Lookupia : An Intelligent Real Estate Search Engine for Finding Houses Optimally Geolocated to Reach Points of Interest

Jonathan Milot[1,3], Patrick Munroe[1,3], Éric Beaudry[1,3], François Grondin[2,3], Guillaume Bourdeau[3]
[1]Département d'informatique, Université du Québec à Montréal
[2]Département de génie électrique et de génie informatique, Université de Sherbrooke
[3]Lookupia
{firstname.lastname}@lookupia.com

ABSTRACT

Lookupia is an intelligent real estate search engine for finding houses optimally geolocated to reach points of interest. It uses data from OpenStreetMap and most of public transit corporations that publish transit schedules in the General Transit Feed Specification (GTFS) format.

Keywords

Geographic Location Optimization; Search Engine; OpenStreetMap; Public Transit Networks; Real Estate; Web Application

1. INTRODUCTION

Looking for a new house might be a daunting task. A plethora of factors have to be taken into account. The location, the lot size, the number of rooms, the price, etc., are generally considered by most future buyers before taking their final decision. With the advent of internet and new technologies, a substantial part of this burden has been alleviated. Nowadays, many websites offer the possibility of listing real estate properties according to the characteristics mentioned above, and even more. However, the techniques involved in those search engines are generally limited to filtering a collection of houses through some fixed criteria (e.g., a minimum and a maximum price) and sorting the resulting list of real estate properties according to the preferences of the user.

Lookupia[1] is a new search engine which goes beyond this classic scheme by putting points of interest at the forefront. A *point of interest (POI)* can be seen as a specific geographical location that a user would be interested in visiting more or less frequently. For example, one's workplace, the nearest elementary school, the local public library, etc., could all be POIs for a future buyer and his family. The system Lookupia provides a simple way to determine optimally the location of someone's future property depending on their POIs.

[1]Website: `https://www.lookupia.com`

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2890485.

Tools that allow their users to find the shortest path between two geographical locations on a map (e.g., Google Maps) exist since already many years. However, to the authors' knowledge, Lookupia is the first computer software that generates a list of real estate properties optimally geolocated based on POIs. Even though Lookupia relies essentially on algorithms to find the shortest path between two points on a map, the system is more complex than it might appear at first. Considering the very large amount of paths that one has to produce in order to determine the optimally located houses, it would be too expensive, both moneywise and timewise, to use already existing APIs such as the Google Maps APIs.

This paper is dedicated to presenting the search engine Lookupia. It begins with a description of how the system works in general. Then, the graphical user interface is presented to show how one would use the website to obtain a list of properties optimally geolocated with respect to their POIs. Finally, a benchmark of the search engine performance is briefly discussed.

2. SYSTEM ARCHITECTURE

Figure 1 shows a diagram explaining the system architecture. In order to produce a list of optimally geolocated houses, Lookupia makes extensive use of open data from OpenStreetMap (OSM) [4] and from the public transit corporations that publish their transit schedules in the General Transit Feed Specification (GTFS) format [3]. The data from OpenStreetMap are essentially used to generate the map, whereas the transit data in the GTFS format allow the search engine to take public transportation into consideration when it determines the optimal location to reach points of interest. In addition to these open data, the system has to be fed with property listings. These data can come, for example, from real estate agencies or independent brokers and, for this reason, are not in general open data.

The core of the search engine is located in the "Back End" part of the system which has a REST architecture [2]. The algorithm used to find the optimally geolocated houses strongly relies on an adapted version of the famous Dijkstra's algorithm [1] for finding the shortest paths between two nodes in a graph. In generating the results, the algorithm takes into account a list of criteria (e.g., minimum and maximum price, minimal living area, number of bathrooms,

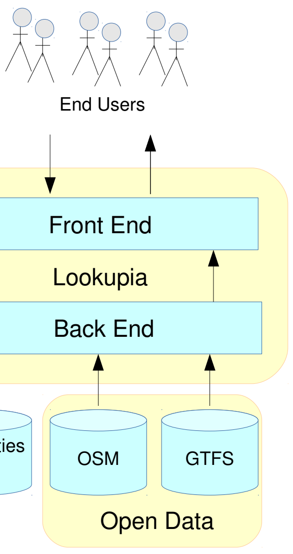


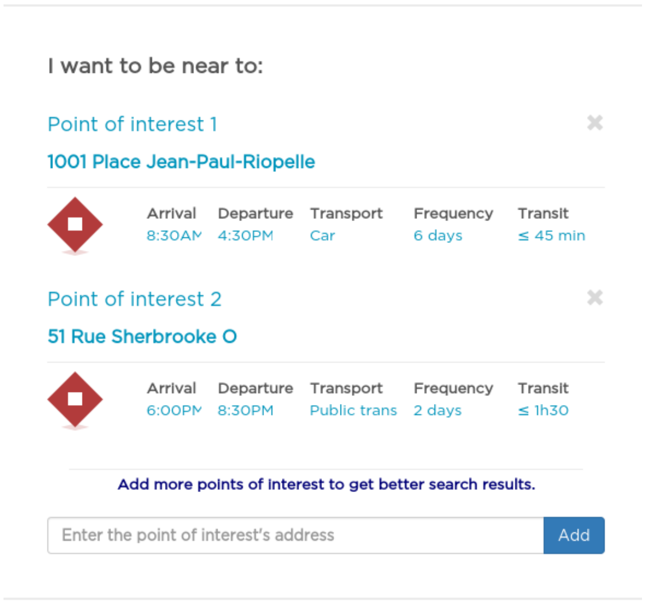
Figure 1: System Architecture

etc.) as well as the possibility to reach each POI on foot, by car, or using the public transit network.

Finally, the end user, e.g., the future buyer of a house, makes queries and see the search results on the webpage, which corresponds to the "Front End" part of the diagram in Figure 1. The procedure followed by the user to obtain the search results is described in the next section.

3. USER INTERFACE

The graphical user interface of Lookupia's website is shown in Figures 2 and 3. Every time a future buyer would do a search, they would go through the following three simple steps:


I want to be near to:

Point of interest 1

1001 Place Jean-Paul-Riopelle

Arrival	Departure	Transport	Frequency	Transit
8:30AM	4:30PM	Car	6 days	≤ 45 min

Point of interest 2

51 Rue Sherbrooke O

Arrival	Departure	Transport	Frequency	Transit
6:00PM	8:30PM	Public trans	2 days	≤ 1h30

Add more points of interest to get better search results.

Figure 2: Points of Interest Form

1. In the search criteria panel accessible through the vertical menu bar, enter the criteria which should be taken into consideration by the search engine. These criteria includes a minimal and a maximal price, the general location, (e.g., the city, the region, the province, etc.) the minimal living area, the number of bedrooms, etc.

2. Place different POIs (one's workplace, the school of their kids, their gym, etc.) on the map by clicking on "Specify your points of interest" in the vertical menu bar or by simply entering the physical address of the POIs. For each POI, specify the time of arrival, the time of departure, the frequency (i.e., the number of days per week that a POI is visited), the mode of transport (walking, by car, or by public transport), and the maximum transit time that one is willing to do (see Figure 2).

3. Select the properties once the algorithm returns the list of optimally geolocated houses. The real estate properties appear both on the map and in a list in the right panel (see Figure 3). This list is sorted according to a score weighted by the frequency each POI is visited. For example, a POI that is visited five times a week is considered to be more important than a POI that is only visited twice a week. The properties can be selected by clicking on them on the map or by clicking on them in the right panel. This shows a new panel containing pictures and a description of the house. Moreover, the shortest paths (under the transport constrained entered by the user in the previous step) from the property to each of the different POIs appear on the map.

4. BENCHMARK

In order to have an idea of the performance of the algorithm, a benchmark of the processing time of the search engine for different numbers of queries was done. The experiment was carried out on a computer equipped with a processor Intel Core i7-4790 at 3.60GHz and with 32GB of RAM on Ubuntu Linux 15.10. The results are given in Table 1. Note that the time is not CPU time, but real time. The benchmark consisted essentially in querying sequentially the search engine using batches of randomly generated points of interest. Moreover, it was done with a list of 1000 properties concentrated in the region of Montreal. The two columns identified with "3 Batches" show what happens when the engine is queried simultaneously by three different parallel batches containing the same number of queries.

Number of Queries per Batch	Processing Time (s)			
	1 POI		3 POIs	
	1 Batch	3 Batches	1 Batch	3 Batches
10	4.685	8.474	9.258	25.015
100	41.949	85.552	103.858	237.733
1000	449.691	846.195	996.564	2425.267

Table 1: The processing time of the search engine with respect to the number of queries and the number of POIs.

It stands out from this table that, as one would expect, adding POIs to the search increases greatly the processing time. With only one POI, the search engine can process sequentially about 133 queries/minute, whereas with three POIs, this rate decreases to 60 queries/minutes. On the

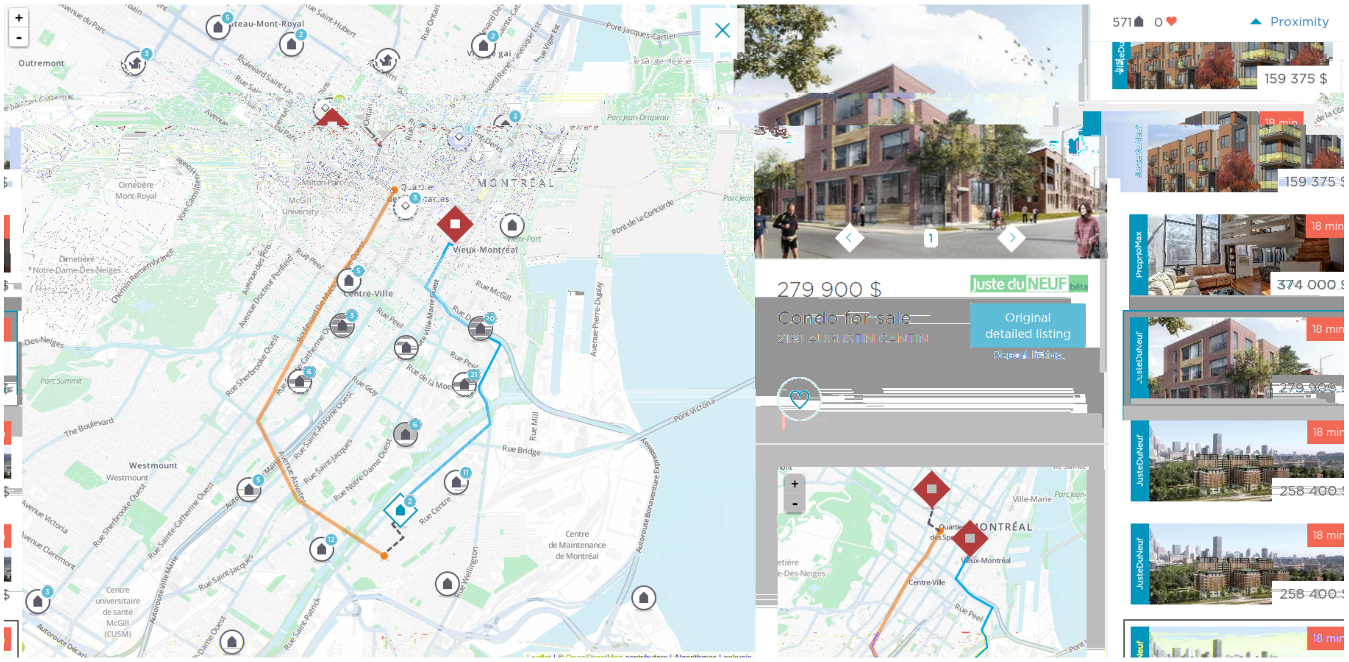


Figure 3: The map with search results.

other hand, when queries are sent at the same time, the multithreaded architecture of the system allows an important gain of time compared to what would happen if the queries were treated sequentially.

5. CONCLUSION

Lookupia is a powerful tool to find real estate properties that are optimally geolocated to reach POIs. For now, the search engine only supports fixed POIs, i.e., the POI is tied to a fixed geographical coordinate. In a near future, Lookupia is also expected to support mobile POIs, i.e., POIs which are not necessarily associated with a specific coordinate, but rather with a type of amenity. For example, a buyer might want her future house to be located close to a public swimming pool, an elementary school, a grocery store, a drugstore, etc., even though the absolute geographical location of those POIs does not really matter.

References

[1] E. W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1(1):269–271, December 1959.

[2] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[3] Google. What is GTFS? `https://developers.google.com/transit/gtfs/`. (accessed January 14, 2016).

[4] M. Haklay and P. Weber. OpenStreetMap: User-Generated Street Maps. *Pervasive Computing, IEEE*, 7(4):12–18, October-December 2008.