Studying public transit API query logs to get an indication of travel flows

Pieter Colpaert
pieter.colpaert@ugent.be

Alvin Chua
alvin.chua@asro.kuleuven.be

Ruben Verborgh
ruben.verborg@ugent.be

Erik Mannens
erik.mannens@ugent.be

Rik Van de Walle
rik.vandewalle@ugent.be

Andrew Vande Moere
andrew.vandemoere
@asro.kuleuven.be

ABSTRACT

In the field of urban planning, researchers need an indication of how people move between cities. Yet, getting statistics of travel flows within public transit systems has proven to be troublesome. We analyzed the query logs of the iRail API, a high expressive route planning API for the Belgian railways, to get an indication of travel flows between cities in Belgium. We were able to study ∼100k to 500k requests for each month between October 2012 and November 2015, which is between 0.5% and 1.7% of the amount of monthly passengers. Using data visualizations, we illustrated the commuting patterns in Belgium and show that Brussels, the capital, acts as a central hub. The Flemish region appears to be polycentric, while in the Walloon region, everything converges on Brussels. The findings correspond to the real travel demand, say experts of the passenger federation Trein Tram Bus. We conclude that query logs of route planners are of high importance in getting an indication of the travel flows. High expressive transport data publishing methods such as route planning API exist, as well as low expressive data dumps or data fragments. In order to be able to gather meaningful logs in all cases, we suggest using a separate POST request containing the entire query.

1. INTRODUCTION

{TODO: Rework introduction at the end}

Getting indications of people flows through public transit networks is a challenge. The data is tedious to get, mainly with public transit systems where passengers don't have to check-in and check-out. Nevertheless, people are calculating their intended routes by using the Web. *Can we get an indication of these transit flows by studying the query logs of Web-services?*

The *iRail* project[1] started in 2008 to make the data of the Belgian railway accessible for developers. Ever since, the project offers developer both a GTFS data dump for third party apps and a route planning API. The query logs of this API has been stored since 2013. We study these query logs by creating a couple of visualizations which illustrate a couple of patterns. Two of the documented patterns in this paper correspond to reality, another does not.

In this paper, we give a small overview of related work to gather data for flow analysis. Next, we study the iRail query logs to find out whether we can find interesting patterns. Finally, we look at Linked Connections [**?**], a proposed new way of publishing queryable public transit data, and whether we would still be able to have an indication of the transit flows with a Linked Data Fragments [**?**] approach.

2. RELATED WORK

{TODO: Write relwork at the end}

Flow analysis is a topic of theoretical interest and practical importance in various disciplines. Flow analysis is conventionally conducted to study spatial dynamics and identify routine patterns in the movement of people. For instance, interest in modelling traffic flows emerged from the strain placed on urban transportation systems during peak hours [**?**, **?**]. Likewise, insight into routine travel patterns is crucial for the conceptualisation of functional urban areas [**?**, **?**], urban hierarchies [**?**] and other territorial structures.

Over the past decade, large datasets have become increasingly commonplace due to the proliferation of sensor networks and portable devices like smartphones. Termed "Big Data" due to the large volume of data records that emerge from real-time sensing[**?**], such datasets typically contain information of activities or processes linked to the space and time where they occur. In the domain of "Smart City" research, much has been accomplished with the use of "Big Data" to monitor human movement. Smart card data from public transport systems [**?**, **?**], taxi journeys [**?**] as well as cellular call data [**?**] have provided planners with new opportunities to develop greater understanding of mobility patterns in urban environments [**?**].

3. THE LOGS

The iRail API is a XML/JSON HTTP API in which one request on *http://api.irail.be* will result in a response that can be used directly in an end-user app. It contains four features: a route planning query (*http://api.irail.be/connections/{?from,to,date,time}*), a query for the next departures in a certain station (*http://api.irail.be/liveboard/{?station,date,time}*), a query for the status of a certain train (*http://api.irail.be/vehicle/{?id,date}*) and a query for a list of all stations (*http://api.irail.be/stations/*). We have gathered the *Apache access.log* files

- the timestamp,
- whether or not the request succeeded,

[1] *http://hello.irail.be*

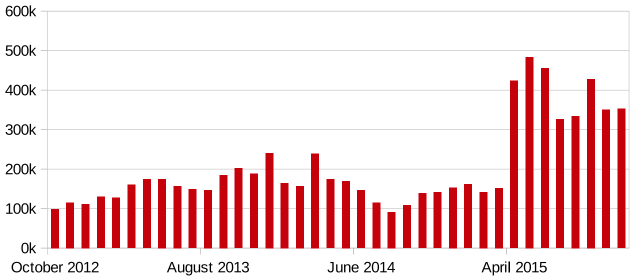

Figure 1: Amount of queries on the route planning part of the iRail API between 2012 and 2016. In April 2015 an official app of the Belgian railways was discontinued, which explains the sudden raise of iRail API queries.

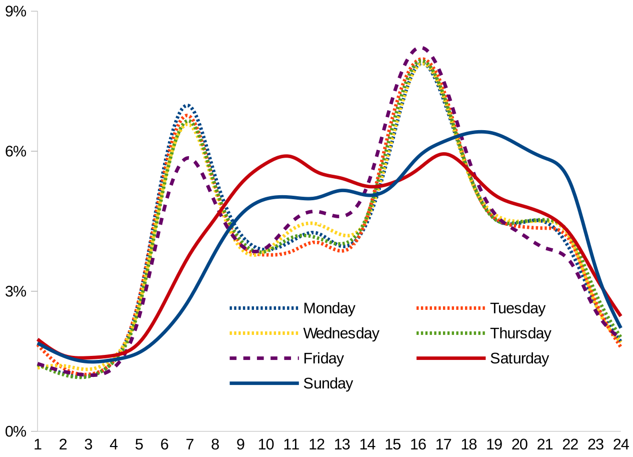

Figure 2: Distribution of iRail route planning API queries on average per day of the week between 2012 and 2016

- the path of the query (e.g., */connections/?from=Ghent&to=Antwerp*) and
- the User-Agent (cfr. RFC2616[2])

From an export of this table, we filtered out only the route plannning queries that succeeded for the entire year. The resulting dataset, used for the rest of this paper, can be downloaded at **{TODO**: *http://datawijs.be/apilog.tar.gz***}**.

As we are dealing with query logs, privacy becomes an issue [**?**]. However, we do not publish the IP addresses with each query, which makes it hard to track single users. To remove all doubt, we have asked and received the consent of the privacy commission of Belgium, which declared we can study the logs without any restrictions.

4. METHOD

4.1 Processing

We cleaned the logs by removing the requests done by harvesters or search engine bots. Thanks to the user agents, this was a fairly straight-forward operation. **{TODO**: The remainder of the file delivers an average of 3000 route planning queries a day, which we can use an an indication of travel intensities. Taking into account that 758,370 people take the train each day on a weekday (according to official number of the Belgian Railway company in 2013[3]) and assuming that each query is a trip done by a certain passenger, this sample equals $\sim$ 0.4% of the railway users in Belgium.**}**[4]

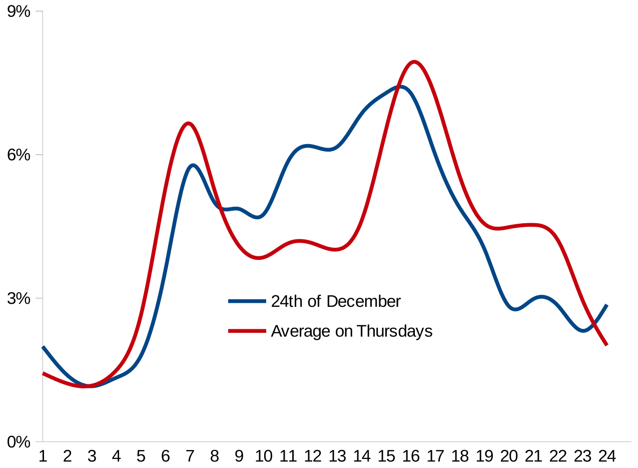

Figure 3: Comparison of route planning queries on December 24th (Christmass eve) 2015 with the average on a Thursday between 2012 and 2016. The illustration shows that people started going home earlier than usual.

Each data record contains the names of both origin and destination (OD) stations, the user agent that made the query, as well as the time when the query was made. As we were specifically interested in studying morning and evening rush hour travel that occur on weekdays, we filtered the dataset to exclude data records created outside the time range of 06:00 hrs and 10:00 hrs as well as 17:00 hrs and 21:00 hrs on weekdays, and all data records created on weekend. Queries made by automated user agents like search engine bots and data harvesters were also excluded from the dataset to remove biases that may over represent connections between certain stations. Similarly, data records with unparsable origin or destination station names due to spelling mistakes or invalid character encoding were removed as well.

The data is then simplified so that province level flow patterns are emphasised. **??** provides a diagrammatic representation of the procedure. First, both origin and destination stations for each data record are spatially aggregated based on provincial administrative boundaries (See Figure xa). If an origin or destination station is a located outside of Belgium, it is considered international travel and aggregated in a distinct group (See Figure xb). Train stations in major cities are excluded from aggregation so that the volume of flow between provinces and major cities are comparable (See Figure xc). The following types of flows are observable from the outcome of aggregation:

- Travel from any provincial station to a major city.
- Travel between any two major cities.

[2] The User-Agent request-header field contains information about the user agent originating the request. This is for statistical purposes, the tracing of protocol violations, and automated recognition of user agents for the sake of tailoring responses to avoid particular user agent limitations – *http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html*

[3] *http://www.treintrambus.be/images/OpstappendeReizigers_2013_2.pdf*

[4] Calculated by using indicators of the Flemish government and extrapolating the numbers for 2015 *http://www4.vlaanderen.be/sites/svr/Cijfers/Exceltabellen/mobiliteit/vervoersprestaties/personenvervoer/MOBIOPEN006.xls*

- Travel between any two provincial stations.
- Travel between any international station to a major city.
- Travel between any international station to a provincial station.

4.2 Visualization

{TODO: Alvin**}**

Visualisation is frequently used to make data analysis tangible, so that the results can be communicated and debated Robinson, 2008 121. This is crucial for our research since the questions presented are exploratory in nature Kraak, 2008 150 and can be addressed in many ways. Movement data has received substantial attention from the visualisation community and a range of techniques has been developed to support analysis Andrienko, 2012 196. Of these techniques, flow visualisations facilitate the comparison of aggregated movement over space and time. There are three types of flow visualisations: Line based representations preserves the complete trajectory while matrix type representations take only the start and end locations into account.

Line based representations indicate movement on a path with the shape of the line. The thickness of each line is scaled to the volume of flow along a path. The direction of flow is commonly indicated with an arrowhead. This representation is detailed and straightforward to understand but becomes cluttered when lines intersect. A number of solutions have been proposed for clutter reduction. Filtering is the most commonly practiced. The same goal can be achieved by clustering lines that have similar properties like common start, end or intermediate locations Andrienko, 2007 198. Hierarchical Guo, 2009 199 and density based Rinzivillo, 2008 201 clustering have also been introduced for this purpose. Both solutions are effective at reducing clutter yet it should be noted that excessive aggregation or filtering removes a significant amount of information from the visualisation. Another solution to reduce clutter is to transform the visual representation. This is achieved by rerouting Phan, 2005 88 or bundling Hurter, 2014 200 lines in close proximity so that they appear grouped. Unlike clustering, this solution changes the shape of lines by way of interpolation to obtain an ideal layout. In most cases, the interpolation is regulated by conditions to avoid intersections and follow key geographic features such as roads, rivers or coastlines so that the layout appears natural. This solution emphasises major flows within the visualisation since large groups gain visual dominance. Nonetheless, there are two drawbacks to this solution. First, short distance flows that may be important at local scales are de-emphasised or lost in the transformation. Next, the lines do not depict actual paths of movement thus this solution may not be suitable for applications that require strict geographic accuracy.

Matrix type representations are frequently used when the origin and destination preside over intermediate locations on the path of movement. This type of representation is referred to as origin-destination (OD) matrix. Rows and columns correspond to locations while cells are coloured to express the volume of flow. OD matrices can be reordered to emphasise connectivity between locations but the lack of geographic context is a distinct disadvantage. Several solutions have been developed to address this shortcoming. These are based on the notion of small multiples: The matrix is arranged in a geographical order so that the cells correspond to locations on the map. Then, a smaller map of the geographical context is nested in each cell to enable more intuitive comparisons Guo, 2006 202. The limitation of this solution is similar to other forms or small multiples in general - the layout space allocated to each nested map becomes smaller as the number of cells increase. In this instance, nesting detailed geographical maps is problematic since majority of the geographical features become illegible due to scaling. OD Map Wood, 2010 86 proposes to nests a series of geographically ordered matrices that are more expressive of spatial relations and remain legible even when a small amount of layout space allocated to each cell.

5. RESULTS

In this chapter, we report on the results of studying visualizations with experts of Trein Tram Bus, a not for profit passenger federation.

5.1 Commuting pattern

We see people leaving in the morning and return in the evening (see **??** and todo)

The pair of chord diagrams in **????** capture the aggregated number of queries made between any two stations on weekdays and weekends between 6 to 9 in the morning. The chords that link two distinct stations, are coloured coded by region and proportionately scaled to the number of queries from a station of origin to a destination. For instance, approximately 3,530 queries were made from Leuven to Brussels than in the opposite direction (approximately 1,530) on weekdays. Reading both diagrams in this manner reveals the complexity of movement on the rail system and the significance of cities in daily travel. Brussels serves as the principal centre of rail activity in general, yet its centrality is more distinctive for the walloon region than Flanders. Queries from Walloonia with exception to Liege, are generally made from provincial stations towards Brussels instead of their respective provincial capitals. Queries in Flanders, on the hand, tend to be distributed among major cities otherwise known as the Flemish Diamond, a network of four metropolitan areas in Belgium comprised of Ghent, Brussels, Antwerp and Leuven. The difference between both patterns appear to correspond with existing measures of population density, providing valuable insight and alternative perspectives into the function of cities in rural and urban settings.

5.2 Flanders is polycentric, Wallonia monocentric

We can also observe the effect of "density" in Flanders where people move from the rural towards cities. Look at weekday-mornings: commuting towards Gent, Brussels, Antwerp. This is not as obvious in the Walloon region (except for Liege). The reverse occurs on weekday-evenings. Local patterns can also be observed and this corresponds to what we know: “cities are the central hubs”. Walloon region: everything converges on Brussels

5.3 Inexplicable results

We also see weird things happening in Antwerp, where a large amount of people go to the province in the morning on weekdays, while the oposite happens in Liege. When looking into the data, the Antwerp pattern appears due to one specific route which is queried each morning. Is there a group of colleagues all using the same app taking the same train daily? Is it someone who checks this trip over and over again for delays? We can only guess. . .

6. PUBLISHING TRANSPORT DATA

The expressiveness of a server affects the way logs can be gathered. We have shown in previous sections that the query logs of a *high expressive server*, such as the iRail API, are interesting: each request contains an entire query which can be interpreted as a travel intention. The possibility that two requests are exactly the same is low, as there are many URLs which can be requested. Nevertheless, we cannot fully guarantee that each HTTP GET request to the server

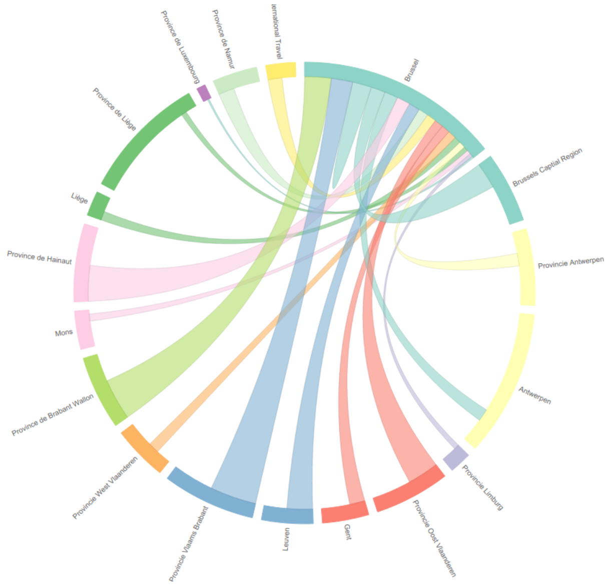

Figure 4: Pair of chord visualization of the city of Brussels: transit flows towards Brussels from elsewhere in Belgium and vice versa.

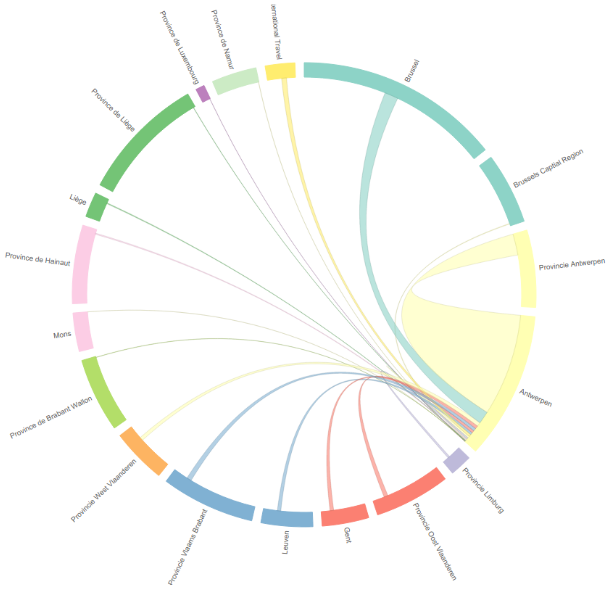

Figure 5: Pair of chord visualization of the city of Antwerp: transit flows towards Antwerp from elsewhere in Belgium and vice versa.

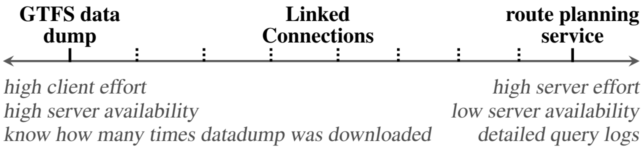

Figure 6: This axis, first introduced by Linked Data Fragments [?], illustrates the possibilities within publishing transport data. Different ticks on the axis illustrate client effort versus server expressivity: on the far left, data dumps offer high server availability, yet the effort needed by data reusers is high, and query logs do not reach the server. Moving to the right, we identify Linked Connections [?] as an in-between solution. On the far right, route planning services require high server effort, leading to detailed query logs.

will trigger a log entry, due to caching mechanisms on the Web [**?**], which might lead to a false representation.

Hosting a route planning API as the only way to publish transport data comes with three identified limitations, as the server will need to handle the requests from different use cases with different needs:

1. When an application developer would like a *new feature*, such as taking wheelchair accessibility information into account, the feature would have to be implemented on the server of the data publisher
2. Keeping the server *high available* is costly, as any question can be asked by anyone for any purpose.
3. Federated querying, which would allow for *intermodal* route planning for route planning APIs, is unexistent up to date.

In order to overcome these limitations, the General Transit Feed Specification (GTFS)[5] can be used. GTFS is a compressed ZIP-file containing a couple of CSV files, describing the rules for when a public transit vehicle will pass by on a certain location. It is supported among all current open source route planning software systems and it is used in products/apps such as *CityMapper*, *Ally*, *Navitia.io*, *Google Maps* and *Bing Maps*. It succesfully enabled reuse for intermodal travel, engineers can rely on the data dumps even if the servers of the transit agency are offline and there is no limitation to the features that can be implemented. The logs are however lost.

In **??** we illustrate these two options as two extremes, with other options that are yet to be discovered. When a *rather low expressive* server only allows to set e.g., a departure station and a departure time, then the server cannot log the arrival station, yet the client is still able to plan a route by executing the algorithm on the client-side [**?**]. Again, we are not able to rely on the query logs

7. PUBLISHING QUERY LOGS

{**TODO**: When the expressiveness of the servers is lower in order to maximize the reuse of the data, such as in the case of Linked Connections or a GTFS datadump, we suggest to, instead of logging each GET-request, to have an analytics server that gathers all the logs.}

{**TODO**: Also in USEWOD2014 there was a paper on LDF and how we could know which SPARQL queries needed to be solved}

{**TODO**: As of the 17th of December 2015, iRail publishes the most recent 1000 requests done on the API as open data, each second, at *http://api.irail.be/logs/*. This amounts today to an average of 86k queries to the API per day, as the API gained popularity thanks

[5] *https://developers.google.com/transit/gtfs/reference*

to apps like *BeTrains*[6] and *Railer*[7]. On average, 14,357 of these are valid route planning queries (other queries include requesting departures in a station, status of a vehicle, or a list of all stations). If each query would amount to indeed one passenger, this would today be equal to almost 2% of all passengers.}

8. CONCLUSION AND DISCUSSION

The contribution of this paper is twofold. For the first time, we studied query logs to find travel patterns. Due to lack of information we could not evaluate our results, yet the visualizations look promising... There are obvious caveats associated with the use of such data as proxy for actual statistical counts. For instance, the data only captures an intention of a passenger: it's unsure whether the person actually took the train. Furthermore, different requests could be done by one person with an intention to travel. On top of that, our sample used for this paper can only represent a maximum of 0.4% of the daily traffic flow.

The second contribution is our position that we suggest to decouple the query logs from the query execution.

[6] *https://play.google.com/store/apps/details?id=tof.cv.mpp*

[7] *http://railer.be/*