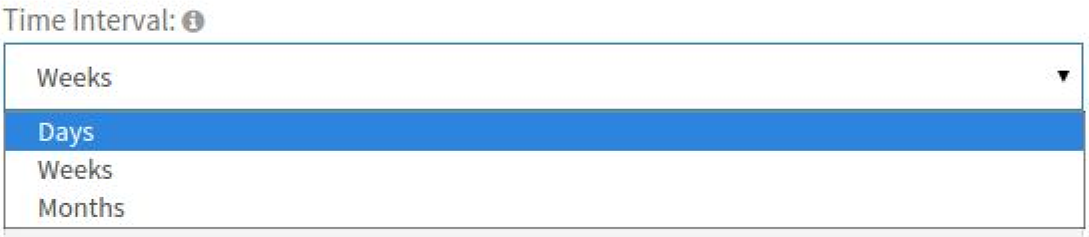
Note: ES doc mappings have been mentioned in the *'mod_twapi.mappings'* module.

❖ Analytics



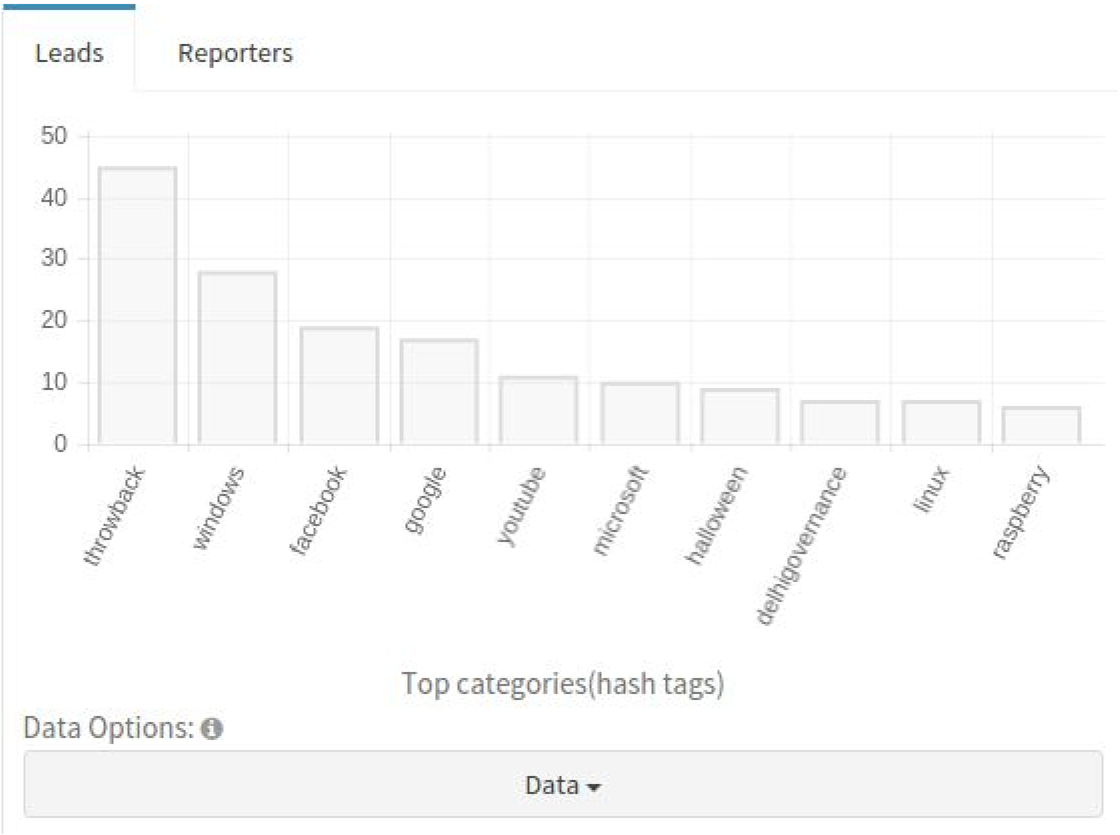
- Analytics tools are powered by ES bucket aggregation queries.
- Facets were used before the advent of aggregations which enabled us to summarize data results from query and create distribution histograms

- ➢ But, even though pretty powerful, they have a limitation of allowing calculation only one level deep. And since, ours is quite nested data, aggregations seemed to be a perfect fit which allows for multi level calculations at query time with single request.
- ➢ The simple leads analytics tab, for example, shows variation in number of leads aggregated per specified *'time interval'* with 'date_histogram' aggregation.
- ➢ Time interval can be also be specified in the ui.

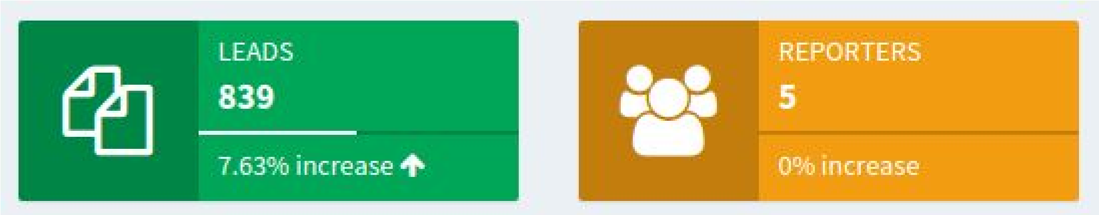


Time Interval: Weeks

- ➢ The leads tab also includes a data option to select lead analytics on categories to know the top trending categories over the ES lead docs.



- The categories histogram data is obtained by using ‘terms’ aggregation.
- The reporters tab displays the top reporters over the entire dataset.
- Time limit functionality has been implemented on the backend using ‘filtered’ query to specify the amount of time, the data needs to be fetched from ES. Currently, it has been limited to 1 month for simplicity of user interface.
- Simple analytics are also present to show count of leads and reporters in ES index using ES count API with % change shown using aggregations.



Notes:

1. UI for time limit options have not been created for the categories and reporter analytics.
2. All datetime quantities are in UTC

❖ ES Indexing

- A separate module is created for obtaining tweets from Twitter Streaming API, process the data and index into ES index. (source : hln.mod_twapi module)
- *“tweepy”* module is used to fetch the tweets which are handler in “hln.twapi_handler” module
- Once the tweets are fetched, relevant information is extracted into python dictionaries and sent to “hln.es_loader” module where data is bulk indexed into Elasticsearch index specified in the config.
- Choice of bulk size depends on the system being used, value of 1000 has been used keeping a quad core system in mind since twitter limits latest tweets to 3200.
- Performance can be improved by using multiprocessing module and making full use of all cores.

- ➢ Currently the module, can be run before starting the server, but can be easily used to load new data into elasticsearch with the use of ajax polling every 15 minutes.

  Note: Check usage for loading new data and starting server using ***“python run.py -h”*** in the repository.

Directions to use

- ❖ Please follow the README in the cloned repository for complete instructions to setup.
- ❖ Use following credentials to login to the application:
  - ➢ Username : holmes@stbart.com
  - ➢ Password : sherlocked

Contact

- ❖ In case of any queries/issues please contact me at *dpkshrma01[at]gmail[dot]com*