HipTrip: Exposing Local Hotspots by Classifying and Linking Geospatial Social Media

Cort Werner[1], Ryo Chiba[1]

[1]Dept. of Computer Science, University of Southern California, USA

Abstract. When traveling, there are many existing services that provide reviews to help travelers decide where to dine, shop, or explore. However, distinguishing between places that tourists frequent and places that locals frequent is a challenge that continues to elude review services. In this paper, we present *HipTrip*, a novel system that utilizes geo-spatial data from various public sources to determine local "hotspots", and finds highly rated points of interest within these areas. This application utilizes Semantic Web technology, named entity recognition, knowledge bases, and information extraction techniques to synthesize traveler behavior and points of interest in order to expose a visual layer of valuable data for travelers.

Keywords: tourist, heatmap, poi, geospatial, travel, named entity recognition

1 Description

Building and exposing a more comprehensive knowledge base of tourist social media activity and combining this with a knowledge base of reviews of local businesses has not been done before, and would be of significant practical value. Businesses that cater to tourists are often more expensive, crowded, and tacky. By collecting and exposing tourist behavior, travelers find cheaper, calmer, and hipper points of interest to patronize. Within a given geographic region, we propose a mashup that retrieves and analyzes posts from:

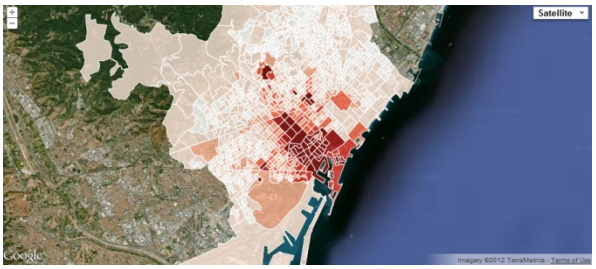
1) Flickr – Tourists generate countless geo-tagged photos that will aid in both visualizing points of interest and determining if touristy or not.

2) Twitter – This be used in the same way as Flickr, but using geo-tagged tweets.

3) Yelp – To find Points of Interests to overlay on tourist behavior data, we will utilize ratings and metadata from Yelp.

To visualize all of this data, *HipTrip* uses OpenStreetMap[1] and Javascript Heatmaps[2]. Colored visualization of frequent tourist and local social media postings allow users to quickly discern which areas are areas of densest tourist activity.

[1] http://www.openstreetmap.org/



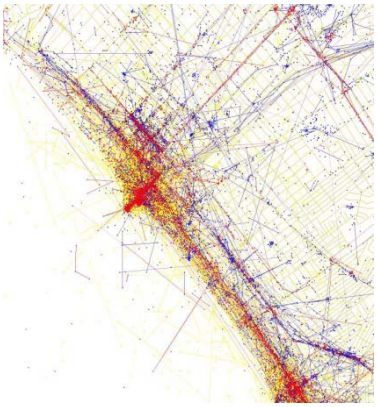


Fig. 2. The Data Republic's visualization of Barcelona

Fig. 3. Eric Fischer's *Locals and Tourists*

3 Previous Work

Previous work has been done on determining tourist hotspots based on available API's. Most notably, Eric Fischer's widely circulated data visualizations[3] and The Data Republic's work using data to model tourist behavior in Barcelona (Fig. 1 and 2)[4]. Heatmaps were used by Ahti Heinla to create a map of the least touristy places, although the data was only granular enough to determine cities of interest, rather than points[5].

3 *HipTrip* User Interface

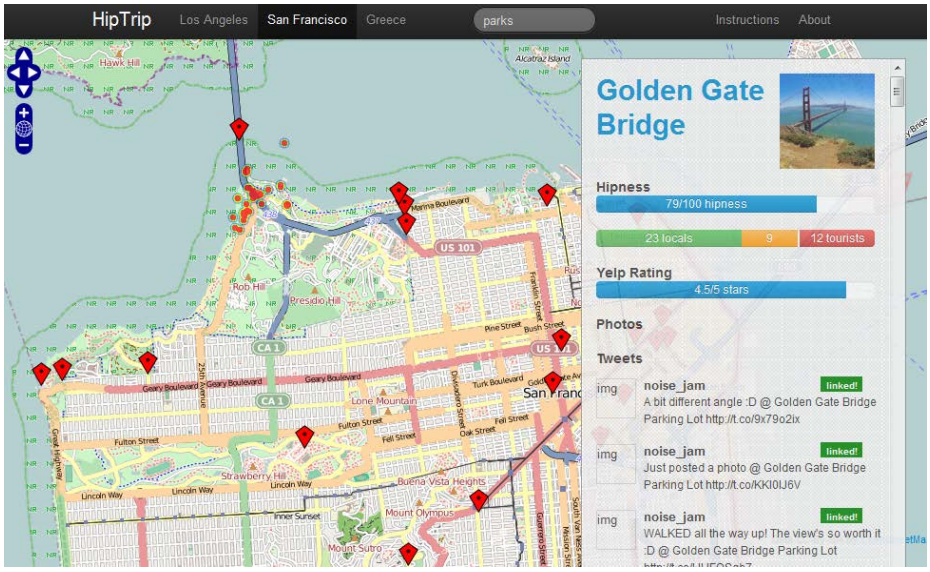


Figure 1: *HipTrip* User Interface – visit at http://www.hiptrip.us

---

[2] http://www.patrick-wied.at/static/heatmapjs/

[3] http://www.flickr.com/photos/walkingsf/sets/72157623971287575/

[4] http://flickrtourism.thedatarepublic.com/

[5] http://maps.google.com/maps/mpl?moduleurl=http://www.bluemoon.ee/~ahti/touristiness-map/touristiness-map.xml

The HipTrip user interface features a map overlayed with either the most linked POIs in a given city by default or POIs that match a search if keywords are entered in the search bar. Clicking on any of these POIs will reveal a heatmap of nearby social media postings, with “local” users being weighted more than tourists, allowing the user to gauge the activity surrounding this POI. In addition, the tweets and photos linked to this, a breakdown of the how nearby social media users were classified, and tags that were associated with this POI are displayed in a side overlay. Users can quickly explore related POIs by clicking on a given POIs tags, and the maker opacity is tied with the “hipness” score of a POI, allowing the user to quickly find the hippest POIs. This score is tied to how nearby social media users are classified, as explained below.

4 User Classification

Our approach in determining the relative “touristyness” of a user first involves linking a user’s account location field to a geographic point. Yahoo! PlaceFinder was used to do this due to the unstructured nature of this field which often included euphemisms for cities, coordinates, and misspellings. For our dataset of 12,000 Twitter users, we were able to find 9,000 matches. To determine hipness scores for each point of interest, *HipTrip* takes a number of nearest Flickr photos and tweets and finds the distance between the point of interest and the user account associated with each social media posting. Using this distance, users are classified as either people within a metropolitan (30km), state (500km), or out-of-state(>500km) scale distance. A perfectly “hip” score indicates that all nearby posting users are from within the metropolitan area while a perfectly “touristy” score indicates that all are from outside of the country. When a point of interest is selected, nearby social media posts by “locals” are shown in a heatmap layer, providing a visual indicator to the hipness of a given region.

5 Record Linkage

One major task of building our knowledge base was the necessity to link together our various data sources to present the user with a richer experience. What made this task particularly difficult is that we were dealing with free form text in the form of tweets and image descriptions on Flickr that may or may not refer to a point of interest.

> **Ex:**
> *Bro Tip: A well groomed stache is the key to any coffee shop meeting. (at Oakside Cafe) [pic] — http://t.co/at2IExoV*
>
> *@Harrison_Lee have you listened? RT@pitchforkmedia: Check out the New Jack White Track "Sixteen Saltines" http://t.co/bIcmzv3U*

A cursory glance at these tweets illustrates several fundamental problems with performing point of interest record linkage using free form text from Twitter. First and foremost, tweets are largely unstructured. Although the first contains a point of interest in San Francisco, the Oakside Cafe, the latter does not and is simply conversation. Next, tweets contain text particular to the Twitter domain; for example, users know that by starting the tweet “@Harrison_Lee” the user is directing his tweet at another user and that the URL at the end refers to some media. However, a program without any special domain knowledge will fail to take this into account

and may disambiguate records erroneously. Lastly, there can be a great deal of tokens in tweets. For example, if we're checking to see if a tweet matches the point of interest "Golden Gate Bridge" then we would have to examine each consecutive trigram in the tweet. Another further complication is that we don't always compare a simple point of interest name like "Golden Gate Bridge." In our Flickr data source, the photo title or photo description may contain useful text for linkage. In that case, we would need to compare unigrams, bigrams, trigrams, and so on until some limit. This is costly, slow, and extremely inefficient.

To resolve the aforementioned problems, we opted to utilize machine learning in the form of named entity recognition in combination with geolocation data to facilitate the linkage process. We used the Natural Language Toolkit(NLTK) package to perform named entity recognition on the generally well-formed text in Flickr titles and descriptions. We noticed that Flickr text tended to be properly capitalized and properly spelled so NLTK's named entity chunker was adequate:

Ex.
DESC: Backflip Studios party at the legendary Great American Music Hall in San Francisco on March 5th, 2012.

Photo by Charlotte Cunningham:
**
charlottecunningham.com

TITLE: Backflip Bash: GDC 2012
NAMED ENTITIES: ['Backflip', 'Studios', 'Great American Music Hall', 'San Francisco', 'Photo', 'Charlotte Cunningham', 'Backflip']

The well-formed nature made the task easy using NLTK's machine learning based named entity chunker. However, this was inadequate for Twitter due to its free-form nature and frequent lack of spelling and capitalization. Fortunately for us, Ritter et al, EMNLP 2011 dealt with this issue and made available publicly their tool that uses Conditional Random Fields to segment named entities[6]. Note that for our task, we were only interested in named entities and not their further classification into Person/Location/Organization; this is because for the sake of named entities, the line between location and organization is not clear. Thus, we were only interested in the segmentation of named entities and non-named entities. For this particular task, their tool yields an F1 score of 0.67 with a precision score of 0.73. Though imperfect, we felt the relatively high precision rate in combination with our geo data for points of interest would yield very accurate results in terms of linkage. We prioritized higher precision over recall; from a utilitarian point of view it matters more to the user that he or she sees information corresponding to the correct point of interest rather than making sure the user sees all possibly related information. We ran the Twitter NER tool to annotate our tweet corpus with BIO tag information:

Ex.

[6] http://malt.ml.cmu.edu/mw/index.php/Ritter_et_al,_EMNLP_2011._Named_Entity_Recognition_in_Tweets:_An_Experimental_Study

Bro/O Tip/O :/O A/O well/O groomed/O stache/O is/O the/O key/O to/O any/O coffee/O shop/O meeting/O ./O (/O at/O Oakside/B-ENTITY Cafe/I-ENTITY)/O [/O pic/O]/O —/O http://t.co/at2IExoV/O

The resulting BIO tags allowed us to filter out mostly irrelevant text and only compare relevant named entities. With the named entity chunks, we had a far smaller list of tokens for comparison and possible record linkage. With this reduced token set, we proceeded to perform the comparisons by leveraging the geocode data. Our data was stored in a database that supported geospatial indexing, so for example in the Twitter to Yelp record linkage phase we queried the 20 closest Yelp points of interest and compared their names against the tweet's named entity chunks. We then determined linkage by applying an edit distance comparison between the named entity chunk and business name. The formula we used was to take the business name and allowed a threshold of 20% edit distance; in other words, if the Yelp name was "McDonald's" with a length of 10 characters, the maximum edit distance for linkage would be two characters. We found the threshold of 20% to properly mitigate the trade-off between allowing for user errors and not wanting to introduce erroneous linkages. By querying the nearby points of interest, we found a high precision rate in the linkage.

We found is that when performing comparisons to determine linkage to Yelp points of interest, there was only one field to examine: the "name" field. This made things straightforward. However, when performing Twitter to Flickr linkage, the process was made significantly more complex because each contained a great deal of free-form text. As a result, each set of named entities from both sources must be compared chunk by chunk and the process was significantly slower than their counterparts involving Yelp.

6 Future Work

We would like to enhance the record linkage step by adding a layer of machine learning to determine linkage. Our current method is good and provides very accurate results, but it is a trade-off of precision at the expense of recall. For example, many businesses contain the city name but people don't necessarily refer to them as such: for example, people may refer to the "San Francisco Museum of Art" as simply the "Museum of Art" when in the city. We propose a machine learning algorithm like a Support Vector Machine that counts substring matches and properly weights their importance would help this problem and greatly increase our recall. The only difficulty in doing so is the need for a comprehensive training corpus that includes a significant amount of examples illustrating what should and should not be linked. This will take significant time and effort to construct.

We plan to augment the featured cities in the near future; we started with San Francsico as a starting point but we will soon apply the same methods to Los Angeles, Chicago, and New York. Furthermore, we want to build a more dynamic version of the application that is able to periodically poll our more live data sources (particularly Twitter) for new data. We feel that stale data will not adequately serve the users' purpose of being able to see the newest and most recent trends of an area. We believe this can be accomplished by running perhaps weekly or bi-weekly jobs to download new Twitter data and apply linkages before updating our linked data source.