Article

Past Railway Networks: the value of Remote Sensing and Volunteer Geographic Information in Computer-assisted geo-coding

Robert Jeansoulin [1]

[1] CNRS UMR 8956 LIGM, Université Gustave-Eiffel: robert.jeansoulin@univ-eiffel.fr

Abstract: Transportation of goods is as old as human civilizations : past networks and their evolution shed light on long term trends. Transportation impact on climate change is measured as major, as well as the impact on spreading a pandemic. These two reasons motivate the importance of providing relevant and reliable historical geographic datasets of these networks. This paper focuses on reconstructing the railway network in France at its maximal extent, a century ago. The active stations and lines are well documented by the French SNCF, in open public data. However, that information ignores past stations (ante 1980), which represent probably more than what is recorded in public data. Additional open data, individual or collaborative (eg. Wikipedia) are particularly valuable, but they are not always geo-coded, and two more sources are necessary to completing that geo-coding: ancient maps and aerial photography. Therefore, remote sensing and volunteer geographic information are the two pillars of past railway reconstruction. The methods developed are adapted to the extraction of information from these sources: automated parsing of Wikipedia Infoboxes, data extraction from simple tables, even from simple text. That series of sparse procedures can be merged into a comprehensive computer-assisted process. Beyond this, a huge effort in quality control is necessary when merging these data: automated wherever possible, or finally visually controlled by observation of remote sensing information. The main output is a reliable dataset, under ODbl, of more than 9100 stations, which can be combined with the information about the 35000 communes of France, for a large variety of studies. This work demonstrates two thesis: (a) it is possible to reconstruct transport network data from the past, and generic computer assisted methods can be developed; (b) the value of remote sensing and volunteered geo info is considerable (what archeologists already know).

Keywords: geographic information fusion; data quality; data consistency checking; historic GIS; railway network; patrimonial data; crowdsourcing open data; volunteer geographic information VGI; wikipedia geo-spatial information extraction.

---

1. Introduction

Transporting people and goods is as old as human civilizations, roman Via Appia, Via Aemilia, Via Flaminia, Via Egnatia, Via Aurelia, have helped structuring the European landscape. What remains from the ancient networks, what has been abandoned or destroyed, shed lights on the evolution of the society itself, the main events and turning points. Over the last century, the evolution of road and rail networks is a well known picture of their competition; however, less known regional details, may help to analyze different planning scenarios at work within broad transportation policies, including reduction of inequalities [1-4].

The impact of transportation on climate change is now measured as one major negative factor, recently surpassing industry emissions. The European Union acknowledges that "Transport represents almost a quarter of Europe's greenhouse gas emissions" and "has not seen the same gradual decline in emissions as other sectors" [5]. This 2011 White paper stated that " 50% of road freight over 300 km should shift to rail or waterborne transport by 2050". Long term goals cannot be monitored without long time-span datasets allowing "a measurable benchmark to assess progress at EU level towards that vision" [3]. In Public Health also, transportation plays a major role, and can transform an epidemic into a pandemic, as it was demonstrated with the 1918-flu [6].

These two reasons demonstrate the importance of the digital reconstruction of past networks, in particular railways, mandatory for understanding how long time trends form, as investigated by several authors [7-9].

The present paper focuses on reconstructing the railway network in France as it existed through the XXth century, at its maximum extent around 1920, a century ago. The public data about the stations presently in activity, or recently closed to traffic (last two decades) are up-to-date, in convenient format (*CSV*, *geojson*), and easy to access through download or API. But most stations closed after WWII or even during the 70's, are not recorded in a public dataset, but many crowd-sources, including Wikipedia, provide a lot of information about them. Geo-coding [10] and structuring this information is important in particular for:

(1) linking rail-transport data with demographic, economic, environmental and social data, allowing socio-economico-politico... studies in relation with rail, (2) building a graph of the railway network with stations as nodes, allowing distance computation, density, connectivity, accessibility, redundancy (fragility), ... and their evolution through a century.

Besides producing a comprehensive dataset, suited for information fusion [11], the objective to develop reproducible procedures, generic enough to be adapted to other countries. Railways share enough semantics to allow such a generalization. Luckily, previous standardization efforts of the UIC (International Union of Railways) have helped building an international ontology, and the translation of terms related to shared concepts. In particular, the processing of the "railway-oriented" entries of Wikipedia, and the specific *Infobox*, can be quite general, as we can see comparing:

```
https://de.wikipedia.org/wiki/Bahnstrecke_Paris%E2%80%93Mulhouse
https://en.wikipedia.org/wiki/Paris-Est%E2%80%93Mulhouse-Ville_railway
https://fr.wikipedia.org/wiki/Ligne_de_Paris-Est_%C3%A0_Mulhouse-Ville
https://hu.wikipedia.org/wiki/P%C3%A1rizs%E2%80%93Mulhouse-vas%C3%BAtvonal
https://nl.wikipedia.org/wiki/Spoorlijn_Paris-Est_-_Mulhouse-Ville
https://pl.wikipedia.org/wiki/Linia_kolejowa_Pary%C5%BC_%E2%80%93_Miluza
https://zh.wikipedia.org/wiki/...chinese_characters ... (get there via one link above)
```

Another generic development concerns data quality. Besides the efforts about the precision and consistency during the data collection process (computer-assisted or manual procedures), several a posteriori monitoring procedures are proposed.

For sure we all know that the debate about the different modes of transportation, for people or for the freight, has restarted because of environmental and health issues. A better knowledge of the past is crucial for the decision for tomorrow: the computer-assisted reconstruction presented and applied in this work is contributing to that.

2. Materials and Methods: public data, open data and how to automate them.

One dataset fits exactly the expectations, and it is a reliable source, originating from *SNCF-Réseau* (the French national operator), though not official: the file named `garedetoustypes` is available on the data.gouv.fr public site [12]. The records from the version that we copied from that website, are time-stamped 08-2014. The schema of the data is sufficient to complying the requirements listed in section 2.1. Therefore this dataset is the baseline resource of this work.

2.1. Shaping the expected target

The overall objective is to build a comprehensive dataset of the stations that were in service at some time in the last century, including the 3029 official stations ("*gares du réseau français*") in service in 2020 [13].

2.1.1. Target schema

For the purpose of enabling further studies, the dataset must contain the minimal necessary information allowing to joining stations with cities or towns (French: *gares* with *communes*), where most statistical info is collected, and to joining stations with railway lines (arcs and nodes of the network graph). The conceptual schema is given in Figure 2.11.

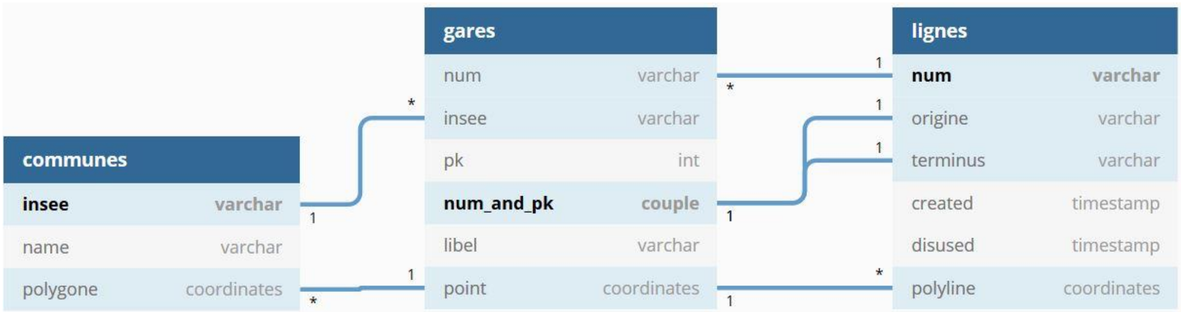


Figure 2.11. Conceptual target schema for stations, lines and towns (primary key: bold font).

Attributes: `insee` = official unique code for a *commune*; `num` = official SNCF number for a *ligne* (or a unique number is created); `pk` = "point kilométrique" (ordered along the line); `name` = toponym for a *commune*; `libel` = for a *gare*; a name for a line can be formed from the couple (`origine`, `terminus`); and finally, `polygone`, `point`, `polyline`, `created`, `disuse`, are spatial or temporal attributes.

The relations between primary keys and foreign keys are obvious, the spatial relations between `point` coordinates and `polygons` or `polylines` are computed intersections: a *commune* may have several *gares*, but a *gare* belongs to a single *commune* (though in a few cases, a *gare* has been built on the limit, intentionally). Finally a *gare* is uniquely identified by the couple `(num, pk)` and not by its `libel`, which is not necessarily unique.

2.1.2. Target notation and presentation

The notation for a "*gare*-object" recorded as a *geojson* feature, is:

```
{"type":"Feature", "geometry":{"type":"Point", "coordinates":[LATITUDE, LONGITUDE]},
"properties":{ "libel":"LIBEL", "num":"NUMERO", "pk":"PK", "insee":"INSEE", "info":"any
additional info such as origine | terminus | ..."}},
```

This is similar to the schema used by the main public sources (the baseline resource). For additional stations, we must build the *geojson* feature step by step:

- the process is generally started by the knowledge that there is some station named `libel`, on some line identified by its `origine` and its `terminus`;
- if we are lucky, there is a Wikipedia page for `'Ligne de` *`origine`* `à` *`terminus`*`'` (see below 2.4.2), and we will get the `num`, the `length` and much more; else, we have to provide a unique `num` for that line (see below 2.5.x);
- the `pk`, if not given, can be approximated by interpolation between two geo-coded stations, at least using the `origine`, `terminus` locations, and the total `length` of the line;
- the `name` of the *commune*, sometimes is given by the `libel` of the station (most of the time in rural places), or the first one if the `libel` associates the `names` of two *communes*, when the station is in between (eg.: `libel = "Ermont - Eaubonne"` for a *gare* in between these two *communes* of the *Val d'Oise département*). In all cases, the `name` and the associated `insee` code, can be obtained (if unknown) or verified (if given: see §3.3 "quality control") by geometric inclusion of the station `point` coordinates into the `polygon` of the *commune*.
- the `point` coordinates must be obtained by any means, unless dropping the feature. This is the most time consuming part of the overall process.

Moreover, the network structure can be made more easily readable, if the features are sorted by `num` and `pk`:

```
{"properties":{"libel":"a", "num":"N_1", "pk":"PK0", "info":"origine of N_1"}},
{"properties":{"libel":"b", "num":"N_1", "pk":"PK+", "info":"connect"}},
{"properties":{"libel":"c", "num":"N_1", "pk":"PK++", "info":"simple"}},
{"properties":{"libel":"g", "num":"N_1", "pk":"PK++", "info":"terminus"}},
{"properties":{"libel":"d", "num":"N_2", "pk":"PK0", "info":"origine of N_2"}},
{"properties":{"libel":"b", "num":"N_2", "pk":"PK+", "info":"connect =="}},
{"properties":{"libel":"e", "num":"N_2", "pk":"PK+", "info":"simple"}},
{"properties":{"libel":"_", "num":"N_2", "pk":"PK+", "info":"bridge over"}},
{"properties":{"libel":"f", "num":"N_2", "pk":"PK+", "info":"simple"}},
{"properties":{"libel":"g", "num":"N_2", "pk":"PK++", "info":"terminus =="}},
```

and the network graph can be drawn, even without the coordinates of the lines:

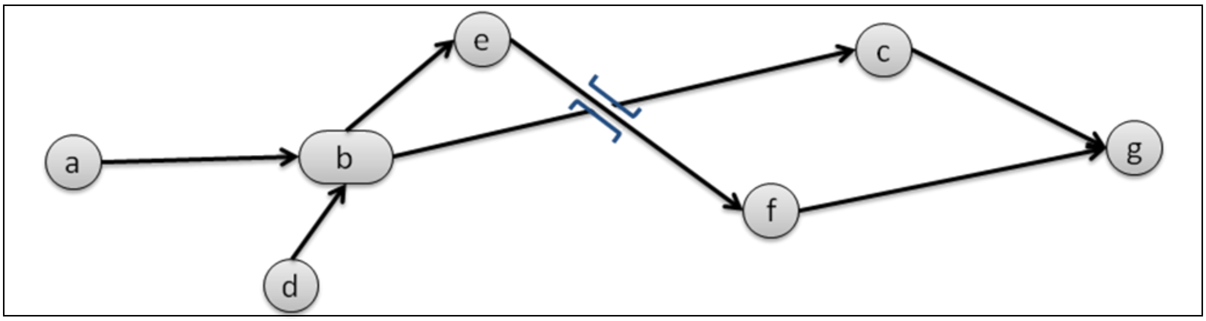


Figure 2.12. Drawing the graph of the network from the *gare*-objects.

Preferably, though not mandatory, *gares* linked to several *lignes* should appear as many times, junctions should appear additionally if they occur outside of a *gare*, as well as line crossing (eg. bridges) to indicate where the graph is non-planar (useful for some path computations).

2.1.3. Baseline goal, baseline resource and baseline cost

In conclusion, we have a goal: the structured dataset targeted above, we know the baseline resource (the "SNCF-Réseau" dataset of *gares*), we have an idea of the baseline cost for building one feature (the total process outlined above), but we don't know precisely how many features are expected It could be 10000, with an error margin of 20%, or more?

2.2. Open data sources: public, and crowdsources

The public datasets, and the version that has been used in this work, are provided in the next Tables 2.2(a)-(c).

Table 2.2(a). Public data sources for stations ('gare'-object)

SNCF1: "Réseau ferré de France", created: jul. 2013, last update: 2017
`https://data.gouv.fr/fr/datasets/gares-ferroviaires-de-tous-types-exploitees-ou-non/`
`Schema (CSV) and counts (6442 features, all attributes: 6442):`
`[CODE_LIGNE, NOM, NATURE, LATITUDE (WGS84), LONGITUDE (WGS84)]`
	version
	2017

SNCF2: "Réseau ferré de France", created: ante 2017, last update: 2020
`https://ressources.data.sncf.com/explore/dataset/liste-des-gares/`
`Schema (geojson) and counts (7702 features, with geometry: 6812 only, other null):`
`{code_ligne, libelle_gare, fret, voyageurs, rang, code_uic, idreseau, pk, departement, commune} +{geometry}`
	version
	2017

Table 2.2(b). Public data sources for cities and towns ('commune'-object)

Insee-new: official codes for *commune* (with 2003-2020 changes), last: 2020
`https://www.insee.fr/fr/information/2028028`
`Schema (CSV) and counts (35589 features, 2577 changes, no geometry):`
`[annee_modif, COM_FIN, COM_INI, LIB_COM_FIN, LIB_COM_INI]`
	version
	2018

Insee-geo: *commune* contours (curated by G. David), last: oct. 2018
`https://france-geojson.gregoiredavid.fr/`
`Schema (geojson) and counts (35798 features, with geometry):`
`{code,nom}+{geometry}`
	version
	2018

Table 2.2(c). Public data sources for railways lines ('ligne'-object)

LignesSNCF: "Réseau ferré de France", created: (before 2017), last: may 2020 https://ressources.data.sncf.com/explore/dataset/formes-des-lignes-du-rfn/ Schema (geojson) and counts (1779 features, with geometry): {libelle, code_ligne, rg_troncon, pk_fin_r, pk_debut_r, mnemo}+{geo}	version 2017

Among the attributes of the different schemes, let's look at those which can be primary keys in the target schema. The attributes code_ligne and pk (including pk_debut and pk_fin) are exactly what we want for num and pk in target schema: we use num in the sequel. For the targeted insee code, it's a bit more tricky (see below §2.3).

The attributes SNCF1:NOM, or SNCF2:libelle_gare, will be used as target libel, and we will notice problems between sources, because toponyms may have variants (eg.: orthograph, accentuation, see §2.31).

The attributes SNCF1:NATURE, SNCF2:(fret,voyageurs) and LignesSNCF:mnemo inform us about the status of the station (eg.: open to voyageur, temporarily closed, etc.). The attributes SNCF2:commune, Insee:LIB_COM* and Insee-geo:nom are expected to match exactly, but they are toponyms too ... (see also §2.31).

2.3. *Confronting public sources*

In a "perfect" world (of data) a simple *DBMS-join* can realize a perfect fusion of information: for instance if (SNCF1:NOM == SNCF2:libelle_gare), we can mix the SNCF1:NATURE and SNCF2:(fret,voyageurs) values into a richer one (eg.: 'status'), and we can compare the coordinates for precision control (see 3.x).

In the "real" world (even a digital one), data do not fit perfectly every time. The difference can be summarized as follows:

let A, B, C, ... be different features (eg.: *gares*), let p be the primary key, let *dist* be some metric, which can be computed for any couple of values for that key, then

join, or perfect fusion of A and B: $$A.p = B.p \quad \textit{and} \quad \forall\ C,\ A.p \neq C.p \qquad (1)$$

imperfect fusion of A and B: $$\forall\ C,\ \textit{dist}(A.p, B.p) < \textit{dist}(A.p, C.p) \qquad (2)$$

2.3.1. Choosing toponyms as primary key for stations (gare):

It would be the best solution, because the libel is in general the first information we can get about a *gare*. Unfortunately, here are a few bad examples:

- accents: Bordeaux-Benauge can be recorded with an accent Bordeaux-Bénauge;
- abbreviations: St-Paul instead of Saint-Paul;
- non-unicité of commune name: "Saint-Prix" exists in six departements, and Wikipedia uses Saint-Prix_(Allier), Saint-Prix_(Ardèche), ... to distinguish them, but it is not standard;
- non-standard notation of libel: a double name can be Laroche - Migennes or Laroche-Migennes, (with space or w/o), also Paris-Nord instead of Paris Gare du Nord;

Excepting the case of accents and "*Saint.e.s*", which are resolved by a special comparator, the great number of different situations, often with a few, or a single occurrence, doesn't deserve the development of as many different routines. The classical string distances, such as *Hamming* or *Levenshtein* distances, didn't prove satisfactory with our examples, also creating false positive such as "Céreste" with "Ceyreste", etc.

Therefore, between the 6442 *gares* from **SNCF1**, and the 7702 from **SNCF2**, we could match about 5800 of them. The same procedure applied within SNCF2, showed that about 900 were variants of a same station (slight different location for different lines, etc.). Remaining 900 were processed another way, using coordinates of the not-null 6800 SNCF2:geometry, to link them with Insee-geo:geometry., or trying to match a SNCF2:libel with a Insee-geo:nom.

2.3.2. Choosing toponyms as a key for commune (cities or towns):

If we don't have the official insee code, actually, the toponym isn't enough as a primary key, instead the couple (departement, commune), as already noticed with "Saint-Prix". Moreover, because of the (french) political incitation to regroup communes: 2500+ fusions in the last 5 years, and 50 separations, the *commune* may never match between datasets from different dates (and the time-stamp of each record is not often provided).

The imperfect solution is to pre-processing all SNCF2:commune and Insee-geo:nom, to have them translated into the 2018 state. And applying the same processing, on-the-fly, when trying to match a SNCF2:libel. This is the reason for introducing the Insee-new dataset.

That processing didn't lead to the fusion of many more *gares*, but it gives a coherent final dataset, where SNCF2:commune and SNCF2:geometry are linked with a single and same Insee-geo:nom and Insee-geo:geometry. This is important for the final quality assurance.

Finally, there are still 650 incomplete stations: either with a SNCF1:NOM and coordinates, but without a num and pk, or with a SNCF2:libel, num and pk, but with a geometry:null.

It is time to investigate more sources of information.

2.4. Gathering information from Wikipedia pages

There are tons of pages related to railway networks, and there is a specific vocabulary, in each country. Luckily, previous standardization efforts of the UIC (International Union of Railways) sharing a common international ontology, what eases the translation of terms related to same concepts (station, junction, ...). For instance, Figure 2.40 compares the Wikipedia pages in two languages:

https://nl.wikipedia.org/wiki/Spoorlijn_Audun-le-Tiche_-_Hussigny-Godbrange and
https://fr.wikipedia.org/wiki/Ligne_d'Audun-le-Tiche_à_Hussigny-Godbrange

Figure 2.40. internationalization of the Wikipedia-Infoboxes for railways (similar ontology)

We can notice that the semantics is quite the same, in the respective *Infoboxes*. Therefore the procedures developed in this paper, for building the dataset of the French stations, would be translatable for many countries.

2.4.1. extracting coordinates from Wikipedia

Given a toponym, `libel` of a *gare*, or `nom` of a *commune*, the procedure is simple: prefix the toponym with `'Gare de '`, or variant (French rules, eg. `Gare des Aubrais` ou `Gare d'Ercé`, etc...) and "*URI*-encode" that string, into the request:

https://fr.wikipedia.org/w/api.php?action=query&prop=coordinates&format=json&titles=Gare%20d%27Erc%C3%A9

whose answer is a *json* text, which, in that case, contains a `'missing'` property:

{"batchcomplete":"","query":{"pages":{"-1":{"ns":0,"title":"Gare d'Erc\u00e9","missing":""}}}}

As it fails, try again without prefixing the toponym, and get:

{"batchcomplete":"","query":{"pages":{"279049":{"pageid":279049,"ns":0,"title":"Erc\u00e9","coordinates":[{"lat":42.85027778,"lon":1.29027778,"primary":"","globe":"earth"}]}}}}

However, the location of the station can be somewhat far from the location assigned to the commune, for instance 5.3km in the example of Figure 2.

Figure 2. the respective coordinates of a commune and its station (5.31 km apart in that example)

Table 2.41. Routine2.41 (pseudo javascript): get geometry from Wikipedia.

```
const Q1 =
"https://fr.wik.../w/api.php?action=query&prop=coordinates&format=json&titles=";
function prefixWithGare (name) { /* returns "Gare_de_", or  "Gare_du_", ...  */ }
const ccOf = (pp) => Object.values(pp).find(x => x==="coordinates")[0]?.coordinates;
const ggOf = (cc) => cc? ({"type":Point", "coordinates": cc}) : null;
const topon = "Martigues";
const P1 = fetch(encodeURI( Q1 + prefixWithGare(topon) )
    .then(a => a.json()).then(b => ggOf( ccOf(b.query.pages) ) ||
               fetch( encodeURI( Q1 + topon) )
               .then(a => a.json().then(b => ggOf( ccOf(b.query.pages) ));
/* then use that promise result in combination with other promises */
const promiseFinal  = Promise.all([P1, promB, promC])
    .then( ([a,b,c]) => combine(a,b,c), reject);
/* reject can be used for any smooth fallback */
```

The returned `geometry` is either set with coordinates (precise, or at least within the territory of the same *commune*), or `"geometry":null`, which is an accepted *geojson* value, denoting that further work has to be done.

2.4.2. from a station to the line(s) it belongs with, using Infobox

Getting coordinates is simple and has been around for a while. A new revision of the Wikipedia API allows to getting more content. In the request query text, let's replace `prop=coordinates` with: `prop=revisions&rvslots=*&rvprop=content&formatversion=2`, and we get a richer content `'{{Infobox}}'` (cf.: `https://en.wikipedia.org/wiki/Wikipedia:List_of_infoboxes`).

For a *commune* we have access to {`latitude`, `longitude`, `insee`}.

For a *gare* we have all the railway `lignes` that connect to that station, see Figure 2.42.

```
/* complies with Modèle:Infobox_Commune_de_France */
{"batchcomplete":true,"query":{"pages":[{"pageid":279049,"ns":0,"title":"Ercé","revisions":[{"
slots":{"main":{"contentmodel":"wikitext","contentformat":"text/x-wiki","content": "{{Infobox
Commune de France\n | nom = Ercé\n | ...abridged... \n | région = [[Occitanie (région
administrative)|Occitanie]]\n | département  = [[Ariège (département)|Ariège]]\n |
arrondissement = [[Arrondissement de Saint-Girons|Saint-Girons]]\n | canton = [[Canton du
Couserans Est]]\n | circonscription législative = [[Première circonscription de l'Ariège|
Première circonscription]]\n | insee = 09113\n | cp = 09140\n | maire = Christian Carrère\n |
mandat maire  = [[Élections municipales de 2020 dans l'Ariège|2020]]-2026\n | intercomm =
[[Communauté de communes Couserans-Pyrénées]]\n | longitude = 1.29027777778\n | latitude =
42.8502777778\n | alt mini = 574\n | alt maxi = 1903\n | superficie = 40.75\n | population =
{{Population de France/dernière_pop}}\n | ...abridged... \n}}\n'''Ercé''' est une [[Commune
(France)|commune française]], située dans le [[Département français|département]] de
l'[[Ariège (département)|Ariège]] en [[Région française|région]] [[Occitanie (région
administrative)|Occitanie]]. \n\nSes [[gentilé|habitants]] sont appelés les Ercéens et les
Ercéennes."}}]}]}}
```

```
/* complies with Modèle:Infobox_Gare */
... ... \n| lignes = [[Ligne de Paris-Montparnasse à Brest| Paris-Montparnasse à Brest]]<br>
[[Ligne de Rennes à Redon|Rennes à Redon]]<br>[[Ligne de Rennes à Saint-Malo - Saint-Servan|
Rennes à Saint-Malo]]<br>[[Ligne de Châteaubriant à Rennes| Châteaubriant à Rennes]] \n|
 ... ...
```

Figure 2.42. Infox examples for a `commune` (Ercé), and for a `gare` (Gare de Rennes). Relevant items are underlined.

The routine for parsing an `Infobox_Gare` (French template) is sketched in Table 2.42.

Table 2.42. Routine 2.42 (pseudo javascript): parsing a wikipedia Infobox (ignores tables or map)

```
const Q2 = Q1.replace( "prop=coordinates",
    "prop=revisions&rvslots=*&rvprop=content&formatversion=2");
const boxOf = (json) => json.query.pages[0]?.revisions[0]?.slots?.main?.content;
const valueOf = (item, box) => /* seek for item in box (cf.243) */;
const addItem = (acc, x, box) => (x && void (acc[x] = valueOf(x, box))) || acc;
const parseBox = (box, list) => box && list && list.length > 0 &&
    list.reduce( (acc,x) => addItem(acc,x,box), {}) || ({"missing":true});
const topon = "Ercé";
const P2 = fetch( encodeURI( Q2 + topon) ).then(a => a.json)
    .then (b => parseBox( boxOf(b), ["latitude", "longitude", "insee", "lignes"]);
 /* as many items as possible: returns value or make nothing */
```

The adaptation of this routine to similar templates from other countries, shouldn't be difficult. The lists of *gares* given by `lignes` are used in the next section 2.4.3.

2.4.3. extracting data for all the stations of a same line.

If the *Infobox* complies with `Modèle:Infobox_Ligne_ferroviaire` or `Template:Infobox_rail_line` or any international equivalent, there is an item named `'schema'`, or `'schema2'`, which contains a `{{BS-table}}` (or similar). "BS" stands for *Bahnstrecke* as the first format was developed for German Wikipedia pages. This *Infobox* provides the `num` plus useful information about the line `length`, the `gauge`, the `status` (in service, disused, dismantled), and associated dates. And the BS-table of the schema gives an ordered list of the stations, with `libel`, `pk`, `nom`, together with junctions, tunnels, viaducts, ... what may prove useful when the line is not easily visible on aerial imagery (see below 2.6).

This is the grail of the railway network data scientist. Because there are thousands of such pages (Figure 2.43), it deserves the development of a specific software.

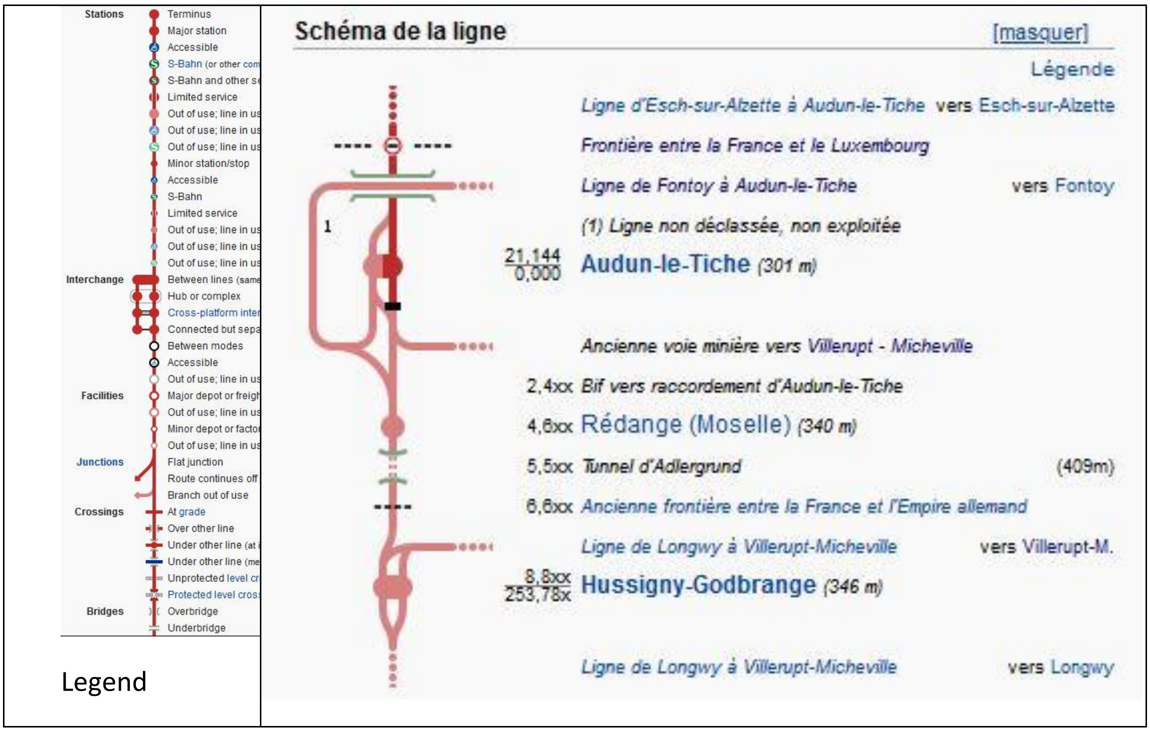
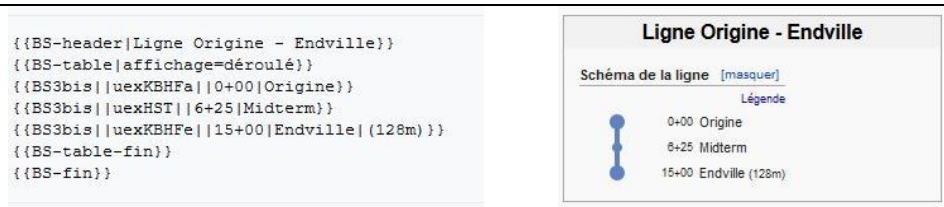


Figure 2.43. Typical *Infobox* `schema` (right), and the legend (left) of its content, from `Template:Railway_line_legend`, or French Modèle: `Infobox_Ligne_ferroviaire/Légende_schéma`

It provides a list of all stations, from `origin` to `terminus`, with `libel`, `pk`, providing also branches towards other lines, bridges and tunnels, which are sometimes useful when the line is not visible anymore on any map or aerial imagery (see § 2.6). Hence, we can build a full list of stations ordered along that line, with most of the information, excepting the coordinates. However, the hard part is twofold:

the *Infobox* table complies with `Wikipédia:Modèles/BS`, or `Wikipedia:Route_diagram_template`. The syntax is somewhat tricky, but can be parsed in most cases. Figure 5 illustrates a simple example of `{{BS-table}}`-code and application.



{"type":"Feature", "properties":{"libel":"Audun-le-Tiche", "num":"196000", "pk":"0+000", "nom":"Audun-le-Tiche", "info":"origine Ligne_d'Audun-le-Tiche_à_Hussigny-Godbrange connec."}, "geometry":null},
{"type":"Feature", "properties":{"libel":"Rédange", "num":"196000", "pk":"4+570", "nom":"Rédange", "info":""},"type":"Feature","geometry":null},
{"type":"Feature", "properties":{"libel":"Hussigny-Godbrange", "num":"196000", "pk":"8+8xx", "nom":"Hussigny-Godbrange", "info":"terminus connec."}, "geometry":null},

Figure 5. *Infobox* sample code (top-left) and display (top-right), and the result of processing the *Infobox* of the Fig.4 example: `'Ligne_d'Audun-le-Tiche_à_Hussigny-Godbrange'` (bottom)

Attributes are directly set from the parsing, `num` is in the *Infobox*, `libel` and `pk` are in the BS-table, and `nom` is often given, if differing from `libel`, or may be set to `libel`. The `insee` and `geometry` are not given, but could be determined later, by next requests to pages 'Gare d'Audun-le-Tiche', 'Gare de Hussigny-Godbrange', and 'Rédange', using the previous routines (routine1 and 2). The quality control will double check the consistency of `geometry`, `insee` and `nom` data (see 3.3).

- The other difficulty to deal with, is the cascade of asynchronous HTTP requests for processing a single line: see section 2.6. The pseudo code in Table 2.43 is not enough to correctly performs that recursion (out of the scope of that paper).

Table 2.43. Routine2.43 (pseudo javascript): from Wikipedia list of stations to list of *geojson* features.

```
/* const Q2, const boxOf(json) from 2.42 */
function prefixWithLigne (A,B) { /* returns "Ligne_de_A_à_B" or "Ligne_d'A_" ...*/ }
const lineName = "Ligne_d'Audun-le-Tiche_à_Hussigny-Godbrange";
/* possibly    = prefixWithLigne("Audun-le-Tiche", "Hussigny-Godbrange"); */
function parseBStableBox(box){
    /* get relevant items using parseBox(box, itemlist) */
    const share = parseBox(box, ["num", ..., "schema", "schema2"]);
    if(share.missing) return null; // page doesn't exist
    if(share.schema) return processBStable(share.schema);
    if(share.schema2) /* there is an indirection to another page */
        return fetch(schema2).then(a => a.json()).then(processBStable); //recursion!
}
const P3 = fetch(encodeURI( Q2 + lineName)).then(a => a.json())
    .then(b => parseBStableBox(boxOf(b)),
          b => reject (b, "some comment"));
```

2.5. *Gathering information from VGI pages*

In any country, railways arouse vocations for Volunteer Geographic Information. A lot of French websites are devoted to keep track of past railway stations, vanished lines, viaducts and tunnels. The next table provides a partial list.

Table 2.50. Some websites about railway lines and stations in France (shortname on the left).

Archef	`http://archeoferroviaire.free.fr` records hundreds of secondary lines, incl. industrial, military lines, at a high-resolution
GeoCF	`http://www.train.eryx.net/hscff/` records hundreds of secondary lines
Ligou	`https://www.lignes-oubliees.com/` "forgotten lines" records 1768 stations and 361 lines
RDrail	`https://www.lignes-oubliees.com/` by the author of "Les 400 profils des lignes voyageurs du réseau" (2 volumes)

Routes	https://routes.fandom.com/wiki/Portail:Transport_ferroviaire_français several tables about ancient lines, about chronology
RuePT	https://rue_du_petit_train.pagesperso-orange.fr / records 6800 street signs mentioning the presence of a former station
Traver	http://chemins.de.traverses.free.fr gathers photographs and cartographic records about ancient lines

The variety of the information provided, and the generally unstructured way this information is represented, make it difficult to code any generic software. But some functions may help in assisting the parsing of some excerpts of html pages. Here are three cases that we face with several time:

2.5.1. html-style tables:

as opposed to a Wikipedia *Infobox*, the `<table>` tag of *html* is not a semantic tab, and it is uneasy to determine the content of such a table, unless by direct reading (no simple AI routine at hand). For decision making, the most important question is: *is there a relevant key in that table? (*eg.: `num`, `insee`, `libel`*)*. Only a few cases, from sources **Routes** or **RDrail**, were treated by this procedure:

- ✓ visual: make a decision about relevance;
- ✓ manual: copy-paste the table text into a spreadsheet; convert into CSV format;
- ✓ software: fetch CSV file; join object with other object sharing same key

Table 2.51. Routine3 (pseudo javascript): get additional info (eg.: dates) from VGI.

```
const A = "http_of_LignesSNCF", B = "http_of_CSV_file";
function CSV2Json(txt) { /* returns JSON from CSV [{num, begdate, enddate},..] */ }
function mergeLigneInfo(a, b){
  const match = (s,r) => r?.enddate?
                        (s.begdate= r.begdate && s.enddate= r.enddate && s) : s;
  return a.map( s => match(s, b.find(r => r.num === s.num)[0] ))
}
const P3 = Promise.all([fetch(A).then(a=>a.json()),
                        fetch(B).then(a=>a.text()).then( CSV2Json) ])
   .then( ([a,b]) => mergeLigneInfo(a,b));
```

2.5.2. coordinates lists in Google KML or GPX format:

Some contributors have digitalized railway lines using, for instance the *Google MyMaps* facility. Though this project doesn't require the full geo-coding of a line, it is useful when trying to localize a station is uneasy by visual means. Sometimes, station data are individualized from the line data, what is even easier. The procedure is:

- ✓ manual: export the line data into KML format (from the Google page);
- ✓ software (or online): convert into *geojson* format;
- ✓ manual: copy and paste the relevant *geojson* features.

This has been used with source **Traver**.

2.5.3. simple lists in plain text:

Hard to detect, unless you read carefully the text. This procedure may help if you get a list of stations along a same line:

- ✓ manual: copy-paste the table text into a string constant;
- ✓ software: build an array of *gare*-objects sharing same `'num'`, use previous routine to check in the same line is documented in Wikipedia, possibly to cross-check data quality, or proceed with a new line number `num`.

Table 2.53. Routine 2.53 (pseudo javascript): from VGI list of stations to list of *geojson* features.

```
const SST = "paste_semi_structured_text", Q2 = "cf 2.42",
   regionCode = "type_railway_region_code", rank = "unique_number";
function prefixWithLigne (A,B) { /* returns "Ligne_de_A_à_B" ...*/ }
function SSTtoJSON (txt) {
   /* parses SST text then => */
   return ({origin, terminus, length, acronym, stations});
}
function stationsAlongLine (json, num) {
   const ptkm = (len,n,tot) => n * len / (tot-1);
   const info = (n,tot) => n===0? "origin" : (n===(tot-1)? "terminus" : "...");
   const mkpr = (a,num,n,tot) =>
      ({ "libel":a.stations[i], num, "pk": ptkm(a.length,n,tot),
"info":info(n,tot) });
   return json.stations.map( (s,i,t) =>
      ({"type":"Feature", "properties": mkpr(json,num,i,t.length),
"geometry":null}));
}
const json = SSTtoJSON(SST);
const num = regionCode + json.acronym + rank; // once for all stations
/* check Wikipedia if line exists, or apply this routine, or both to compare data */
fetch(Q2 + prefixWithLigne(json.origin, json.terminus))).then(a => a.json())
     .then(b => parseBStableBox(boxOf(b)), /* resolve with Routine2.43 */
           b => stationsAlongLine (json, num)); /* process the reject */
```

Example: the text extracted from one page of source **`Archef`**, is an ordered list from origin to terminus, for a line which has a title, a length, an operator, and two dates for its lifetime. The example and parsing using Routine 2.53 is illustrated in Figure 2.53.

```
Ligne :Chantelle - Ebreuil, 23 km
Gares : Chantelle (correspondance pour Saint-Pourçain et Commentry), Ussel-d’Allier, Charroux-d’Allier, Saint-Bonnet-de-Rochefort (correspondance pour Montluçon et Gannat), Vicq, Ebreuil
Ligne ouverte en 1892 fermée en 1939, Société générale des chemins de fer économiques (CFE)
```


```
{"origin":"Chantelle", "terminus":"Ebreuil", "length":23, "lifetime": [1892, 1939], "operator":
"Société générale des chemins de fer économiques", "acronym":"CFE", "stations": [
{"libel":"Chantelle", "connection":true}, {"libel":"Ussel-d'Allier"}, {"libel":"Charroux-
d'Allier"}, {"libel":"Saint-Bonnet-de-Rochefort", "connection":true}, {"libel":"Vicq"},
{"libel":"Ebreuil"}] }
```

{"type":"Feature", "properties":{ "libel":"Chantelle", "num":"700CFE01 ", "pk":"0+xx", "info":"origine Ligne Chantelle - Ebreuil cnxn"}, "geometry":null},
{"type":"Feature", "properties":{ "libel":"Ussel-d'Allier", "num":"700CFE01 ", "pk":"5+xx", "info":""}, "geometry":null},
{"type":"Feature", "properties":{ "libel":"Charroux-d'Allier", "num":"700CFE01 ", "pk":"9+xx", "info":""}, "geometry":null},
{"type":"Feature", "properties":{ "libel":"Saint-Bonnet-de-Rochefort", "num":"700CFE01 ", "pk":"14+xx", "info":"cnxn"}, "geometry":null},
{"type":"Feature", "properties":{ "libel":"Vicq", "num":"700CFE01 ", "pk":"18+xx", "info":""}, "geometry":null},
{"type":"Feature", "properties":{ "libel":"Ebreuil", "num":"700CFE01 ", "pk":"23+xx", "info":"terminus"}, "geometry":null},

Figure 2.53. Example of semi-structured input text with list of stations (top); result of the parsing (middle); and resulting *geojson* list of features (bottom)

2.6. Remote sensing information for the assisted visual geo-coding of stations.

For geo-coding a *gare*, we have a two (or one) steps procedure:

- ✓ software: use a geo-coder API, eg.: *Google geocode*, in order to get coordinates, for instance, with the example of *Cardet* (*Gard*), and try a few variants in Table 2.60.

Table 2.60. example of geo-coding queries for a gare-object.

Query	[lat,lon]	dist
`[lat,lon] =geocode("Cardet, France");`	[44.026042, 4.081059]	672m
`[lat,lon] =geocode("gare, Cardet, France"); //same result`	[44.026042, 4.081059]	672m
`[lat,lon] =geocode("rue de la gare, Cardet, France");`	[44.010420, 4.084597]	1515m
`[lat,lon] =geocode("chemin de la gare, Cardet, France");`	[44.019122, 4.086764]	536m
... true coordinates (see below: Fig.2.61(b) and §2.63) are	[44.023710, 4.088814]	0

... no way to know exactly the best query, nor at which distance will be the answer. Conclusion: use only the nom of the commune as for a general approximation;

- ✓ manual and visual: input the geo-coded result, or skip the first step and type directly the toponym, into the generated URI of an online map facility that provides -preferably- ancient maps and aerial photo.

In France, the main, and certainly best data source for historic maps, and old Remote sensing (aerial before satellite) is collected on a single portal, operated by IGN, the French national geographic organization.

This website is "`https://remonterletemps.ign.fr/`" and allows access to:

Present: (2016-2018)	Aerial photography (≈meter resolution) Vector cartography IGN (≈1/10000 compatible)
Past: (1950-1960)	Aerial photography (≈10 m resolution)~ Digitized map (≈1/25000 compatible)

Figure 2.60 illustrates how to use this website: type the toponym `'Chantelle'` (cf. 2.53) use a zoom level of 16, with two past sources (here: 1954-aerial and 1952-map), and you

can visually detect (at least on-line) where is the station (big arrow) and where is the junction between the two lines (yellow arrow), about 200 m SW of the station (the scale for 200m is given at the bottom-left corner).

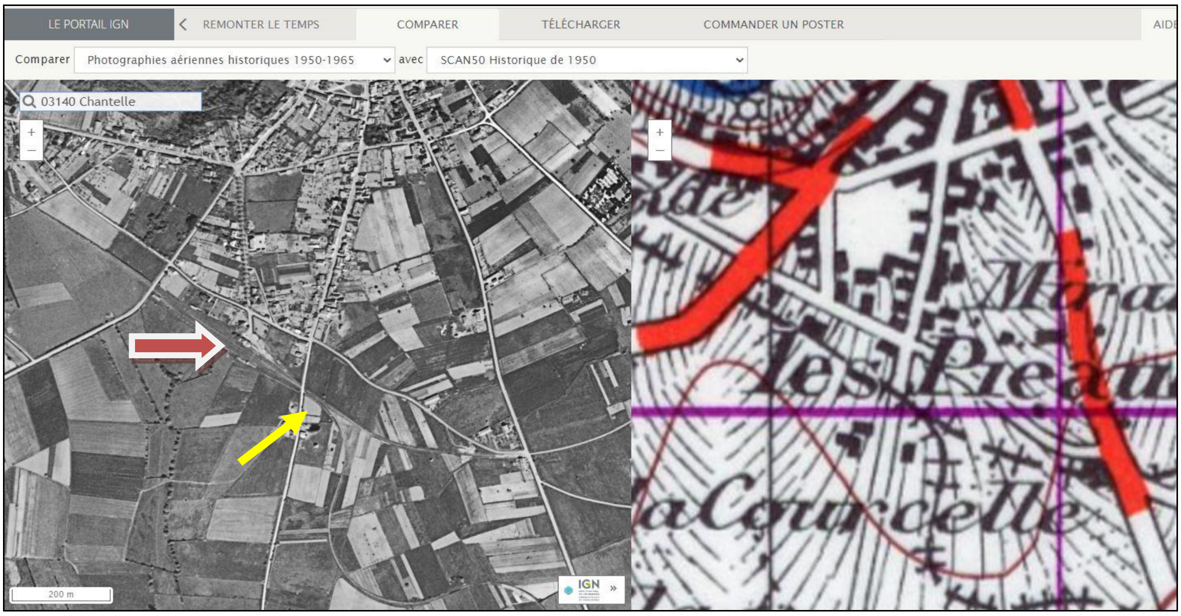


Figure 2.60. Website `remonterletemps.ign.fr`: geo-coding *Chantelle (Allier)* at roughly 1/20000, using a 1954 aerial image, and the scanned map (1952). The double arrow shows the station, the smaller arrow points to a junction of two lines (the line to *Ebreuil* makes a bend to the south).

2.6.1. Dual use of the website RemonterLeTemps, (a) direct geocoding of toponyms:

Table 2.61 sketches out the general procedure, and can be used to geo-code the station of *Chantelle*, as we did in Figure 2.60.

Table 2.61. Geo-location of an expected *gare* around a given *commune* toponym.

```
/* GEO-LOCATION: from toponym to coordinates */
const RLT = "https://remonterletemps.ign.fr/comparer/basic?mode=doubleMap" +
    "&layer1=ORTHOIMAGERY.ORTHOPHOTOS.1950-1965" +
    "&layer2=GEOGRAPHICALGRIDSYSTEMS.MAPS.SCAN-EXPRESS.STANDARD";
/* open a window with RLT, type a toponym: the image moves, the coordinates appear
in the URI: x for longitude, y for latitude, and a zoom (z=16 generally) */
```

For geo-coding a `commune`: type the `nom`, then try to detect if there is a feature that may be a station, generally at the junction of several linear features (rail and road). Here are a few examples: not the easy ones, but once you have find the right location and the most appropriate zoom, you can be quite confident.

(a) *Jaligny-sur-Besbre (Allier):* a small road is visible. But where is the station? The answer is given by using the present IGN map: toponym "la Gare" is there (toponymy memory!), and we can "see" the ancient railway bending N-E.	 

(b) *Cardet (Gard):* the railway line is well visible, as well as two roads converging towards the center of village, and crossing the railway in two places. Where is the station? at the intersection with the biggest road (east?), or closer to the center (west?).	
(c) *Kirchheim (Bas-Rhin):* the railway is well visible, but where is the station? North or South of the central road? No answer. Probably South, but it's just 100m apart anyway.	
(d) *Le Cateau-Cambrésis (Nord):* railway very well visible, half-circling the city. But where is the stop "Le Cateau-Halte"? Comparing the PK with previous and next stops, by rough interpolation, the stop should be at the bidge, near "*Faubourg de Landrecies*". And there is a station just South of the picture, where lines meet.	

Figure 2.61. Some examples where the visual geo-coding of a station is not obvious.

Therefore, the different steps of the procedure can be summarized as:

- ✓ manual: geo-coding initial step: type the toponym as in Table 2.61
- ✓ visual: start with an intermediate zoom (eg.:z=16) and focus on the section where the line is expected to reach the commune (eg.:S-W corner)
- ✓ visual: seek help on associated maps: old- or recent-map can be used, as we noticed from Fig.2.61(a). Pay attention to road networks, as it certainly intersects the rail where the *gare* is, as in Fig.2.61(b) or (d)
- ✓ manual: play with zoom if you have a candidate location: the trace of the ancient line may become visible only at a certain level (eg.z=18) as in Fig.2.61(a)

- ✓ visual: for a good candidate, seek for a 'voyageur building' (if any) among several, and point to the place where people are expected to cross the rails, as in Fig.2.61(c).
- ✓ manual: copy-paste the xy-location from the website bar to the *geojson* feature.

There is no place for more automation in that part of the overall procedure, unless well beyond the scope of this study. But the care brought to this already time consuming task, is a protection against quality deficiencies later, which will be much more difficult to detect.

2.6.2. Dual use of the website RemonterLeTemps: (b) visual reverse-geocoding:

The "visual" reverse geo-coding is useful if we want to inspect what a couple of geographic coordinates designates in different representations of the geographic space at different dates. Table 2.62 sketches the initial step of that procedure.

Table 2.62. Reverse Geo-location of a couple of coordinates.

```
/* REVERSE GEO-LOCATION: display the coordinates such as Remotely represented */
const RLT = "as above";
const lon = 4.11488, lat = 44.04241, z = 20;  // high zoom for a 10m. location
const nom = reverseGeocode(lon,lat);  // from any online geocoder (or via API)
const mkURI = (x,y) => RLT + "&x="+x +"&y="+y +"&z=20";
/* check 'nom' against any toponym related with (lat,lon) */
/* open window at mkURI(lon,lat) and inspect with available aerial pictures/maps */
```

For reverse geo-coding, use the information that you want to control, which associates `coordinates` with a `libel`, and possibly a `nom`:

- ✓ software: reverse-geo-coding initial step: generation of the URI as in Table 2.62
- ✓ software: cross-check with a *geocoder*, to get *commune* name;
- ✓ visual: start with a rather high zoom (eg.:19), and inspect one or several (map-aerial photo) combinations. You may have to reduce the zoom to be able to read one of the toponyms that you expect (including words linked to railway terminology);
- ✓ manual: confirm, or infirm, the quality of the information;
- ✓ manual: bring casual correction to properties of the feature(s).

This procedure can be applied every time a problem occurs: (a) during the reconstruction phase, in particular around the junction of the investigated line and a previously recorded one: some discrepancies may emerge; (b) during the quality assurance phase. An extensive example is provided in section 3.3.

2.6.3. Hints for alternatives

Cartography isn't an "exact science" even at "*the scale of a mile to the mile!*" (*Lewis Carroll*), different maps may display different toponyms. For instance, with the help of Google maps, or Bing maps, we can solve the *Cardet* indecision in Figure 2.61(b).

Figure 2.63. Using alternative maps to resolve indecision: there is a "*Chemin de la Gare*" indication.

3. Results: the overall procedure, the reconstructed network, and its quality control

There are three main results out of this study:

1. series of routines that can be combined into a global Computer-Assisted process of reconstructing a fully geo-coded network;

2. the production of a national dataset, whose structure is similar to the public dataset, an can serve similar purposes, back in the past and not just today;

3. series of posterior quality controls that should help to work in confidence, or help to point out imprecision, or missing data, to be further corrected.

3.1. Computer-Assisted Reconstruction of past railway networks

There is no miracle solution to the problem: we may consider building a tagged collection of small pictures showing what a station should look like on old aerial maps, and input that collection into some deep-learning software. But it may be as time consuming than the computer-assisted solution proposed here. Not counting a posteriori control for assessing the quality of the result.

The procedures described in section 2, are a mix of manual and software steps: this is what we call "Computer-Assisted" procedure. Here are the main general steps for building the graph representing the network. Figures 3.11 and 3.12 are sketched with ANSI symbols (process, manual operation, visual display, decision, storage, multi-documents ...).

3.1.1. From lines (ligne identity) to list of stations (identified gare-objects)

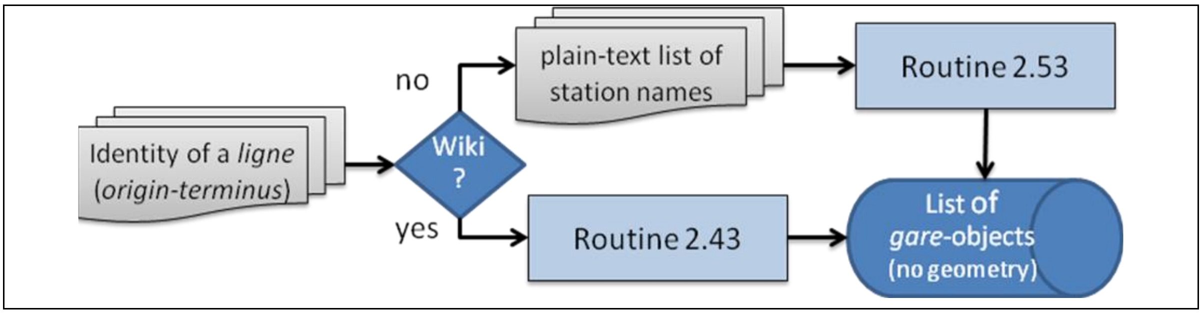


Figure 3.11. the computer-assisted procedure for *lignes*.

3.1.2. From station identity to geo-coded gare-object

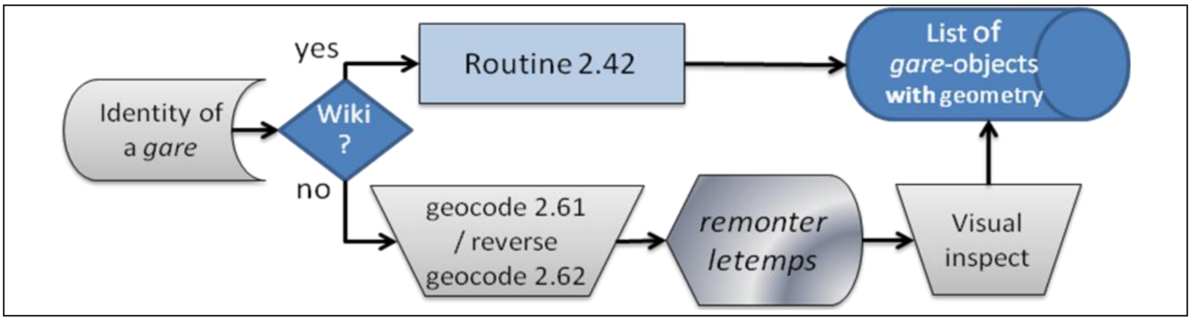


Figure 3.11. the computer-assisted procedure for *gares*.

3.2. The Reconstruted network

The dataset, as of August 2020, contains 9100 features, with Point geometry, and:

{"properties":{ "libel":_, "uic":_, "num":_, "pk":_, "nom":_, "insee":_, "info":_ }}

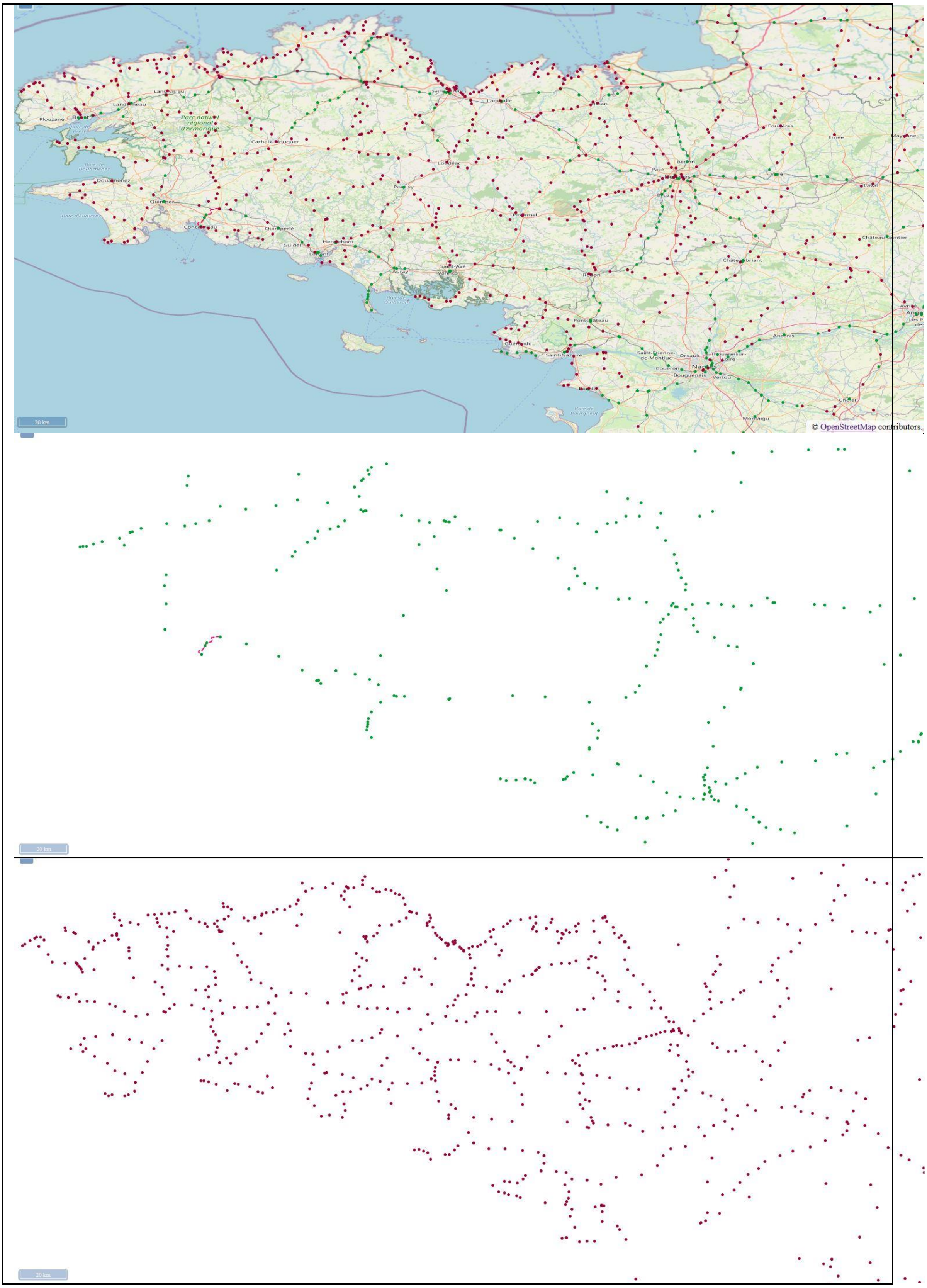


Figure 3.20. Bretagne station: (a:top) full set of stations on OSM map; (b:middle) active stations only; (b:bottom) inactive stations, disused or demolished.

3.3. Computer-Assisted Quality Control

One advantage of the computer-assisted procedure, is that the quality control is part of the data collection process, and every feature that is added to the dataset, receives an `info` property indicating which property is recorded with which uncertainty. So far this is simple text added into the `info` string: such as '`approx.`' if the point cannot be located precisely, or '`pk?`' if that data is interpolated.

Moreover, an automated systematic a posteriori control is performed, checking the consistency of each gare-object feature. The rules to be checked are:

- ✓ the stations along a same line (same `num`) should have `pk` in strict increasing order (no duplicates); otherwise a warning is issued (incompleteness);
- ✓ the stations along a same line (same `num`) should have the first one tagged as `origin` and the last as `terminus`; otherwise a warning is issued (eg. incomplete information), and more if origin or terminus are known to be different (inconsistency and/or incompleteness);
- ✓ a station without a `nom` for its *commune*, is checked with "point-in-polygon" routine, and information is automatically completed (`nom` and `insee`); next rule applies;
- ✓ a station with `nom` and `insee` must be consistent with its `libel`. The `libel` is checked in a geo-coder together with `nom`: it can be a part of the *commune* (eg. la Souys, is a borough in Bordeaux), what is generally noticed by a geocoder. But if it is the name of a different commune, a warning is issued. See example and Figure 3.30.
- ✓ when a line with same `num` exists in **LignesSNCF** or among crowdsources, the station point can be checked with closest-point-to-polyline or equivalent, and if the distance overpasses a threshold, a warning is issued.

Only rules 1-3 are implemented so far, and it already helped to bring correction. Here is one particular example, which shows different aspects of possible errors: the two features come from the "baseline" **SNCF2** dataset (version 2017):

{"type": "Feature", "geometry": {"type": "Point", "coordinates": [4.11488, 44.04241]}, "properties": {"departement": "Gard", "commune": "Lézan", "voyageurs": "N", "code_uic": 87775445.0, "libelle_gare": "Lézan", "pk": "690+672", "code_ligne": "**814000**", "fret": "N"}},

{"type": "Feature", "geometry": {"type": "Point", "coordinates": [4.11488, 44.04241]}, "properties": {"departement": "Gard", "commune": "Lézan", "voyageurs": "N", "code_uic": 87775445.0, "libelle_gare": "Lézan", "pk": "690+673", "code_ligne": "**815000**", "fret": "N"}},

are related to the same *gare*, because their original attributes are all identical: `libelle_gare`, `commune` and `departement`, and same `uic`, the UIC code. Only two different `code_ligne`, what is not surprising if a *gare* is a connecting station. No problem can be detected at first glance.

But a reverse geo-coder answers *commune* = *Vézénobres*, which is about 7 km East of *Lézan*. A visual inspection is then mandatory.

There is a well visible railway (N/S) and a possible branch going west, over a bridge. This is the symptom of a junction. An old station (or simple halt) may have disappear. Or: the record is the mark of the junction (French *bifurcation*) towards *Lézan*, which was actually a *gare* on the old line `num=814000`.

The Wikipedia `Infobox:schéma` of that line indicates the junction just before the viaduct over "*Le Gardon d'Alès*".

Figure 3.30. Quality control after automated identification of a geographic inconsistency (*gare* `point` not in *commune* `polygon`).

Example end: here are the two new features, in the target dataset:

{"type":"Feature","geometry":{"type":"Point","coordinates":[4.05285,44.01741]},{"properties":{"libel":"Lézan","uic":"87775445","num":"814000","pk":"690+672","frevoy":"NN","nom":"Lézan","insee":"30147"}},

{"type":"Feature","geometry":{"type":"Point","coordinates":[4.11488,44.04241]},{"properties":{"libel":"Lézan (**bifurcation**)","uic":"87775445**x**","num":"815000","pk":"**684+3xx**","frevoy":"NN","nom":"Vézénobres","insee":"30348","info":"**libel and uic change, pk approx.**"}},

including updates of the `libel` and `uic` of the bifurcation: distinguished from *Lézan*, and of the `pk`: approximated during the visual inspection, about 400m after *Mas-de-Gardiès*.

4. Discussion

The paper focuses on the realization of the computer-assisted reconstruction procedure. Not every improvement has been considered, several software solutions have been dropped as requiring much too important development with respect to the expected benefit. Not all a post-processing have been implemented either: more consistency rules must be added and software coded.

The additions have been brought to only a dozen French *départements* (among 95), mainly in Bretagne, and South-West of France (more than a thousand), and in some places purposely spread all over France (about a thousand), and, also purposely, for every national border crossing with neighbor countries (about 50 such places). What can be noticed, without further, and necessarily trans-disciplinary, investigation, is that local patterns would certainly emerge in the process of opening lines (eg.: in the XIXth century, or just before WWI), then closing lines (eg.: just before WWII, in the late 90's).

This is important, because the deviation between the recommendations made by the EU, and the decisions made by a member state such as France, cannot be understood without the combination of multiple factors, depending on multiple actors, over several decades where different political orientations followed one another.

Let's conclude with the EU White paper [5], which observed, ten years ago that the freight transported by road is about 100 times more polluting than by rail, and observed that present trend is clearly that the goal (shifting 30% from road to rail/maritime by 2030) would not be reached.

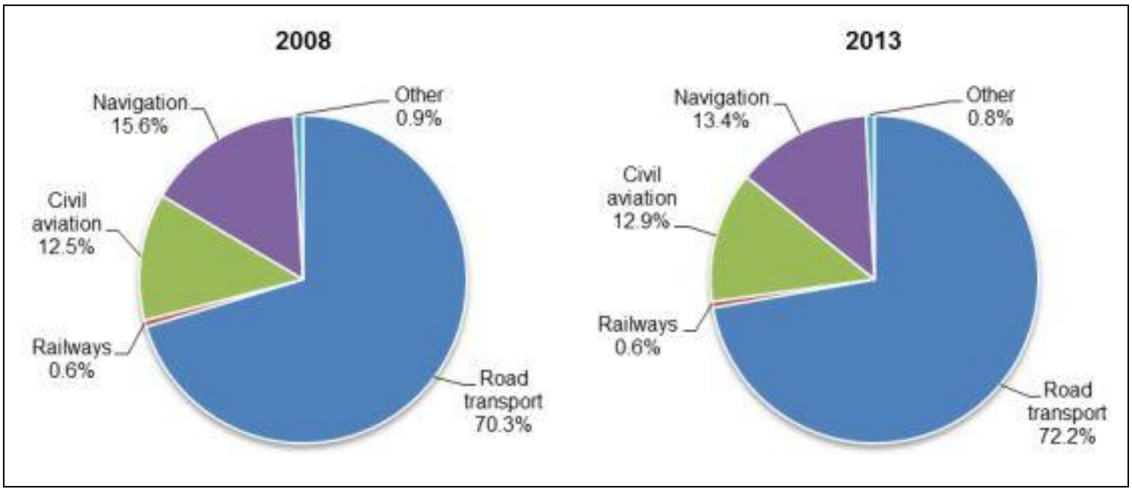


Figure 4.00: Greenhouse gas emissions from transport by mode in 2008 and 2013 [14]

A better understanding of the multiple causes of the rise and decline of the rail versus road transportation, may advise the EU legislators to propose more efficient policies today, because "states played, and continue to play, a fundamental role in the establishment of Europe's national railway networks, regardless of EU guidelines" [9]. The 2016 report executive summary first sentence notes: " *The fact that only limited data are available and that the impacts of most of the initiatives cannot yet be observed do not allow the proper assessment of the effectiveness of the measures adopted so far and their contribution to reaching the goals. Moreover, at this stage it would be difficult to assess the impacts of the long-term transport strategy, given that even where initiatives have led to the adoption of corresponding legislation, the latter has often not yet been fully implemented on the ground and it would be premature to expect any significant impacts* ". It can't be more crystal clear.

Funding: This research received no external funding.

Acknowledgments: work initiated with the help of the sophomores 2016 and 2017 at IUT-Marne-la-Vallée, and pursued with the encouraging comments from IGN and some VGI contributors and railways lovers.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Evans, A.K.B., Gough, J.V. *The impact of the railway on society in Britain : essays in honour of Jack Simmons*. Ashgate Publishing: Burlington, VT, USA, 2003; 315 p.

2. De Block, G.; Polasky, J. Light railways and the rural–urban continuum: technology, space and society in late nineteenth-century Belgium. *Journal of Historical Geography* **2011**, Vol 37: 3, pp 312-328.

3. Enflo, K.; Alvarez-Palau, E.; Marti-Henneberg, J. Transportation and regional inequality: the impact of railways in the Nordic countries, 1860–1960. *Journal of Historical Geography* **2018**, ...

4. Martí-Henneberg, J. European integration and national models for railway networks (1840–2010). *Journal of Transport Geography* **2013**, Vol 26, pp 126-138. https://doi.org/10.1016/j.jtrangeo.2012.09.004

5. Report on implementation of the 2011 EU White Paper on Transport ("*Roadmap to a Single European Transport Area – towards a competitive and resource-efficient transport system*"), five years after its publication: achievements and challenges. **2016**, https://ec.europa.eu/transport/sites/transport/files/themes/strategies/doc/2011_white_paper/swd(2016)226.pdf

6. Byerly, C.,R. The U.S. military and the influenza pandemic of 1918-1919. *Public Health Reports*. **2010**, 125: 3, pp 82-91. PMCID: PMC2862337.

7. Siebert, L. Using GIS to Map Rail Network History. *The Journal of Transport History* **2004**, Vol 25: 1, pp: 84-104. Dept. Geography and Planning, University of Akron, Akron, OH, USA. siebert@uakron.edu

8. Gregory I.,N.; Ell, P., S. *Historical GIS: Technologies, Methodologies and Scholarship* (Cambridge Studies in Historical Geography). Published February **2008** by Cambridge University Press. 227 p.

9. Morillas-Torné, M. Creation of a Geo-Spatial Database to Analyse Railways in Europe (1830-2010). A Historical GIS Approach. *Journal of Geographic Information System*. **2012**, 4. 176-187. 10.4236/jgis.2012.42023.

10. Tian Lan, Longley, P. Geo-Referencing and Mapping 1901 Census Addresses for England and Wales. *ISPRS Int. J. Geo-Inf*. **2019**, 8(8), 320; https://doi.org/10.3390/ijgi8080320

11. Edwards, G.; Jeansoulin R. Data fusion—from a logic perspective with a view to implementation. *International Journal of Geographical Information Science* **2004**, 18 (4), 303-307.

12. SNCF Réseaux, public data. Available online, https://ressources.data.sncf.com/explore/dataset/liste-des-gares/download?format=geojson

13. SNCF Gares & Connexions, (Wikipedia page), available online https://fr.m.wikipedia.org/wiki/SNCF_Gares_%26_Connexions

14. Greenhouse-Gases calculator. Available online, https://www.eea.europa.eu/data-and-maps/data/data-viewers/greenhouse-gases