Demonstration of Taqreer: A System for Generating Spatio-temporal Analysis Reports on Microblogs

Amr Magdy[#] Mashaal Musleh[§] Saif Al-Harthi[#] Louai Alarabi[#]
Kareem Tarek[§] Thanaa M. Ghanem[⋆] Sohaib Ghani[§]
Hicham G. Elmongui[§] Mohamed F. Mokbel[#]

[#]Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN
[§]KACST GIS Technology Innovation Center, Umm Al-Qura University, Makkah, KSA
[⋆]Department of Information and Computer Sciences, Metropolitan State University, Saint Paul, MN
{amr,alhar035,louai,mokbel}@cs.umn.edu, {mmusleh,ktarek,sghani,elmongui}@gistic.org,
thanaa.ghanem@metrostate.edu

***Abstract*—This demonstration presents *Taqreer*; a system that automates generation of a wide variety of spatio-temporal analysis reports on different attributes of microblogs data, e.g., tweets, reviews, and news comments. *Taqreer* consists of two components: *Taghreed* engine and a *Report Generation Tool*. *Taghreed* supports efficient indexing for incoming fast microblogs in main-memory and scalable indexing for older data in disk. On top of its indexing, *Taghreed* provides efficient query processing for spatio-temporal keyword queries. *Taqreer* exploits such data management capabilities to automate generating different kinds of interactive reports from large archives of microblogs. To this end, *Taqreer* adds a *Report Generation Tool* to automate generation of: (1) Comparative reports: that compare microblogs of arbitrary keywords and arbitrary ranges in space and time. (2) Categorical reports: that analyze categorical attributes of microblogs, e.g., language attribute, over space and time. (3) Image gallery reports: that summarize events and local regions with image galleries. *Taqreer* framework is also extensible to support other types of reports. Throughout this paper, we highlight the system design and components' internals. In addition, we demonstrate *Taqreer* using a rich set of reports that are generated based on a large archive of real Twitter data.**

I. Introduction

Analyzing microblogs data, e.g., tweets, reviews on Yelp and Amazon, and comments on news websites and Facebook, has got a considerable attention of several commercial and research efforts. For example, microblogs are used to analyze user behavior for geo-targeted advertising purposes [7], detecting and analyzing events [1], [3], [8], and extracting news stories [2]. Although the research community has addressed a large set of queries and applications on microblogs, none of those presents an end-to-end microblog data management solution that enables generic microblog analysis. The closest to this goal is TweeQL [6], which is proposed as a general query language for Twitter data; a prime example of microblogs. However, TweeQL just provides a wrapping interface for Twitter streaming APIs without addressing the actual data management issues for microblogs data.

In this demo, we present *Taqreer* [5]; a system for generic spatio-temporal data analysis and interactive report generation on large numbers of microblogs. *Taqreer* consists of two main components: *Taghreed Engine* and a *Report Generation Tool*. The *Taghreed* engine [4] is a scalable query engine for indexing and querying microblogs streams. *Taghreed* employs a scalable indexer that is able to digest newly incoming microblogs with high rates in main-memory indexes. When the allocated memory is filled, older microblogs are flushed from memory to a disk-resident spatio-temporal indexer. Equipped with highly efficient in-memory index, a scalable secondary storage index, and a smart flushing policy, *Taghreed* efficiently retrieves microblogs that satisfy a combination of spatial, temporal, and keyword constraints. Meanwhile, the *Report Generation Tool* receives users' requests and parses the report parameters to determine a set of queries that retrieve the report data and a set of analysis tasks on this data. Then, all queries are submitted to the *Taghreed* query engine to retrieve the data. Based on report type and settings, a set of analysis tasks are performed before the final report is submitted to the requesting user.

Taqreer reports mainly analyze and track the appearance of certain attribute(s) over space and time. Such reporting functionality is important, for example, in tracking interest in diseases (e.g., Ebola) or natural disasters (e.g., earthquakes). Currently, *Taqreer* supports three types of reports: (1) *Comparative reports*, that compare how various keywords are trending over space and time. Such reports can be used in several domains, like analyzing the status of election candidates, the interest in sports teams, or other comparisons based on social media discussions. (2) *Categorical reports*, that analyze categorical attributes of microblogs over space and time, e.g., analyzing the language attribute to understand the diversity of various countries and cities. (3) *Image gallery reports*, that exploit the plenty of images posted on social media to summarize events with image galleries. In addition, *Taqreer* is extensible and provides a platform that allows adding other types of reports.

We demonstrate *Taqreer* through an actual implementation of the system showing its different reports. The reports are generated based on a large archive of 6+ billion real tweets that are being collected through Twitter Streaming APIs since October 2013. The demo attendees will be able to generate and interact with different reports, of the three supported types, based on real social media discussions.

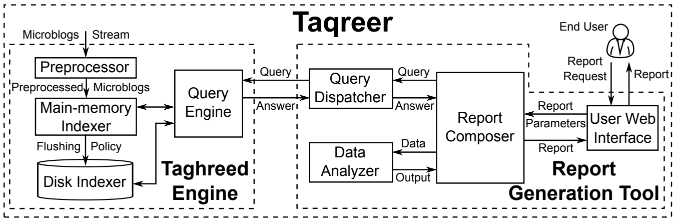


Fig. 1. Taqreer Architecture.

II. Taqreer Overview

Figure 1 gives *Taqreer* system architecture, which is composed of two main system components, the *Taghreed* query engine [4] and a *Report Generation Tool*. *Taqreer* users submit their report generation requests to a *User Web Interface* module. Then, the report parameters are forwarded to a *Report Composer* module, which parses the parameters and divides the report into: (a) a set of spatio-temporal keyword queries that retrieve the necessary data to generate the requested report, and (b) a set of analysis tasks to run on the retrieved answer of the spatio-temporal keyword queries. All spatio-temporal keyword queries are sent to the *Taghreed* query engine through a *Query Dispatcher* module.

Meanwhile, the *Taghreed* query engine continuously receives an incoming stream of microblogs with high arrival rates. The incoming stream is processed and digested in highly scalable and efficient main-memory index structures. Once the memory becomes full, a flushing policy is invoked to move a portion of memory contents to the disk storage in another highly scalable and efficient disk-based index structure. The *Taghreed* query engine answers its incoming spatio-temporal queries from both in-memory and disk-based index structures, based on where the data needed for the query answer reside. Once the query answer is collected, it is sent back to the *Report Generation Tool*, which invokes the *Data Analyzer* module to perform the required data analysis tasks. Finally, the report composer gets the analysis output, e.g., certain attribute count over space and time, composes the report in its final form as an interactive web page and sends it back to the requesting user. We next highlight the internals of both system components.

A. Taghreed Query Engine

Taghreed [4] is the query engine behind *Taqreer*, which has two main responsibilities: (1) digesting incoming stream of microblogs with high arrival rates, and (2) efficient support for spatio-temporal keyword queries over large set of microblogs. *Taghreed* is composed of four main components, namely, *main-memory index*, *disk-based index*, *query optimizer*, and *recovery manager*, described below.

Main-memory index. *Taghreed* employs two in-memory index structures; a keyword index and a spatial index. Both indexes are segmented into temporally disjoint segments that partition data based on arrival timestamp, where each segment includes the data of T hours, where T is a system parameter. Incoming microblogs are digested in the most recent segment. Once the segment spans T hours of data, it is concluded and a new empty segment is introduced to digest the new data. Index segmentation has two main advantages: (a) new microblogs are digested in a smaller index and hence higher digestion rates are supported, and (b) it makes it easier to flush data from memory to disk.

Disk-based index. Similar to main-memory index structures, disk-based indexing supports both spatial and keyword indexes, where each index embeds the temporal aspect in its organization. However, the disk-based index structures are a bit different from the main-memory ones. Each disk index is organized in temporally partitioned segments that are maintained at different levels of temporal granularity, day, week, and month levels. This enables supporting queries with arbitrarily large periods of time, e.g., several months, to access minimal number of index segments.

Query optimizer. *Taghreed* query optimizer selects which index segment(s) should be accessed to retrieve the query answer. Specifically, *Taghreed* provides two index structures in both main-memory and disk: a keyword index and a spatial index. In addition, disk-resident data is replicated at different temporal granularity. Consequently, the query processor may have different ways to process the same query based on: (1) either the keyword or the spatial index would be hit first, and (2) the temporal granularity of disk index segments to hit. The query optimizer employs cost models to select a cheap query plan so that *Taghreed* queries are processed efficiently.

Recovery manager. With hundreds of millions of microblogs managed in main-memory, *Taghreed* accounts for memory failures that may lead to data loss. *Taghreed* employs a simple, yet effective, triple-redundancy model where the main-memory data is replicated three times over different machines. The core idea of this model is similar to Hadoop redundancy model that replicates the data three times. Replicating the data three times significantly reduces the probability of having the three machines down simultaneously and lose all data.

B. Taqreer Report Generation Tool

Taqreer report generation tool is composed of four modules, namely, *User Web Interface*, *Report Composer*, *Query Dispatcher*, and *Data Analyzer*. The user web interface receives input report parameters from end users, forwards them to the report composer module to sync the work among other modules. In particular, the composer goes through four steps: (1) Based on the report type and parameters, the composer determines the set of queries that retrieve the required data and a set of analysis tasks to be performed on that data. (2) The report composer calls the query dispatcher module to submit spatio-temporal keyword queries to *Taghreed* query engine. (3) The retrieved query answers are forwarded to the data analyzer module to perform the required analysis. (4) The report composer adds all the output to an interactive web page and sends it as the final report to the user.

The above steps represent a framework that can be used to support various report types. Currently, *Taqreer* supports three types of reports, namely, *comparative reports*, *categorical reports*, and *image gallery reports*. Table I gives the parameters, queries, and analysis tasks for the three report types, that are briefly described below.

Comparative reports. The first row in Table I shows the parameters, queries, and analysis tasks of comparative

Report Type	Parameters	Queries	Analysis Tasks
Comparative Reports	• n spatial regions R_i, $1 \leq i \leq n$ • m keywords (topics/entities) W_j, $1 \leq j \leq m$ • Time range $[T_s,T_e]$	$n \times m$ queries, each takes: • Spatial region R_i • Keyword W_j • Time range $[T_s,T_e]$	None
Categorical Reports	• Spatial region R, auto divided into n subregions of fixed default area • Time range $[T_s,T_e]$ • Categorical attribute A • Optional m keywords W_j, $1 \leq j \leq m$	n queries, each takes: • Spatial region $R_i \subset R$ • Time range $[T_s,T_e]$ • Optional m keywords W_j, $1 \leq j \leq m$	• Count categories of attribute A for each query microblogs • Aggregate counts over less granular spatial levels
Image Gallery Reports	• m keywords W_j, $1 \leq j \leq m$ • Time range $[T_s,T_e]$ • Optional spatial region R	One query that takes: • m keywords W_j, $1 \leq j \leq m$ • Time range $[T_s,T_e]$ • Optional spatial region R	• Extract photos

TABLE I. PARAMETERS, QUERIES, AND ANALYSIS OF DIFFERENT REPORT TYPES

reports. The user inputs n spatial regions of interest R_i, $1 \leq i \leq n$, m entities or topics (identified by keywords W_j, $1 \leq j \leq m$), and an arbitrary time range $[T_s,T_e]$. For example, a user may input three Spanish cities, Madrid, Barcelona, and Cordoba, and two keywords representing the famous two soccer teams #RealMadrid and #FCBarecelona, during the week of their soccer game. A set of $n \times m$ queries are submitted to *Taghreed* query engine to retrieve the report data, each query takes a spatial region R_i, a keyword W_j, and the time range $[T_s,T_e]$. In our example, this give six queries, with all the combination of three cities and two hashtags. The retrieved data is displayed in an interactive web page that allows arbitrarily inclusion/exclusion of microblogs of certain spatial regions. Also, the report allows to navigate along a timeline, either for a single point of time or on a time range. The report may have optional components, like a heatmap for the microblogs, pie charts that show percentage analysis, and locating and displaying individual microblogs on a geographical map with full text and user information.

Categorical reports. For categorical reports, as the second row in Table I shows, the user inputs a spatial region of interest R, an arbitrary time range $[T_s,T_e]$, a categorical attribute A, and an optional set of keywords. For example, a user may want to analyze the tweet language usage in Saudi Arabia during the annual season of Muslim pilgrimage (September 22 to 27, 2015) for hashtags related to this event. The report composer divides the space into n small spatial regions of default fixed size. Then, a set of n queries are submitted to *Taghreed*, each query takes one of the small regions, the time range $[T_s,T_e]$, and the set of keywords. Each query retrieves individual microblogs that lie within the query parameters. The retrieved data is forwarded to the data analyzer module to count microblogs in different categories of attribute A, e.g., different languages. After counting is performed for all data, the counts are then aggregated at higher levels of spatial granularity to support zoom in/out analysis in the final report. Finally, the report composer puts all the aggregates on pie charts aligned with latitude/longitude coordinates of a geographical map and embed all of this in a web page. This forms an interactive web page that is sent as the final report to the user.

Image gallery reports. The third row in Table I describes image gallery reports. Users input m keywords W_j, $1 \leq j \leq m$, a time range $[T_s,T_e]$, and an optional spatial region R. For example, a user may ask for image gallery for Boston Marathon (hashtag #BostonMarathon) that is held on April 20, 2015 in Boston city. A single query with the input parameters is submitted to *Taghreed* to retrieve the report data. The retrieved microblogs are scanned to extract their images. Extracted images are organized and displayed in an interactive web page that allow users to navigate, enlarge, and share portions of the report on social media websites.

III. DEMONSTRATION SCENARIOS

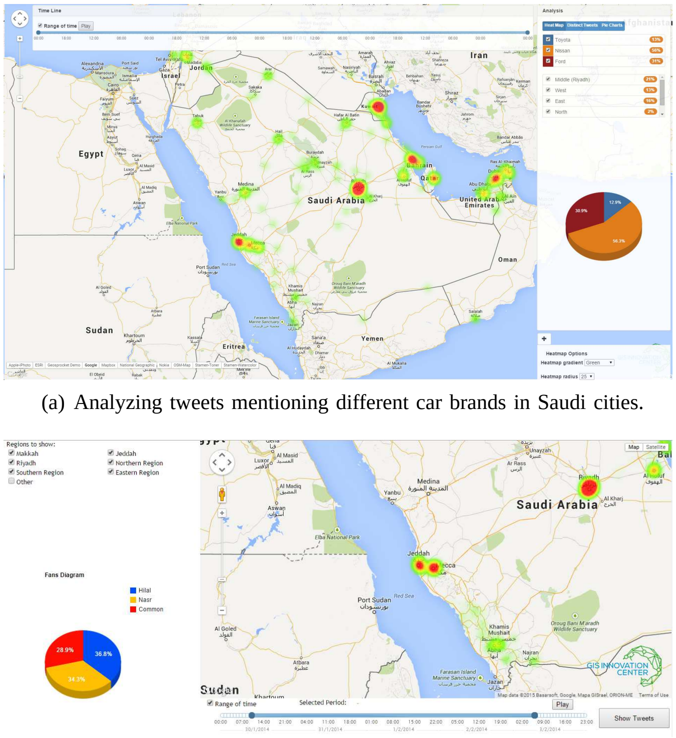
We demonstrate *Taqreer* reports using real data collected from Twitter, as a prime example of microblogs. Our dataset contains more than six billions of real tweets that are being collected from Twitter streaming APIs since October 2013. Our demo attendees would be able to generate/interact with the three types of *Taqreer* reports as follows.

A. Scenario 1: Comparative Reports

In this scenario, users can use the web interface of *Taqreer* to generate comparative reports. Users can compare the popularity of two or more terms of interest in different spatial regions within a certain time range. For example, a user may know about the popularity of the two Spanish soccer teams *Real Madrid C.F.* and *FC Barcelona* in different cities in Spain during the week of their soccer game. This can also include analysis related to presidential candidates, political parties, product trademarks, or events. Figure 2 gives two examples of comparative reports. Figure 2(a) shows tweets that mention different car brands in Saudi Arabia cities during the period January-March 2015. Figure 2(b) gives tweets of a soccer game in Saudi Arabia, where tweets are visualized based on local cities and show percentage of tweets that support each team. In both examples, users can navigate through a timeline, and limit the displayed tweets to a certain time point or range. Also, they can exclude/include any city to have finer granular analysis. The compared terms, spatial regions, and time range are arbitrary. This gives a generic and powerful tool for analyzing social media contents to get insights from the public discussions in different contexts and applications, e.g., elections, products reviews, or planned events.

B. Scenario 2: Categorical Reports

In this scenario, users can generate categorical reports to analyze the spatial distribution of categorical attribute in Twitter data. Prime examples of categorical attributes are the language attribute, that indicates the language used in



(a) Analyzing tweets mentioning different car brands in Saudi cities.

(b) Soccer game tweets

Fig. 2. Comparative Reports.

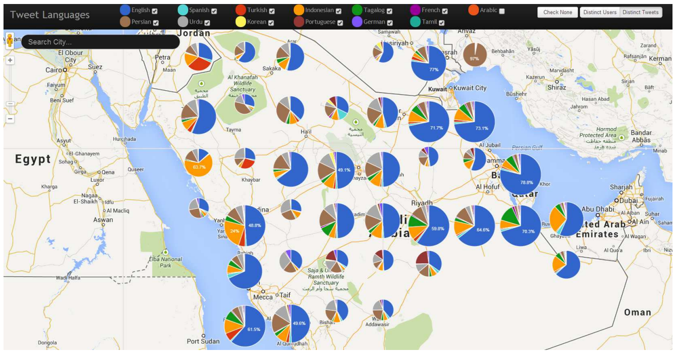


Fig. 3. Tweets Languages Spatial Analysis in Arab Gulf Countries.

each tweet, and the tweet source attribute, which determines from which operating system, device, or application the tweet is posted. Figure 3 gives an example of analyzing tweets languages in Arab Gulf countries. The figure gives a pie chart for each sub-region/city. Each pie chart shows the distribution of languages in its region. Zooming in/out gives a finer/coarser granular analysis for language distributions up to the street level. Users can arbitrarily include/exclude languages from the top bar to focus on a subset of the languages (Figure 3 excludes Arabic language for clarity). This language analysis gives deep insights on language diversity and minority groups in local communities. Similar analysis on *tweet source* attribute shows the distribution of different mobile devices brands and mobile applications usage in the country/city levels. This could be useful for local marketing purposes.

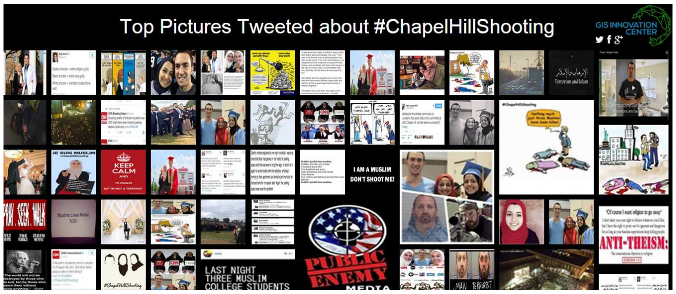

Top Pictures Tweeted about #ChapelHillShooting
GIS INNOVATION CENTER
KEEP CALM
I AM A MUSLIM DON'T SHOOT ME!
PUBLIC ENEMY
LAST NIGHT THREE MUSLIM COLLEGE STUDENTS
ANTI-THEISM

Fig. 4. Tweets Image Gallery for 2015 Chapel Hill Shooting.

C. Scenario 3: Image Gallery Reports

In this scenario, users can exploit the availability of many photos on the social media to generate image galleries that summarize certain topics, entities, or regions using their microblogs. An example of such reports is to extract and organize photos that are posted in response to a certain event, e.g., human crisis like Nepal earthquake, shooting attack like the one happened in Chapel Hill, elections, or sports game. Events and entities are defined by a set of keywords/hashtags. Figure 4 gives an image gallery for the event of 2015 Chapel Hill Shooting. The shown images are extracted for the hashtag #ChapelHillShooting for 11 days after the accident happened, from February 10 to February 20, 2015, and hashtag #ChapelHillShooting. Such gallery gives a quick understanding about how the virtual community think about certain accidents. This is shown by the popularity of certain photos and comments on them. Users can also navigate, enlarge, and share parts of the gallery on their social media accounts.

References

[1] Hamed Abdelhaq, Christian Sengstock, and Michael Gertz. EvenTweet: Online Localized Event Detection from Twitter. In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, 2013.

[2] After Boston Explosions, People Rush to Twitter for Breaking News. http://www.latimes.com/business/technology/la-fi-tn-after-boston-explosions-people-rush-to-twitter-for-breaking-news-20130415,0,3729783.story, 2013.

[3] Wei Feng, Jiawei Han, Jianyong Wang, Charu Aggarwal, and Jianbin Huang. STREAMCUBE: Hierarchical Spatio-temporal Hashtag Clustering for Event Exploration Over the Twitter Stream. In *Proceedings of the IEEE International Conference on Data Engineering, ICDE*, 2015.

[4] Amr Magdy, Louai Alarabi, Saif Al-Harthi, Mashaal Musleh, Thanaa Ghanem, Sohaib Ghani, and Mohamed Mokbel. Taghreed: A System for Querying, Analyzing, and Visualizing Geotagged Microblogs. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM GIS*, 2014.

[5] Amr Magdy, Mashaal Musleh, Saif Al-Harthi, Louai Alarabi, Kareem Tarek, Thanaa Ghanem, Sohaib Ghani, Hicham Elmongui, and Mohamed Mokbel. Taqreer: A System for Generating Spatio-temporal Analysis Reports on Microblogs. *IEEE Data Engineering Bulletin*, 2015.

[6] Adam Marcus, Michael S. Bernstein, Osama Badar, David R. Karger, Samuel Madden, and Robert C. Miller. Tweets as Data: Demonstration of TweeQL and TwitInfo. In *Proceedings of the ACM International Conference on Management of Data, SIGMOD*, 2011.

[7] New Enhanced Geo-targeting for Marketers. https://blog.twitter.com/2012/new-enhanced-geo-targeting-for-marketers.

[8] TweetTracker: track, analyze, and understand activity on Twitter. tweettracker.fulton.asu.edu/, 2014.