Demonstration of Taghreed: A System for Querying, Analyzing, and Visualizing Geotagged Microblogs

Amr Magdy[#], Louai Alarabi[#], Saif Al-Harthi[§], Mashaal Musleh[§],
Thanaa M. Ghanem[⋆], Sohaib Ghani[§], Saleh Basalamah[§], 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,louai,mokbel}@cs.umn.edu, {sharthi,mmusleh,sghani,sbasalamah}@gistic.org,
thanaa.ghanem@metrostate.edu

Abstract*—This paper demonstrates *Taghreed*; a full-fledged system for efficient and scalable querying, analyzing, and visualizing geotagged microblogs, such as tweets. *Taghreed* supports a wide variety of queries on *all* microblogs attributes. In addition, it is able to manage a large number (billions) of microblogs for relatively long periods, e.g., months. *Taghreed* consists of four main components: (1) indexer, (2) query engine, (3) recovery manager, and (4) visualizer. *Taghreed* indexer efficiently digests incoming microblogs with high arrival rates in light main-memory indexes. When the memory becomes full, the memory contents are flushed to disk indexes which are managing billions of microblogs efficiently. On memory failure, the recovery manager restores the memory contents from backup copies. *Taghreed* query engine consists of two modules: a query optimizer and a query processor. The query optimizer generates an optimized query plan to be executed by the query processor to provide low query responses. *Taghreed* visualizer features to its users a wide variety of spatio-temporal queries and presents the answers on a map-based user interface that allows an interactive exploration. *Taghreed* is the first system that addresses all these challenges collectively for geotagged microblogs data. The system is demonstrated based on real system implementation through different scenarios that show system functionality and internals.

I. Introduction

Online social media services become very popular in the last decade which has led to explosive growth in size of microblogs data, e.g., tweets, Facebook comments, and Foursquare check in's. Everyday, over a billion of active users generate billions of microblogs on Twitter and Facebook. As user-generated data, microblogs form a stream of rich data that carries different types of information including text, location, photos, users, and language information. This richness in data enables new queries and applications on microblogs that were not applicable earlier on traditional data streams, e.g., snapshot queries on keywords and spatial information, event detection, and news extraction. Such kinds of applications are so important that major IT companies are spending millions of dollars to enable them to their customers [3], [9].

Although the research community has addressed a large set of newly emerging queries and applications on microblogs, none of those provides a full fledged system that facilitates microblogs data management to support arbitrary queries on multiple attributes. Only TweeQL [8] 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 big data. On the other hand, microblogs can be considered a kind of big data as it comes with high volume and high velocity. High volume data could be managed with Big Data Management Systems (BDMS) like AsterixDB [2] while high velocity data could be managed by Data Streams Management Systems (DSMS). However, microblogs come with the two aspects, i.e., high volumes and velocity, while both BDMS and DSMS are not equipped to handle both aspects simultaneously. BDMS are not equipped to digest streaming data in real-time while DSMS are not equipped with indexes and query processing techniques that can handle the microblogs queries. Moreover, none of this work addresses interactive visualization with end users.

In this paper, we demonstrate *Taghreed* [7]; a full-fledged system for efficient and scalable querying, analyzing, and visualizing geotagged microblogs. On the contrary to all other works on microblogs, *Taghreed* system goals are to manage, query, analyze, and visualize microblogs data for relatively long periods that go up to several months. In addition, *Taghreed* provides data management techniques that are able to support interactive query responses on different microblogs attributes.

The main technical challenges in *Taghreed* comes from three main sources: (a) the large number (billions) of microblogs, (b) the continuous arrival of microblogs with high rates (thousands per second), and (c) the new types of queries on the rich microblogs data which cannot be supported by Data Stream Management Systems (DSMS) and are so popular on microblogs . Unlike the traditional streaming data, the combination of fast-arriving data and selective queries, e.g., keyword search, requires *Taghreed* to employ both real-time and disk-based efficient and scalable data indexing to be able to digest streaming data and support such queries interactively, i.e., the query answer is provided instantly. Thus, *Taghreed* main components are: (1) Efficient and scalable data indexing, (2) efficient interactive query processing, (3) low-overhead recovery management, and (4) effective data visualization.

We demonstrate *Taghreed* as an actual on-going system

*This work is supported by KACST GIS Technology Innovation Center at Umm Al-Qura University, under project GISTIC-13-06, and was done while the first, second, fifth, and last authors were visiting the center.

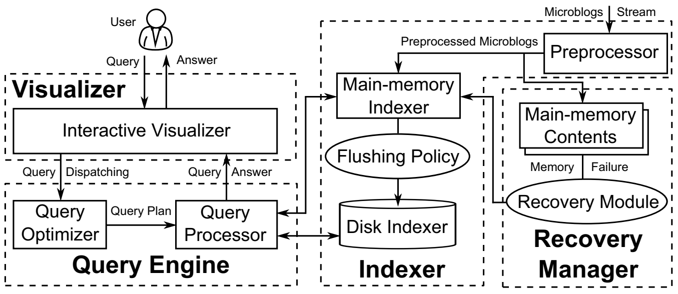


Fig. 1. Taghreed Architecture.

implementation that manages a large archive of real tweets and is also fed with live streaming tweets from Twitter streaming APIs.

II. System Overview

This section gives an overview of *Taghreed* design principles, system architecture, and supported queries.

A. Design Principles

Taghreed is designed based on two principles:

1) **Dominance of the temporal, spatial, and keyword attributes**: All microblogs queries have to be temporal, and then it mostly involves spatial and keyword dimensions. The real-time nature of microblogs data makes the temporal dimension a must to be included in all queries (see [5]). In addition, the richest attributes in microblogs are the spatial and keywords attributes which are involved in most of the queries and applications. Consequently, *Taghreed* promotes the three attributes, spatial, temporal, and keyword, as first-class citizens and supports native system indexes on them. In addition to their importance, indexing spatial, temporal, and keyword attributes provides effective pruning for the microblogs search space.
2) **Importance of queries on recent microblogs**: Due to the rich real-time content of microblogs, e.g., real-time updates on ongoing events [4], [6], important queries are posted on recent microblogs data, i.e., data of the last few seconds, minutes, or hours. This triggers all the literature work on handling queries on real-time microblogs (e.g., [1], [5]). Consequently, *Taghreed* is designed to support queries on real-time microblogs with all its subsequent requirements of main-memory indexing, recovery management, and flushing management.

B. System Architecture

Figure 1 gives *Taghreed* system architecture that consists of four main components, namely, indexer, query engine, recovery manager, and visualizer. We briefly introduce each of them below, more details can be revised in *Taghreed* full paper in [7].

Indexer. *Taghreed* indexer is the main component that is responsible for handling the microblogs data. First, the real-time microblogs are going through a preprocessor that extracts location and keyword information. Then, the microblogs are continuously digested in main-memory indexes in real time. When the memory becomes full, a subset of the main-memory microblogs are selected, through a flushing policy module, to be consolidated into scalable disk indexes that are able to manage billions of microblogs. *Taghreed* employs its indexes on spatial, temporal, and keyword attributes of microblogs which are considered the first-class attributes based on the design principles. The detailed indexes structure and organization can be revised in [7].

Query Engine. *Taghreed* query engine is mainly concerned with supporting a wide set of generic interactive queries such that the framework could be easily adapted to support different types of queries with almost no loss in system performance. To this end, *Taghreed* supports efficient retrieval of individual microblogs that lie within certain spatio-temporal range and satisfy certain keyword expressions. Then, other types of filtering, e.g., based on users, and aggregation, e.g., frequent keywords, are performed with efficient distributed data scanners. To accomplish low query responses, the query engine consists of two main modules: (1) *A query optimizer*, that generates an optimized query plan to hit the system indexes based on different cost models. (2) *A query processor*, that executes the query plan to perform efficient data retrieval from the system indexes. Then, it employs efficient distributed data scanners to answer extended queries that involve attributes other than spatial, temporal, and keywords.

Recovery Manager. With dense main-memory contents, *Taghreed* accounts for memory failures by incorporating a recovery manager component. The recovery manager employs a triple-redundancy model for backing up the main-memory contents. When the memory fails, the backup copies are used to restore the system status.

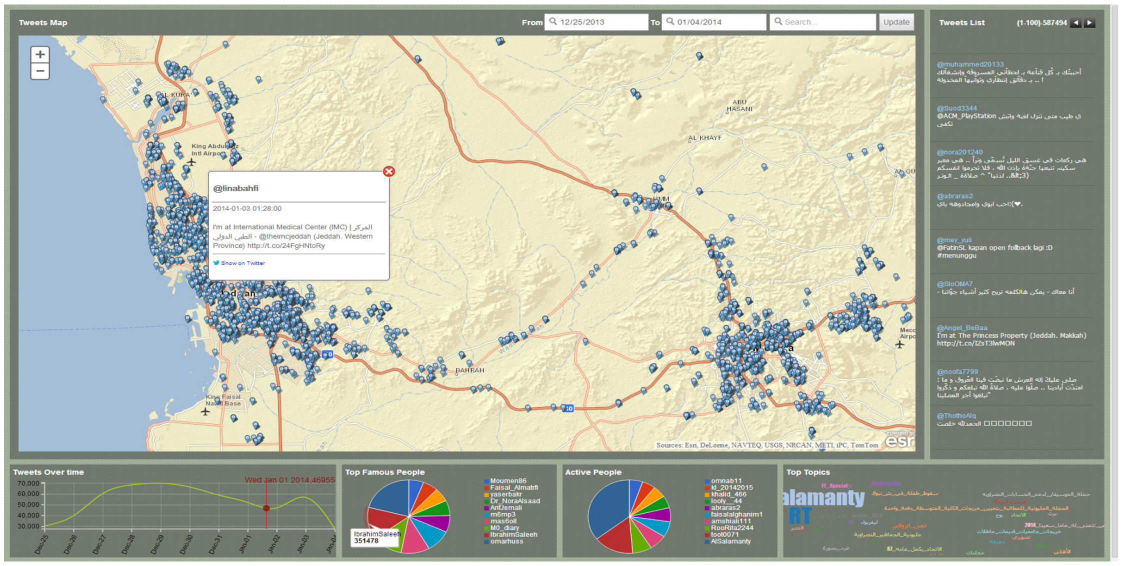
Visualizer. With all the technical details of managing and querying microblogs data in the system back-end, *Taghreed* provides end-to-end solution with an interactive front-end that takes users queries through web-based interfaces, dispatches them to the query engine, and receive the answers back to visualize. The visualization module comprises of an integrated interface that presents answers of a rich set of queries and also provides seeds for comprehensive applications on microblogs data. Section III presents different scenarios to demonstrate *Taghreed* front-end.

C. Supported Queries

Taghreed supports any query on microblogs that involves the spatial, temporal, and keyword attributes. However, the query latency depends on the required processing beyond the three main attributes. For any query, the temporal dimension is mandatory while spatial and keyword dimensions are optional. The queries are answered by pruning the search space through hitting the system indexes for the three main attributes. If the query involves other attributes, generic distributed data scanners are used to refine the answer. This enables *Taghreed* to support generic queries that satisfy a wide variety of applications as Section III shows.

III. Demonstration Scenarios

In this section, we demonstrate some applications that can be powered by *Taghreed* back-end. However, the potential



@linabahfi

2014-01-03 01:28:00

I'm at International Medical Center (IMC) | المركز الطبي الدولي - @theimcjeddah (Jeddah. Western Province) http://t.co/24FgHNtoRy

Show on Twitter

Tweets List (1-100)-587494

@muhammed20133
أحببتُك بـ كُل قناعة بـ لحظاتي المسروقة وإنشغالك .. بـ دقائق إنتظاري وثوانيها المخدولة !

@Suod3344
@ACM_PlayStation ي طيب متى تنزل لعبة واتش تكفى

@nora201240
هي ركعات في غسق الليل تُسمّى وتراً .. هي معبر سكينه تتبعها جنّةٌ بإذن الله ، فلا تحرموا انفسكم لذتها" ^ صلاةُ _ الوتر ..<3)

@abraras2
احب ابوي وامجادوهه باي:(❤.

@mey_yuli
@FatinSL kapan open follback lagi :D #menunggu

@SloOMA7
أنا معاك - يمكن هالكلمه تريح كثير أشياء جوّاتنا -

@Angel_BeBaa
I'm at The Princess Property (Jeddah. Makkah) http://t.co/IZsT3lwMON

@noofa7799
صلى عليكَ إله العرش ما نبضَت فينا العُروق و ما امتدّت أيادينا .. صلّوا عليه ، صلاةُ الله تبلغكم و ذكّروا تبلغوا أجر المصلينا"

@ThothoAlq
الحمدلله خلصت □□□□□□□

Top Famous People: Moumen86, Faisal_Almatrfi, yaserbakr, Dr_NoraAlsaad, ArifJemali, m6mp3, mas6oll, M0_diary, IbrahimSaleeh, omarhuss (IbrahimSaleeh 351478)

Active People: omnab11, kl_20142015, khalid_466, looly__44, abraras2, faisalalghanim1, amshiali111, RooRita2244, toot0071, AlSalamanty

Fig. 2. Multi-dimensional Queries on Microblogs.

applications are endless and bounded only with the richness of microblogs, on which *Taghreed* provides flexible, efficient, and scalable querying abilities. In addition, we demonstrate system measurements that show *Taghreed* scalability in indexing and querying microblogs. We use real data collected from Twitter, as a prime example of microblogs. In particular, our dataset contains more than two billion real geotagged tweets obtained from Twitter streaming APIs for the last few months and is increasing everyday. The real tweets are temporally partitioned, in a daily basis, into disjoint segments and indexed on keyword and spatial attributes as described before. We then power a set of web-based applications using *Taghreed* RESTful APIs. Our demo attendees would be able to interact with *Taghreed* through one or more of the following scenarios.

A. *Scenario 1: Interactive Multi-dimensional Querying*

In this scenario, we show the powerful ability of *Taghreed* to perform: (1) multi-dimensional queries, i.e., querying multiple attributes simultaneously, and (2) interactive querying, where the results are changing instantly based on user navigation on the geographical map interface. Thanks to the flexible query processing technique and the parallel scalable processing, *Taghreed* demo attendees would be able to provide a set of input parameters to get results for multiple queries very fast.

Figure 2 shows the main user interface for multi-dimensional queries. The user interface facilitate the user to provide: (1) a spatial range bounded by the visible map borders, (2) time range, (3) optional filtering keywords, and (4) optional filtering user through top bar text boxes. The application then submits a multi-dimensional query to *Taghreed* back-end to get the following from the input spatio-temporal boundaries: (1) individual tweets, (2) tweet frequency over time in a daily step, both filtered by the optional keywords and user, (3) top-30 frequent keywords, (4) top-10 active users, and (5) top-10 popular users. The numbers thirty and ten have been chosen for visualization space limitations. The application interface then displays the output as shown in Figure 2. Demo attendee navigation through the geographical map interface changes the query spatial extent and hence changes the results interactively for the above five outputs. Figure 2 shows the output for Makkah and Jeddah cities in Saudi Arabia.

B. *Scenario 2: Multi-dimensional Spatial Comparison*

Exploiting the power of multi-dimensional querying abilities in *Taghreed*, we show a scenario of comparing Twitter activity in two different arbitrary spatial regions, e.g., two cities or a city versus a country, as shown in Figure 3. The user interface facilitates submitting two multi-dimensional queries given two different spatial regions and common time range, filtering keywords, and user. The outcomes then, provides a side-to-side comparison for number of tweets over time, frequent keywords, active users, and popular users in the two regions. With arbitrary time window and keywords, we can explore user activities towards specific events in different areas, e.g., Twitter activity about US elections in Seattle and Chicago. Figure 3 gives a user activity comparison between two Saudi cities, Medina and Hail, during October and November 2013.

C. *Scenario 3: Temporal Analysis on Interest Changes*

In this scenario, using a top-k frequent keywords query, we show an application to analyze the change in topics of interest, in certain region, over time. To this end, the user inputs a spatial region R, a temporal window T, and a temporal step t, e.g., six hours. Then, a series of top-30 frequent keywords queries are submitted to *Taghreed* back-end, each covers R and a period of t time units within T. The answers of each two consecutive queries are compared to analyze the newly introduced keywords and the rank changes for existing keywords. Using these comparisons, we show the evolution in topical interest of users within the region. For each keyword, we show a profile of rank changes within the periods of time it exists. In addition, the most trending keywords are identified, using approximation of regression line slope, and reported to the user.

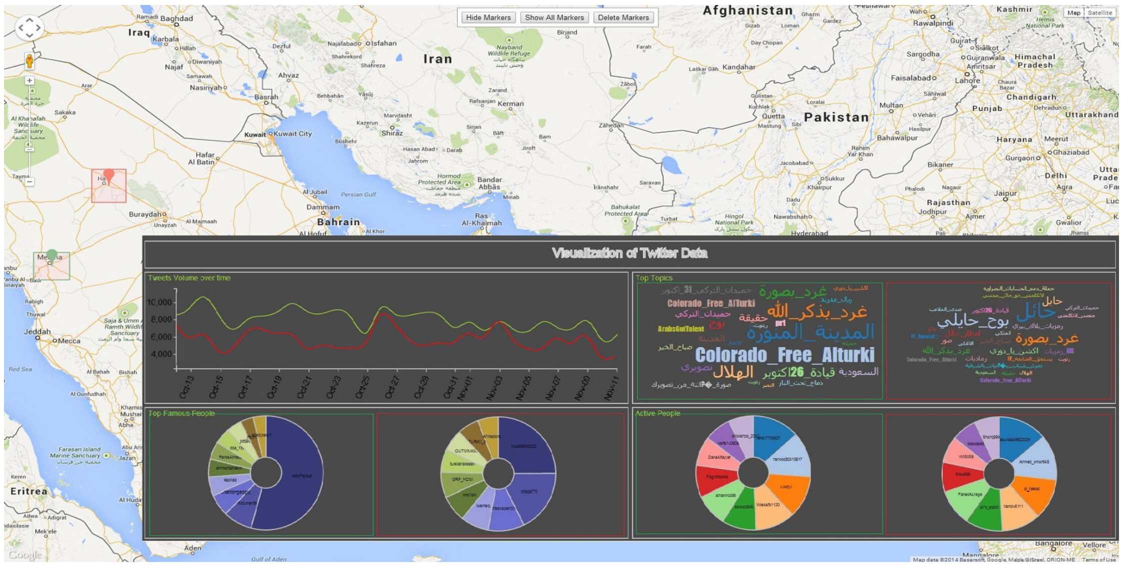


Fig. 3. Multi-dimensional Spatial Comparison. (Green color represents Medina while red color represents Hail)

D. Scenario 4: Spatio-temporal Interactive Heatmaps

In this scenario, we show an application that is built on top of *Taghreed* while its functionality is not built-in in the system front-end. Using the non-aggregate query that retrieves individual tweets that are related to certain keywords, time, and place, we develop an application that provides spatio-temporal visual analysis for tweets that are related to certain topic, event, etc. This visual analysis includes showing tweet distribution in the form of heatmap through the time and space. The heatmap can be played versus time in form of animation, as an interactive video for tweets distribution. In addition, individual tweets are still available for selective investigation at any point of time. Figure 4 shows the heatmap of Oscars 2014 tweets at a certain time instant. Such tweets can be easily obtained from *Taghreed* back-end by filtering the results based on Oscars hashtags. Then, the tweets are fed to our visual spatio-temporal analysis tool depicted in the figure, that facilitates easy visual analysis in the spatio-temporal space.

E. Scenario 5: System Internals

With the scalable distributed design of *Taghreed* system, monitoring the system internals would be of interests for demo attendees as well as system developers. Monitoring system internals could help in identifying performance bottlenecks and even provide rooms for improvement in the system design itself. In our demo, we develop a simple interactive interface that shows two aspects of system internals: (1) Digesting the new microblogs in the system and loading them to the segmented index. (2) Handling the incoming queries and the performance of our query processor in terms of distributed node utilization, throughput, intermediate storage consumption, etc. Such measurements show *Taghreed* efficiency and scalability to the demo attendees.

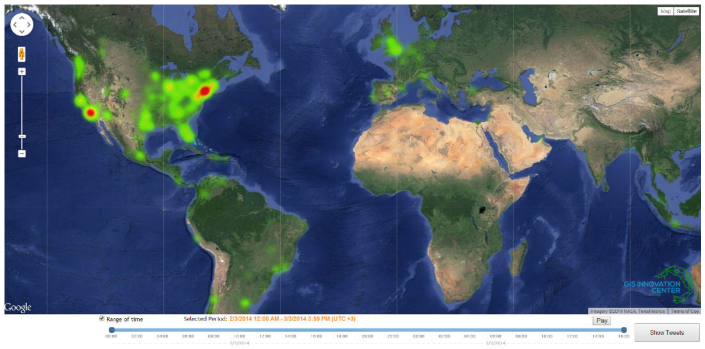


Fig. 4. Oscars 2014 Tweets Heatmap.

REFERENCES

[1] H. Abdelhaq, C. Sengstock, and M. Gertz. EvenTweet: Online Localized Event Detection from Twitter. In *VLDB*, 2013.

[2] S. Alsubaiee, Y. Altowim, H. Altwaijry, A. Behm, V. R. Borkar, Y. Bu, M. J. Carey, R. Grover, Z. Heilbron, Y.-S. Kim, C. Li, N. Onose, P. Pirzadeh, R. Vernica, and J. Wen. ASTERIX: An Open Source System for "Big Data" Management and Analysis. *PVLDB*, 5(12):1898–1901, 2012.

[3] Apple buys social media analytics firm Topsy Labs. http://www.bbc.co.uk/news/business-25195534, 2013.

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

[5] M. Busch, K. Gade, B. Larson, P. Lok, S. Luckenbill, and J. Lin. Earlybird: Real-Time Search at Twitter. In *ICDE*, 2012.

[6] Sina Weibo, Chinas Twitter, comes to rescue amid flooding in Beijing. 2012. http://thenextweb.com/asia/2012/07/23/sina-weibo-chinas-twitter-comes-to-rescue-amid-flooding-in-beijing/.

[7] A. Magdy, L. Alarabi, S. Al-Harthi, M. Musleh, T. Ghanem, S. Ghani, and M. Mokbel. Taghreed: A System for Querying, Analyzing, and Visualizing Geotagged Microblogs. In *SIGSPATIAL*, 2014.

[8] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller. Tweets as Data: Demonstration of TweeQL and TwitInfo. In *SIGMOD*, 2011.

[9] New features on Twitter for Windows Phone 3.0, 2013. https://blog.twitter.com/2013/new-features-on-twitter-for-windows-phone-30.