Enriching Geolocalized Dataset with POIs Descriptions at Large Scale

Ibrahima Gueye[1(✉)], Hubert Naacke[2], and Stéphane Gançarski[2]

[1] Ecole Polytechnique de Thiès, LTISI, Thies, Senegal
igueye@ept.sn

[2] Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6,
75005 Paris, France
{hubert.naacke,stephane.gancarski}@lip6.fr

Abstract. We present an efficient method to enrich a geolocalized dataset with contextual description about Points of Interest (POI). We implemented our solution using two large scale datasets: YFCC [14] and Geonames [2]. A practical problem we have encountered is the size of the manipulated data. Actually, the YFCC geolocalized dataset accounts for 45 million entries that we propose to cross with 12 millions of Geonames POIs. We show that using the Apache Spark cluster computing platform and the GeoSpark [18] spatial join library as-is lead to inefficient computation because of the important bias in the data. We propose a method to distribute the data non uniformly according to the data bias, which greatly improves the spatial join performance. Moreover, we propose a method to select among a set of close POIs, those which are the most relevant with the YFCC entries. The resulting enriched dataset will be made publicly available and should contribute to better validate future works on large scale POI recommendation.

Keywords: YFCC large scale dataset · Distributed query processing · Spatial join · Apache Spark · POI recommendation

1 Introduction

Photos and videos extracted from social media are used in many contemporary studies and research that are as varied as they are numerous. These works range from the Point Of Interest (POI) recommendation system for tourist tours [6–8,13] to helping systems for destination promotion [1]. These photos and videos are also used as training data for unsupervised deep neural networks [11] or supervised classifications [5]. Such data originating from social media is usually provided with some contextual information: at least every photo has a geolocation (GPS coordinates), a user id, and a timestamp.

However, there is an increasing demand for richer contextual information that could be captured in new recommendation models, expecting to improve

J. P. R. Thorn et al. (Eds.): InterSol 2020, LNICST 321, pp. 264–273, 2020.
https://doi.org/10.1007/978-3-030-51051-0_19

the overall quality of recommendation systems. It appears that recent research efforts about POI recommendation fall into two distinct lines of works, depending on the model complexity and the dataset scale. There are either complex (sophisticated) models validated using rich but small-scale datasets, or simpler models validated through large-scale datasets but containing few contextual information [3]. We are not aware of much work that would explore the two above-mentioned dimensions together, *i.e.*, works that would propose a sophisticated model taking into account rich semantic information and validated at large scale using millions of POI from several continents all around the world. If a complex recommendation model have a low theoretical complexity that qualify it as scalable, its large scale performance validation remains an open issue as long as there is no publicly available rich and large dataset for such validation.

Our goal is to contribute to the research community about POI recommendation and deliver a rich and large-scale dataset. This will help to improve both the training phase as well as the test phase of new POI recommendation models. We consider a large-scale dataset about geo-located photos originating from social media, and focus on enriching it with contextual POI descriptions. We intend to join two large-scale datasets on their geo-location, which raises two difficult problems: (1) a photo location almost never exactly match any POI location, it generally requires to approximately match 0 or many POIs within a fixed radius. (2) Among all the possible close POIs that approximately match a photo location, only few of them are relevant with the photo.

We show that using the Apache Spark cluster computing platform and the GeoSpark [18] spatial join library as-is lead to inefficient computation because of the important bias in the data. We propose a method to distribute the data non uniformly according to the data bias, which greatly improves the spatial join performance. Moreover, we propose a method to select among a set of close POIs, those which are the most relevant with the photo tags. We implemented our solution using two large scale datasets: YFCC [14] and Geonames [2].

In the following Sect. 2, we provide some background knowledge. Section 4 details our enrichment method. Section 3 gives an overview of related works.

2 Background: YFCC100M, Geonames Datasets and Spatial Queries

2.1 Yahoo Flickr Creative Commons 100M : YFCC100M

The exponential growth of Instagram has brought photo sharing to the mass, generalizing the use-case that Flicker early initiated through its photo sharing platform. By now, Flicker contains an impressive amount of photos that users published since 2004, and it still remains an active platform for sharing photos over the Internet and to manage personal galleries on the cloud. In particular, the availability of Flicker's data has made it academically recognized as a source of pictorial research [7]. In July 2015, Yahoo released and made available a visual content dataset for researchers called "Yahoo Flickr Creative Commons

100M” *YFCC100M* [14]. The Dataset contains more than 100 million multimedia metadata published on Flickr between 2004 and 2014, consisting of 99.2 million photos and 0.8 million videos. Each entry of the YFCC100M dataset contains the user ID, the date when the photo was taken, and up to 24 optional fields among them the most relevant for our work are the geo-location as GPS coordinates, and the users’ tags. The YFCC dataset comes with 3 kinds of supplementary information: attributes related to the camera that took the picture such as the photo definition, or the camera brand; auto-generated concepts (*e.g.*, people, animal, food, outdoor) obtained by an unsupervised image processing approach, and geographic attributes (*e.g.*, street, town, country) associated with the GPS coordinates of a photo. Photos taken with digital devices usually carry descriptive information, called metadata. This information often appears as the exchangeable image contained in the JPG photos. Metadata can be parsed and saved when the photo is uploaded to a website.

2.2 Geonames Dataset

The YFCC dataset does not contain detailed information about the POI categories, which is what we aim to add by matching the photo locations with the POI categories included in the Geonames dataset [2].

The Geonames is a geographical database that covers all countries and contains over eleven million unique features (see Fig. 1) whereof 4.8 million populated places and 13 million alternate names. These place names correspond to Point of Interest (POI). All features are categorized into one out of nine feature classes and further sub categorized into one out of 645.

The usefulness of the dataset of Geonames with respect to the YFCC100M is that their crossing will allow to find the points of interest associated with each photo, or those which are closest. But most importantly, it gives us the ability to access the POI categories; rarest data and most difficult to obtain.

2.3 Spatial Queries and Quadtree

When crossing these two large datasets, we need to process spatial queries that require specific methods.

A spatial query is a set of spatial conditions characterized by spatial operators that form the basis for the retrieval of spatial information from a spatial database system. Moreover, we express some combination (longitude and latitude) for extracting specific information from the datasets without actually changing these data. Considering this, we aim dealing with spatial queries.

To efficiently process spatial queries, we will rely on indexes. Spatial indices are used by spatial databases to optimize spatial queries by accessing a spatial object efficiently. Conventional index types do not efficiently handle spatial queries such as how far two points differ, or whether points fall within a spatial area of interest. Without indexing, any search for a feature would require a sequential scan of every record in the data, resulting in much longer processing time. In a spatial index construction process, the minimum bounding rectangle serves as an object approximation [16]. Many common spatial index methods exist [10,16]. In our work, we choose the Quadtree [16].

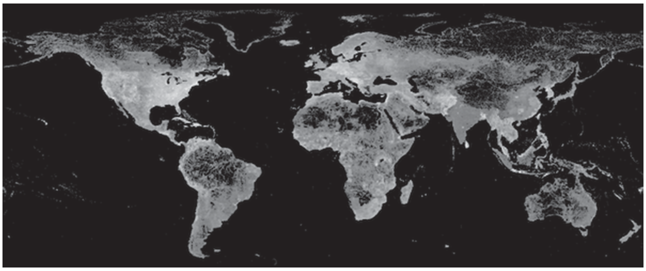

Fig. 1. GeoNames feature density map (source: www.geonames.org)

3 Related Work

The YFCC Dataset release opened many possibilities in data-driven research fields. It gave new opportunities exploiting geolocalized data and crossed with identified existing POIs for recommendation [4,6,7,9,17].

We motivate our work by reminding that recent works on POI recommendation using YFCC dataset enriched with POI categories is restricted to small scale data:

[7] considers 250 POIs located in 8 towns, and 153 K visits (*i.e.*, posts). [6] considers 118 POI in one city, and 332 K visits. This is 10 to 100 order or magnitude less than the dataset scale we are considering.

In [7] authors present PERSTOUR, an algorithm for recommending personalized tours. This recommending uses POIs and existing real-life travel sequences to align with users interests. They formulate the problem using orienteeriing problem and consider two mains constraints: time budget and specific starting and ending POIs. The authors validate their proposition using the YFCC100M dataset and show promising results. They enhanced the PERSTOUR algorithm in [8] by updating user interests through visit recency and visit duration of their POIs. The authors propose other extensions. In [6] they propose an other enhanced personalized itinerary recommendation that considers queuing times at recommended POIs, while in [15] they propose a Personalized Crowd-aware Trip recommendation.

The YFCC data is also used in many other case studies. In [1], the authors used it in a context of photo recommendation for posters of touristic destination promotions. The authors highlight the dual character of information about photos and videos posted on social networks. On the one hand there is the metadata which is the set of information that describes the media; for example a title, some tags, a description. On the other hand, there is all comments published by social network users. These comments have the particularity of containing explicit emotions on the media, but also sometimes tags. Based on this observation, the authors propose following two concepts: This duality in the characterization of photos and videos is related to their metadata and their comments. Authors of this contribution propose a model between these two concepts, using a naive Bayesian classifier, which is a machine learning algorithm. The purpose of this model is to help the management agents of the tourist destination promotion to choose the most suitable photo for the promotion of a destination. For their experimental validation, the authors used a final dataset containing 21 K posts in New York city; which is a quite small scale.

Many other works focusing on POIs for touristic or travel recommendation are available. But they use other datasets [12,19] in their experimental validation. While these datasets are different from YFCC, they still are at relatively small scale.

4 Enriching a Photo Dataset with POI Categories

Consider the simplified schemas for the two datasets to be joined: **YFCC***(user, date, photoID, latitude, longitude)*, and **GN***(POI, latitude, longitude, categories)*. We aim to process the query J that associates a photo y with a POI o if the distance d between y and o is less than a given upper bound b. We have:

$$J = \{(y, o) | y \in YFCC, o \in GN \wedge d(y, o) \leq b\} \tag{1}$$

We investigate three methods to process J in a distributed environment such as Apache Spark.

The *Cartesian* Method. The first baseline method named *Cartesian* is to translate J in SQL and submit it to the Spark framework which is able to process J in parallel. This requires to define d as a user defined function, to compute the cartesian product between $YFCC$ and GN then to apply d and select the (y, o) couples that satisfy the condition. This method has a high complexity: the number of d invocations is $|YFCC| \times |GN|$, which is the order of 10^{14} in our case.

The *IndexJoin* Method. The second method named *IndexJoin* relies on the GeoSpark [18] additional library to compute the spatial join. Note that GeoSpark is a dedicated library for processing large-scale spatial data in a cluster computing system. GeoSpark extends Apache Spark/SparkSQL with a set of out-of-the-box Spatial Resilient Distributed Datasets (SRDDs)/ SpatialSQL that efficiently load, process, and analyze large-scale spatial data across machines. The spatial IndexJoin is computed in a parallel and distributed way as follows:

- The GN dataset is partitioned based on the GPS locations of POIs, using a quad-tree like approach. Each partition boundary covers a specific rectangular area of the world. The area assigned to a partition has a lower bound that depends on the distance b of the join operation to process. Therefore even in dense areas with many POIs, the smallest partition area still has edges of minimal size around $2 * b$. As a consequence, some partitions are rather unbalanced in terms of number of POIs they contain.

- The $YFCC$ dataset is partitioned using the partitioning that has been defined for the GN dataset. This means that $YFCC$ has the same partitions as GN in terms of geometry.
- The two spatial-partitioned datasets are distributed over the machines such that each $YFCC$ partition is located on the same machine as the corresponding GN partitions that contains POIs in the same area or close areas (less than b meters).
- Each machine computes the $YFCC/GN$ join a distinct set of $YFCC$ partitions.

We empirically observed that the J spatial join processing had the expected degree of parallelism (*i.e.*, 200 cores are fully utilized) during the processing of 995 partitions out of 1000. However, processing the last 5 partitions raised important performance degradation as it was highly unbalanced: most of the query time was spent in processing those last 5 partitions, using only 5 CPU cores out of 200 available ones. Therefore the *indexJoin* method suffers from an unbalanced partitioning which is due to data bias. Indeed, few famous small areas (having an edge size lesser than b) actually contain most of the POIs and photos. The vast majority of other areas have very few POIs and photos.

We solve this major performance issue by proposing the next method.

The *Bias-aware* Method. This method aims to balance a join workload in the case of high data skew on the join attribute values. It relies on the idea to partition the dataset according to the expected result size within a partition. This differs from the previous method which partitions the data according to the partition size. It enables to handle data skew in an efficient way by ensuring that the number of (y, o) results is rather uniform among the partitions.

We divide the photo dataset into two parts: a bias-free part denoted Y_{free} and a biased part denoted Y_{bias}. We defined the biased areas as the top-N world most visited cities denoted C. Notice that in the following section, we empirically tune N. We use the city information contained in the dataset to put into Y_{bias} the photos located in C; the remaining part of the dataset is put in Y_{free}. We process Y_{free} using the above *IndexJoin* method. Then to process Y_{bias} efficiently, we re-distribute it into 1000 partitions and process it using the *Cartesian* method.

5 Experiments

5.1 Experimental Set Up

We implemented the above described join methods and ran them on the Spark cluster of the LIP6 Lab, which consists of 1 driver machine and 10 worker machines. Each worker machine has 20 CPU cores (Intel Xeon processors with hyperthreading) and 50 GB memory, totaling 200 cores and 500 GB memory in use for join processing.

We had to clean the YFCC dataset. We only kept entries with valid GPS coordinates and date.

5.2 Join Performance with Spatial Index

We first run the YFCC/Geonames join on the entire dataset. It lasts 2 h. Then we study the impact of most visited cities on the performance. We successively removed from the YFCC dataset the top-N most visited cities. We report the response time on Fig. 2, the x-axis gives the number of cities that have been skipped, it has a log scale. For example, we see that when we removed 20 cities the response time is 322 s. It decreases below 50 s when more than 200 most visited cities have been removed. The index should only be used for the remaining cities (approx 32 800). We explain the poor performance for most visited cities as follows. When using the spatial index to join YFCC with Geonames location on the condition that they are distant of less than 500 m, the spatial partitioning keeps some dense areas (with a radius about 500 m). The join computation within each area is sequential which explains the high response time.

Therefore, the most visited cities must be treated apart, as detailed in next section.

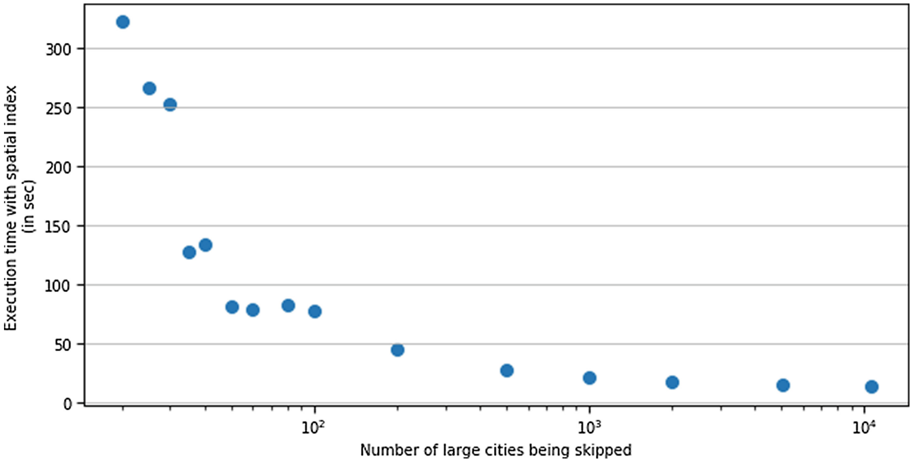

Fig. 2. Execution time for YFCC/Geonames join with spatial index. Log scale

5.3 Join Performance Without Index for the Most Visited Cities

We now target on computing the YFCC/Geoname join for the 200 most visited cities. Indeed, we observe that cities with a lot of visits also have the highest number of POIs.

We run the YFCC/Geonames join for each city with many visits using the cartesian method explained in Sect. 4. We report the response time on Fig. 3. For readability we only show the top-80 most visited cities, the response time being less than 0.5 s for the remaining cities. We observe that even for New-York, the most visited city, the response time is rather low (11.8 s), thanks to the high degree of parallelism. The total response time for processing the 200 most visited cities is less than 125 s. This means that we can proceed the full set of 33000 cities in less than 175 s.

In comparison, we also measured the performance using a single join method. The response time to join the entire dataset is 2 h (respectively 157 h) for the spatial index method (resp. the cartesian join method). Thus our *Bias-aware* method is at least 40 times more efficient than when using a single join method.

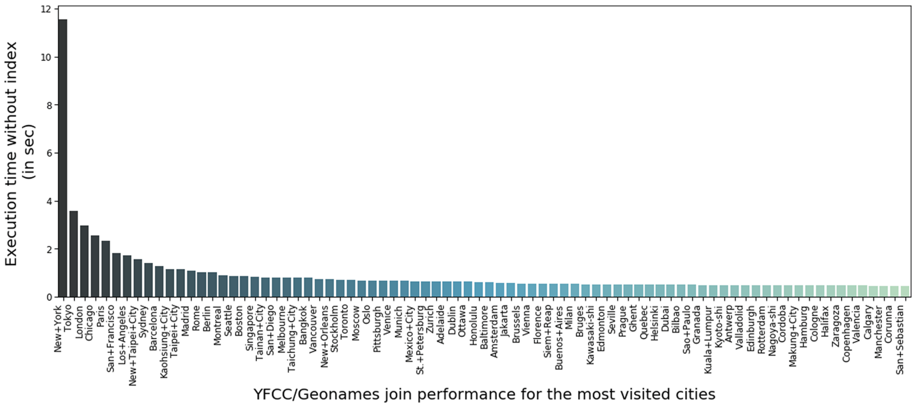

Fig. 3. YFCC/Geonames join execution time without index, for the top 80 cities with the highest number of visited locations. Cities are ordered by decreasing join time

6 Conclusion and Future Works

Considering the lack of publicly available rich and large-scale dataset for POI recommendation research, we have proposed an efficient method to join YFCC with Geonames datasets. We propose a efficient approximate spatial join method that performs on top of Apache Spark and takes into account the important bias on the spatial distribution of locations to better distribute the join workload on a cluster of machines.

The main use we will make of this enriched dataset is to extract more accurately users' information and define a new method for POI recommendation that is expected to leverage on rich and world-wide travel information to improve the recommendation quality. To achieve this goal, we are improving our POI selection method. Actually, We are facing the problem of choosing, for a given photo y a small set of best matches among a possible large set of (y, o) resulting from the spatial join described in Sect. 4. The d distance function might not help here since every o candidate is close to y (by definition of J we have $d(y, o) \leq b$). Therefore, we are investigating a method to select the best matches among a set of close candidate POIs, by comparing the photo tags with the POI description.

References

1. Deng, N., Li, X.R.: Feeling a destination through the "right" photos: a machine learning model for dmos' photo selection. Tour. Manag. **65**, 267–278 (2018)
2. Geonames: The geonames dataset. http://www.geonames.org/export. Accessed 26 Nov 2019
3. Griesner, J., Abdessalem, T., Naacke, H., Dosne, P.: Algeospf: a hierarchical factorization model for POI recommendation. In: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM, pp. 87–90 (2018)
4. Griesner, P.-B.: Scalable models for Points-Of-Interest recommender systems. Ph.D thesis, Telecom ParisTech, Paris, tel-02085091, 7 2018. Artificial Intel-ligence [cs.AI] (2018)
5. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Valencia, Spain, April 2017, pp. 427–431. Association for Computational Linguistics (2017)
6. Lim, K.H., Chan, J., Karunasekera, S., Leckie, C.: Personalized itinerary recommendation with queuing time awareness. In: ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 325–334 (2017)
7. Lim, K.H., Chan, J., Leckie, C., Karunasekera, S.: Personalized tour recommendation based on user interests and points of interest visit durations. In: International Joint Conference on Artificial Intelligence, IJCAI, pp. 1778–1784 (2015)
8. Lim, K.H., Chan, J., Leckie, C., Karunasekera, S.: Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. Knowl. Inf. Syst. **54**(2), 375–406 (2017). https://doi.org/10.1007/s10115-017-1056-y
9. Liu Shudong, G.V.L.J.: User modeling for point-of-interest recommendations in location-based social networks: the state of the art. Mob. Inf. Syst. (2018)
10. Manolopoulos, Y., Theodoridis, Y., Tsotras, L., Vassilis, J.: Spatial indexing techniques. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems, pp. 2702–2707. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-39940-9
11. Ni, K., et al.: Large-scale deep learning on the YFCC100M dataset. CoRR, abs/1502.03409 (2015)
12. Tang, L., Cai, D., Duan, Z., Ma, J., Han, M., Wang, H.: Discovering travel community for poi recommendation on location-based social networks. Complexity, 2019:8503962:1–8503962:8 (2019)
13. Taylor, K., Lim, K.H., Chan, J.: Travel itinerary recommendations with must-see points-of-interest. In: Companion Proceedings of the The Web Conference 2018, WWW 2018. International World Wide Web Conferences Steering Committee, pp. 1198–1205 (2018)
14. Thomee, B., et al.: Yfcc100m: the new data in multimedia research. Commun. ACM **59**(2), 64–73 (2016)
15. Wang, X., Leckie, C., Chan, J., Kwan Hui, L., Vaithianathan, T.: Improving personalized trip recommendation to avoid crowds using pedestrian sensor data. In: Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM 2016), pp. 25–34 (2016)
16. Xiaoyi Zhang, Z.D.: Spatial index. Geographic Information Science and Technology Body of Knowledge (2017)
17. Yonghong Yu, X.C.: A survey of point-of-interest recommendation in location-based social networks. In: Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence. AAAI (2015)

18. Yu, J., Wu, J., Sarwat, M.: Geospark: a cluster computing framework for processing large-scale spatial data. In: SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages 70:1–70:4 (2015)
19. Zhao, S., Zhao, T., Yang, H., Lyu, M.R., King, I.: Stellar: spatial-temporal latent ranking for successive point-of-interest recommendation. In: AAAI 2016: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. AAAI Press (2016)