SoMap: Dynamic Clustering and Ranking of Geotagged Posts

Marie Al-Ghossein
LTCI CNRS, Télécom ParisTech,
Université Paris Saclay
marie.alghossein@telecom-paristech.fr

Talel Abdessalem
LTCI CNRS, Télécom ParisTech,
Université Paris Saclay
talel.abdessalem@telecom-paristech.fr

ABSTRACT

This demo presents *SoMap*, a web-based platform that provides new scalable methods to aggregate, analyse and valorise large collections of heterogeneous social data in urban contexts. The platform relies on geotagged data extracted from social networks and microblogging applications such as Instagram, Flickr and Twitter and on Points Of Interest gathered from OpenStreetMap. It could be very insightful and interesting for data scientists and decision-makers.

SoMap enables dynamic clustering of filtered social data in order to display it on a map in a combined form. The key components of this platform are the clustering module, which relies on a scalable algorithm described in this paper, and the ranking algorithm that combines the popularity of the posts, their location, and their link to the points of interest found in the neighbourhood. The system further detects mobility patterns by identifying and aggregating trajectories for all the users. *SoMap* will be demonstrated through several examples that highlight all of its functionalities and reveal its effectiveness and usefulness.

1. INTRODUCTION

It is well-known that social media became an important source of information, not only for individuals but also for the companies. Users are producing large amounts of heterogeneous data that can describe their activities and their opinions. Researchers, data scientists and decision-makers analyse social data in order to infer interesting tendencies and insights on users' behaviour, searching also for causes and implications.

With the increasing availability of location-aware devices, geotagged posts are frequently published on online social networks and microblogging applications, such as Instagram and Twitter. By collecting the posts related to a specific user and sorting them in a chronological order, we may be able to draw the physical individual trajectory taken by the user. In an aggregated form, geotagged data enhance the ability to understand data trends and provide significant insights about urban dynamics, mobility between locations and frequent venues.

Furthermore, textual and visual content of geotagged posts make them more informative than ordinary check-ins that only reveal users' locations. Gathering large sets of data can tell us for example where specific terms are frequently mentioned and therefore which aspect of a particular region is so popular and well-known.

Aggregating geotagged posts is an important step in the process of data analysis and has to be carefully accomplished. We wouldn't want to give wrong intuitions and we wouldn't want to hide valuable information and important common trends shown by social data.

In this demo, we present *SoMap*, a web-based platform that considers geotagged posts extracted from social media and provides scalable methods to aggregate, analyse and visualize the extracted data. Our system dynamically clusters social data based on its geographic distribution, displays the clusters and the mobility flows on a map. The most relevant posts can be displayed for each location.

The main components of *SoMap* are its scalable clustering algorithm, its location-aware ranking algorithm, and the mobility patterns tracking module.

Dynamic clustering of the crawled data. The clustering is essential to aggregate data and to clearly visualize large-scale collections on the map. The developed algorithm clusters the filtered posts on-the-fly, doesn't require an a-priori decision about the number of clusters and is sensitive to the level of zoom of the map (the clusters are adjusted dynamically, according to the zoom level).

Location-aware ranking of the posts. Every cluster represents a group of posts tagged in the same region. How could we select among these posts, the most relevant ones? The originality of our approach lies in the fact that our algorithm combines the popularity of the posts with the relevant elements (Points Of Interest) found in the neighborhood. It relies on the tags used to describe the post, the geolocation of the posts, the number of likes expressed on the social network, and the geographic distribution of the Points Of Interest (POI) in the area.

Mobility patterns. Users' trajectories, observed for a given time interval, can also be clustered dynamically and displayed at different levels of zoom. Thus, we can analyse visually the mobility patterns (origin/destination) inferred from users' trajectories. Each cluster of patterns is represented on the map by a path linking two clusters of points, inferred from users locations (geotags of their posts).

WWW 2016, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4143-1/16/04.
http://dx.doi.org/10.1145/2872518.2890550.

Our platform enables also the filtering of the input data, in order to refine the data analysis. We can choose the time frame, the targeted location, and the data sources (Twitter, Instagram or Flickr). Then, the input data is automatically filtered according to the chosen filters.

The remaining sections are organised as follows. In section 2, we present the approaches and algorithms used to implement each functionality of the system. Section 3 describes the handled datasets, some pre-processing steps and the system architecture. Finally, section 4 exposes the demonstration scenario.

2. SOMAP

2.1 Filtering Social Data

Our initial datasets consist of geotagged posts crawled from Instagram, Twitter and Flickr. Each post is characterised by a geographic location (latitude,longitude), the publication date, the identifier of the user who published it, its content (text for Twitter or photo for Instagram and Flickr), the number of likes it got, and the list of tags (keywords) associated to the post.

Displaying the crawled posts on a map can be very interesting and insightful. However, due to the large number of posts, we should be able to specify some characteristics of the data that has to be analysed and displayed. The platform gives the user the possibility to filter the input data according to multiple criteria including :

1. Time frame : when specified, only posts published in the required day, month, or year will be retrieved and shown on the map;
2. Spatial filter : only posts located in one or several specific regions are displayed. Countries and big cities can be usually divided into regions and districts. The spatial filtering in *SoMap* is based on the division of the area into rectangles defining each region;
3. Source filter : it is possible to indicate the source of social data (e.g., only display Instagram posts).

2.2 Dynamic Clustering

Our algorithm relies on a hierarchical division of the space in rectangular cells, enhanced with additional statistical information in each cell. Our work has been inspired by [4], which was used in spatial databases to answer efficiently region-oriented queries, such as finding connected regions which satisfy a density condition and possible additional conditions on the properties of the objects contained in the cells. In order to fasten the re-computation of the clusters according to the zoom level, we use a hashing technique for the hierarchy of space cells. At the end, we obtain a scalable clustering approach, dynamic and sensitive to the zoom level of the map. Our algorithm handles the input data without any need to analyse the posts individually.

Grid cell hierarchy. As described in [4], the considered space is divided hierarchically into rectangular cells. At the first level of the hierarchy we have one cell (root node), covering the whole region of the crawled data (the Parisian region, for the demonstration). Each cell at a higher level l is partitioned into 4 cells at the next lower level $l-1$. The number of levels in the cells hierarchy corresponds to the number of zoom levels allowed on the map. When we zoom in, we move from one level to the next lower one. Thus, the area displayed on the map corresponds to the space covered by one of the cells at a certain level of the hierarchy.

Statistical information. In our algorithm, we compute and store for each couple *(cell, day)* the following values that are essential for the clustering:

- n – number of geotagged posts located in the cell with respect to their publication date;
- C – average position of the centroid, regrouping the n posts.

Aggregating and indexing cells. The values associated to a higher level cell are calculated from the values of the next lower level cells, by computing the sum of the number of posts n and the average of the centroid C. In order to find rapidly the lower cells that has to be aggregated and retrieve their values, we rely on a hashing technique similar to the one used in the geoHash system [1]. We assign to each cell a hash key whose length is determined by the cell's level in the hierarchy. The hash of a cell child is equal to the hash of the parent cell plus "0","1","2" or "3" if the child cell corresponds, respectively, to the top left, top right, bottom left or bottom right quarter of the parent cell. The children of a cell having a hash h at level l are the cells located at level $l' < l$ and having a prefix of their hash equal to h.

Clustering algorithm. In order to cluster the geotagged posts, we use a top-down approach based on the hierarchical structure of the grid cells. Starting with the root cell, we compute the density of posts geolocated in the region covered by the cell c as follows:

$$Density(c) = \frac{n}{Area(c)}$$

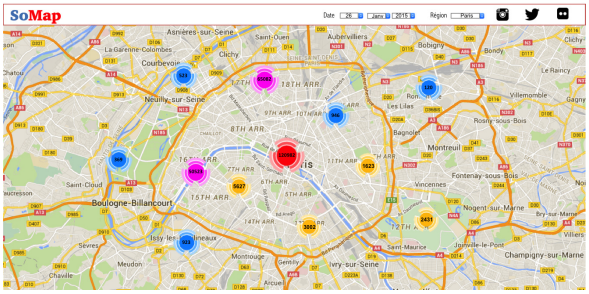
where n is the number of posts located in c and $Area(c)$ is the area of c. Moving to the next lower level, we compute again the density of posts in each cell. When we find a cell with a higher density than its parent cell, we consider it as a cluster and use its centroid to locate it on the map. For the other cells, and in case they include at least one post, we proceed to the next lower level and repeat the same process. This procedure continues until one of the following two conditions is met : *1.* all the geotagged posts are clustered; *2.* we reach the maximum level of zoom (no child cells). Finally, the last step of the algorithm consists in the merge of the clusters found at the same level and located within a small fixed distance.

The clustering algorithm is executed whenever the user zooms in or out. Its computation time does not depend on the number of geotagged posts we have in the input dataset, and it uses only the statistical information associated to the cells. It is therefore very fast and scalable.

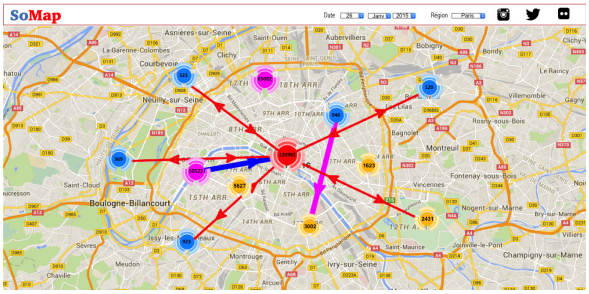
2.3 Mobility Flows

The trajectories of each user are inferred from the geotagged posts he or she published. The posts are sorted in a chronological order for this purpose. Then, the trajectories are aggregated in patterns of mobility. These patterns show the flows of mobility between couples of origin and destination clusters of users' positions.

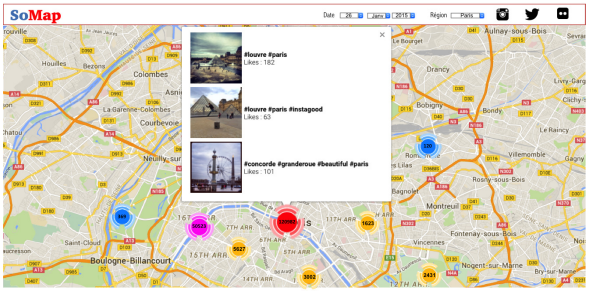
[1] http://www.geohash.org/



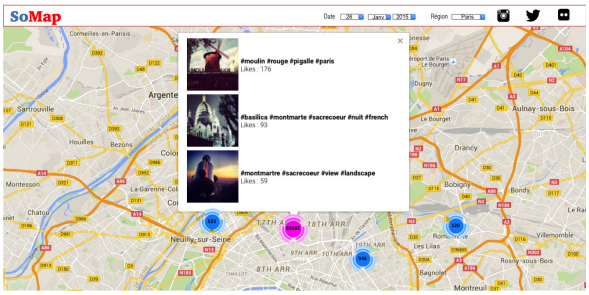
(a) Visualising clusters of posts geotagged in Paris region



(b) Visualising mobility flows between clusters of posts



(c) Displaying most relevant posts geotagged in the center of Paris



(d) Displaying most relevant posts geotagged in the north of Paris

Figure 1: Examples of data visualisations of the *SoMap* platform

2.4 Location-aware ranking of the posts

Each cluster gathers a number of geotagged posts. The objective of their ranking is to be able to show on the map the most representative and relevant ones. To do so, we implemented in *SoMap* a ranking algorithm that selects the top-k most relevant ones (k is set to 3 for this demonstration).

Different efforts have been made in order to rank posts extracted from social media and choose the appropriate metrics for this ranking. Some of these works focused on the frequency of tags used to describe the posts [1], others on the visual content of photos associated to the posts [2], on the popularity of the posts, on the reputation of the authors of the posts, etc.

In *SoMap*, we chose to combine the popularity of the posts with the information we are able to obtain on the points of interest located nearby. Our solution relies on the tags associated to the posts, their locations, the number of likes (or any other metrics that can indicate how much a post is appreciated), and the geographic distribution of the POIs found in the neighborhood. Our algorithm can handle posts containing only text, as well as posts with associated images (e.g., Instagram posts).

Motivation. When dealing with tags provided by users, it is important to note some essential points: *1.* The tags are usually imprecise; *2.* They are not always related to the image or the topic discussed; *3.* The order of the tags in the list is not necessarily correlated to their degree of relevance. Therefore, only relying on the tags for the ranking of the posts could not give a good result.

Intuitively, the most relevant posts published in a particular place should be related to the POIs located nearby. For instance, in the neighbourhood of the Eiffel Tower in Paris, users on Instagram would normally post pictures of the Eiffel Tower, the Trocadero or some popular restaurants or gardens found in the area. A photo of a burger posted near the Eiffel Tower, should be less relevant, even if it is posted by a famous star or if it is liked by a lot of friends.

Associate posts to POIs. Therefore, the first step of the ranking algorithm would be to associate every geotagged post to the POIs found nearby. For each geotagged post, we look at the POIs located within a small fixed radius. Among these POIs, we evaluate the number of words describing the POI and appearing in the tags. Then, we associate the post to the POI with the highest matching (if many, we choose the closest POI).

Ranking the posts. At this point of the algorithm, we propose to rank the posts associated to each POI. In other words, we would like to compute, for each post, the scores of the tags linked to the considered POI. Initially, the more unique a tag is for a specific POI, the more relevant it is for that POI, and then the higher its assigned score should be. The approach is based on the TF-IDF scoring system used in [1]. The initial score of each tag t used in a post p linked to the POI poi is defined as follows:

$$S_i(t, poi) = tf.idf.uf$$

where tf is the normalised term frequency of the tag t among the posts of poi, idf is the overall ratio of the tag t among all the posts geotagged in the considered region, and uf is the proportion of users that used the tag t among the posts related to poi.

Notice that the initial score of a specific tag is independent of the context where it is used and gives the same importance to any post mentioning it. In order to explore the relationship between tags, we refine the initial score of each tag by performing a random walk on the graph of tags as detailed in [3]. This process promotes the tags that are closely related and weakens the isolated ones. We construct the graph for each post.

The nodes of the graph are the tags used to describe the considered post and the edges are weighted with concurrence similarity that is based on the co-occurrence of each couple of tags. In order to compute the concurrence similarity, we first estimate the distance between two tags t_i and t_j as follows:

$$d(t_i, t_j) = \frac{\max(\log(f_t(t_i)), \log(f_t(t_j))) - \log f_t(t_i, t_j)}{\log(N) - \min(\log(f_t(t_i)), \log(f_t(t_j)))}$$

where $f_t(t_i)$ and $f_t(t_j)$ are the number of posts mentioning the tags t_i and t_j respectively, $f_t(t_i, t_j)$ is the number of posts mentioning both t_i and t_j, and N is the total number of posts. The concurrence similarity is therefore defined as:

$$\varphi_c(t_i, t_j) = \exp(-d(t_i, t_j))$$

At the end of the random walk process, the score of the post is equal to the score of the tag mentioning the POI and is marked $S_t(p, poi)$. (If many tags are linked to the POI, we choose the tag with the highest score).

$S_t(p, poi)$ is then used to rank the posts associated to each POI. If two posts score the same, we order them based on the number of likes.

Ranking the POIs. It is obvious that different POIs can have different importance and not all of them can impact people in the same way. We introduce a popularity metric that highlights the importance of each POI *poi* in a given region R, as follows:

$$Popularity(poi, R) = \frac{f_u(poi, R)}{N_u}$$

where $f_u(poi, R)$ is the number of users posting about *poi* in R, and N_u is the total number of users. The popularity metric allows us to rank the POIs in each specific region.

Final ranking. Let's go back to our initial problem. We want to find and display the most relevant posts published in a given region R (a cluster of posts' locations). We generate for R two lists: *1.* A ranked list of POIs located in R; *2.* A ranked list of posts associated to each POI in R. The aggregation of these two lists would lead us to the final ranking. This is done using the following simple formula:

$$S_f(p, R) = S_t(p, poi) \times Popularity(poi, R)$$

In summary, the ranking approach proposed takes into consideration the geographic distribution of POIs in the region, the popularity of POIs, and the relevance of the tags contained in the post.

3. DATASET AND SYSTEM OVERVIEW

The used dataset gathers 15 million Instagram posts, 1 million tweets and 500,000 Flickr posts. The posts are geotagged in the area of Île-de-France (almost 40 kilometers around Paris). The data is stored in a MongoDB database.

The POIs dataset is extracted from OpenStreetMap where a location and a descriptive name are associated to each POI. In the pre-processing step, we remove punctuation and stop words from the description of the POIs in addition to the redundant entities. We obtain at the end approximatively 40,000 POIs in the region of Île-de-France.

The platform is implemented as a client-server web application. The client part allows the users to define their data filtering criteria and displays the results (clusters, mobility patterns, ranked posts) on the map. The server processes the users' requests based on the algorithms mentioned in the previous section. It partially relies on pre-processed and indexed data. The updated and added data are handled with a minimum cost, without needing to recompute all pre-processed information.

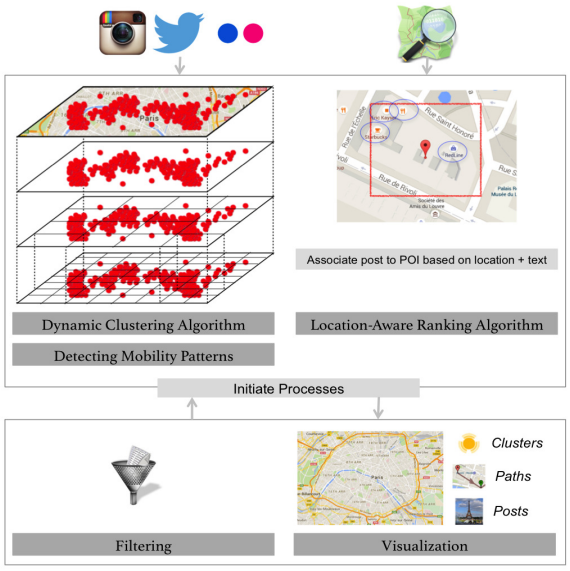


Figure 2: System architecture

See an illustration of the system architecture in Fig. 2.

4. DEMONSTRATION SCENARIO

We will use the following scenario for our demonstration. First, the user specifies the data filtering criteria by selecting on the interface a particular time frame, one or many regions of the Île-de-France area, and the data source. Once the choice is confirmed, the clusters of the geotagged posts are displayed on the map. The user can then zoom in and out and notice the merging and division of clusters. After that, the user enables the display of mobility flows between the clusters. Whenever the user clicks on a cluster, the most relevant posts geotagged in the cluster will be displayed in a small box. The user can modify the filtering parameters at any point of the scenario and therefore repeat it.

The demonstration show that our system retrieve and display processed data effectively and accurately, and provide relevant information about aggregated data.

5. ACKNOWLEDGEMENTS

This work was supported by the NormAtis ANR project.

6. REFERENCES

[1] S. Ahern, M. Naaman, R. Nair, and J. Yang. World explorer: Visualizing aggregate data from unstructured text in geo-referenced collections. In *JCDL*, 2007.

[2] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In *WWW*, 2009.

[3] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang. Tag ranking. In *WWW*, 2009.

[4] W. Wang, J. Yang, and R. Muntz. STING : A statistical information grid approach to spatial data mining. In *VLDB*, 1997.