Dipartimento di Ingegneria e Scienza dell'Informazione

– KnowDive Group –

Geospatial Knowledge Graph

Document Data:	Reference Persons:
October 23, 2020	Bortolon Matteo
	Mattedi Daniele
	Povoli Sergio
	Tessari Michele

KnowDive (internal) reports are for internal only use within the KnowDive Group. They describe preliminary or instrumental work which should not be disclosed outside the group. KnowDive reports cannot be mentioned or cited by documents which are not KnowDive reports. KnowDive reports are the result of the collaborative work of members of the KnowDive group. The people whose names are in this page cannot be taken to be the authors of this report, but only the people who can better provide detailed information about its contents. Official, citable material produced by the KnowDive group may take any of the official Academic forms, for instance: Master and PhD theses, DISI technical reports, papers in conferences and journals, or books.

Contents

Revision History:

Revision	Date	Author	Description of Changes
0.1	12.10.2020	Sergio Povoli	Initial personas description
0.2	13.10.2020	Matteo Bortolon	Scope description
0.3	15.10.2020	Sergio Povoli	CQs persons 1,2
0.4	15.10.2020	Matteo Bortolon	CQs persons 3,4
0.5	18.10.2020	Michele Tessari	Initial dataset description
0.6	20.10.2020	Matteo Bortolon, Sergio Povoli	Data Objects
0.7	20.10.2020	Daniele Mattedi	Datasets metadata documentation
0.8	21.10.2020	Matteo Bortolon	Inception level evaluation

1 Knowledge Graph Development Process

This part of the document aims of describe the development process of the knowledge graph. For the development of the knowledge graph, we decided to use iTelos methodology because it will guide us step by step during the development process. At the start we describe the contest and the problem we need to solve and the personas that can benefit from our knowledge graph. In the inception phase we describe the Competence Query and a initial list and description of the datasets used.

1.1 Scope Definition

In this section we describe the contest of the knowledge graph and some stereotypes of person that represent some potential user of our knowledge graph.

1.1.1 Problem contest definition

With GeoSpatial data we intend all the data that are explicit or implicit connect to a location. And without thinking about it a lot of our everyday life are connected to location information, at example where has my wife gone or when we want to try a new restaurant. The public administration is one of the entities that mostly used geospatial data, a business estimate that 90% of the PA data are bound with a location. [1]
While the Geospatial field is very large, we decide to focus on the real estate market. In Italy, based on the report from the Italian Tax Bureau, in 2019 603'541 residential real estate units were sold and bought [2] [3]. Combined they are equal to the surface of the city of Bolzano. The real estate trading sector is valued 97.5 billions of euros per year. These numbers do not consider the market of the rents. In 2019 the new rents agreements touch the staggering amount of 1 million and 7 hundred units. In 2019 the 10% of residential rental units in Italy was rented, generating a market of around 10 billion per year. [4]
In Italy, the biggest part of residential real estate units rented and traded are owned by physical persons. The Italian real estate market status is very different from other nations in Europe, where rented and traded residential buildings are mostly owned by enterprises. In particular, these include banks or investment funds, normally behind the shields of ad-hoc front company. This difference has taken in Italy to the proliferation of the real estate agents. To make a comparison in Italy there are 43'698 real estate agents for 59 million people, while in Germany where most of the building is rent or buy from businesses the agents are 23'780 for a population of 80 million [5]. Searching between a large number of real estate operators can represent a considerable expenditure of energy for who want to rent or buy a unit. For resolve this problem various intermediation portal was born. In these portals, the real estate agents (or also the owner directly) list the house. These portals are mainly own and operate from national companies if compared to other sectors like search engines where near all the companies are foreign.
Listings on this portals normally describe very well the features of the real estate unit, but they normally failed to describe the connected services. Connected services include, for example, the internet connectivity available, the quality of schools in the nearby of the real estate unit or the presence of parking where leave the car. These data are sometimes sparse and difficult to retrieve, as a consequence the real estate agents sometimes do not know or have

[1] https://philly.curbed.com/2017/11/7/16617296/philadelphia-properties-land-vacant-lot-atlas-map-tool

[2] https://www.agenziaentrate.gov.it/portale/documents/20143/262485/StatisticheOMI$_RES_12020_20200605.pdf/3af07746 - 906c - 45af - 5853 - da521605de03$

[3] The given data do not include the Italian territories that use the Grundbuch (at the moment they are the provinces of Trento and Bolzano)

[4] https://www.idealista.it/news/immobiliare/residenziale/2020/05/07/140165-affitti-italia-focus-su-trend-del-mercato-proprietari-e-riduzione-del-canone-di

[5] https://www.liberopensieroimmobiliare.com/evoluzione-di-un-agente-immobiliare-italiano-dati-e-considerazioni/

little knowledge about these connected services. Sometimes agents know the information but avoid to give them to the customer because they can lower the value of the property. These considerations can in some case represent the decisive factor for the choosing of a unit over another.
Our idea for the project is to build a knowledge graph that can help the people in search of to buy or rent a real estate unit to easily consult the list and quality of these connected services. While the knowledge graph can be useful in near every location in the world (excluding Antarctica), we decide to focus to the province of Trento in Italy, in the first phase, for the amounts of publicly available datasets. For some type of data we decide to focus only on the city of Trento, because they are not available at the province level. While our final target for the project is the people that want to rent or buy a real estate unit, we target the intermediation portals as a potential customer for our knowledge graph. The intermediation portals can use our knowledge base to help the customers compare the various listings available on the portal, potentially taking to AI-based tools that suggest the best unit for the customer needs.

1.1.2 Personas

In this subsection is presented the scenario of competence of the project and in particular the different types of Personas that could be the actors of this scenario. All the 4 different types of user presented are characterized by the need to look for an apartment or house to buy or rent in the geographical area of competence of the project having, however, different attitudes and consequently different needs.
Giuseppe is 24 years old, he is a student of industrial engineering at the University of Trento in Povo. Despite attending university in Povo, it would be very important for Giuseppe to find accommodation in the city in order to be close to the essential services and university social life, without neglecting the need to move with public transport every day to reach the university outside the city. Giuseppe is not able to have a constant source of income and his accommodation expenses are borne entirely by his family, for this reason he is looking for a very economical situation and like most of his fellow students he is easily satisfied with a place in a room shared by other students and since he does not have a private transport he is not looking for an apartment with a private or public parking nearby. In choosing the apartment to share his priorities are guided by his university life and his hobbies. He would prefer to find accommodation in a home away from elementary schools and kindergartens and possibly away from the railway to have a quiet place to study. To be effective in the studio Giuseppe would need the house to be equipped with a broadband internet connection allowing him to increase his productivity in studying together with quick access to the many online multimedia content he watches in his spare time. Giuseppe also has a passion for fitness and is used to training 3 times a week. For this reason he would like to live near a gym.
Paola and Mario, respectively 43 and 45 years old, are the parents of Luigi and Angela twins of 8 years old and need to find a new housing near the city to get closer to the workplace. Paola is a lawyer in a firm based in the city center and prefers to move with his private vehicle having a covered parking near work, even his spouse Mario, a computer consultant, is used to go to work by car even if during the summer months would be inclined to take advantage of bike-sharing services offered in the city due to its strong sporting nature. For this reason the couple is looking for accommodation with a large garage but at the same time not too far from the bike-sharing services possibly. The couple's children are lively twins and for this reason Paola would like to find a house with a private garden and possibly in a quiet and safe area near the essential services for children (playground, elementary and middle school). Economically, the family is wealthy and prefers to spend an important amount of money for their new home as long as it meets their safety, size and proximity of services for children. Mario would also like the house to be located in a geographically interesting location for evening excursions, in fact he is a fan of running

and trekking and after work would have the pleasure of leaving home a couple of hours to keep trained and run in a natural and unpolluted environment.

Giovanna is 35 years old and works remotely for a large software house in Milan. Giovanna, having the possibility to stay at home and work, is used to change house periodically and would like to find an apartment in the surroundings of Trento, which is a very dear place for her family origins. The accommodation she is looking for is an apartment to rent where she can not only live with her beloved dog but also work remotely. For these reasons Giovanna is looking for an apartment in a quiet area of the city and has no needs for private or public parking as she is a lover of a green lifestyle and would prefer to have the essential services within walking or cycling. For this reason she is looking for a home that is well equipped with essential services such as supermarkets and pharmacies but would also like, given her innate passion for art and culture, that places of culture such as libraries, cinemas and theaters were easily accessible from her home. Given her working position, it is essential for Giovanna that her home be equipped with good network coverage and Internet connection. Last but not least, Giovanna loves outdoor sports and in particular she has been practicing climbing for a few years. For this reason a useful information when buying the house would be to have an overview of the rock cliffs from the nearest to the most distant in Trentino Alto Adige.

Luciana and Corrado are a loving elderly couple, she seventy-three years old and he seventy-five years old, retired. The couple are looking for a setting for their winter and summer vacations in Trentino-Alto Adige. The house they are looking for is not a classic vacation home, but a place where they can feel at home even when they go on vacation as they plan to spend almost a third of the year in this residence. The couple, being very well off, are looking for a spacious and comfortable house in a quiet neighborhood and as panoramic as possible because, although they have their own means of transport in Trentino, they want to relax and enjoy the peace of a small town. In the search for a house some features are very important to them. In addition to a house in a quiet area and with a private or public parking nearby they would like to find an accommodation well equipped with essential services and with some ad hoc services to make them feel at home. For Luciana and Corrado, given their strong faith, it is in fact important that the house is close to a church so that they can participate in Christian community life. In addition, they would like to have a few old people's homes nearby to spend a few days in company and participate in the activities organized by these centers. Luciana and Corrado have been passionate about winter sports since their youth and this was one of the reasons why I want to look for a house in Trentino-Alto Adige. In particular, in order to choose a house, they are interested in knowing where they are and what ski facilities are available within a few dozen kilometers from their future vacation home.

In the following a table that summarize the personas is presented.

Persona	**Description**	**Usage**
Giuseppe	24 y.o. engineering student that loves university social life and going to the gym but he hasn't a private car.	Giuseppe is looking for a student room in a quiet environment in the city (away from the railway and main roads) where he can study, with a fast internet connection and with the following facilities nearby: essential services (supermarket, bank, pharmacy); the bus stop 5/; a gym; library and nightclubs

Paolo and Mario	Paola and Mario, respectively 43 and 45 years old, are the parents of Luigi and Angela twins of 8 years old.	Paola and Mario are, together with their children, looking for a house in the city in an uncrowded place. When choosing a house, the couple carefully assess whether the services nearby meet their needs as parents and their hobbies. In particular, they would like to have them nearby: playground, elementary and middle schools, public parking and bike-sharing service to facilitate their mobility, routes and excursions to do after work
Giovanna	Giovanna is 35 years old and works remotely for a large software house in Milan. She loves culture.	Giovanna works from home, that's why she doesn't need a car and prefers to reach interesting places for her on foot or by bicycle. Besides having a good Internet connection and being located in a not too chaotic environment she would like to look for a house close to essential services (supermarkets, drugs, banks) and close to the places she prefers to attend in her free time: cinemas, theaters and theaters. To keep fit Giovanna practises climbing and it would therefore be very useful for her to know which cliffs are available in Trentino in order to plan her weekend adventures.
Luciana and Corrado	Luciana and Corrado are a loving elderly couple, she seventy-three years old and he seventy-five years old, retired. They serach for a vacation home.	Luciana and Corrado are looking for a vacation home that is in an uncrowded place. They would like the house to be close to a parking lot, churches and old people's homes. In addition, the couple has a passion for skiing and would like to know which skiing areas are near their home.

1.2 Inception

This section first describe the Competency Queries in a informal format, then the datasets collected and the relative metadata will be described.

1.2.1 CQs definition

In this section we try to think what our example users can ask to the knowledge graph. Based on the datasets selected, we formalized a complete list of competency queries. This queries describe the need of the people as write in the storytelling.

Person	**Number**	**Question**	**Action**
Giuseppe	1.1	Where is the closest library ?	The system given the unit position, returns the position of the nearest library
Giuseppe	1.2	How many library are there within a 5km radius ?	The system given the unit position, returns the number of library within the 5 km
Giuseppe	1.3	How far is the nearest supermarket ?	The system given the unit position, returns the distance of the nearest supermarket
Giuseppe	1.4	What kind of internet connection is available in the apartment?	The system given the unit position, returns the level of available internet connection
Giuseppe	1.5	Is there a pharmacy within a radius of 1Km?	The system given the unit position, returns the presence or absence of a pharmacy within a radius of 1Km

Giuseppe	1.6	Is the house in a quiet enough place ?	The system given the unit position, returns the major roads and railways closest to the house indicating the distance from the house.
Giuseppe	1.7	How far is it to the nearest bus stop where bus number 5 runs?	The system, given the location of the unit, returns the distance to the nearest bus stop where bus number 5 goes by.
Giuseppe	1.8	Where is the closest gym?	The system given the unit position, returns the position of the nearest gym
Giuseppe	1.9	How far is the nearest bank to the apartment?	The system given the unit position, returns the distance of the nearest branch for each bank
Giuseppe	1.10	Where is the nearest bar ?	The system given the unit position, returns the position of the nearest bar
Giuseppe	1.11	How far is the train station ?	The system given the unit position, returns the distance of the train station
Paola and Mario	2.1	Is the apartment in a quiet place?	The system given the unit position, returns the major roads and railways closest to the house indicating the distance from the house.
Paola and Mario	2.2	Where is the closest elementary school to the house?	The system given the unit position, returns the position of the nearest elementary school
Paola and Mario	2.3	How far is the nearest middle school?	The system given the unit position, returns the distance of the nearest middle school
Paola and Mario	2.4	How many playgrounds are there within 2 km from the house?	The system, given the location of the unit, returns the number of playing fields within 2 km and for each playground it lists the present games
Paola and Mario	2.5	Is the house in a crowded area?	The system given the unit position, counts the number of house in 3 km
Paola and Mario	2.6	Is there a bike-sharing service nearby?	The system given the unit position, returns the presence or absence of a bike-sharing service within a radius of 1Km and returns also the number of available bike in those racks
Paola and Mario	2.7	What are the bike paths within 1 km?	The system given the unit position, returns the bike paths within 1km
Paola and Mario	2.8	How many excursion or bike paths are there in the vicinity of the house (maximum 3 km)?	The system given the unit position, returns the number of different excursion and bike paths within 3 km.
Paola and Mario	2.9	What are the excursion paths within a radius of 3 km?	The system given the unit position, returns the excursion paths within 3km
Paola and Mario	2.10	Where is the closest parking lot to the house?	The system given the unit position, returns the position of the nearest parking lot
Paola and Mario	2.11	How far is the center of the city?	The system given the unit position, returns the distance of the city center.
Paola and Mario	2.12	Where is the closest middle school to the house?	The system given the unit position, returns the position of the nearest middle school
Paola and Mario	2.13	Which high schools are within a radius of 2 km ?	The system given the unit position, returns the the name of the high schools within a radius of two kilometers
Giovanna	3.1	Know what type of Internet is available in the apartment	The system given the unit address, return the level of available internet connection
Giovanna	3.2	I want to live in a quiet place. Where is the nearest railway line ?	The system given the unit position, search for the nearest piece of rail and return the distance between the unit and it
Giovanna	3.3	I want to live in a quiet place. Where are the nearest main roads ?	The system given the unit position, search for the nearest pieces of major road and return the distance between the unit and the type of this major road
Giovanna	3.4	Which are the nearest banks?	The system given the unit position, search the nearest bank branches, maximum one for each bank
Giovanna	3.5	What are the post offices and their timetables within a radius of 3 km ?	The system given the unit position, search for the post offices within 3 km and their opening hours
Giovanna	3.6	Where is the nearest ambulatory?	The system given the unit position, returns the position of the nearest ambulatory

Giovanna	3.7	Which are the supermarkets within a radius of 2 km ?	The system given the unit position, search the supermarkets within the 2km and return for each one the area and the name
Giovanna	3.8	How far is the nearest pharmacy?	The system given the unit position, returns the distance of the nearest pharmacies
Giovanna	3.9	Which libraries are the closest?	The system given the unit position, returns the library within 1 km
Giovanna	3.10	Where is the nearest cinema?	The system given the unit position, search for the nearest cinemas
Giovanna	3.11	Where is the nearest theaters?	The system given the unit position, search for the nearest theaters
Giovanna	3.12	What are the cliffs within a radius of 20 km ?	The system given the unit position, search for the nearest cliffs and return the position of the cliffs within of 20 km
Luciana and Corrado	4.1	I want to live in a quiet place. Where is the nearest railway line ?	The system given the unit position, search for the nearest piece of rail and return the distance between the unit and it
Luciana and Corrado	4.2	I want to live in a quiet place. Where are the nearest main roads ?	The system given the unit position, search for the nearest pieces of major road and return the distance between the unit and it
Luciana and Corrado	4.3	i want live in a small neighborhood	The system given the unit position, count the number of house in 5 km
Luciana and Corrado	4.4	How far is the nearest public parkings?	The system given the unit position, search for the nearest public parkings and return the distance between the unit and the parking. It also return the capacity of the parking.
Luciana and Corrado	4.5	Where is the closest church?	The system given the unit position, search for the nearest church and return the position of the church
Luciana and Corrado	4.6	Ho far is the nearest center for elderly?	The system given the unit position, search for the center for elderly and return the distance between the unit and the center for elderly
Luciana and Corrado	4.7	Which ski areas are available within a radius of 35 km ?	The system given the unit position, search for ski area within 35 km and return the position and the distance between the unit and the ski area

Each query consult one or more type of data. The types of data needed to resolve the query and the connection between each type and the query that use it are given in the following table:

Types:	**Properties:**	**Notes:**	**CQ using the type:**
Library	Position [coordinate]		**1:**1,2 **3:**9
Pharmacy	Position [coordinate], Opening hours [string]		**1:**5 **3:**8
School	Position [coordinate], Name [string], Type [enum]		**2:**2,12,13
Sports Facility	Position [coordinate]		**1:**8
Bar	Position [coordinate]		**1:**10
Building	Position [coordinate]		**2:**5 **4:**3
Park	Position [coordinate], types of games [list of string]		**2:**4
Bike path	Initial Point [coordinate], End point [coordinate]	It combine dataset of province of Trento with that of the city of Trento	**2:**7,8
Excursion path	Initial Point [coordinate], End Point [coordinate]		**2:**8,9
City Center	Position [coordinate], name [string]		**2:**11
Major Road	Initial Point [coordinate], End Point [coordinate], Type [enum]		**1:**6 **2:**3 **3:**3 **4:**2
Post Office	Position [coordinate], Opening hours [string]		**3:**5
Ambulatory	Position [coordinate]		**3:**6
Theater	Position [coordinate]		**3:**11
Church	Position [coordinate]		**4:**5
Elderly center	Position [coordinate]		**4:**6
Ski area	Position [coordinate], Name [string]		**4:**7
Cinema	Position [coordinate], Name [string]		**3:**10
Bank Branch	Position [coordinate], Name of the bank [string]		**1:**9 **3:**4

Internet	Address [composed field], Level_at_2019 [enum])		**1**:4 **3**:1
Supermarket	Position [coordinate]		**1**:3 **3**:7
Train line	Initial Point [coordinate], End Point [coordinate]		**1**:6 **2**:1 **3**:2 **4**:1
Train Station	Position [coordinate]) [Connected to Linee Ferroviarie]		**1**:11
Bus Stop	Position [coordinate], Lines [list of string]	Lines are the bus lines that go through this stop	**1**:7
Bike-sharing	Position [coordinate], rack slots [int]		**2**:6
Cliff	Position [coordinate]		**3**:12
Car parking	Position [coordinate], capacity [int]		**2**:10 **4**:4

1.2.2 Initial Datasets description

The first source of datasets considered, *OPENdata Trentino*[6], is a big web portal with a lot of different public data of the province of Trento. From there we took these datasets:

- 257 datasets (for 149 municipalities) of the **locations and points of interest**[7]: for every town in the province of Trento is provided a list in json of the relevant public locations like schools, parks, libraries, parking lots etc. with some additional information attached (name, coordinates, address, descriptions, photos, telephone number) and a GEO version (not present in all towns) with a lot more locations but with less information (only name, coordinates, address). Because of the high number of datasets to download, it has been written a script in python to automate the operation, and another to manage the files.

- 7 datasets of **Bikesharing stations**[8]: the list in json of all the bikesharing stations in 7 towns of Trentino provided by the public transport service containing the name, position, address and the number of bike slots for every station.

- 3 datasets with **nursery**,[9] **elementary**[10] and **middle school**[11] information: the list of Trento's school with name, position, address, number of subscribers, number of staff for every school.

- piste_ciclabili.csv for the **bike trails**[12] in Trento with the position, type, address and the length of the paths.

- civici_web.json for **civic numbers**[13] of buildings in the Municipality of Trento.

However the **high school** data is not present in the openData Tretino website, and to complete the missed data we did scraping in *comuniecittà.it*[14] containing less but still relevant informations (private or public school information, study paths) and saving the content in csv format.

The second big source of datasets used to get a large amount of geo-spatial informations is *OpenStreetMap*[15], the biggest free license world map collaborative project. All the data were extracted using *Overpass Turbo*[16]: a tool to make query for specific data extraction from OpenStreetMap. The datasets extracted (all exported in geojson):

- **areaski.geojsoni:** all the areas where is possible to make ski activities in Trentino.

- **bank.geojson:** all the banks located in Trentino.

- **building.geojson:** the buildings polygons divided in municipality

- **busstops.geojson:** all the bus stops with coordinates, the operator and (sometimes) the bus line.

- **cinema.geojson:** the list of the Trentino's cinema with coordinates and some relevant information where present (3D available, contact informations, number of rooms and other)

- **citycenter.geojson:** provide informations about cities or villages boundary in Trentino and where each city centre is located.

- **climb.geojson:** the cliffs in Trentino where is possible to do climbing activities.

- **parking.geojson:** car parks with the capacity.

[6] dati.trentino.it
[7] dati.trentino.it/dataset?tags=luoghi+e+punti+di+interesse
[8] dati.trentino.it/dataset/stazioni-bike-sharing-emotion-trentino
[9] dati.trentino.it/dataset/scuole-dinfanzia
[10] dati.trentino.it/dataset/scuole-elementari
[11] dati.trentino.it/dataset/scuole-media-inferiori
[12] dati.trentino.it/dataset/piste-ciclabili-open-data
[13] dati.trentino.it/dataset/comune-di-trento-numeri-civici
[14] www.comuniecitta.it/scuole-secondarie-di-secondo-grado/comune-di-trento-22205
[15] www.openstreetmap.org
[16] overpass-turbo.eu

- **park.geojson:** all different types of public parks (also dog parks).
- **pharmacy.geojson:** list pf pharmacies with position and (where present) the timetables.
- **post_office.geojson:** post offices with position and (where present) the timetables
- **railway.geojson:** train stations and railways of Trentino.
- **roads.geojson:** all the roads of Trentino.
- **skislopes.geojson:** the ski slopes location in the Trentino.
- **supermarket.geojson:** supermarkets of Trentino.
- **trails.geojson:** list of path (open to all non-motorized vehicles) or trails of Trentino

In order to satisfy the request of a good internet connection of some personas, we took a dataset from *Infratel Italia s.p.a.*[17] with all the information about the availability and the **quality of internet**[18] per home in Trentino.

1.2.3 Datasets metadata documentation

- **locations and point of interest:**

Fields	Type	Description
Titolo	string	name of the location
address	string	street address
description	string	short description of the location
image	image	image of the location
email	string	official email of the location
phone number	string	official phone contact of the location
url	string	website of the location
info	string	short infos
gps	string	coordinates and accurate address

- **locations and point of interest (GEO version):**

Fields	Type	Description
id	int	identification code
coordinates	float[2]	latitude and longitude of the location
properties	object	main properties of the location (name, address)

- **bikesharing_METADATA.json:**

Fields	Type	Description
name	string	name of the bike station
address	string	geographical address of the bike station
id	string	identifier
bikes	int	realtime data with the number of the present bikes
slots	int	realtime data with the number of the free slots
totalSlots	int	total number of slot bikes
position	int[2]	coordinates of the station

- **piste_cliclabili_METADATA.json:**

Fields	Type	Description

[17] www.infratelitalia.it

[18] www.infratelitalia.it/archivio-documenti/documenti/esiti-consultazione-2020–conclusa-una-prima-analisi-dei-dati-forniti-dagli-operatori

WKT	list int	coordinates of the paths
tipo	string	name of the typology of path (not useful)
fumetto	string	name of the path
descrizione	string	name of the path (equivalent to 'fumetto')
tipologia	string	path dedicated only to cycle or also pedestrian
tratto isolato ciclabile	int	meters without path dedicated to cycle
tratto isolato ciclopedonale	int	meters without path dedicated to cycle or pedestrian
anno	int	building year of cycle path

- **nursery_elementary_and_middle_school_METADATA.json:**

Fields	**Type**	**Description**
WKT	string	position in WKT system
civico_alf	int	civic number
destra	string	address
sobborgo	string	suburb
scuola	string	name of the school
fumetto	string	name of the school
iscritti	int	number of subscribers
personale	int	number of staff
mail	string	e-mail information
telefono	string	telephone number

- **high_school_METADATA.json:**

Fields	**Type**	**Description**
name	string	name of the school
address	string	address
school type	string	private or public school
study paths	string	study path offers

- **internet_quality_trento_METADATA.json:**

Fields	**Type**	**Description**
cod_egonciv	int	EGON civic code
regione	string	Only the TRENTINO-ALTO ADIGE region
provincia	string	only Trento province
comune	string	town
frazione	string	fraction of the town
via	string	street address
civico	int	civic code
barrato	char	sub-civic code
class_19	string	2019 internet coverage
class_22	string	2022 internet coverage estimation

- **areaski_METADATA.json:**

Fields	**Type**	**Description**
@id	string	name that identifies a unique object
landuse	string	describes the primary use of areas of land
description	string	describes the type of object
area	string	date rarely existing (value - *yes*/*no*)
name	string	name of data
alt_name	string	Another name or names by which the feature is known
name:de	string	german name of data
name:it	string	italian name of data
name:lld	string	Ladin name of data
type	string	type of data

website	string	website
wikidata	string	The ID of the Wikidata item about the feature
ref:pat:cat	string	identification
ref:pat:id	string	identification
@relations	map	describes the commercial relations with other ski areas

- **bank_METADATA.json:**

Fields	**Type**	**Description**
@id	string	identification code
building	string	type fo building
name	string	name of the facility
type	string	type of element
atm	boolean	indicate the presence of an atm
automated teller machine	boolean	see atm
operator	string	indicate the operator of the facility
brand	string	indicate the brand of the facility
addr:housenumber	string	the house number component of the address
addr:street	string	the street component of the address
source	string	the source of the data
opening_hours	string	the opening hours of the facility
contact:website	string	the website where contact the facility
building:levels	integer	the level above ground of the facility
brand:wikidata	string	wikidata identifier code
brand:wikipedia	string	the wikipedia page of the bank branch
wheelchair	boolean	describe if it is accesible for wheelchair users
addr:city	string	the city component of the address
addr:postcode	string	the postcode component of the address
roof:levels	integer	the level of the roof
roof:shape	string	the shape of the roof
drive_through	boolean	describe if the facility offer drive through
addr:full	string	the full address of the facility
addr:country	string	the country component of the address
website	string	the website of the facility
phone	string	the phone of the facility

- **busstop_METADATA.json:**

Fields	**Type**	**Description**
@id	string	name that identifies a unique object
highway	string	name that identifies objective type
public_transport	string	name that denotes stop positions and platforms of public transport
name	string	name that identifies the name of the bus-stop
shelter	boolean	identifies the presence or absence of the shelter
bus	boolean	Access values describe restrictions or not of the bus in the bus-stop
bench	boolean	identifies the presence or absence of the shelter
alt_name:lld	string	another unofficial but locals feequntly name in local dialects
name:it	string	Italin name of the bus-stop
name:lld	string	Ladin name of the bus-stop
departures_board	string	name that identifies the type of time table or departures present in the bus stop
operator:old_name	string	name that identifies company who is in charge of the service at the bus stop
route_ref	liststring	name that identifies a list of all bus lines that serve the stop
bus_routes	liststring	name that identifies a list of Flixbus that serve the stop

created_by	string	name that identifies the computer program (editor or script) which made the changes
wheelchair	string	identifies if the bus stop is suitable to be used with a wheelchair and a person with a disability who uses another mobility device (like a walker)
note:tactile_paving	boolean	identifies if there is Tactile paving
covered:source	string	identifies how the information was collected
ref	liststring	name that identifies a list of all bus lines that serve the stop
name:de	string	German name of the bus stop
addr:city	string	name identifying the name of the city where the bus stop is located
addr:postcode	int	number identifying the postcode of the city where the bus stop is located
addr:street	string	name identifying the name of the street where the bus stop is located
alt_name	string	alternative name of the bus stop
amenity	string	Describes useful and important facilities for visitors and residents
bin	boolean	identifies presence or absence of a waste basket

- **cinema_METADATA.json:**

Fields	**Type**	**Description**
@id	string	Identifies univocally the object
amenity	string	Descirbes useful and important facilities for visitors and residents
building	boolean	Is used to mark a given object as a building
name	string	Identifies the name of the cinema
email	string	Identifies the e-mail ralated to the cinema
phone	string	Identifies the e-mail ralated to the cinema
source	string	Identifies the information source
website	string	Identifies the website related to the cinema
addr:city	string	Identifies in term of cites the position of the cinema
addr:country	string	Identifies in term of state the position of the cinema
addr:housenumber	int	Identifies the hosenumber of the cinema
addr:postcode	int	Identifies the postcode of the cites where the cinema is located
addr:street	string	Identifies the street name where the cinema is located
cinema:3D	boolean	Indentifies the possibility or not to watch 3D projection
contact:phone	string	Identifies the hosenumber of the cinema
contact:website	string	Identifies the website related to the cinema
drive_in	boolean	Indicate if a cinema is a drive-in cinema
operator	string	Indicate the main operator of the cinema
ref:vatin	string	VAT identification number of an object
screen	int	indentifies the number of screen
wheelchair	string	Identifies the possibility or not to access with a wheelchair

- **city_center_METADATA.json:**

Fields	**Type**	**Description**
admin_level	integer	the level of the administration roles tree
boundary	string	the type of boundary
name	string	the name of the area
ref:ISTAT	string	the reference to the ISTAT code
ref:catasto	string	the reference to the codice catastale
type	string	the type of the object
wikidata	string	link to the related wikidata object
wikipedia	string	the wikipedia page of the facility
@id	string	name that identifies a unique object

name:cim	string	the name in Cimbro
name:it	string	the name in Italian
population	string	the population of this administrative area
old_name:de	string	the old name in german
natural	string	describe the type of nature inside the boundary
name:de	string	the new name in german
name:mhn	string	the name in mócheno
alt_name:de	string	an alternative name in german
reg_name	string	the name of the region
name:lld	string	the name in ladin
postal_code	string	the postal code of the administrative area
old_name	string	the old name of the administrative area
name:left	string	the name from the left side
name:right	string	the name from the right side
source	string	the source of the information
created_by	string	who created this set

- **climb_METADATA.json:**

Fields	**Type**	**Description**
@id	string	Used to univocally identifies the object
climbing:rock	string	Identifies the typology of rock
name	string	Identifies the name of the climb
natural	string	Is used to describe wide variety of physical geography, geological and landcover features
sport	string	Is used to identify one or more sports which can be played
type	string	Is used to set a relation's type
building	string	Indicates the presence or abesnce and the type of a bulding.
leisure	string	is for places people go in their spare time.
addr:country	string	Indicates the city in the address information
addr:housenumber	string	Indicates the housenumber in the address information
addr:postcode	int	Indicates the postcode in the address information
addr:street	string	Indicates the postcode in the address information
email	string	Indicates the email ralted to the climb
fax	string	Indicates the fax ralted to the climb
opening_hours	string	Indicates the openning hours of the climb spot
phone	string	Indicates the phone ralted to the climb
tourism	string	Indentidies the specific interest to tourists including places to see, places to stay, things and places providing information and support to tourists.
website	string	Indicates the website ralted to the climb
climbing:bolted	string	Provides a first indication how well the routes are secured in terms of anchors
climbing:multipitch	string	Indicates if there are multi-pitch routes and how many multi-pitch routes there is
climbing:orientation	string	Identifies orientation of the rock/wall face.
climbing:quality	string	Indentifies the quality of the rock/ice.
climbing:sport	string	Indicates the practicable sports
fee	boolean	Indciates the presence or absensce of fee
outdoor	boolean	Indicate if the climb is outside or not
climbing:boulder	boolean	Indicates the presence or absence of boulder

- **park_METADATA.json:**

Fields	**Type**	**Description**
@id	string	name that identifies a unique object
leisure	string	type of facility inside
type	string	type of data
name	string	the name of the facility

wikidata	string	link to the related wikidata object
landuse	string	describe the usage of the land
access	string	describe the access description
operator	string	the operator of the facility
area	boolean	describe if the area do not have any physical boundary
source	string	the source of the data
alt_name	string	an alternative name for the park
short_name	string	a short name for the park
start_date	date	the opening date of the facility
amenity	string	this variable describe the facility present at the park
created_by	string	the person that created the facility in openstreetmap
old_name	string	the old name of the facility
barrier	string	indicate the type of barrier that limit the facility
tourism	string	indicate why this facility can be of particular interest for the tourist
wikipedia	string	the reference to the wikipedia page
ele	integer	the elevation on the sea level
natural	string	describe the type of nature inside the boundary
sport	string	describe one or more sport that can be played in the facility
surface	string	type of surface in the park
leaf_type	string	describe the type of leave in the facility
wheelchair	boolean	wheelchair service is present
baby	string	specify if the item is tag friendly

- **parking_METADATA.json:**

Fields	**Type**	**Description**
@id	string	identification code
amenity	string	For describing useful and important facilities for visitors and residents. Facilities include for example toilets, telephones, banks, pharmacies, prisons and schools.
fee	boolean	The fee tag is for specifying whether a fee is usually charged for a service, or for access.
parking	string	specify the type of parking facility.
access	string	describe restrictions on the use of highways and other transportation routes
park_ride	boolean	Park and ride facilities are parking lots with public transport connections that allow commuters and other people heading to city centres to leave their vehicles and transfer to a bus, rail system (rapid transit, light rail, or commuter rail), or carpool for the remainder of the journey.
surface	string	provide additional information about the physical surface
capacity	int	max capacity of the parking lot
name	string	name of the location
barrier	string	type of barrier of the parking lot
source	string	source of the data
capacity:disabled	boolean	Defines whether or not dedicated disabled parking spaces are available
highway	string	main key used for identifying any kind of road, street or path
operator	string	company of the parkig lot
supervised	string	indicates that there is a person who supervises the place
wheelchair	boolean	is wheelchair accessible?
service	string	describe details about types of 'service' roads, railways or waterways; or the service or services offered by a business.
layer	int	describe vertical relationships between crossing or overlapping features
landuse	string	describe the primary use of land by humans
addr:housenumber	int	civic code
addr:postcode	int	postal code of the location

addr:street	string	street address of the location
opening_hours	string	day and hours of opening
smoothness	string	a classification scheme regarding the physical usability of a way for wheeled vehicles, particularly regarding surface regularity/flatness.
website	url	wbesite of the location
motorcycle	string	Legal access restriction for motorcycles.

- **pharmacy_METADATA.json:**

Fields	**Type**	**Description**
@id	string	identification code
amenity	string	For describing useful and important facilities for visitors and residents. Facilities include for example toilets, telephones, banks, pharmacies, prisons and schools.
building	string	type of building
name	string	name of the location
addr:city	string	city of the location
addr:housenumber	int	civic code
addr:postcode	int	postcode of the location
addr:street	string	street address of the location
drive_through	boolean	do it has the drive through?
healthcare	string	type of healtcare service
dispensing	boolean	specify whether a pharmacy dispenses prescription drugs or not
opening_hours	string	day and hours of opening
operator	string	name of the company
ref:vatin	string	VAT identification code
wheelchair	boolean	is wheelchair accessible?
addr:country	string	country of the location
email	string	email of contact
phone	string	phone number contact
contact:website	url	website of the location
source	string	source of the data
website	url	website of the location

- **post_office_METADATA.json:**

Fields	**Type**	**Description**
@id	string	name that identifies a unique object
addr:city	string	city component of the address
addr:country	string	the country component of the address
addr:housenumber	string	the house number component of the address
addr:postcode	string	the postal code component of the address
addr:street	string	the street component of the address
amenity	string	this variable describe the usage of the building
building	boolean	describe if the facility is a building
contact:website	string	the website where contact the facility
name	string	the name of the facility
operator	string	the name of the facility operator
atm	boolean	if the facility have an atm
building:colour	string	the colour of the building
building:levels	integer	the number of above-ground levels of the buildings
building:part	string	describe a part of the building where the attribute are different
height	double	the height of the building
opening_hours	string	the opening hours of the facility
roof:colour	string	the color of the facility roof
roof:shape	string	the shape of the facility roof
brand	string	brand of the post office

brand:wikidata	string	wikidata identifier code
brand:wikipedia	string	brand associated with country
addr:suburb	string	the suburb component of the address
contact:fax	string	the fax to contact the facility
contact:phone	string	the phone to contact the facility
description	string	a description of the building

- **railway_METADATA.json:**

Fields	**Type**	**Description**
@id	string	identification code
addr:city	string	city of the locaiton
addr:postcode	string	postcode of the location
addr:street	string	street address of the location
building	string	type of building
name	string	name of te location
network	string	network administration name
operator	string	operator administration name
railway	string	railway or station classification
public_transport	string	type of public transport of the location
train	boolean	it's a trainstation
electrified	string	description of the object electrified
maxspeed	int	speed limit of the railway
name:de	string	name in German
passenger_lines	int	number of lines
railway:track_class	string	The Track class of the railway line
railway:traffic_mode	string	Lines dedicated to passenger trains (might have some exceptions, especially temporally i.e. freight only during 0200-0330).
ref	int	The reference number of the railway line the track belongs to.
usage	string	usage of the rail line
voltage	int	Voltage of the railway electrification system
bridge	string	What kind of bridge carrying the track is
layer	int	To describe the vertical relationship to other bridges and features.
start_date	string	Date the Interlocking Tower was put into service
embankment	boolean	Is the track elevated using earth or concrete dam
service	string	type of the track
tunnel	boolean	The track is in a tunnel

- **roads_METADATA.json:**

Fields	**Type**	**Description**
@id	string	identification code
highway	string	type of road (prymary, secondary, tertiary)
name	string	name of the road
oneway	boolean	is oneway
ref	string	road code
junction	string	type of junction (roundabaout)
surface	string	type of road surface (asphalt, paved)
lanes	int	number of lanes
maxspeed	int	speed limit
nat_ref	string	road code
toll	boolean	toll present
cycleway:both	boolean	have 2 way cycleway
old_ref	string	road code
bridge	string	it has a bridge
source:maxspeed	string	source of the speed limit data
maxweight	float	max weight in tons

tunnel	boolean	it has a tunnel
foot	boolean	can go by foot
tunnel:name	string	name of the tunnel
access	string	type of access
horse	boolean	horse can go
length	int	length of the road in mt
incline	string	road inclination infromation
tunnel:length	int	tunnel length in mt
disabled	boolean	the road is disable

- **skislopes_METADATA.json:**

Fields	**Type**	**Description**
@id	string	identification code
colour	color	graphic color
name	string	name of the ski slopes
network	string	network administration of the ski slopes
piste:type	string	type of the skiing available (nordic, downhill)
website	url	link of the website of the skislope
name:de	string	name in German
operator	string	operator of the skislope
distance	float	distance between skislopes
piste:difficulty	string	difficulty level of the skislope (easy, intermediate, advanced)
piste:grooming	string	tpye of skiing available (classing, skating)
source	string	source of the data
piste:name	string	name of the piste
sport	string	sport available in the location
bridge	boolean	it's present a bridge

- **supermarket_METADATA.json:**

Fields	**Type**	**Description**
@id	string	identification code
building	string	type of supermarket
name	string	name of the supermarket
opening_hours	string	days and hours of opening
payment:cash	boolean	accept cash
payment:debit_cards	boolean	accept debit card
shop	string	type of shop
source	string	source of data
addr:city	string	city of the location
addr:country	string	country if the location
addr:housenumber	int	civic code
addr:postcode	int	postcode
addr:street	string	street of the location
operator	string	operator of the supermarket (coop, conad, despar ...)
ref:vatin	string	value added tax identification number
brand	string	brand of the supermarket (coop, conad, despar ...)
brand:wikidata	string	wikidata identifier code
brand:wikipedia	string	brand associated with country
toilets:wheelchair	boolean	wheelchair toilets is present
wheelchair	boolean	wheelchair service is present
website	url	official website of the brand
addr:province	string	province of the location
addr:state	string	state code of the location
building:levels	int	number of levels of the building
parking	string	type of parking (underground)
toilets	boolean	it has toilets

- **trails_METADATA.json:**

Fields	**Type**	**Description**
id	string	name that identifies a unique object
cai_scale	string	Identifies the cai scale difficulty classification
distance	int	Identifies the distance of the trials
duration:backward	string	Identifies the duration of the backward
duration:forward	string	Identifies the duration of the backward
foot	boolean	identifies the possibility to complete the path by foot
from	string	Indicates the start position
highway	string	path which has been look for these dataset
mtb	bool	Indicates if the path is practicable of mtb
mtb:scale	int	Indicates the difficult scale of the mtb path
network	string	specifies the hiking route as a local route (=lwn)
note:project_page	string	Indicates the project page url
operator	string	The name of the organization that maintains the path
osmc:symbol	string	describe route symbol that is used as waymarker or on guideposts
sac_scale	int	A difficulty rating scheme for hiking trails.
source	string	where these data came from
to	string	where the path ends
trail_visibility	string	describes attributes regarding trail visibility (not route visibility) and orientation
type	string	type of the path
website	string	link to the website of the organization
@id	string	identification number
ascent	int	meters of ascent
descent	int	meters of descent
name	string	name of the path
surface	string	Surface values provide additional information about the physical surface of roads/footpaths.
bicycle	bool	affordable for byclicle

- **building_METADATA.json:**

Fields	**Type**	**Description**
id	string	identification code
amenity	string	For describing useful and important facilities for visitors and residents. Facilities include for example toilets, telephones, banks, pharmacies, prisons and schools.
building	string	The building tag is used to mark a given object as a building
name	string	name of the location
type	string	The key type is commonly used to set a relation's type
wikidata	string	The ID of the Wikidata item about the feature
@id	string	identification code
historic	string	used to identify features that are of historic interest
addr:city	string	city of the location
addr:country	string	country of the location
addr:housenumber	string	civic code
addr:postcode	string	postcode of the location
addr:street	string	street address
contact:email	string	email contact
contact:fax	string	fax contact
contact:phone	string	phone number contact
contact:website	url	website of the location
email	string	email contact
operator	string	used to name a company, corporation, person or any other entity who is directly in charge of the current operation of a map object.
ref:vatin	string	VAT code information

castle_type	string	used to distinguish between the various type of historic castles - stately vs defensive etc
building:levels	string	used for marking the number of above-ground levels of a building
building:use	string	describes what kind of function a building=* serves
roof:levels	string	For stating the number of levels in the roof of a building
access	string	describe restrictions on the use of highways and other transportation routes (railways, waterways), as well as facilities such as buildings, building entrances, amenities and leisure entities.
layer	string	one of several methods used to describe vertical relationships between crossing or overlapping features

- **civici_web_METADATA.json:**

Fields	**Type**	**Description**
civico_num	string	civic number (without slash)
civico_let	string	slash, if exist
civico_alf	string	numero civico con barra (se presente)
desvia	string	street description
strada	string	street code
cap	int	postal code
tipo_num	string	type of civic: 'principale' or 'secondaria' (language: italy)
tipo_en	string	type of civic: 'principal' or 'secondary' (language: english)
ingresso	string	entrance type: 'abitazione', 'cancello' or 'altro' (language: italy)
ingr_en	string	entrance type: 'abitation', 'gate' or 'other' (language: english)
fumetto	string	complete adress
url	string	not useful
sobborgo	string	Cadastral community

1.2.4 Datasets collection process

In order to download the 257 locations and point of interests datasets from open data Trentino we made a script in python to automate the process: the script "points_of_interests_download.py"[19] looks in the search page of the website with the tag "luoghi e punti di interesse"; starting from page 1 and going to 8, it looks all the datasets containing "luoghi" and "interesse" in the title, then for every dataset found it download in the dataset page the json, (half of the times) the GEO version of the dataset (which it has more records) and the metadata. After the download process it's been used "points_of_interests_reorganize.py"[20] to move the metadata files in the right folder.

Because of the missing high school data in open data Trentino, we made scraping in comuniecittá.it: the script "comuniecitta_download.py"[21] it take all the information in the main page (name, address, if it's private or public) and then for every school it take the study path information. Finally the script saves the data in csv format.

The information has been exported from **OpenStreenMap** thanks to **Overpass turbo**, a web based data mining tool for OpenStreetMap. The Overpass API query for each dataset are listed down here:

- **areaski.geojsoni:**

```
[out:json][timeout:25];
// fetch area \Trentino" to search in
area(3600045756)->.searchArea;
// gather results
(
  // query part for: \landuse=winter_sports"
  node["landuse"="winter_sports"](area.searchArea);
```

[19] https://github.com/UNITN-KDI-2020/GeoSpatial-Domain-project/blob/master/code/points_of_interests_download.py
[20] https://github.com/UNITN-KDI-2020/GeoSpatial-Domain-project/blob/master/code/points_of_interests_reorganize.py
[21] https://github.com/UNITN-KDI-2020/GeoSpatial-Domain-project/blob/master/code/comuniecitta_download.py

```
  way["landuse"="winter_sports"](area.searchArea);
  relation["landuse"="winter_sports"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

- **bank.geojson:**

```
[out:json][timeout:25];
// fetch area \Trentino" to search in
area(3600045756)->.searchArea;
// gather results
(
  // query part for: \amenity=bank"
  node["amenity"="bank"](area.searchArea);
  way["amenity"="bank"](area.searchArea);
  relation["amenity"="bank"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

- **busstops.geojson:**

```
[out:json][timeout:25];
// fetch area \Trentino" to search in
area(3600045756)->.searchArea;
// gather results
(
  // query part for: \highway=bus_stop"
  node["highway"="bus_stop"](area.searchArea);
  way["highway"="bus_stop"](area.searchArea);
  relation["highway"="bus_stop"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

- **building/*nameMunicipality*.geojson:**

```
  [out:json][timeout:25];
  // fetch area \Trentino" to search in
  ( area["name"="Provincia di Trento"]; )->.externalBoundary;
  ( area["name"="{}"]; )->.searchArea;
  (
    // query part for: \building=yes"
    node["building"="yes"](area.searchArea)(area.externalBoundary);
    way["building"="yes"](area.searchArea)(area.externalBoundary);
    relation["building"="yes"](area.searchArea)(area.externalBoundary);
  );
  // print results
  out body;
  >;
  out skel qt;
```

- **cinema.geojson:**

```
[out:json][timeout:25];
// fetch area \Trentino" to search in
area(3600045756)->.searchArea;
// gather results
(
  // query part for: \amenity=cinema"
  node["amenity"="cinema"](area.searchArea);
  way["amenity"="cinema"](area.searchArea);
  relation["amenity"="cinema"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

- **citycenter.geojson:**

```
// fetch area "Trentino" to search in
( area["name"="Provincia di Trento"]; )->.searchArea;
(
  // query part for: "building=yes"
  node["admin_level"=8](area.searchArea);
  way["admin_level"=8](area.searchArea);
  relation["admin_level"=8](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

- **climb.geojson:**

```
[out:json][timeout:25];
// fetch area \Trentino" to search in
area(3600045756)->.searchArea;
// gather results
(
  // query part for: \sport=climbing"
  node["sport"="climbing"](area.searchArea);
  way["sport"="climbing"](area.searchArea);
  relation["sport"="climbing"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

- **park.geojson:**

```
[out:json][timeout:25];
// fetch area \Trentino" to search in
area(3600045756)->.searchArea;
// gather results
(
  // query part for: \leisure=park"
  node["leisure"="park"](area.searchArea);
  way["leisure"="park"](area.searchArea);
  relation["leisure"="park"](area.searchArea);
  // query part for: \leisure=playground"
  node["leisure"="playground"](area.searchArea);
  way["leisure"="playground"](area.searchArea);
  relation["leisure"="playground"](area.searchArea);
  // query part for: \leisure=dog_park"
  node["leisure"="dog_park"](area.searchArea);
  way["leisure"="dog_park"](area.searchArea);
  relation["leisure"="dog_park"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

- **parking.geojson:**

```
[out:json][timeout:25];
// fetch area \Trentino" to search in
area(3600045756)->.searchArea;
// gather results
(
  // query part for: \amenity=parking"
  node["amenity"="parking"](area.searchArea);
  way["amenity"="parking"](area.searchArea);
  relation["amenity"="parking"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

- **pharmacy.geojson:**

```
[out:json][timeout:25];
// fetch area \Trentino" to search in
area(3600045756)->.searchArea;
// gather results
(
  // query part for: \amenity=pharmacy"
  node["amenity"="pharmacy"](area.searchArea);
  way["amenity"="pharmacy"](area.searchArea);
  relation["amenity"="pharmacy"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

- **post_office.geojson:**

```
[out:json][timeout:25];
// fetch area \Trentino" to search in
area(3600045756)->.searchArea;
// gather results
(
  // query part for: \amenity=post_office"
  node["amenity"="post_office"](area.searchArea);
  way["amenity"="post_office"](area.searchArea);
  relation["amenity"="post_office"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

- **railway.geojson:**

```
[out:json][timeout:25];
// fetch area \Trentino" to search in
area(3600045756)->.searchArea;
// gather results
(
  // query part for: \railway=rail"
  node["railway"="rail"](area.searchArea);
  way["railway"="rail"](area.searchArea);
  relation["railway"="rail"](area.searchArea);
  // query part for: \railway=station"
  node["railway"="station"](area.searchArea);
  way["railway"="station"](area.searchArea);
  relation["railway"="station"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

- **roads.geojson:**

```
[out:json][timeout:25];
// fetch area \Trentino" to search in
area(3600045756)->.searchArea;
// gather results
(
  // query part for: \highway=trunk"
  node["highway"="trunk"](area.searchArea);
  way["highway"="trunk"](area.searchArea);
  relation["highway"="trunk"](area.searchArea);
  // query part for: \highway=primary"
  node["highway"="primary"](area.searchArea);
  way["highway"="primary"](area.searchArea);
  relation["highway"="primary"](area.searchArea);
  // query part for: \highway=secondary"
  node["highway"="secondary"](area.searchArea);
  way["highway"="secondary"](area.searchArea);
  relation["highway"="secondary"](area.searchArea);
  // query part for: \highway=tertiary"
  node["highway"="tertiary"](area.searchArea);
  way["highway"="tertiary"](area.searchArea);
  relation["highway"="tertiary"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

- **skislopes.geojson:**

```
[out:json][timeout:25];
// fetch area \Trentino" to search in
area(3600045756)->.searchArea;
// gather results
(
  // query part for: \"piste:type"=downhill"
  node["piste:type"="downhill"](area.searchArea);
  way["piste:type"="downhill"](area.searchArea);
  relation["piste:type"="downhill"](area.searchArea);
  // query part for: \"piste:type"=nordic"
  node["piste:type"="nordic"](area.searchArea);
  way["piste:type"="nordic"](area.searchArea);
  relation["piste:type"="nordic"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

- **supermarket.geojson:**

```
[out:json][timeout:25];
// fetch area \Trentino" to search in
area(3600045756)->.searchArea;
// gather results
(
  // query part for: \shop=supermarket"
  node["shop"="supermarket"](area.searchArea);
  way["shop"="supermarket"](area.searchArea);
  relation["shop"="supermarket"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

- **trails.geojson:**

```
[out:json][timeout:25];
// fetch area \Trentino" to search in
{{geocodeArea:Trentino}}->.searchArea;
// gather results
(
  // query part for: \highway=path"
  node["highway"="path"](area.searchArea);
  way["highway"="path"](area.searchArea);
  relation["highway"="path"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

1.2.5 Inception level evaluation

Considering the dimension of the group on the work on this project (4 persons), we had maintained a near-constant communication between the knowledge group and the data scientist group. This result that before developing the CQs both the components of the knowledge team have a very good knowledge of the datasets available. This has permitted to model the query based on it, with only a couple of queries out of 43 developed CQs (around 4%) that we had to rewrite to fit the metadata available inside the dataset.

Our main concern is instead in the quality of the data we collected, as an example while the metadata inside the data objects is present for near all the collected data, other possible useful metadata are at end available only for one or two data row.