A Demonstration of QARTA: An ML-based System for Accurate Map Services

Sofiane Abbar[1], Rade Stanojevic[1], Mashaal Musleh[2], Mohamed ElShrif[1], Mohamed Mokbel[1,2]
[1]Qatar Computing Research institute, Hamad Bin Khalifa University, Doha, Qatar
[2]Department of Computer Science and Engineering, University of Minnesota, USA
(sabbar, rstanojevic, melshrif)@hbku.edu.qa (musle005, mokbel)@umn.edu

ABSTRACT

This demo presents QARTA; an open-source full-fledged system for highly accurate and scalable map services. QARTA employs machine learning techniques to: (a) construct its own highly accurate map in terms of both map topology and edge weights, and (b) calibrate its query answers based on contextual information, including transportation modality, underlying algorithm, and time of day/week. The demo is based on actual deployment of QARTA in all Taxis in the State of Qatar and in the third-largest food delivery company in the country, and receiving hundreds of thousands of daily API calls with a real-time response time. Audience will be able to interact with the demo through various scenarios that show QARTA map and query accuracy as well as internals of QARTA.

PVLDB Reference Format:
Sofiane Abbar, Rade Stanojevic, Mashaal Musleh, Mohamed ElShrif, Mohamed Mokbel. A Demonstration of QARTA: An ML-based System for Accurate Map Services. PVLDB, 14(12): 2723-2726, 2021.
doi:10.14778/3476311.3476329

1 INTRODUCTION

Maps services, e.g., routing, shortest path finding and travel time estimation have been ubiquitous in widely used applications. Examples including ride-sharing (e.g., Uber and Lyft), food delivery (e.g., Uber Eats and Doordash), and last-mile delivery (e.g., Amazon, UPS, and FedEx). This results in several popular commercial (e.g., Google Maps, Bing Maps, Waze) and open-source (e.g., OSRM [6]) map services as an infrastructure support for such important applications. Unfortunately, there is an apparent accuracy gap between commercial and open-source map services, making it very hard to rely on open-source map services for important applications. The reason is that map services (e.g., shortest path queries) rely on three important components: (1) efficient shortest path algorithm, (2) having access to accurate map topology, and (3) having access to accurate traffic information. While open-source map services can deploy the state-of-the-art shortest path algorithms (e.g., [7, 12, 13]), and obtain accurate map topology from publicly available governmental data, it falls short in terms of accurate traffic data. Such data is usually a proprietary of commercial map services, and key to their accuracy. So, no matter how efficient an algorithm is, if the underlying map has inaccurate traffic information, the result will be inaccurate, which is the case with open-source map services.

This demos presents QARTA [5]; an open-source full-fledged system that employs machine learning techniques to provide highly accurate map services. QARTA fills in the gap of current open-source map services by its ability to build its own accurate map, and hence provide highly accurate map services. QARTA is currently responding to hundreds of thousands of daily API calls coming from its actual deployment in: (a) all Taxis in the State of Qatar (around 4K vehicles), and (b) a food delivery company (around 3K motorbikes). In both cases, QARTA has successfully replaced commercial map services that were in use for a long time. QARTA was triggered by a real need from the Taxi and delivery companies that not only the commercial service is expensive, but more importantly, it is outdated in both the topology and traffic metadata. The main reason for having such stale maps is that the underlying map is rapidly changing due to major country-wide constructions preparing for FIFA 2022 world cup [1].

QARTA builds a machine learning pipeline [8] to: (1) construct its own highly accurate map in terms of both map topology and edge weights, and (2) to calibrate its query answers based on contextual information, including transportation modality, underlying algorithm, and time of day/week. In terms of map construction, QARTA utilizes machine learning techniques to: (a) learn accurate edge weights, where each road network edge is annotated by a set of 168 edge weights corresponding to hours of the week. An edge weight is an indication of the time needed to travel through the edge at a certain hour of the week. Inaccurate edge weights will result in inaccurate route recommendation. (b) learn when to trust incoming GPS traces and uses it to update the map topology and when to trust the underlying topology and match GPS traces over it, and (c) learn road metadata, including maximum speed, number of lanes, road directions, and road type.

In terms of query calibration, QARTA continuously monitors the quality of its traffic model to understand the error margins of all deployed algorithms, and then use it to calibrate the query results. In particular, we build a model that maps each trip feature vector to its error margin. We then use this model to calibrate the query answers for shortest path, k-NN, and range queries. QARTA maintains different models for different transportation modalities (vehicles and motorbikes) and for different underlying algorithms. Experimental evaluation of QARTA, based on real data and actual deployment in all Taxis in the State of Qatar and in a food delivery company, shows that QARTA has significantly higher accuracy than currently available open-source solutions, and has a comparable to slightly better performance than commercial map services.

Proceedings of the VLDB Endowment, Vol. 14, No. 12 ISSN 2150-8097.
doi:10.14778/3476311.3476329

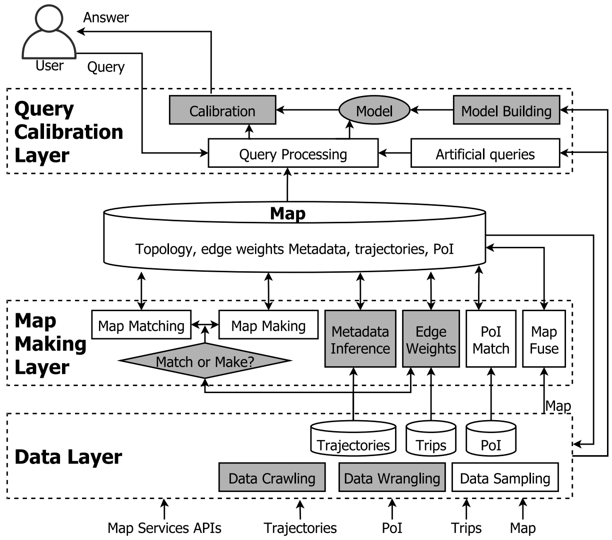

Figure 1: QARTA Architecture

This demo shows QARTA in action. Conference audience will be able to see QARTA receiving live traffic requests and respond to them in real time. Audience will be able to see QARTA map for the city of Doha, Qatar, annotated with edge weights, computed by QARTA machine learning techniques for anytime in the future. To have the demo more interactive, audience will be able to do a live performance comparison between QARTA and commercial and open-source map services. To understand the internals of QARTA, audience will be able to enable/disable various components of QARTA, which will show the impact of each component.

2 QARTA ARCHITECTURE

Figure 1 depicts the map-centric QARTA system architecture, composed of three layers, *Data layer*, *Map Making layer*, and *Query Calibration layer*. The main input to QARTA is various sorts of datasets that goes to the *Data Layer*, which processes and stores the input. The *Map Making* layer uses this data to build its own rich map including map topology, edge weights, edge metadata, and points of interest (PoIs). The *Query Calibration* layer receives input queries (e.g., shortest path, k-nearest-neighbor and range queries) from the user, answer them using the map built by the *Map Making* layer, and calibrate the answer using a built-in model based on input ground truth data obtained from the *Data Layer.*

QARTA completely separates map construction from query processing, where updating or constructing the map is a background process that does not affect the query response time. Map construction digests input live GPS traces through a light process that continuously and accurately updates the map. Meanwhile, query calibration relies on models built offline. Hence, it does not put any overhead on the underlying query processing, making QARTA response time as real time and scalable as the underlying query processor. A resilient feature of QARTA is that it does not come up with new query processing or indexing techniques. Instead, QARTA significantly boosts the accuracy of existing techniques by feeding them with an accurate map, and calibrating their answers. In particular, the gray modules in each layer of Figure 1 are those that QARTA has identified as critical to significantly boost the accuracy of map services. For other modules, QARTA mainly uses off-the-shelf solutions. This makes QARTA a favored choice for researchers, developers, and practitioners to use as a vehicle for their new efficient query processing and indexing techniques, without worrying about the result accuracy.

3 DATA LAYER

The *data layer* is responsible for data digestion and collection efforts while laying out the storage infrastructure for all data. Some of QARTA input data are straightforward in terms of richness and cleanness, e.g., official maps obtained from governmental websites. For such data, we just store it using off-the-shelf spatial data warehousing and indexing. Meanwhile, a major part of QARTA input is noisy for various reasons, e.g., inaccurate GPS readings, missing data, or misinterpreted data. QARTA forwards such data to its *Data Wrangling* module that produces a cleaned version. For parts of the map, where there is data shortage, we employ a *Data Crawling* module to fill in the gaps.

Data Wrangling. QARTA employs its own rule-based data wrangling module that addresses specific trajectory problems that came out from its actual deployment. Examples of such problems and rules include: (1) discovering trajectories with non-traffic related stops, and eliminating the parts of such stops, (2) discovering and eliminating outlier trajectory points that could be a result of erroneous GPS readings, (3) discovering cases where there is missing GPS reading, and split trajectories accordingly.

Data Crawling. QARTA crawls several governmental and open-access sites to enrich its repository of maps and PoIs, which is a straightforward development process. For trajectories and trip information, QARTA continuously collects them through its real deployment. However, in cases where QARTA is newly deployed or there is a shortage of trips, we may acquire traffic information either from UBER-movement-like platforms [11] or explicitly from commercial services through API calls. For commercial services, it is crucial to optimize the number of API calls to accommodate low-budget enterprises. Hence, QARTA employs several rules to maximize the benefit and coverage of a certain number of API calls.

4 MAP MAKING LAYER

The *Map Making* layer is responsible for building QARTA map, with its rich set of information including map topology (road network), edge weights, road metadata, and points of interest. The layer already deploys off-the-shelf algorithms for map fusion [9], building the map topology, and map matching. The layer is also equipped with a *map fusion* module that merges map updates to an existing map [9]. Since there is already a plethora of techniques for building the map topology (e.g., [3]) and map matching (e.g., [2, 4]), we just employ state-of-the-art of these techniques. Then, QARTA focuses on the following three main modules:

Match or Make. Given a road network R and GPS points P, this module decides on the part(s) of the map where R would be considered more accurate than P and vice versa. For parts where R is more accurate, we call *map matching* to match P on R. For other parts, we call *map making* to update R based on P. To do so, we employ machine learning supervised learning to learn the features

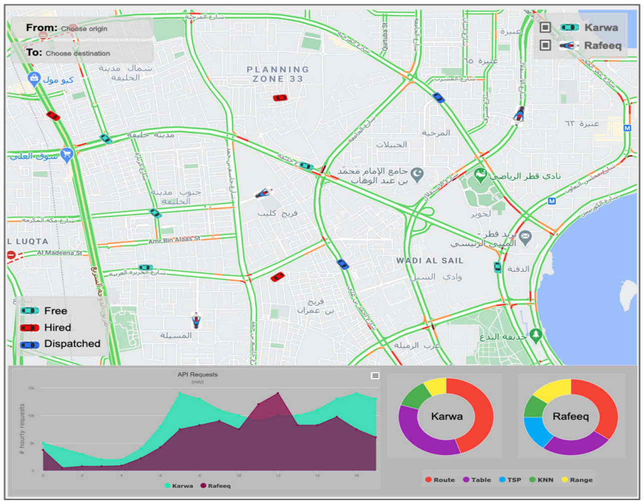

Figure 2: Demo Scenario 1: QARTA Dashboard

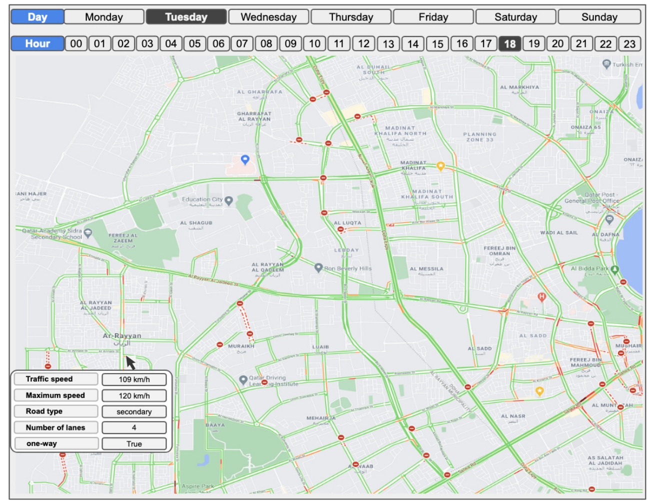

Figure 3: Demo Scenario 2: QARTA Map

of accurate maps and points, then use it to classify parts of the map and points that we are in doubt about their correctness.

Edge Weight Inference. Unlike the case of edge length and the maximum speed, that are static road edges attributes, and can be publicly available, accurate edge weights are usually considered as proprietary information. This is the most crucial information needed to give an accurate answer for shortest path queries, travel estimation, range and k-nearet-neighbor queries. Given a road network topology and a large number of QARTA historical trips, this module finds 168 weights for each edge, as one value for each hour of the week. We do so by employing machine learning techniques, namely, the Constrained Ridge Regression analysis to find non-zero edge weights that would minimize the error offset of each trip from its actual travel time reported by Taxi trips.

Metadata Inference. Annotations (a.k.a metadata) constitute an important component of maps. Several real-life applications, e.g., traffic modeling, driver behavior analysis, road safety, and telematics heavily rely on map metadata such as the number of lanes, maximum speed, directions, and road types (e.g., highway, service road, bridge). Unfortunately, these annotations are very poor in most cities around the world, especially in openly available maps. QARTA frames metadata inference as a supervised learning problem, in which the task is to first find the *best* models that would map road features to each metadata (e.g., road category), then, use these models to predict the metadata values for each road segment.

5 QUERY CALIBRATION LAYER

The query calibration layer is responsible for responding to query APIs from QARTA users, with a special focus on Estimated Time of Arrival (ETA), which is crucial for several applications [10]. For example, trip fares are decided based on their ETAs, scheduling multiple trips and dispatching drivers to customers is based on their ETAs from each other, finding PoIs that are k-NN or within a range also depends on their ETAs from the reference point. The *query* part of this layer includes off-the-shelf algorithms for shortest path, range, and k-nearest neighbor queries. Meanwhile, the *calibration* part sets QARTA apart from other map services, where it aims to understand the ETA error margins of all deployed algorithms, and then use it to calibrate the results. In particular, we use machine learning to build a model that maps the trip feature vector to its ETA error. We then use this model to calibrate the query answers for shortest path, k-NN, and range queries. The *model building* produces a model $\mathcal{M}$, which is used by the *query operation* to adjust its result. Note that we maintain several models based on transportation modality and underlying algorithm. For example, we have a model for a fleet of vehicles, used by our Taxi company partner, and another model for a fleet of motorbikes, used by the food delivery company we are working with. It is important to have this distinction, as they encounter different error distribution.

6 DEMO SCENARIO

This section shows several demo scenarios where the conference audience can interact with QARTA to understand its operation from points of view of both: (a) a user who just needs to query QARTA, and (b) a developer who needs to understand the internals of QARTA system. The demo will be based on QARTA server that is actually in deployment, supporting all Taxis in Qatar (~4K vehicles) and a food delivery company (~3K drivers). As of March 2021, our QARTA server receives 235K API calls and 977K new GPS tracking points per day for various business operations including routing, ETA, driver dispatching, fare estimation, and tracking.

6.1 Demo Scenario 1: QARTA Dashboard

Figure 2 depicts the main dashboard of QARTA that runs 24/7, where conference audience will be able to see the rate of incoming data and queries received by QARTA from its two main customers (Karwa vehicles and Rafeeq motorbikes). The dashboard shows the map of Doha, Qatar, where taxis and motorbikes GPS points are continuously received and plotted on the screen. The bottom panel shows the rate of API queries received by QARTA from its two partners. Color-coded donuts shapes on the corner show the ratio of each query kind per customer.

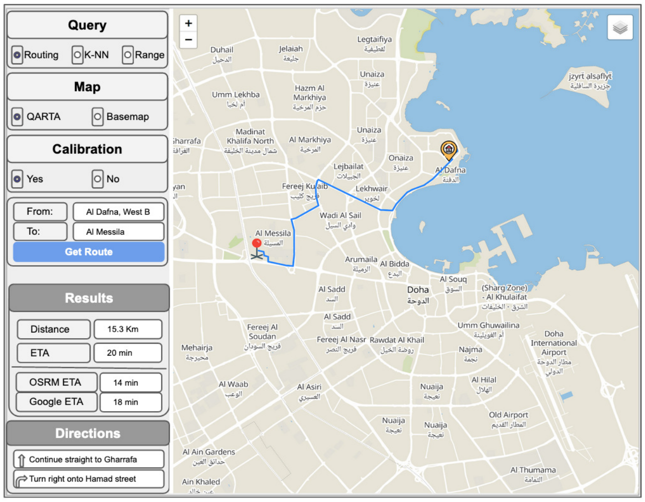

Figure 4: Demo Scenario 3: Query Accuracy

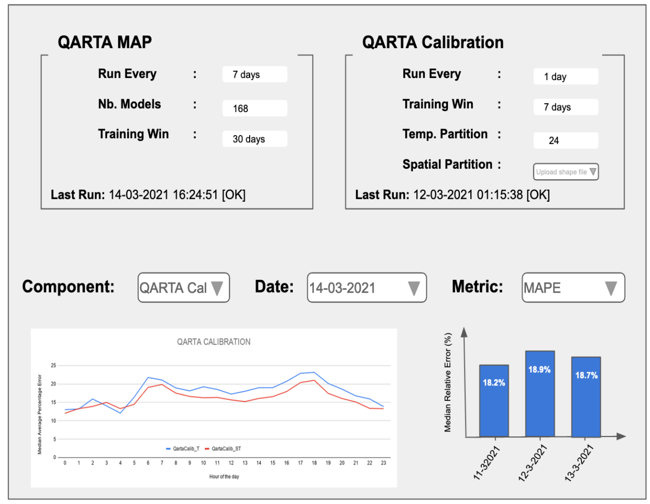

Figure 5: Demo Scenario 4: QARTA Internals

6.2 Demo Scenario 2: QARTA Map

In this scenario, audience will be able to navigate through QARTA map, checking its rich information in three aspects: (1) *Map topology*. Audience will be able to check the difference between QARTA map and other commercial maps for certain parts of Doha, where the map was recently updated, (2) *Edge weights and metadata*. Audience will be able see the predicted traffic map for Doha for any hour in the future. Figure 3 gives the query interface for this functionality, where audience can specify a certain hour in the future using controls on the top of the screen. A color-coded traffic map is shown in which red segments indicate predicted high traffic. Hovering over each road segment will show the edge weight (traffic speed) and metadata (maximum speed, road type, number of lanes, and road directions) computed by QARTA. Users can also see a fast forward map where each hour takes only half a second frame showing the traffic changes across the day.

6.3 Demo Scenario 3: Query Accuracy

Figure 4 gives the query interface of QARTA, where conference audience can specify the query type as either routing, k-NN, or range queries, and (a) compare the answer with Google Maps as an example of a commercial map service and OSRM [6] as an example of an open-source map service, and (b) disable/enable parts of QARTA and see their effect on query accuracy. For example, users will be able to disable QARTA map, and just run the calibration part on top of an open-source map that is not traffic-aware. Also, users can disable the calibration part, and just show QARTA answer with high quality map. this will help the audience to understand more of QARTA internals and the effect of each component.

For routing queries, users would click on the source and destination and specify a specific time. For k-NN and range queries, users would need to click on a reference point and input a number k or a range distance as a query parameter. It will be interesting to see that the answer is slightly different for each system, mainly as the rank of the PoIs within the top-k list. We will show the reasoning of this difference and how sensitive it is to map accuracy.

6.4 Demo Scenario 4: QARTA Internals

This scenario shows some of QARTA internals, where system administrators can set up various QARTA parameters and monitor the system accuracy. Figure 5 depicts the dashboard for such scenario. A system administrator can set up how frequently QARTA map should be updated (default 7 days), and how many models (edge weights) are built for that period (default 168), and how many days are included in the training data (default 30 days). The administrator can also set the query calibration parameters: frequency of training the model, window of training data, and granularity of temporal/spatial partitioning. The administrator can also monitor the running accuracy of the queries and calibration models.

REFERENCES

[1] S. Abbar, R. Stanojevic, S. Mustafa, and M. Mokbel. The Road to 2022 FIFA World Cup: Traffic Routing in the Ever-Changing City of Doha. *CACM*, 64(4):67–68, 2021.
[2] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On Map-Matching Vehicle Tracking Data. In *PVLDB*, pages 853–864, Aug. 2005.
[3] L. Cao and J. Krumm. From GPS traces to a routable road map. In *SIGSPATIAL*, pages 3–12, Jan. 2009.
[4] S. He, F. Bastani, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, and S. Madden. RoadRunner: Improving the Precision of Road Network Inference From GPS Trajectories. In *SIGSPATIAL*, pages 3–12, Nov. 2018.
[5] M. Musleh, S. Abbar, R. Stanojevic, and M. Mokbel. QARTA: An ML-based System for Accurate Map Services. *PVLDB*, 14(11):2273–2282, 2021.
[6] Open Source Routing Machine (OSRM). http://project-osrm.org/.
[7] D. Ouyang, L. Yuan, L. Qin, L. Chang, Y. Zhang, and X. Lin. Efficient Shortest Path Index Maintenance on Dynamic Road Networks with Theoretical Guarantees. *PVLDB*, 13(5):602–615, 2020.
[8] I. Sabek and M. Mokbel. Machine Learning Meets Big Spatial Data (Tutorial). In *ICDE*, pages 1782–1785, Apr. 2020.
[9] R. Stanojevic, S. Abbar, S. Thirumuruganathan, G. D. F. Morales, S. Chawla, F. Filali, and A. Aleimat. Road Network Fusion for Incremental Map Updates. In *LBS*, pages 91–109, Jan. 2018.
[10] L. Tran, M. Mun, M. Lim, J. Yamato, N. Huh, and C. Shahabi. DeepTRANS: A Deep Learning System for Public Bus Travel Time Estimation using Traffic Forecasting. *PVLDB*, 13(12):2957–2960, 2020.
[11] Uber Movement Data. https://movement.uber.com/.
[12] Y. Wang, G. Li, and N. Tang. Querying Shortest Paths on Time Dependent Road Networks. *PVLDB*, 12(11):1249–1261, 2019.
[13] L. Wu, X. Xiao, D. Deng, G. Cong, A. D. Zhu, and S. Zhou. Shortest Path and Distance Queries on Road Networks: An Experimental Evaluation. *PVLDB*, 5(5):406–417, 2012.