ARC '16

مؤتمر مؤسسة قطر
السنوي للبحوث
QATAR FOUNDATION
ANNUAL RESEARCH
CONFERENCE

Towards World-class
Research and Innovation

مؤسسة قطر
Qatar Foundation
لإطلاق قدرات الإنسان
Unlocking human potential

Information Communications Technology Pillar

http://dx.doi.org/10.5339/qfarc.2016.ICTPP3046

Real-Time Location Extraction for Social-Media Events in Qatar

Sofiane Abbar, Noora Al Amadi, Francisco Javier Guzman, Fabrizio Sebastiani, Javier Borge-Holthoefer

Qatar Computing Research Institute, QA

Email: sabbar@qf.org.qa

1. Introduction

Social media gives us instant access to a continuous stream of information generated by users around the world. This enables real-time monitoring of users' behavior (Abbar et al., 2015), events' life-cycles (Weng and Lee. 2010), and large-scale analysis of human interactions in general. Social media platforms are also used to propagate influence, spread content, and share information about events happening in real-time. Detecting the location of events directly from user-generated text can be useful in different contexts, such as humanitarian response, detecting the spread of diseases, or monitoring traffic. In this abstract, we define a system that can be used for any of the purposes described above, and illustrate its usefulness with an application for locating traffic-related events (e.g., traffic jams) in Doha.

The goal of this project is to design a system that, given a social-media post describing an event, predicts whether or not the event belongs to a specific category (e.g., traffic accidents) within a specific location (e.g. Doha). If the post is found to belong to the target category, the system proceeds with the detection of all possible mentions of locations (e.g. "Corniche", "Sports R/A", "Al Luqta Street.", etc.), landmarks ("City Center", "New Al-Rayyan gas station", etc.), and location expressions (e.g. "On the Corniche between the MIA park and the Souq)". Finally, the system geo-localizes (i.e. assigns latitude and longitude coordinates) to every location expression used in the description of the event. This makes it useful for placing the different events onto a map; a downstream application will use these coordinates to monitor real-time traffic, and geo-localize traffic-related incidents.

2. System Architecture

In this section we present an overview of our system. We first describe its general "modular" architecture, and then proceed with the description of each module.

Cite this article as: Abbar S, Al Amadi N, Guzman FJ, Sebastiani F, Borge-Holthoefer J. (2016). Real-Time Location Extraction for Social-Media Events in Qatar. Qatar Foundation Annual Research Conference Proceedings 2016: ICTPP3046 http://dx.doi.org/10.5339/qfarc.2016.ICTPP3046.

2.1. General view

The general view of the system is depicted in Figure 1. The journey starts by listening to some social media platforms (e.g., Twitter, Instagram) to catch relevant social posts (e.g., tweets, check-ins) using a list of handcrafted keywords related to the context of the system (e.g., road traffic). Relevant posts are then pushed through a three-steps pipeline in which we double-check the relevance of the post using an advanced binary classifier (Content Filter). We then extract location names mentioned in the posts if any. Next, we geo-locate the identified locations to their accurate placement on the map. This process allow to filter undesirable posts, and augment the relevant once with precise geo-location coordinates which are finally exposed for consumption via a restful API. We provide below details on each of the aforementioned modules.

Figure 1: Data processing pipeline.

2.2. Content filter

The Content Filter consists of a binary classifier that, given a tweet deemed to be about Doha, decides whether the tweet is a real-time report about traffic in Doha or not. The classifier receives as input tweets that have been tweeted from a location enclosed in a geographic rectangle (or bounding box) that roughly corresponds to Doha, and that contain one or more keywords expected to refer to traffic-related events (e.g., “accident”, “traffic”, “jam”, etc.). The classifier is expected to filter out those 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.). We build the classifier using supervised learning technology; in other words, a generic learning process learns, from a set of tweets that have been manually marked as being either real-time reports about traffic or not, the characteristics that a new tweet should have in order to be considered a real-time report about traffic. For our project, 1000 tweets have been manually marked for training purposes. When deciding about a new tweet, the classifier looks for “cues” that, in the training phase, have been found to be “discriminative”, i.e., helpful in taking the classification decision. In our project, we used the Stanford Maximum Entropy Classifier (Manning and Klein, 2003) to perform the discriminative training. In order to generate candidate cues, the tweet is preprocessed via a pipeline of natural language analysis tools, including a social-media-specific tokenizer (O’Connor et al., 2010) which splits words, and a rule-based Named-Entity Simplifier which substitutes mentions of local entities by their corresponding meta-categories (for example, it substitutes “@moi_qatar” or “@ashghal” for “government_entity”).

2.3. NLP components

The Location Expression Extractor is a module that identifies (or extracts) location expressions, i.e., natural language expressions that denote locations (e.g., “@ the Slope roundabout”, “right in front of the Lulu Hypermarket”, “on Khalifa”, “at the crossroads of Khalifa and Majlis Al Taawon”, etc.). A location expression can be a complex linguistic object, e.g., “on the Corniche between the MIA and the underpass to the airport”. A key component of the Location Expression Extractor is the Location Named Entity Extractor, i.e., a module that identifies named entities of Location type (e.g. “the Slope roundabout”) or Landmark type (e.g., “the MIA”). For our purposes, a location is any proper name in the Doha street system (e.g., “Corniche”, “TV roundabout”, “Khalifa”, “Khalifa Street”); landmarks are different from locations, since the 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 that is right in front of it).

The Location Named Entity Extractor receives as input the set of tweets that have been deemed to be about some traffic-related event in Doha, and returns the same tweet where named entities of type Location or of type Landmark have been marked as such. We generate a Location Named Entity Extractor via (again) supervised learning technology. In our system, we used the Stanford CRF-based Named Entity Recognizer (Finkel et al., 2005) to recognize named entities of type Location or of type Landmark using a set of tweets where such named entities have been manually marked. From these “training” tweets the learning system automatically recognizes the characteristics that a natural language

expression should have in order to be considered a named entity of type Location or of type Landmark. Again, the learning system looks for “discriminative” cues, i.e., features in the text that may indicate the presence of one of the sought named entities. To improve the accuracy over tweets, we used a tweet-specific tokenizer (O’Connor et al., 2010), a tweet-specific Part-of-Speech tagger (Owoputi et al., 2013) and an in-house gazetteer of locations related to Qatar.

2.4. Resolving location expression onto the map

Once location entities are extracted using the NLP components, we use the APIs of Google, Bing and Nominatim to request the geographic coordinates of the map location entities into geographic coordinates. Each location entity is geo-coded by the Google Geolocation API, Bing Maps REST API and Nomination gazetteer individually. We use multiple geo-coding sources to increase the robustness of our application, as a single API might fail to retrieve geo-coding data. Given a location entity, the result of the geo-coding retrieval is formatted as a JSON object containing the name of the location entity, its address, and the corresponding geo-coding results from Bing, Google or Nominatim. The geo-coding process is validated by comparing the results of the different services used. We first make sure that the location returned falls within Qatar’s bounding box. We then compute the pairwise distance between the different geographic coordinates to ensure their consistency.

2.5. Description of the Restful API

In order to ease the consumption of the relevant geo-located posts and make it possible to integrate these posts 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 endpoint provides an interface to request the latest posts identified by our system. It supports two parameters: Count (maximum number of posts to return) and Language (the language of posts to return i.e., English or Arabic.) The later endpoint enables querying the posts for specific keywords and return only posts matching them. This endpoint supports three parameters: Query (list of keywords), Since (date-time of the oldest post to retrieve), From-To (two date-time 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.

3. Target: single architecture for multiple applications

Our proposed platform is highly modular (see Figure 1). This guarantees that relatively simple changes in some modules can make the platform relevant to any applicative context where locating user messages on a map is required. For instance, the content classifier – the first filtering element in the pipeline – can be oriented to mobility problems in a city: accident or congestion reporting, road blocking or construction sites, etc. With the suitable classifier, our platform will collect traffic and mobility tweets, and geo-locate them when possible. However, there are many other contexts in which precise location is needed. For instance, in natural disaster management, it is well admitted that people involved in catastrophic events (floods, typhoons, etc.) use social media as a means to create awareness, demand help or medical attention (Imran 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 crowd-sourced to volunteers; we foresee that, in such time-constrained situations, our proposed technology would represent an advancement. Likewise, the system may be oriented towards other applications: weather conditions, leisure, etc.

4. System Instantiation

We have instantiated the proposed platform to the problem of road traffic in Doha. Our objective is to sense in real-time the traffic status in the city using social media posts only. Figure 2 shows three widgets of the implemented system. First, the Geo-mapped Tweets Widget shows a Doha map with different markers: the yellow markers symbolize the tweets

geo-located by the users, the red markers represent the tweets geo-located by our system; the large markers come from tweets that have an attached photo, while the small markers represent the text-only tweets. Second, the Popular Hashtags Widget illustrates hashtags mentioned by the users, where the large font size shows the most frequent one. Third, the Tweets Widget lists the traffic-related tweets which are collected by our system.

Figure 2: Snapshot of some System's frontend widgets.

5. References