

QT2S: A System for Monitoring Road Traffic via Fine Grounding of Tweets

Noora Al Emadi[‡], Sofiane Abbar[‡], Javier Borge-Holthoefer[◊], Francisco Guzmán[‡], Fabrizio Sebastiani[♡]

[‡]Qatar Computing Research Institute, Hamad bin Khalifa University, Qatar

[◊] Internet Interdisciplinary Institute, Universitat Oberta de Catalunya, Spain

[♡]Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Italy

E-mail: nalemadi@hbku.edu.qa, sabbar@hbku.edu.qa, jborghe@uoc.edu, guzmanhe@gmail.com, fabrizio.sebastiani@isti.cnr.it

Abstract

Social media platforms provide continuous access to user generated content that enables real-time monitoring of user behavior and of events. The geographical dimension of such user behavior and events has recently caught a lot of attention in several domains: mobility, humanitarian, or infrastructural. While resolving the location of a user can be straightforward, depending on the affordances of their device and/or of the application they are using, in most cases, locating a user demands a larger effort, such as exploiting textual features. On Twitter for instance, only 2% of all tweets are geo-referenced. In this paper, we present a system for zoomed-in grounding (below city level) for short messages (e.g., tweets). The system combines different natural language processing and machine learning techniques to increase the number of geo-grounded tweets, which is essential to many applications such as disaster response and real-time traffic monitoring.

Introduction

Social media platforms provide continuous access to information generated by users around the world. This enables real-time monitoring of user behavior (Abbar, Mejova, and Weber 2015), events (Weng and Lee 2011), and urban dynamics (Abbar, Zanouda, and Borge-Holthoefer 2016). The geographical dimension of such user behavior and events has recently caught a lot of attention, and for different reasons: commercial, humanitarian, or infrastructural. In some cases, resolving the location of a user is straightforward, depending on the affordances of their device and/or of the application they are using (e.g. GPS-enabled activity). However, in most cases, locating a user demands a larger effort, such as exploiting textual content of web entries (Rae et al. 2012) or social media messages to estimate location (Han, Cook, and Baldwin 2014).

Researchers have dealt with this problem with varying success. State-of-the-art techniques for dealing with free text (Rae et al. 2012) reach above 90% precision at coarse-grained *grounding* of a user's activity (e.g. country, city levels), which suffices for many applications (say, to place ads online). However, *zoomed-in* grounding (below city level) is essential if the purpose is to profile traffic conditions or to assess damage resulting from a natural disaster.

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

In this paper we present a system which instantiates this fine grounding. We do so by presenting a dedicated implementation, QT2S (Qatar Traffic Social Sensing), which illustrates its usefulness to locate traffic-related events (e.g., traffic jams) in Doha, the capital of Qatar. While some location based social networks such as Instagram present higher fractions of geo-coded posts (30%) compared to Twitter (2%) (Mejova, Abbar, and Haddadi 2016), it is important to notice that these services do not open their data to third parties. The idea then is to exploit the user generated content widely available in open social media platforms (Twitter) in order to enrich with more semantics traditional traffic monitoring systems based on physical counting sensors, which often lack the semantics of what is causing congestion to happen (accidents, traffic deviations, malfunctioning signals). Our work first aim is to fill the gap of below-city-level geo-location, and we do so in a challenging setting: First, the population of Qatar is linguistically heterogeneous and is composed by over 85% of migrants (many of whom do not speak Arabic); second, there is no standard for transliteration from Arabic into English, which adds ambiguity (esp. when referring to local named entities) to the already challenging informal content of social media; third, the rapid growth of the city and the difficulties that migrants experience in dealing with Arabic names of public spaces, have led to a landmark-driven navigation, which results in many places being referred to by multiple expressions.

System Architecture

In this section we present an overview of our system, first describing its architecture and then proceeding with the description of each module.

The general view of the system is depicted in Figure 1. The process starts by listening to the Twitter feed in order to catch relevant social posts using a handcrafted list of 70 keywords related to the context of the system (e.g., road traffic). Returned posts are then pushed through a three-steps pipeline in which: (i) we double-check the relevance of the post using a binary classifier (the *Content Filter*), (ii) we extract location names mentioned in the posts, if any, and (iii) we geo-locate the identified locations to their accurate placement on the map. This process allows to filter undesirable posts, and augment the relevant ones with precise geo-location coordinates which are finally exposed for consump-

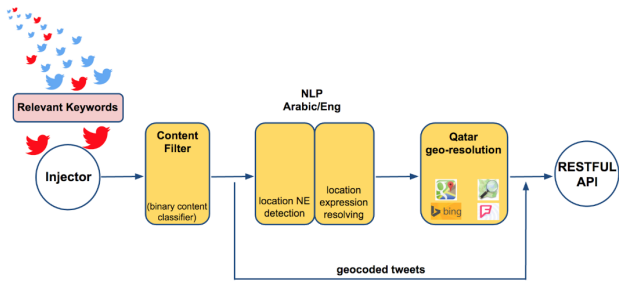


Figure 1: The data processing pipeline.

tion via a RESTful API.

Content filter

The *Content Filter* takes the stream of tweets as input and preserves only traffic-related tweets. It consists of two different modules: a keyword-based filter, and a binary classifier that further refines the selection. The keyword-based filter selects tweets that match one or more keywords that refer to traffic-related events (e.g., *accident*, *traffic*, *jam*, etc.). However keyword-based filtering is a noisy criterion to select events related to traffic. Therefore, we further train a binary classifier to filter out tweets that are not real-time reports about traffic (e.g., tweets that mention *jam* as a type of food, tweets that complain about the *traffic* in general, etc.). To build the classifier, we use the Stanford maximum entropy (Max-Ent) classifier¹, using with n-gram features. For its training, we manually annotated 1200 tweets. Each tweet is tokenized using a Twitter-specific tokenizer (O’Connor, Krieger, and Ahn 2010). Next, a rule-based local named entity simplifier is used to substitute mentions of local entities by their corresponding meta-categories (for example, it substitutes *@moi_qatar* – which stands for the Qatar Ministry of Interior – with the tag *government_entity*).

In Table 1 we present a performance comparison between the keyword-based system and the system in which the results of the former are fed into the Max-Ent classifier. Results are calculated using 10-fold cross-validation on the training set. Initially, the keyword-based filter incorrectly classifies as traffic-related two-thirds of the tweets. Using the Max-Ent classifier we double the precision for the positive tag, thus avoiding to push many irrelevant tweets downstream. The overall accuracy increases from 0.337 to 0.772.

	Prec	Rec	F_1	Acc
Keyword-based	0.337	1.000	0.505	0.337
Maximum-entropy	0.668	0.646	0.657	0.772

Table 1: Positive label precision, recall, F_1 and overall accuracy for the two types of content filter.

¹<http://nlp.stanford.edu/software/classifier.html>

Location Expression Recognition

Qatar is a region where locations are rarely indicated by their street addresses. One of the main challenges in such a region is that locations and directions are thus given in relation to landmarks or distinctive points of interest. Therefore, to geocode an event (e.g., a traffic jam) described in a specific tweet, we first need to identify all the possible locations involved. The *Location Expression Extractor* is a module that identifies (or extracts) location expressions, i.e., natural language expressions that denote locations (e.g., *at the Slope roundabout*, *on Khalifa St*, etc.). Location expressions can be complex linguistic objects that involve several named entities of type Location, e.g., *on location_A between location_B and location_C*. A key component of the Location Expression Extractor is the *Location Named Entity Recognizer*, which identifies named entities of type Location (e.g., *the Sports roundabout*) or Landmark (e.g., *the Museum of Islamic Art*).

For our purposes, a location is any proper name in the Doha street system (e.g., *Corniche*, *TV Roundabout*, *Khalifa Street*, *Khalifa*); landmarks are different from locations, since locations are only functional to the Doha street system, while landmarks have a different purpose (e.g., the MIA is primarily a museum, although its whereabouts may be used as a proxy of a specific location in the Doha street system, i.e., the portion of the Corniche – the main Doha seafront road – that is right in front of it). A key difference, for our purposes, is that a landmark can be mentioned in a tweet for a purpose other than indicating a place, as in *The Traffic Department ruled that ...* or *the MIA hosts a new exhibition*, while the same cannot be said of entities such as *the Majlis Al Taawon Intersection*. Names of streets, roads, intersections, roundabouts, squares, are all location names; names of buildings, shops, parks, institutions, are all landmark names.

The Location Named Entity Extractor receives as input the set of tweets about traffic-related events in Doha, and returns the same tweets where named entities of type Location or of type Landmark have been marked as such. We generate a Location Named Entity Extractor via the Stanford CRF-based Named Entity Recognizer (Finkel, Grenager, and Manning 2005), using POS tags from the CMU-Ark Twitter POS tagger (Gimpel et al. 2011; O’Connor, Krieger, and Ahn 2010) using the tag-set from (Ritter et al. 2011) and a location-specific gazetteer containing more than 700 locations and street names in Doha. The gazetteer is obtained by leveraging different sources, such as Foursquare locations and Open Street Map data. For training we used 400 manually annotated tweets, containing 7.3K words and 1.3K location tags. For testing we used 170 tweets, containing 3K words and 560 location tags. In Table 2 we display the results for the Twitter-specific NER, which show how the performance of the recognizer improved dramatically from a previous version which was based on gazetteer matching only and used no CRF learning.

Resolving location expression onto the map

Given a location entity identified by the previous module, we first check whether or not the location exists in the gazetteer

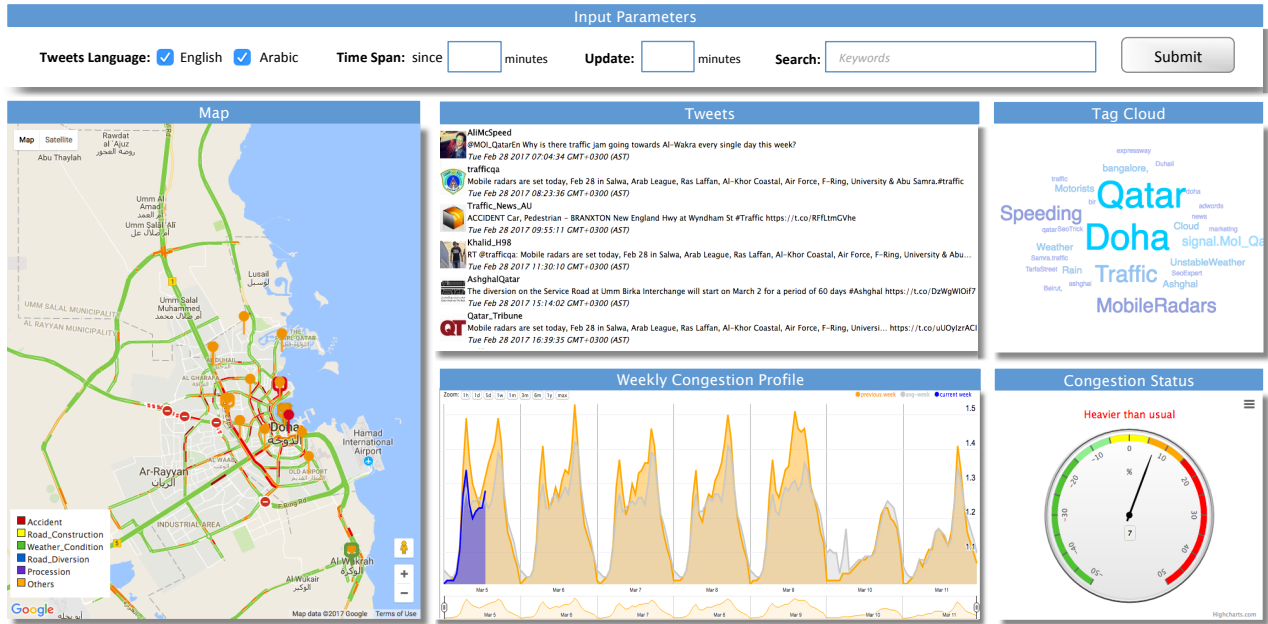


Figure 2: Snapshot of some of QT2S frontend widgets.

	Prec	Rec	F_1
Gazeteer	0.810	0.120	0.210
CRF	0.896	0.674	0.769

Table 2: Word-level precision, recall and F_1 for Location NER.

by using approximate string matching. If the location is not found, the geocoding APIs of Google², Bing³ and Nominatim⁴ are used to map the new location into a pair of latitude / longitude coordinates. We use multiple geocoding sources to increase the robustness of our application, since a single API might fail to retrieve the geographic coordinates of the identified location.

The result geocoding is formatted as a JSON object containing the name of the location entity, its address, and the corresponding geocoding results from Bing, Google and Nominatim. The geocoding process is validated by comparing the results of the different services used. We first make sure that the location returned falls within Doha’s bounding box. We then compute the pairwise distance between the different geographic coordinates to ensure their consistency. If the coordinates returned by the three services are far apart from each other, then the location is ignored.

While it is hard to evaluate the accuracy of such a system, since it relies on many external services, we found that

²<http://bit.ly/1NZn2wa> (accessed on November 15, 2015)

³<https://msdn.microsoft.com/en-us/library/ff701711.aspx> (accessed on November 15, 2015)

⁴<http://wiki.openstreetmap.org/wiki/Nominatim> (accessed on November 15, 2015)

combining our NLP components with geocoding services allows geo-grounding many traffic-related tweets. E.g., from 2015/10/09 to 2015/11/12 the system processed $\approx 25M$ tweets matching one of the keywords we used to filter the Twitter stream. Among them, 3K only have made it to the map, with only 143 gps-geocoded tweets and 2857 tweets grounded by our system. That is, our NLP pipeline augmented the number of available tweets with precise locations 20x. Note that for gps-geocoded tweets, location is about the place where the tweet was issued from; whereas detected locations by QT2S are those referred-to in the text. Obviously, those two locations may not coincide.

Description of the RESTful API

To facilitate the consumption of the relevant geo-located posts and make it possible to integrate them in a comprehensive way with other platforms, we have built a RESTful API. In the context of our system, this refers to using HTTP verbs (GET, POST, PUT) to retrieve relevant social posts stored by our back-end processing. Our API exposes two endpoints: Recent and Search. The former provides an interface to request the latest posts identified by our system, and supports two parameters: Count (maximum number of posts to return) and Language (the language of posts to return.) The latter endpoint enables querying the posts for specific keywords and return only posts matching them. It supports three parameters: Query (list of keywords), Since (datetime of the oldest post to retrieve), From-To (two datetime parameters to express the time interval of interest). In the case of a road traffic application, one could request tweets about “accidents” that occurred in “West Bay” since the 10th of October.

Repurposing the architecture for different applications

Our proposed platform is highly modular (see Figure 1). This guarantees that relatively simple changes can make the platform relevant to any application context where anchoring user messages to specific locations on a map is needed. For instance, the content filter can be oriented to mobility problems in a city: accident or congestion reporting, road blocking or construction sites, etc. With a suitable classifier, our platform can collect traffic and mobility tweets, and geolocate them when possible. However, there are many other contexts in which precise location is needed. E.g., in natural disaster management it is well known that people who have witnessed catastrophic events (floods, typhoons, etc.) use social media as a means to create awareness, demand help or medical attention (Meier et al. 2013). Quite often, these messages may contain critical information for relief forces, who may not have enough knowledge of the affected place and/or accurate information of the level of damage in buildings or roads. Often, the task to read, locate on a map and mark is crowdsourced to volunteers (Imran and Castillo 2014); we foresee that, in such time-constrained situations, our proposed technology would represent a way to support the work of crowdsourcers, or to replace them when they are not available.

Demonstrator script

Our demo is about an instantiation of the proposed architecture to real-time sensing of the status of traffic in Doha (Qatar). Figure 2 shows a screenshot of QT2S dashboard and highlight five relevant visualization widgets along with a section for parameter selection and tuning:

- The *Input Parameters*. Users can decide the data they want to render on the map by setting the following parameters: i) language of the tweets (so far the system supports Arabic and English). ii) Time interval of tweets and traffic data. iii) Refreshing frequency of the widgets. vi) Topic and keywords of interest. The last parameter allows users to explore a subset of Tweets that match some keywords.
- The *Map Widget*. Displays a map of Doha with different markers. Colors of the markers represent different categories of tweets as labeled by our incident detection classifier (i.e. Accident, Road Construction, Weather Condition, ... etc). Large markers are used for tweets with attached images, whereas small markers are used for text-only tweets.
- The *Tag Cloud Widget*. Shows hashtags mentioned in the tweet collection corresponding to the input parameters of users. The font size is correlated with the hashtag frequencies.
- The *Tweets Widget*. Lists the traffic-related tweets which are collected by our system and that satisfy the input parameters of the user.
- The *Weekly Congestion Profile Widget*. Displays three weekly time series representing the congestion levels observed in Doha. Gray color is used for the time series is

for the typical week, aggregated over the last six months. Orange color is used for the time series of the previous week. The blue colored time series correspond to the current week. Having the three time series superposed in one widget allows a user to better get a sense of the current traffic overhead.

- The *Congestion Status Widget*. Shows the current overall congestion status in the city compared to the typical traffic congestion observed at the same hour of the weekday in the past six months. This value is updated every hour.

References

- Abbar, S.; Mejova, Y.; and Weber, I. 2015. You tweet what you eat: Studying food consumption through Twitter. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI 2015)*, 3197–3206.
- Abbar, S.; Zanoua, T.; and Borge-Hoehoefer, J. 2016. Robustness and Resilience of cities around the world. In *The 5th International Workshop on Urban Computing (UrbComp)*. Held in conjunction with the 22th ACM SIGKDD. ACM.
- Finkel, J. R.; Grenager, T.; and Manning, C. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, 363–370.
- Gimpel, K.; Schneider, N.; O’Connor, B.; Das, D.; Mills, D.; Eisenstein, J.; Heilman, M.; Yogatama, D.; Flanigan, J.; and Smith, N. A. 2011. Part-of-speech tagging for Twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, 42–47.
- Han, B.; Cook, P.; and Baldwin, T. 2014. Text-based Twitter user geolocation prediction. *Journal of Artificial Intelligence Research* 49:451–500.
- Imran, M., and Castillo, C. 2014. Volunteer-powered automatic classification of social media messages for public health in AIDR. In *Proceedings of the 23rd International World Wide Web Conference (WWW 2014)*, Companion Volume, 671–672.
- Meier, P.; Castillo, C.; Imran, M.; Elbassuoni, S. M.; and Diaz, F. 2013. Extracting information nuggets from disaster-related messages in social media. In *Proceedings of the 10th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2013)*.
- Mejova, Y.; Abbar, S.; and Haddadi, H. 2016. Fetishizing food in digital age: #foodporn around the world. In *Proceedings of the Tenth International Conference on Web and Social Media, Cologne, Germany, May 17-20, 2016.*, 250–258.
- O’Connor, B.; Krieger, M.; and Ahn, D. 2010. Tweetmotif: Exploratory search and topic summarization for Twitter. In *Proceedings of the 4th AAAI International Conference on Weblogs and Social Media (ICWSM 2010)*, 384–385.
- Rae, A.; Murdock, V.; Popescu, A.; and Bouchard, H. 2012. Mining the Web for points of interest. In *Proceedings of the 35th ACM Conference on Research and Development in Information Retrieval (SIGIR 2012)*, 711–720.
- Ritter, A.; Clark, S.; Mausam; and Etzioni, O. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, 1524–1534.
- Weng, J., and Lee, B.-S. 2011. Event detection in Twitter. In *Proceedings of the 5th AAAI International Conference on Weblogs and Social Media (ICWSM 2011)*.