Università di Genova
Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi

The role of geographic knowledge in sub-city level geolocation algorithms

Laura Di Rocco

Theses Series DIBRIS-TH-2018-1

Università di **Genova**

Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi

Ph.D. Thesis in
Computer Science and Systems Engineering
Computer Science Curriculum

The role of geographic knowledge in sub-city level geolocation algorithms

by

Laura Di Rocco

December, 2018

Ph.D. Thesis in Computer Science and Systems Engineering (S.S.D. INF/01)
Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi
Università di Genova

Candidate
Laura Di Rocco
laura.dirocco@dibris.unige.it

Title
The role of geographic knowledge in sub-city level geolocation algorithms

Advisors
Giovanna Guerrini
DIBRIS, Università di Genova
giovanna.guerrini@unige.it

Michela Bertolotto
Department of Computer Science, University College Dublin
michela.bertolotto@ucd.ie

External Reviewers

Location
DIBRIS, Univ. di Genova
Via Opera Pia, 13
I-16145 Genova, Italy

Submitted On
December 2018

"Give the girl the right shoes and she can conquer the world."

— Marilyn Monroe

Dedicated to my beautiful colleagues that were fundamental to reach the end of this Ph.D.: Sara, Andrea, Ulde, Samu, Fede, Tommi, Angelo, Vero, Vane, Chiara, Luca.

Abstract

Geolocation of microblog messages has been largely investigated in the literature. Many solutions have been proposed that achieve good results at the city-level. Existing approaches are mainly data-driven (i.e., they rely on a training phase). However, the development of algorithms for geolocation at sub-city level is still an open problem also due to the absence of good training datasets. In this thesis, we investigate the role that external geographic knowledge can play in geolocation approaches. We show how different geographical data sources can be combined with a semantic layer to achieve reasonably accurate sub-city level geolocation. Moreover, we propose a knowledge-based method, called Sherloc, to accurately geolocate messages at sub-city level, by exploiting the presence in the message of toponyms possibly referring to the specific places in the target geographical area. Sherloc exploits the semantics associated with toponyms contained in gazetteers and embeds them into a metric space that captures the semantic distance among them. This allows toponyms to be represented as points and indexed by a spatial access method, allowing us to identify the semantically closest terms to a microblog message, that also form a cluster with respect to their spatial locations. In contrast to state-of-the-art methods, Sherloc requires no prior training, it is not limited to geolocating on a fixed spatial grid and it experimentally demonstrated its ability to infer the location at sub-city level with higher accuracy.

Publications

Conference Proceedings

Di Rocco, Laura (2016). "Semantic Enhancement of Volunteered Geographic Information." In: *In Doctoral Consortium in International Conference of the Italian Association for Artificial Intelligence - DC@AI*IA*.

Di Rocco, Laura, Michela Bertolotto, Davide Buscaldi, Barbara Catania, and Giovanna Guerrini (2019). "The role of Geographic Knowledge for sub-city level Geolocation". In: *GeoInformation Analytics at ACM/SIGAPP Symposium On Applied Computing - GIA@SAC (To appear)*.

Di Rocco, Laura, Michela Bertolotto, Barbara Catania, Giovanna Guerrini, and Tiziano Cosso (2016). "Extracting Fine-grained Implicit Georeferencing Information from Microblogs Exploiting Crowdsourced Gazetteers and Social Interactions". In: *AGILE International Conference on Geographic Information Science*.

Di Rocco, Laura, Roberto Marzocchi, Barbara Catania, Tiziano Cosso, and Giovanna Guerrini (2017). "Exploiting multiple heterogeneous data sets for improving geotagging quality". In: *Symposium on Advanced Database Systems - SEBD*.

Mauro, Noemi, Laura Di Rocco, Liliana Ardissono, Michela Bertolotto, and Giovanna Guerrini (2018). "Impact of Semantic Granularity on Geographic Information Search Support". In: *IEEE/WIC/ACM International Conference on Web Intelligence - WI*. Santiago de Chile.

Robino, Camilla, Laura Di Rocco, Sergio Di Martino, Giovanna Guerrini, and Michela Bertolotto (2018a). "A Visual Analytics GUI for Multigranular Spatio-Temporal Exploration and Comparison of Open Mobility Data". In: *Information Visualization - IV*.

– (2018b). "Multigranular Spatio-Temporal Exploration: An Application to On-Street Parking Data". In: *Web and Wireless Geographical Information Systems - W2GIS*.

Acknowledgements

I want to say thank you to all the people that helped me both professionally as well as personally. First of all, Giovanna and Michela, my supervisors, for their advice and continuous support. No less important, my unofficial supervisor and co-author Barbara. She was the best reviewer that we could have.
During my Ph.D. I had the possibility to visit three diferent universities. In all these experiences I learnt a lot both from a personal and a professional point of view. A first thank you is to Themis Palpanas and Pavlos Paraskevopulos for providing me the opportunity to study with them. My second thank you is for Davide Buscaldi who gave me my second experience in France. This helped me conclude my Ph.D. I also have to thank Noemi Mauro and Liliana Ardissono for our collaboration in Dublin on their project on recommender systems.

A special thank you is for Sergio Di Martino, my co-author on the visualization project. It was a pleasure to have the possibility to work with him.

I want to say thank you to "my students" Camilla and Federico. It was an honor being able to help them with their master's theses.
Last but not least, this thesis would never have existed without my incredible DIBRIS colleagues and without the support of my external colleague Kostas.

Contents

List of Figures

List of Tables

Introduction

CONTEXT AND MOTIVATION Microblog data are being produced at unprecedented speeds, causing a data deluge that makes their efficient mining an overwhelming task. These data represent a valuable source for the extraction of new types of information patterns and knowledge. Their multifaceted nature is being exploited to better understand social dynamics and propagation of information. One of the facets that analysts are interested in exploring is the semantics behind each message, i.e., what does it refer to? Another one is the spatial component, i.e., where does it come from? These facets are most of the times not independent. For example, the thing that a post is referring to most probably has a location in space and inferring this location has been a very active research topic over the recent years. This intense interest is mainly due to the large amount of applications involved. For example, knowledge of the positions of microblog messages enables the detection of outbreaks of diseases and emergency situation response. Moreover, microblog message mining has recently gathered a lot of attention as a viable approach for identifying social trends and even predicting various physical and social phenomena (Castillo, 2016; Gelernter, Ganesh, et al., 2013). Other applications also include flash mobs, or short-term event recognition.

However, one of the greatest challenges is that they rely on explicitly geotagged messages. Explicitly geotagged messages contain a set of coordinates that indicate where the user was located when the message was sent. Nevertheless, for either technical, privacy or energy management reasons (deactivated location services), a large fraction of the messages do not include any such spatial coordinates. It is indicative that only a small percentage of microblog messages (e.g., only 1% of Twitter messages (Graham, Hale, and Gaffney, 2014)) is explicitly geotagged. This is a significantly disruptive for a large part of the proposed methods, as they rely on already geotagged messages in order to provide accurate insights. Nevertheless, recent research is exploring the path of inferring these coordinates by analysing the content of the message (e.g., if it contains names of specific places, etc.). To exploit the intuition that users often mention places that are near to their current location, several approaches attempt to automatically geolocate nongeotagged messages using their textual content (Eisenstein, O'Connor, et al., 2010; Gelernter, Ganesh, et al., 2013; Hulden, Silfverberg, and Francom, 2015; Zhang and Gelernter, 2014).

Most of these methods rely on a training phase, during which they construct language models, in order to probabilistically infer the location of unseen messages. These types of models can very accurately geolocate microblog messages at a city level (Eisenstein, Ahmed, and Xing, 2011; Hulden, Silfverberg, and Francom, 2015)(but suffer from problems related to text noise for sub-city level geolocation. This is mainly due to the fact that messages do not only

contain proper natural language, but they also often contain mis-spellings, links, etc. As a result, such methods are prone to overfitting noisy data and as a result fail to identify the fine-grained features required for sub-city level geolocation. Moreover, during classification, the finer-grained the grid used to geolocate is (i.e., the higher sub-city detail), the higher the number of classification classes. This situation significantly harms performance. For the aforementioned reasons, while there has been a lot of recent work in the field (Cheng, Caverlee, and Lee, 2010; Eisenstein, Ahmed, and Xing, 2011), most methods presented so far operate best at the city level, i.e., are only able to infer the city from within which a tweet came.

To alleviate the problem of noise in the data, several methodologies try to address the problem of geolocating microblog messages by looking at explicitly mentioned location information in the message content (Gelernter, Ganesh, et al., 2013; Gelernter and Mushegian, 2011; Zhang and Gelernter, 2014). However, these solutions require large training sets in order to be able to achieve good results. Such training sets are notoriously difficult to construct, not only because pre-processing requires time, but most importantly because it requires people that are willing to manually label them. Finally, the majority of those methods relies on authoritative data sources. This is good solution but, in general authoritative data source does not contains a lot of fine-grained information.

INTUITION AND MAIN CONTRIBUTION Our intuition is that, to alleviate these problems, one has to rely on external information sources (Di Rocco, Bertolotto, Catania, et al., 2016). We claim that in order to provide an accurate sub-city level geolocation pipeline, data-driven methods should be only used at the first stage to reach city-level geolocation. However, as such methods fail when higher detail is required, external information sources should be exploited to reach sub-city accuracy. This approach is facilitated by the fact that there have been a plethora of geographic information sources developed recently, which are publicly available on the web.

In this thesis, we take a different approach and propose a knowledge-based method for inferring the location of a microblog message. Our method is complementary to existing research both in the goal (i.e., improving performance at the sub-city level only) and in the fact that it requires no prior training. This is because it does not perform prediction relying on annotated data, but instead, it performs location inference by analyzing the contents of each specific message. To achieve that it utilizes external knowledge in the form of semantic geospatial data, targeted to a given geographical area of interest. We propose an algorithm that, relying only on the information contained in a single message, can infer, with a reasonable accuracy, where this message came from.

Our approach is based on the following intuitions:

1. People tend to contextualize their messages by providing a semantic description of geographical objects. For example, a Twitter message like:

"Notre Dame De Paris. A masterpice of architecture and sculpture."

contains the information about *Notre Dame De Paris* underlining the fact that is an important piece of architecture.

2. Attributes of geographical objects, i.e., labels, have an important semantic aspect that characterizes the object itself. For example, Starbucks is the label of a geographical object that is also a cafeteria.

3. Geographical objects in a target area are often geographically clustered based on their semantics. For example, in a city, specific areas are full of restaurants or are mainly shopping districts.

4. In a city, in general, specific neighborhoods share common names. For example, *Liberty Island* in New York City, USA, is the island with the well known *Statue of Liberty*. In this island, we have the *Statue of Liberty Museum*, the *Statue of Liberty café* and so on.

5. A target area could be characterized by a specific Point of Interest that is of extraordinary importance for this location. For example, in general, there is only one important monument in a given neighborhood.

Based on these intuitions, we develop an approach that allows for generic multi-faceted filtering to be performed in an efficient way. The approach captures different facets that correspond to the semantic and spatial aspects of microblog data. For the semantic aspect, we use the external knowledge bases studied and we transform these semantic gazetteers into a metric space, maintaining their hierarchical structure as one aspect and use location of their toponyms in the physical world as the second aspect. The structured semantic dimension is embedded in a metric space and indexed by a spatial data structure. An efficient algorithm identifies the location corresponding to a message by looking at the semantics of geographical terms in the message and their respective locations in the physical world.

In this thesis, we start by studying diverse geographic information sources and their effect on geolocation accuracy. Specifically, we analyze both semi-authoritative data sources such as Geonames [1], and crowdsourced geographical data, such as OpenStreetMap [2] (OSM). Our second intuition is that an additional semantic level (e.g., an ontology) will allow us to increase the accuracy of geolocation estimation. While typically authoritative and semi-authoritative data repositories contain an associated semantic level, this is often not the case for crowdsourced data (e.g., OSM). For this reason, we examine a semantically enriched version of OSM, called LinkedGeoData (Stadler et al., 2012) (LGD), as well as an external ontology for conceptualized cities called OpenStreetMap Facet Ontology (Di Rocco, 2013, 2016).

[1] http://geonames.org
[2] http://openstreetmap.org

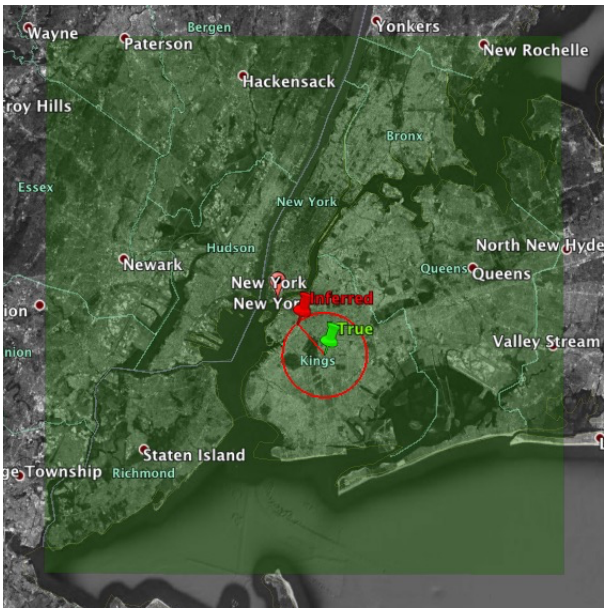

FIGURE 0.1. An example of geolocation inference made by Sherloc.

Our algorithm is called Sherloc [3]. It takes as input a microblog message and returns its coordinates at the sub-city level, i.e., with an accuracy which is within a given percentage of the area covered by the city. In Fig. 0.1, we show an example of a tweet geolocated by Sherloc. The red circle is the area that represents the inferred location contrasted to the green point that is the real location of the message. The highlighted light green bounding box is the target area taken into account.

EXPERIMENTAL RESULTS The experimental evaluation, made on Twitter datasets, demonstrates that our approach outperforms the state-of-the-art when working at a sub-city level. We demostrate, by the end of this thesis, that data-driven approaches on twitter location prediction are not the best methods to solve sub-city level location prediction problems. Instead, the idea to use a knowledge-driven method give very good results.

Indeed, through our experimental evaluation we show that, while knowledge-based methods have a lower recall, as they exploit the explicit geographical mentions in the text, they have a much higher precision for the subset of messages that contain such explicit mentioned terms. Thus, we validate our intuition that knowledge-based methods can be used as an additional step after a data-driven pipeline, in order to increase accuracy.

SUMMARY OF CONTRIBUTIONS The thesis main contributions are:

- The use of geographic data sources combined with a semantic level in order to structure an external geographical knowledge that describes a city at fine-grained level.

[3] A simple word pun between Sherlock Holmes and location.

- The representation of a geographical knowledge using embeddings to find the most similar terms in an easy and efficient way.
- The definition of an algorithm, Sherloc, that exploits semantic geographic knowledge for geolocating messages inside a target area, thus enabling geolocation inference for microblog messages without prior training (knowledge-driven approach).
- The experimental evaluation of Sherloc using different geographic knowledge bases, showing that it can achieve a sub-city mean distance error and accuracy distance error geolocation inference that cannot be obtained by data-driven algorithms.

OUTLINE The thesis is organized as follows:

- In Chapter 1, we introduce some preliminaries. In this chapter, we present all the basic notions useful to understand the thesis. We discuss geospatial data in general, we define the main terminology involved in the geolocation field. We then introduce the domain of this thesis: microblog data. At the end of the chapter, we briefly discuss two relevant techniques exploited in the thesis: embeddings and clustering.
- In Chapter 2, we discuss the state-of-the-art in the two fundamental fields dealt by the thesis: geospatial knowledge and microblog location inference. We discuss in detail, the literature on geospatial data and ontologies. Then, we discuss the state-of-the-art in microblog geolocation inference, with specific reference to Twitter.
- In Chapter 3, we formalize the geographic knowledge. We propose a specific format that a body of geographic knowledge must have to be used as input to our algorithm.
- In Chapter 4, we present Sherloc, our proposed algorithm to geolocate a microblog message. We provide detailed information on every step of the algorithm motivating every choice and proposing our specific instantiation for every step. Moreover, we propose an extended version of Sherloc, called Sherloc-DR that uses two different embeddings to represent the geographic knowledge.
- Chapter 5 introduces information about datasets, metrics and algorithms involved in the evaluation. Moreover, in this chapter, we provide details about the implementation.
- Chapter 6 contains the evaluation results. We start by evaluating geospatial knowledge. We then evaluate the embeddings. The previous evaluations provide us the best parameters to run Sherloc and Sherloc-DR. We finally comparatively evaluate both algorithms in terms of accuracy w.r.t. a state-of-the-art method.

In the end, Conclusions & Future Work conclude the thesis and provide an overview on the current limitations and future work directions.

1 Preliminaries

The thesis addresses the problem of geolocation of microblog message using geographic knowledge. This chapter has the goal to introduce all the preliminaries useful to understand the thesis. We start the discussion focusing on geospatial data, what they are and how they can be defined. Then, we discuss geolocation and we introduce the definitions and the terminology used in the literature and in the thesis. We provide an overview on microblog data, the target of our geolocation and we introduce Twitter as the most famous example of a microblog. We conclude the chapter with an overview of the main exploited techniques, methods for constructing embeddings for hierarchical data, and clustering algorithms.

1.1 *An Overview of GeoSpatial Data*

Geospatial data have two components: a spatial component which specifies their location in space, and a nonspatial component which specifies other characteristics. For instance, for a road these characteristics could include the road name, the road type, the company in charge of road maintenance, the traffic volume, etc. These two types of information (spatial and non-spatial) are called the geometry and the attributes of the spatial datum under examination, respectively.

The characteristics of geospatial data and the different sources generating them, make them very relevant in the Big Data contest. Geospatial data are principally generated from authoritative institutions. These geographical data are called authoritative datasets. Nowdays, different types of geographical data exist, namely crowdsourced datasets (e.g., OpenStreetMap[1]).

1.1.1 Spatial Component: Geometry and Topology

Geometry and topology are specific properties of spatial data. Geometry is concerned with shape (point, line, or region), extension (coordinates defining the point, line, region), position (in a geographic reference system).

Topology studies the rules of relationships between the geometry objects: point, line and regions, independently of their coordinates (e.g., two entities intersect, are disjoint, tangent to each other etc.).

Topology is qualitative rather than quantitative (e.g., if two entities intersect, the shape, extension, and position of the intersection do not matter; if they are disjoint, the distance separating them does not matter, etc.). Topological relations are invariant under continuous transformations (we can deform the

[1] openstreetmap.org

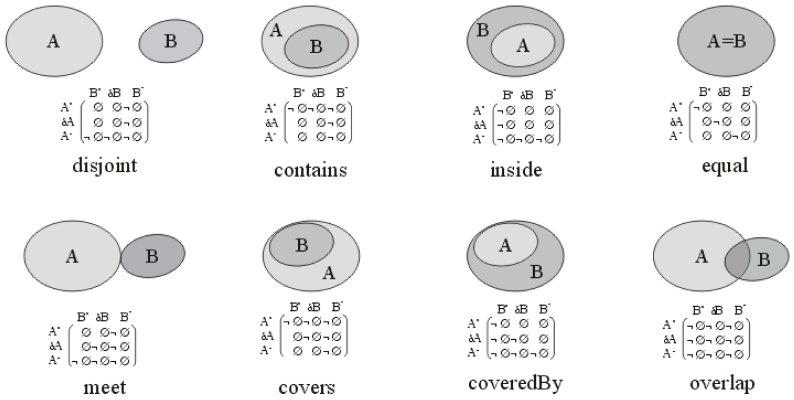

FIGURE 1.1. An interpretation of the eight relations between two regions with connected boundaries. Source: Egenhofer and Herring, (1991).

shape of the two entities while preserving the fact that they intersect, or they are disjoint).

The most popular model for representing topological relations between two spatial objects (i.e., two geometries in $\mathbb{R}^2$) is the 9-Intersection Model (9IM) (Egenhofer and Franzosa, 1991; Egenhofer and Herring, 1991).

The 9IM is based on a 3×3 intersection matrix defined as:

$$9\text{IM}(A, B) = \begin{bmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B & A^\circ \cap B^- \\ \partial A \cap B^\circ & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^\circ & A^- \cap \partial B & A^- \cap B^- \end{bmatrix}$$

where A and B are two spatial regions and A°, ∂A and A^- are respectively the interior, boundary and exterior of the region A (it is the same for B).

The dimension of empty sets ($\emptyset$) are denoted as F (false). The dimension of non-empty sets ($\neg\emptyset$) are denoted with the maximum number of dimensions of the intersection, specifically zero for points, 1 for lines, 2 for areas. Then, the domain of the model is $0, 1, 2, F$.

If we consider values empty/non-empty for the entries of the matrix, there are $2^9 = 512$ configurations, but some of them are not possible in practice. In Section 1.1.1, we show the common topological relations between two regions with connected boundaries: disjoint, contains, inside, equal, meet, covers, coveredBy, and overlap.

Notice that this is just a brief overview of geometry and topology of geospatial data. Detailed information can be found in Egenhofer and Franzosa, (1991) and Egenhofer and Herring, (1991). The examples and the summary are derived from Egenhofer and Herring, (1991).

1.1.2 NonSpatial Component

The attributes of geospatial data which do not describe the geometrical and topological aspects of the objects are called non-spatial components. This component can be in general classified as semi-structured data. It represents the additional information that we want to associate with geospatial data like name, type, etc.

In some geographic datasets, this information can be represented in the form of tags. Each tag has a key and a value. Tags can thus be written as `key=value`.

- The key describes a broad class of features (for example, highways or names).
- The value details the specific feature that is generally classified by the key (e.g., `highway=motorway`).

In some geographic datasets, there is no fixed list of tags. New tags can be invented and used as needed. Everybody can come up with a new tag and add it to new or existing objects. This makes these datasets enormously flexible, but sometimes also hard to work with. Other geographic datasets use an ontological level to describe the non-spatial components. In this case, geographical data has a semantic level to describe their relationships.

1.2 *Geolocation: Meaning and Terminologies*

Geolocation, also called geolocalization or geotagging, is the process of adding geographical information to various kinds of data such as photos, videos or documents (Viana, Filho, et al., 2008; Viana, Miron, et al., 2011). In this section, we focus on textual data and we provide an overview on the specific geolocation problem studied in this thesis and set the used terminology.

IMPLICIT AND EXPLICIT GEOGRAPHIC INFORMATION Data are said to be implicitly (or indirectly) geolocated (Hill, 2009) when they are not associated with explicit geospatial references (such as positioning on maps or spatial/physical coordinates), rather they are referenced by place names, geocodes, and addresses. The terms highlight the fact that additional steps are required to identify the locations on maps. Data are said to be explicitly geolocated if they are associated with a spatial/physical component.

Geolocating by place name is the most common form of referencing a geographic location and it is an informal means of georeferencing. We use place names in conversations, correspondence, reporting, and documentation. Dictionaries of placenames are called gazetteers (Goodchild and Hill, 2008). Gazetteers contain descriptive information about named places, which can include their geographic locations, types/categories, and other information.

Implicit geographic information has been exploited, for instance, for localizing news on maps (Teitler et al., 2008).

GEOLOCATION OF TEXTUAL DATA Geolocating data can be in form of points-of-interest (e.g., hotels or shopping mall) or simply geographical coordinates (latitudes and longitudes).

Focusing on geolocation of text, this consists of toponym recognition and toponym resolution. Toponym recognition is the process of finding a word/bag of words that should be a toponym. The latter, instead, is eliminating the geo/non-geo ambiguity (e.g., Washington can be a city in the USA or the name of a person). There are different strategies to do this:

1. finding names in the text that appear in a gazetteer (Amitay et al., 2004; Grover et al., 2010; Lieberman, Samet, and Sankaranarayanan, 2010; Lieberman, Samet, Sankaranarayanan, and Sperling, 2007; Samet et al., 2014);

2. using Name Entity Recognition techniques (Moncla et al., 2018; Purves et al., 2007; Stokes et al., 2008);

3. using a geographic ontology in order to understand the context of the text (Purves et al., 2007; Stokes et al., 2008).

In the rest of the thesis, we refer to our approach as a geolocation algorithm without focusing only on recognition or resolution.

ALTERNATIVE TERMINOLOGIES There are a lot of different terminologies used in the area. A lot of them share the same meaning or the differences are subtle and difficult to interpret.

Geolocation is a synonym of geotagging and geolocalization. Georeferencing is similar but there is a subtle difference. Georeferencing means that the internal coordinate system of a map or aerial photo image can be related to a ground system of geographic coordinates. Geolocation and georeferencing can achieve the same result: to obtain objects with coordinates and information related to them. The important difference is that geolocation is a post-process on objects and georeferencing is a process performed during the acquisition phase. This is why, when we talk about georeferencing, we can subdivide the data as implicitly/explicitly georeferenced and afterwards as implicitly/explicitly geotagged.

We highlight that geolocation algorithms are also called location inference or location prediction algorithms. This terminology helps to highlight the main goal of geolocation algorithms: prediction. In the next chapter, we refer to state of the art geolocation approaches as location prediction/inference algorithms. Notice that prediction and inference are synonyms that we will use interchangeably.

1.3 An Overview of Microblog Data

In this section, we provide an overview of microblog data and, then, we introduce Twitter as the most famous example of it.

MICROBLOGS Microblogging is an online broadcast medium that exists as a specific form of blogging. A microblog differs from a traditional blog in that its content is typically smaller in both actual and aggregated file size. Microblogs allow users to exchange small elements of content such as short sentences, individual images, or video links, which may be the major reason for their popularity. These small messages are sometimes called microposts.

As with traditional blogging, microbloggers post about topics ranging from the simple, such as "what I'm doing right now", to the thematic, such as "sports cars". Commercial microblogs also exist to promote websites, services, and products, and to promote collaboration within an organization.

Some microblogging services offer features such as privacy settings, which allow users to control who can read their microblogs or alternative ways of publishing entries besides the web-based interface. These may include text messaging, instant messaging, E-mail, digital audio or digital video.

The emergence of microblogging services is changing the form people share information on the web. People often access a social network such as Tumblr or Twitter to retrieve news, videos, or comments from their friends. In such systems, a large number of posts or tweets are posted every day. According to recent statistics[2] (October 2018), there are around 500 million of tweets per day and around 326 million of active users per month.

Due to the large volume of data available on microblogging sites, it is natural to consider the automatic methods of information extraction to capture semantic meaning of entities in the data. For example, in order to extract the abovementioned three types of information covered in our work (i.e., personal, social, and travel information), the most straightforward way is to treat personal information as attributes of entities, social information as relationships between entities, travel information as lists of location entities, and then directly apply traditional information extraction technologies to the collection of microblog posts. However, this attempt has been proven to be difficult and unsuccessful for the following reasons:

- *Ungrammatical Sentence*. Unlike documents on the web, posts on microblogging services are always length-limited. For example, Twitter allows users to post only 280-character messages. This length-limitation often leads posts to be noisy and ungrammatical, which makes traditional Natural Language Processing (NLP) tools such as Part Of Speech (POS) tagger inappropriate to use.
- *Informal Writing*. Microblog posts often contain noisy texts such as abbreviations, symbols, and misspellings, which consequently brings great difficulties in analyzing the content and meanings of posts. The sentence "I don't knw wht s gona happen" is an example of a message which people briefly write in abbreviated form on a microblogging site.

[2]Source: omnicoreagency.com/twitter-statistics/

TWITTER AS A SPECIFIC EXAMPLE Twitter is an American online news and social networking service on which users post and interact with messages known as "tweets". Tweets were originally restricted to 140 characters, but on November 7, 2017, this limit was doubled for all languages except Japanese, Korean, and Chinese. Registered users can post tweets, but those who are unregistered can only read them. Users access Twitter through its website interface, through Short Message Service[3] (SMS) or mobile-device application software ("app").

Users may subscribe to other users' tweets—this is known as "following" and subscribers are known as "followers". Individual tweets can be forwarded by other users to their own feed, a process known as a "retweet". Users can also "like" (formerly "favorite") individual tweets.

As a social network, Twitter revolves around the principle of followers. When you choose to follow another Twitter user, that user's tweets appear in reverse chronological order on your main Twitter page. If you follow 20 people, you'll see a mix of tweets scrolling down the page: breakfast-cereal updates, interesting new links, music recommendations, even musings on the future of education.

Users can group posts together by topic or type by use of hashtags – words or phrases prefixed with a "#" sign. Similarly, the "@" sign followed by a username is used for mentioning or replying to other users. To repost a message from another Twitter user and share it with one's own followers, a user can click the retweet button within the Tweet.

In late 2009, the "Twitter Lists" feature was added, making it possible for users to follow *ad-hoc* lists of authors instead of individual authors.

The tweets were set to a constrictive 140-character limit for compatibility with SMS messaging, introducing the shorthand notation and slang commonly used in SMS messages. The 140-character limit also increased the use of URL shortening services such as bit.ly, goo.gl, tinyurl.com, tr.im, and other content-hosting services such as TwitPic, memozu.com, and NotePub to accommodate multimedia content and text longer than 140 characters. Since June 2011, Twitter has used its own t.co domain for automatic shortening of all URLs posted on its site, making other link shorteners unnecessary for staying within Twitter's 140 character limit.

Due to the Twitter structure, we can extract three types of information:

- *Content*: short and noisy tweets, e.g., "I don't knw wht s gona happen". Tweets have 280 characters length. They can contain everything that a user wants: mood, events, news, etc. A user can also retweet others (s)he read. When a user composes a tweet (s)he can include hashtags or mentions. A mention is notified to another user and it is possible to start a conversation.
- *Network*: a massive Twitter network established among users. A user can follow or be followed by another user. If a user u_i follows a user u_j, we call

[3]Not available for every country.

u_i the follower and u_j the followee. It is important highlight that the *follow* relation is undirectional, i.e., if u_i follows u_j it is not necessary that u_j follows u_i. If this happens, they become mutual friends. Differently from other social media, being a friend in Twitter does not imply being a friend in real life. Some of the users represent official accounts of celebrities or journals, etc. It also happens that some users are bots.

- *Context*: rich types of contextual information for both users and tweets. A tweet is not only a message. It has different metadata attached to it. An important piece of information is the timestamp. Another one is location if the device has the GPS enabled. If timestamp always exists, location depends on the device and privacy settings. Moreover, a user can declare profile information like his/her home, personal website, etc. Therefore, context information can be summarized as follows: user profile information (declared home location that it is optional), third-party sources (Foursquare, Yelp, etc.), timezone, tweet timestamp and location.

In this thesis, we exploit only Twitter content. Notice that state of the art approaches exploit also the other information. We will discuss this in the next chapter.

1.4 Exploited Techniques

In this section, we provide an overview of the techniques that we use in the algorithm that we present in Chapter 4. We start with an introduction to word embeddings, which we use for representing geospatial ontologies. We then discuss clustering algorithms, which are a central part of our geolocation algorithm.

1.4.1 Word Embeddings

Here, we explain word embeddings first, and then we define some interesting state-of-the-art embedding algorithms.

WORD EMBEDDING VS VECTOR SPACE MODEL A word embedding, sometimes called word representation, is a collective name for a set of language models and feature selection methods. Its main goal is to map words into a low-dimensional continuous space. Conceptually it involves a mathematical embedding from a space with one dimension per word to a continuous vector space with a lot of fewer dimensions.

The most used methods to represent text data are Vector Space Models (VSMs) where a vector is the representation of a document. Information Retrieval proposed models to facilitate text categorization (Yang and Pedersen, 1997). Then the concept of word embedding using Neural Networks Language Models was introduced late in 2000 (Xu and Rudnicky, 2000). These approaches belong to the classes of unsupervised learning methods.

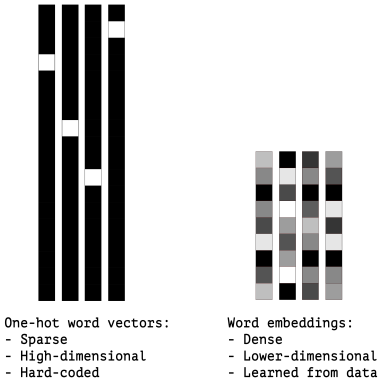

FIGURE 1.2. Graphical representation of one-hot word vectors model and word embeddings.

Neural networks take as an input a numeric tensor. Therefore we need to process raw data transforming them in a tensor. To do so, we can apply three different techniques:

- Segment text into words, and transform each word into a vector.
- Segment text into characters, and transform each character into a vector.
- Extract n-grams of words or characters and transform each n-gram into a vector.

In general, the different units that you can choose to break down text are called *tokens* and the process is called *tokenization*.

The easiest way to represent text through a sparse vector is called one-hot encoding. It consists of associating a unique integer index with every word and then turning this integer index i into a binary vector of size N (the size of the vocabulary). The vector contains all zero except for the ith entry, which contains 1.

Another popular and powerful method is *word vectors*, also called a *word embedding*. Vectors obtained with one-hot encoding are binary, sparse and high-dimensional (same dimensionality of the number of words in the vocabulary). Vectors obtained with a word embedding are low-dimensional floating-point vectors, i.e., dense vectors. Word embeddings, unlike the word vectors obtained with one-hot encoding, are learned from data.

In Figure 1.2 the difference between one-hot encodings and word embedding is graphically illustrated.

There are two ways to obtain word embeddings:

1. Learning word embeddings jointly with the main task you were trying to solve, e.g., document classification or sentiment analysis.
2. Loading into your model word embeddings in conjunction with different machine learning tasks and apply them for the task that you are trying to solve. This one is called *pretrained word embeddings*.

There are multiple pre-trained word embedding vocabularies, which are ready to be used. Those are well known in the literature and freely available. We describe the most frequently used.

GLOVE GloVe, coined from Global Vectors, is a model for distributed word representation. The model is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. It was developed as an open-source project at Stanford.

WORD2VEC Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of hundreds of dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

Word2vec was created by a team of researchers led by Tomas Mikolov at Google. The algorithm has been subsequently analyzed and explained by other researchers. Embedding vectors created using the Word2vec algorithm have advantages compared to earlier algorithms such as latent semantic analysis.

FASTTEXT fastText is a library for learning word embeddings and text classification created by Facebook's AI Research (FAIR) lab. The model is an unsupervised learning algorithm for obtaining vector representations for words. Facebook makes available pre-trained models for 294 languages. Differently from word2vec and GloVe, fastText includes the sub-word information. To have this additional information, they split all words into a bag of n-gram characters in general in the size of 3-6. They aggregate all these sub-words together to create a whole word as a final feature. fastText uses Neural networks for word embedding. Details about fastTest can be found in Joulin et al., (2016).

HIERARCHICAL EMBEDDING Hierarchical embedding algorithm is a specific embedding algorithm used to represent hierarchical data. The algorithm is presented in Nickel and Kiela, (2017). They decide to use hyperbolic geometry to capture hierarchal properties of words, i.e., their semantics. To do it, they embedded words into a hyperbolic space. In this way, they can use properties of hyperbolic space to use distance to encode similarity and the norm of vectors to encode hierarchal relationships.

The end result is that fewer dimensions are needed in order to encode hierarchal information. They show that the algorithm is a good way to represent data that have a hierarchical structure, like ontologies.

We will see in more detail the algorithm and how we use it in Chapter 4.

1.4.2 Unsupervised Learning: Clustering

The goal of clustering is that of grouping similar objects together. Objects that are in the same group, called *cluster*, are grouped by some definition of *similarity*. Clustering is an unsupervised learning task. In machine learning there are, classicaly, two paradigms: supervised learning, i.e., we learn from labels and unsupervised learning, i.e., we learn without labels. In unsupervised learning, we learn properties or recurrent patterns from unlabeled samples.

Clustering is not a specific algorithm but a general task to solve. Algorithms differ on how they compute clusters, i.e., how they find the space partitions, and how they efficiently identify clusters.

Three main different algorithms have been proposed:

- K-means: one of the most intuitive algorithms.
- Hierarchical clustering: an algorithm that proposes as a solution a hierarchy of clusters for data.
- DBSCAN: a clustering algorithm that has the particularity of handling "noise" samples.

In the following we briefly present these three algorithms. As we will discuss and motivate in Chapter 4, among these algorithms DBSCAN is the most suitable for our context and it will be exploited in the final stage of Sherloc.

K-MEANS CLUSTERING One of the most common clustering algorithms is K-Means clustering.

Given a set of input data $D = \{x_i\}_{i=1}^{n}$, with samples represented as d-dimensional vectors $x_i \in \mathbb{R}^d$ $(\forall i = 1, \ldots, n)$, the goal of K-means is to partition the data space into K predefined non overlapping groups that contain similar samples.

One nice property of K-Means clustering is that the clusters will be strict, spherical in nature, and converge to a solution.

The K-means algorithm starts with random centroids. Then the points are assigned to the cluster having the nearest centroid. Successively, the centroids position are updated. At each step the value of centroid changes following a function that assures the convergence of K-means. Nevertheless, the algorithm may converge to a local minimum, achieving a suboptimal solution.

In Figure 1.3, an example of an application of this algorithm on 2-dimensional data is shown.

HIERARCHICAL CLUSTERING Hierarchical clustering is a general family of clustering algorithms that build nested clusters by merging or splitting them successively. This hierarchy of clusters is represented as a tree (or dendrogram). The root of the tree is the unique cluster that gathers all the samples, the leaves being the clusters with only one sample. In general, the merges and splits are determined in a greedy manner.

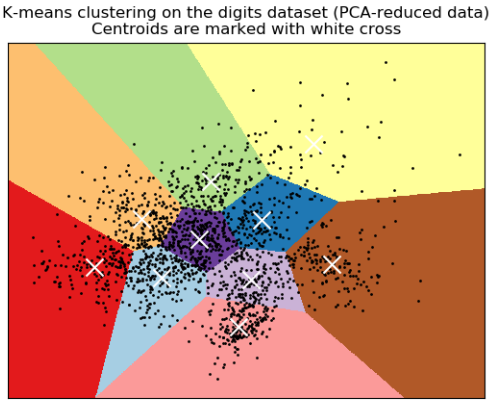

FIGURE 1.3. An example of k-means result. Source: scikit-learn.org

In data mining and statistics, hierarchical clustering (also called hierarchical cluster analysis or HCA) is a method of cluster analysis which seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types:

- Agglomerative: This is a "bottom-up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
- Divisive: This is a "top-down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

Figure 1.4 shows an example of hierarchical clustering using different agglomerative methods.

DBSCAN CLUSTERING Density-based spatial clustering of applications with noise (DBSCAN) is a data clustering algorithm (Ester et al., 1996). It is a density-based clustering algorithm: given a set of points in some space, it groups together points that are closely packed together (points with nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature.

For the purpose of DBSCAN clustering, the points are classified as core points, (density-)reachable points and outliers, as follows:

- A point p is a core point if at least minPts points are within distance ϵ (ϵ is the maximum radius of the neighborhood from p) of it (including p). Those points are said to be directly reachable from p.
- A point q is directly reachable from p if point q is within distance ϵ from point p and p must be a core point.

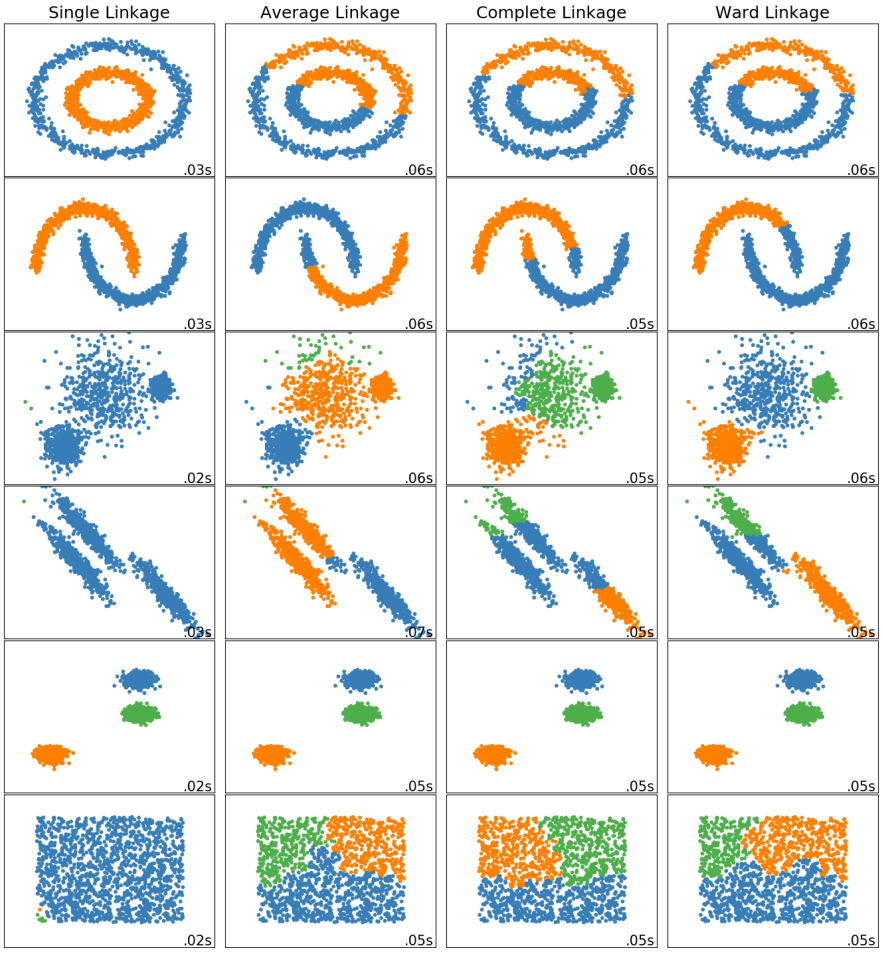

FIGURE 1.4. Example of hierarchical clustering result. Source: scikit-learn.org

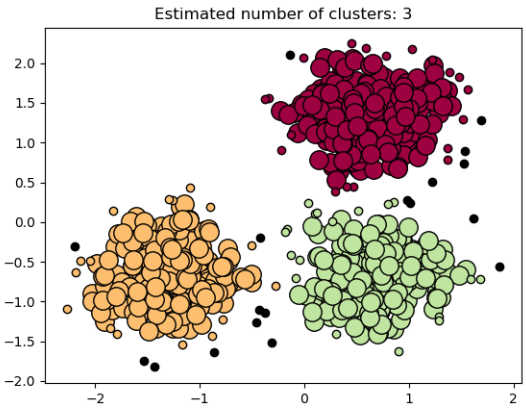

FIGURE 1.5. Example of DBSCAN clustering. Source: scikit-learn.org

- A point q is reachable from p if there is a path $p_1, \ldots, p_n$ with $p_1 = p$ and $p_n = q$, where each $p_i + 1$ is directly reachable from p_i (all the points on the path must be core points, with the possible exception of q).

All points not reachable from any other point are outliers (black points in Figure 1.5). Now if p is a core point, then it forms a cluster together with all points (core or non-core) that are reachable from it. Each cluster contains at least one core point; non-core points can be part of a cluster, but they form its "edge" since they cannot be used to reach more points. DBSCAN can guarantee that the cluster is not always a circular shape, given the possibility to individuate streets or geographical features in general.

2 Related Work

In this thesis we propose an approach for location inference from microblog messages, which relies on semantic geographic gazetteers. For this reason, in this chapter we discuss work related to both location prediction and geographic gazetteers.

In detail, we provide an in-depth study of external information that can be used to serve the task. Such information sources include semantic gazetteers, which lie at the core of our novel approach. We also present in detail the geographic data sources chosen to address the problem. Moreover, we provide an overview of state-of-the-art algorithms for gelocating non geolocalized tweets. We conclude the chapter with a positioning of our proposed approach towards the state-of-the-art.

2.1 Geospatial Knowledge

The most important novelty of the work presented in this thesis is the use of external geographical knowledge for sub-city geolocation. Within this context, in this section, we present the relevant research work on geospatial ontologies.

Since the principal focus of this thesis is geolocation of microblog messages, discussing and understanding external knowledge semantics is fundamental. This is because such knowledge bases can be used to augment current geolocation algorithms. In this section we drill into the details of relevant research work and identified GeoNames [1], LinkedGeoData (Stadler et al., 2012) and OSM facet ontology (Di Rocco, 2013, 2016) as the three most significant geospatial ontologies that can be used in order to achieve our goal. Moreover, since the aim is to do geolocation at sub-city level, we need do have an ontology that conceptualize or can conceptualize the concept of city. We will see later on that the different conceptualizations of the city domain produce different results on sub-city level geolocation algorithms.

2.1.1 Geospatial Ontologies

An ontology is defined as: *"a specification of a conceptualization."* (Gruber, 1993). Ontologies have been used for a set of tasks: improving communication between agents (human or software), reusing data models, developing knowledge schemas, etc. All these tasks deal with interoperability issues and can be applied in different domains.

Ontologies of the geographic world are important to allow the sharing of geographic data among different communities of users. A geospatial ontology

[1] geonames.org

provides a description of geographical entities. A lot of geospatial ontologies have been defined and can be used by different applications. We first discuss some well-known ontologies that we analyze for our work, then we present specific state of the art approaches related to ontologies for gazetteers and Volunteered Geographic Information (VGI).

We can classify geographic data sources as authoritative and crowdsourced. One of the most famous examples of a semi-authoritative data source is GeoNames, and one famous example of a crowdsourced data source is OSM. While authoritative data sources are mainly non-open data, we focus our analysis on open data sources where GeoNames is the most famous data source close enough to an authoritative data sources since it was born from authoritative data. GeoNames and OSM are different geographical data sources not only w.r.t. this classification. GeoNames is structured with its explicit ontology level, while OSM is not associated with an explicit ontology level, however, it is possible to extract a sort of semantic information from tags associated with toponyms. In the state-of-the-art, this problem is largely investigated and it is possible to attach a semantic level to OSM.

In any case, other knowledge bases exist. Generic knowledge bases can contain information useful to describe geospatial data or can contain geospatial data. An example is DBpedia. DBpedia[2] is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web. DBpedia allows you to ask sophisticated queries against Wikipedia, and to link the different data sets on the Web to Wikipedia data. The goal is to make it easier for the huge amount of information in Wikipedia to be used in some new interesting ways. Furthermore, it might be inspired by new mechanisms for navigating, linking, and improving the encyclopedia itself. The DBpedia Ontology is a shallow, cross-domain ontology, which has been manually created based on the most commonly used infoboxes within Wikipedia. The ontology currently covers 685 classes which form a subsumption hierarchy and are described by 2,795 different properties.

RELATED WORK ON GEOSPATIAL ONTOLOGIES In the geospatial domain, there are a lot of approaches that try to use ontologies for semantifying geographic data (Janowicz and Keßler, 2008; Janowicz, Maue, et al., 2008). We hereby analyze research works that involve OSM, the most famous example of VGI. OSM offers an open and easy to use platform that enables contributors to upload geographic information collected from mobile devices or aerial images. There is no formal ontology or vocabulary of predefined tags that have to be adopted by the users. For this reason, there exists a lot of work about semantifying OSM. For instance, Baglatzi, Kokla, and Kavouras, (2012) shows a way to bridge the gap between ontological and crowdsourcing practices. To create this bridge they use an ontological alignment approach. They align OSM tags to the DOLCE Ultralite top level ontologies(Masolo et al., 2002).

There is also an important project called OSMOnto[3] (Codescu et al., 2011).

[2] http://dbpedia.org/

[3] It is possible to find more information here: http://wiki.openstreetmap.org/wiki/

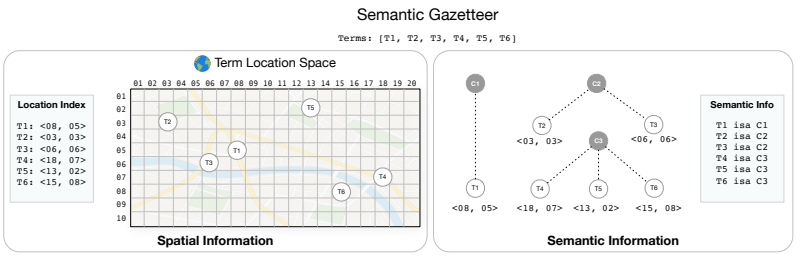

FIGURE 2.1. A model of a semantic gazetteer.

This is an ontology for tags. The purpose of the ontology of tags is to stay as close as possible to the structure of the OSM files in order to facilitate database querying. This means that they do not try to correct any possible conceptual mistakes in the taxonomy of OSM tags, but rather have it reflected faithfully in the structure of the ontology.

Another example of semantic research work on OSM is the OpenStreetMap Semantic Network[4] (Ballatore and Bertolotto, 2011; Ballatore, Bertolotto, and Wilson, 2013). It is a Semantic Web resource extracted from the OSM Wiki website, encoded as a SKOS vocabulary [5]. It contains a machine-readable representation of OSM tags and several semantic relationships among them.

Beard, (2012) introduces an ontology-based gazetteer model for organizing VGI. The aim of such work is to try to associate VGI contribution to place such that overtime places may be characterized through these contributions.

Fonseca et al., (2002) present an ontology that can be used to classify geographic elements with respect to not only their geometrical features but also their attribute values, i.e. their semantic features. In this work, they integrate vector-based GIS (geometrical feature) that imports raster data (attribute values) or raster-based GIS that imports vector data. The result of this classification with an ontology support is a set of images indexed not only by its semantics but also by its attribute values. In this way, they obtain not only a static polygon but also a set of semantic features and its corresponding values. This approach is very useful in cases in which we need to analyze and manage observations on continuous spatiotemporal data.

Another problem that we have to cope with is language. When we search for information about something, we usually use a specific language. For instance, we are Italian, so we usually search for information in Italian. But, if we search "Londra" and not "London", we would find less information in Italian than in English. There are several approaches designed to solve this kind of problems. For instance, Laurini, (2015) uses ontologies among gazetteers to try to create a bridge connecting the same concept in different languages. In Figure 2.1, we sketch the model of geographic ontologies.

Notice that, in our problem we need a geographic knowledge that describes

OSMonto

[4]It is possible to find more information here: http://wiki.openstreetmap.org/wiki/OSM_Semantic_Network

[5]Simple Knowledge Organization System - http://www.w3.org/2004/02/skos/

geographic data at fine-grained level. After this analysis on geospatial knowledge in general, we focus the discussion on three ontologies that we consider as best candidates for this work. The three knowledge bases are: GeoNames, LinkedGeoData and OSM facet ontology. They are ontologies that describe geographic data coming from GeoNames itself and OpenStreetMap. In the following, we describe them in detail.

2.1.2 A Semi-Authoritative Data Source and Gazetteer: GeoNames

The GeoNames Ontology makes it possible to add geospatial semantic information to the World Wide Web. Over 11 million GeoNames toponyms now have a unique URL with a corresponding RDF web service. Other services describe the relation between toponyms. The Ontology for GeoNames is available in OWL.

GeoNames is a geosemantic data source web service. GeoNames contains over 11,000,000 geographical names corresponding to over 7,500,000 unique features. All features are categorized into one of 9 feature classes and further subcategorized into one of 645 feature codes. Each GeoNames feature is represented as a web resource identified by a stable URI. This URI provides access, through content negotiation, either to the HTML wiki page, or to an RDF description of the feature, using elements of GeoNames ontology. This ontology describes the GeoNames features properties using the web ontology language, the feature classes, and codes are described in the SKOS language. GeoNames consists of various locations of all countries. It includes geographical data: place names in various languages, latitude, longitude, altitude, and class.

The nine classes are represented by alphabetic letters. The classes are:

- A country, state, region, etc.
- H stream, lake, etc.
- L parks, area, etc.
- P village, city, etc.
- R road, railroad
- S sport, building, farm, etc.
- T mountain, hill, rock, etc.
- U undersea
- V forest, heat, etc.

The GeoNames ontology describes point-of-interest and high-level information as a lake, village, forest, etc. However, the information at fine-grained level is not so exhaustive compared to other ontologies, as we will see in detail soon.

In Figure 2.2 we show the classification of Greater London in GeoNames. Greater London has class `Feature` with feature A and a specific code ADM2.

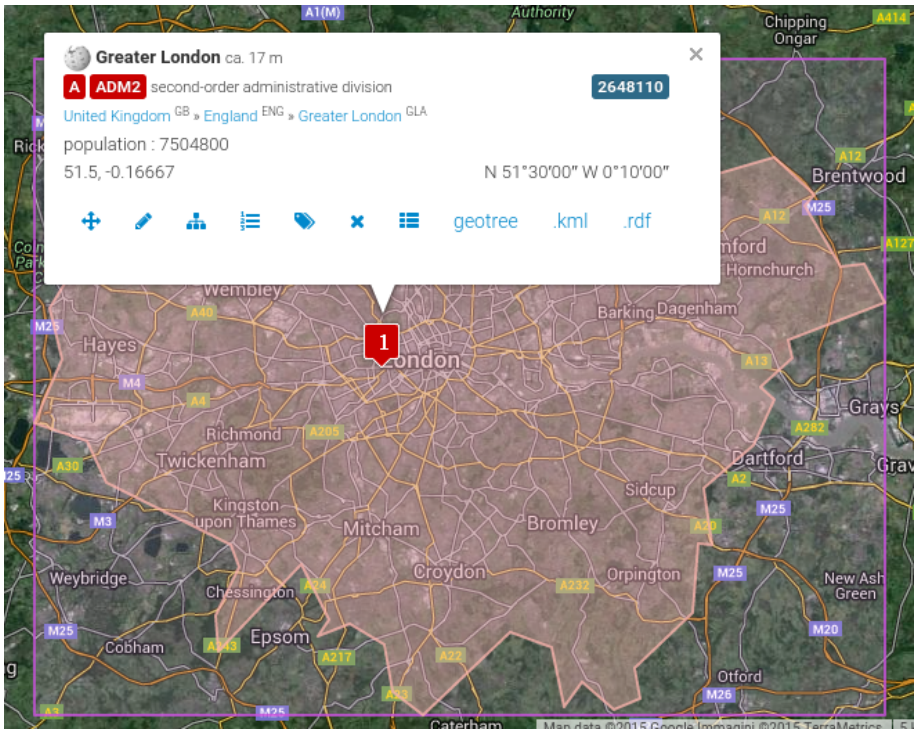

FIGURE 2.2. Greater London in GeoNames.

We hereby report some other information and examples about the GeoNames ontology from the official documentation (`geonames.org/ontology/documentation.html`). In the GeoNames Semantic Web we use these URIs for Berlin:

1. `http://sws.geonames.org/2950159/`
2. `http://sws.geonames.org/2950159/about.rdf`

If you want to express the fact that you are living in Berlin you use the first URI, if you want to make a remark about the information GeoNames has about Berlin then you use the latter.

The GeoNames web server is configured to return a 303 redirection response for a request of the URI for the concept (Item 1) and give Item 2 as the new location of the document. This is one way how the Technical Architecture Group (TAG) of the W3C has decided to resolve the ambiguity between concept and document. The other accepted way to remove the ambiguity is the use of hashes in the URI.

The Features in the GeoNames Semantic Web are interlinked with each other. Depending on applicability the following documents are available for a `Feature`:

- The children (countries for a continent, administrative subdivisions for a country, ...). As an example the children of France: `http://sws.geonames.org/3017382/contains.rdf`

- The neighbors (neighboring countries). As an example the neighbors of France : `http://sws.geonames.org/3017382/neighbours.rdf`
- Nearby features. Nearby to the Eiffel Tower are Champ de Mars, Trocadéro - Palais de Chaillot, ...: `http://sws.geonames.org/6254976/nearby.rdf`

2.1.3 Crowdsourced Data Source: OpenStreetMap

OpenStreetMap (OSM) is a collaborative project to create a free editable map of the world via crowdsourced data.

The basic components of the OSM's conceptual data model are:

- *node*: consists of a single point in space defined by its latitude, longitude and node id. A third, optional dimension (altitude) can also be included.
- *way*: is an ordered list of nodes which usually also has at least one tag or is included within a relation. A way can have between 2 and 2000 nodes, although it is possible that faulty ways with zero or a single node exist.
- *relation*: consists of one or more tags and also an ordered list of one or more nodes, ways or geographic relationships between other elements. It is sometimes used to explain how other elements work together.

OSM data can have associated tags that describe the meaning of the particular element they are attached to. A tag consists of a "Key" and a "Value". Each tag describes a specific feature of a data element or changesets[6]. Both Key and Value are free-format text fields. The Key describes a broad class of features. The Value details the specific feature that is generally classified by the key. However, the tagging approach used in OSM is not very well structured and therefore does not adequately support information retrieval.

Here are a few examples of how keys and values are used in practice:

- `highway=residential` a tag with a key of "highway" and a value of "residential" which should be used on a way to indicate a road along which people live.
- `name=Park Avenue` a tag for which the value field is used to convey the name of the particular street.
- `maxspeed=50` a tag whose value is a numeric speed in km/h (or in miles per hour if the unit is provided with a suffix "mph"). Metric units are the default.

In Figure 2.3 we can see how OSM is structured. In what follows we discuss the different components in greater detail.

[6] A changeset consists of a group of changes made by a single user over a short period of time. One changeset may, for example, include the additions of new elements to OSM, the addition of new tags to existing elements, changes to tag values of elements, deletion of tags and also deletion of elements.

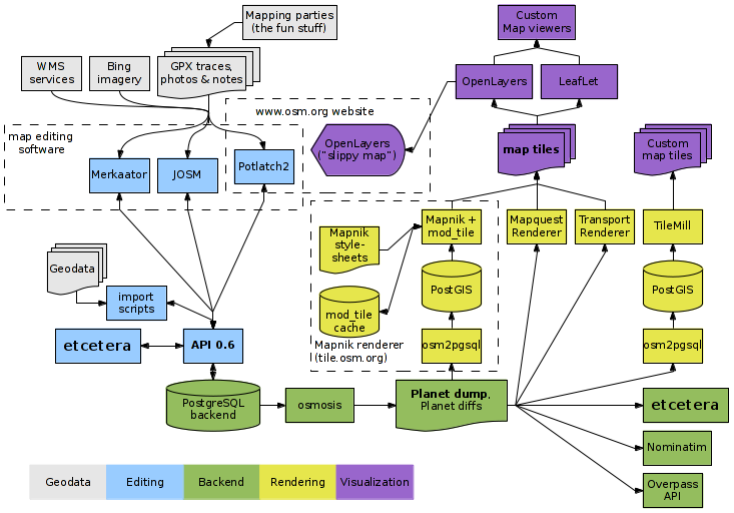

FIGURE 2.3. OSM components.
Source: https://wiki.openstreetmap.org/w/images/1/15/OSM_Components.png

NODE A node is one of the core elements in the OpenStreetMap data model. It consists of a single point in space defined by its latitude, longitude and node id. A third, optional dimension (altitude) can also be included: `key:ele` (i.e. elevation).

Nodes can be used on their own to define point features. When used in this way, a node will normally have at least one tag to define its purpose. Nodes may have multiple tags and/or be part of a relation. For example, a telephone box may be tagged simply with `amenity=telephone`, or could also be tagged with `operator=*`.

We discuss the structure of Node with an example.

LISTING 2.1. An example of node.

```
<node id="25496583" lat="51.5173639" lon="-0.140043" version="1" changeset="203496"
    user="80n" uid="1238" visible="true" timestamp="2007-01-28T11:40:26Z">
    <tag k="highway" v="traffic_signals"/>
</node>
```

The main elements are:

- **id**: 64-bit integer number ≥ 1. Node ids are unique among nodes (however, a way or a relation can have the same id number as a node.). Editors may temporarily save node ids as negative to denote ids that have not been saved yet to the server. Node ids on the server are persistent, meaning that the assigned id of an existing node will remain unchanged each time data are added or corrected.
- **lat**: decimal number ≥ -90.0000000 and ≤ 90.0000000 with 7 decimal places. Latitude coordinate in degrees (North of the equator is positive) using the standard WGS84 projection. Some applications may not accept

latitudes above/below ±85 degrees for some projections.

- **lon**: decimal number ≥ -180.0000000 and ≤ 180.0000000 with 7 decimal places. Longitude coordinates in degrees (East of Greenwich is positive) using the standard WGS84 projection. Note that the geographic poles will be exactly at latitude ±90 degrees but in that case, the longitude will be set to an arbitrary value within this range.
- **tags**: A set of key/value pairs, with a unique key.

WAY A way is an ordered list of nodes which usually also has at least one tag or is included within a Relation. A way can be open or closed. A closed way is one whose the last node on the way is also the first on that way. A closed way may be interpreted either as a closed polyline, or an area, or both.

There are different types of ways:

- **Open way**: An open way is a way describing a linear feature which does not share a first and last node. Many roads, streams, and railway lines are open ways. Sometimes, in these cases, it is important to identify the direction of a "way" (with an appropriate tag).
- **Closed way**: A closed way is a way where the last node of the way is shared with the first node with suitable tagging. A closed way that also has a `area=yes` tag should be interpreted as an area (but the tag is not required most of the time).
- **Area**: An area (also polygon) is an enclosed filled area of territory defined as a closed way. Most closed ways are considered to be areas even without a `area=yes` tag. Some exceptions are: `leisure = park` and `amenity = school`. The first defines the perimeter of a park and the second defines the outline of a school. For tags which can be used to define close polyline, it is necessary to also add a `area=yes` if an area is desired. For instance, `highway = pedestrian + area = yes` to define a pedestrian square or plaza. Areas can also be described using one or more ways which are associated with a multipolygon relation.
- **Combined closed-polyline and area**: It is possible for a closed way to be tagged in a way that it should be interpreted both as a closed-polyline and as an area.

For instance, a way is:

LISTING 2.2. An example of way.

```
<way id="5090250" visible="true" timestamp="2009-01-19T19:07:25Z" version="8"
    changeset="816806" user="Blumpsy" uid="64226">
    <nd ref="822403"/>
    <nd ref="21533912"/>
    <nd ref="821601"/>
    <nd ref="21533910"/>
    <nd ref="135791608"/>
    <nd ref="333725784"/>
```

```
    <nd ref="333725781"/>
    <nd ref="333725774"/>
    <nd ref="333725776"/>
    <nd ref="823771"/>
    <tag k="highway" v="residential"/>
    <tag k="name" v="Clipstone_Street"/>
    <tag k="oneway" v="yes"/>
  </way>
```

The nodes defining the geometry of the way are enumerated in the correct order and indicated only by reference using their unique identifier. These nodes must have been already defined with their coordinates.

RELATION A relation is one of the core data elements that consists of one or more tags and also an ordered list of one or more nodes, ways and/or relations as members. A member of a relation can optionally have a role which describes the part that a particular feature plays within a relation.

Relations are used to model logical (and usually local) or geographic relationships between objects. They are not designed to bind loosely associated and widely spread items. It would be inappropriate, for instance, to use a relation to group 'All footpaths in East Anglia'.

It is recommended to use no more than about 300 members per relation. If you have to handle more than that amount of members, it is better to create several relations and combine them with a Super-Relation [7]. This is because the more members are stuffed into a single relation, the harder it is to handle, the easier it breaks, the easier conflicts can show up and the more resources it consumes at database and server level.

A role is an optional textual field describing the function of a member of the relation. For example, in North America, `role:east` indicates that a way would be posted as East on the directional plate of a route numbering shield. Or, multipolygon relation, `role:inner` and `role:outer` are used to specify whether a way forms the inner or outer part of that polygon.

There are several types of relation[8].

An example of relation is shown in Listing 2.3.

LISTING 2.3. An example of relation.

```
<relation id="3503470" visible="true" version="19" changeset="33628327"
    timestamp="2015-08-27T19:01:18Z" user="vpettenati" uid="834086">
    <member type="way" ref="265375775" role="outer"/>
    <member type="way" ref="367867538" role="outer"/>
    <member type="way" ref="309967328" role="outer"/>
    <member type="way" ref="339565174" role="outer"/>
    <member type="way" ref="339565178" role="outer"/>
    <member type="way" ref="339565177" role="outer"/>
    <member type="way" ref="339565175" role="outer"/>
    <member type="way" ref="339565176" role="outer"/>
    <member type="way" ref="339565180" role="outer"/>
```

[7] We have to admit, anyway, that though "super-relations" is a good concept on paper none of the many OSM software applications is working with them.

[8] There are listed http://wiki.openstreetmap.org/wiki/Types_of_relation

```
    <member type="way" ref="367867539" role="outer"/>
    <member type="way" ref="339565173" role="outer"/>
    <tag k="allocation:it" v="Il suo territorio si estende per oltre 22.000 ettari
        lungo la dorsale appenninica tra l'Emilia-Romagna e la Toscana interessando le
        province di Massa-Carrara, Lucca, Reggio Emilia e Parma"/>
    <tag k="boundary" v="protected_area"/>
    <tag k="governance_type" v="government_managed"/>
    <tag k="iucn_level:ref" v="2"/>
    <tag k="leisure" v="nature_reserve"/>
    <tag k="name" v="Parco nazionale dell'Appennino Tosco-Emiliano"/>
    <tag k="operator" v="Ente Parco Nazionale dell'Appennino tosco emiliano"/>
    <tag k="protect_class" v="2"/>
    <tag k="protection_title:" v="geomorphological"/>
    <tag k="ref:EUAP" v="EUAP1158"/>
    <tag k="ref:SIC" v="n.a."/>
    <tag k="related_law"
        v="Decreto del Presidente della Repubblica del 21 maggio del 2001"/>
    <tag k="site_ownership" v="national"/>
    <tag k="site_status" v="obtained"/>
    <tag k="source"
        v="http://servizigis.regione.emilia-romagna.it/wms/areeprotette_natura2000"/>
    <tag k="type" v="multipolygon"/>
    <tag k="WDPA_ID:ref" v="178782"/>
    <tag k="wikipedia"
        v="it:Parco nazionale dell'Appennino Tosco-Emiliano"/>
</relation>
```

In Listing 2.3 there is a multipolygon relation. A multipolygon relation can have any number of ways in the role outer (the outline) and any number of ways in the role inner (the holes), and these must somehow form valid rings to build a multipolygon from.

TAG Each tag has only a key and value. Tags are written in OSM documentation as `key=value`.

- The key describes a broad class of features (for example, highways or names).
- The value details the specific feature that was generally classified by the key (e.g. `highway=motorway`). If multiple values are needed for one key the semi-colon value separator may be used in some situations.

OpenStreetMap uses tags to add meaning to geographic objects. There is no fixed list of tags. New tags can be invented and used as needed. Everybody can come up with a new tag and add it to new or existing objects. This makes OpenStreetMap enormously flexible, but sometimes also a bit hard to work with.

Since OSM has no explicit semantic level associated with it, a lot of researchers studied this problem proposing a different solution. In Chapter 2 we have already shown some solutions. Analyzing them, we choose two representative solutions that we explain in the following. The first one is LinkedGeoData(Stadler et al., 2012) (LGD), and the second one is OSM facet

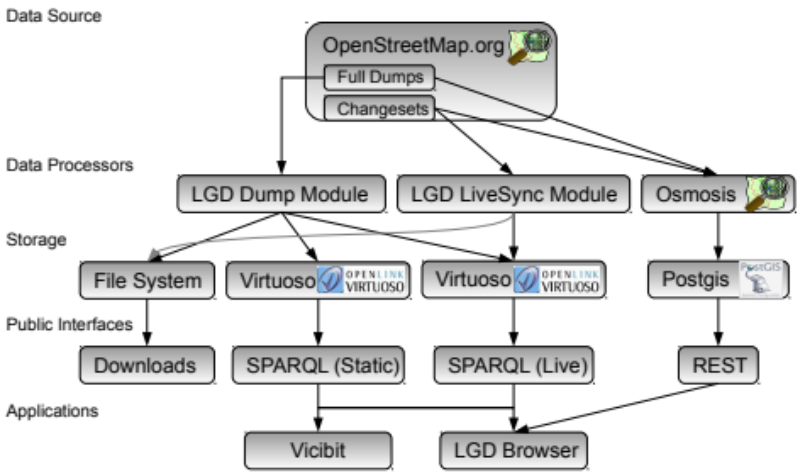

FIGURE 2.4. LGD components.
Source: Stadler et al., (2012)

ontology, an ontology for OSM that we develope with the aim to have a better classification of OSM tags conceptualizing the city.

2.1.4 OSM-Based Ontologies

Since OSM is a geographic data sources with only an implicit semantic level obtained with tags, there are solution in literature to add a semantic structure to OSM. In this thesis, we introduce LinkedGeoData and OSM facet ontology.

LINKEDGEODATA Stadler et al., (2012) presented an ontology called LinkedGeoData (LGD) that links OSM information to DBpedia, GeoNames and others ontologies. LGD uses the comprehensive OpenStreetMap spatial data collection to create a large spatial knowledge base. It consists of more than 3 billion nodes and 300 million ways and the resulting RDF data comprises approximately 20 billion triples. The data is available according to the Linked Data principles and interlinked with DBpedia and GeoNames.

In Figure 2.4 we show the LGD components as described by Stadler et al., (2012). The LGD ontology is derived from OSM tags. An OSM object can have more than one tag. They consider each tag separately. In some cases, they split the the key from the related value and both became classes with the subclass relation between them, e.g., an object with `amenity = restaurant` became the instance of `lgdo:Restaurant`[9] that is *SubClassOf* `lgdo:Amenity`. In other situations, they give a specific class to a set of tags that represent the same concept, e.g., an object with tags `amenity = place_of_worship` and `religion = christian` became instance of `lgdo:Church` . In this way, they create an ontology structure very close to OSM structure. Stadler et al., (2012)

[9] `lgdo` is the prefix of LGD ontology resources.

declare in the research paper that in the first release of LGD there were 50 object properties. After a better analysis, they considered useful only 9 object properties.

OSM FACET ONTOLOGY The OSM Facet ontology (Di Rocco, 2013, 2016) allows us to use the data present in OSM as instances of an ontology. It is extracted from the non-spatial information from geospatial data in order to create a classification hierarchy. This ontology aims at classifying geospatial objects that are relevant in urban contexts, thus, that may appear in a generic city, trying to avoid, whenever possible, to focus on the specificities of a particular city. For this reasons, the OSM Facet ontology is structured to represent geographic information on a city using three different facets: Point of Interest (PoI) facet; geoPolitical facet; geoPhysical facet.

The OSM Faceted Ontology is used to filter OSM data associating specific classes with tags of OSM objects. It contains 97 classes.

First of all, we need to create a class that represents the target domain, i.e. the core of the ontology on which we implement the facets. The obtained pattern is shown in Listing 2.4.

LISTING 2.4. Faceted Ontology pattern in Turtle

```
# Object Properties
:hasFacet_i rdf:type owl:ObjectProperty ;
            rdfs:domain :TargetDomainConcept ;
            rdfs:range :Facet_i .

:isPartOf   rdf:type owl:ObjectProperty ;
            rdfs:domain :TargetDomainConcept ;
            rdfs:range :F_iTerm_jTDC .

# Classes
:F_iTerm_j rdf:type owl:Class ;
           owl:equivalentClass :F_iTerm_jTDC ;
           rdfs:subClassOf :Facet_i .

:F_iTerm_jTDC rdf:type owl:Class .

:Facet_i rdf:type owl:Class .

:SpecificTDC_x rdf:type owl:Class ;
               rdfs:subClassOf :TargetDomainConcept
                   .

:TargetDomainConcept rdf:type owl:Class .
```

Referring to Listing 2.4, we can better explain our ontology:

- `TargetDomainConcept` is a class that represents a domain;

- `Facet_i` is a class that represent a different facets. In our case $i = 3$ then we have three different classes.
- `SpecificTDC_x` is a class to represent a specific feature of domain. In our case, $x = 1$. This means that we have only one specification of facet domain.
- `F_iTerm_j` and `F_iTerm_jTDC` are equivalent classes and they are respectively subclasses of `Facet_i` and `isPartOf` relation of `TargetDomainConcept`. In our case represent the first child of our facets.

This pattern is useful for each domain in which we can perform a faceted searching. We use this pattern to implement our ontology.

Based on the introduced pattern, we develop our ontology to classify OSM tags.

Our `TargetDomainConcept` is a city. We want to search geographic information on a city using the three different facets. In Figure 2.5 we can see the root of the ontology: Target Domain and Facets. We have a class `City` that represents the target domain. This class is related to the facets through a relation `City -> hasFacet -> *_facet` (yellow arrow) and it is related to the specific concept of facets through a relation `*_facet -> hasSubClass -> * -> isPartOf -> City` (orange arrow). The figure also shows `subClass` relations between classes (purple arrows).

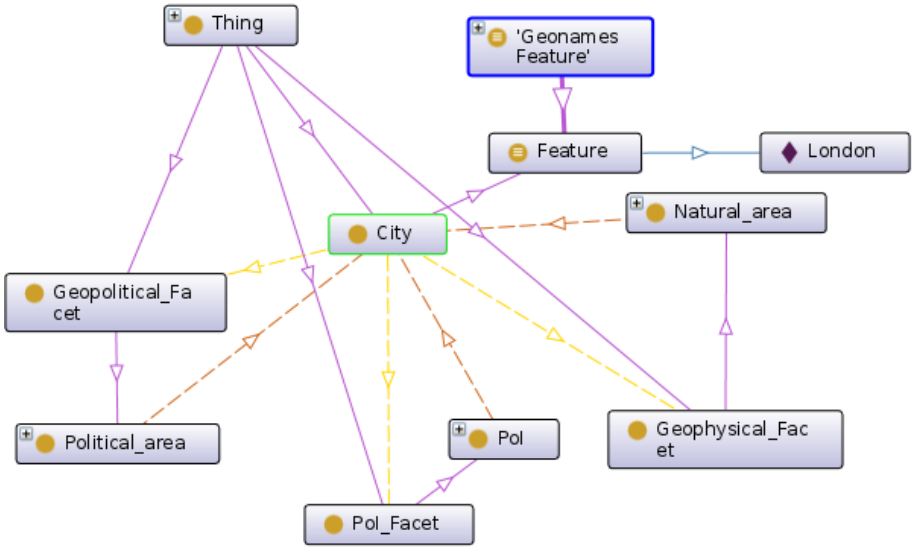

FIGURE 2.5. Screen of a part related to `City` class.

2.2 Location Prediction

There are multiple different techniques proposed for extracting implicit geographic information from social media messages. In this section we investigate the difference between multiple different approaches. We also highlight that

a strong part of our approach is its ability to be used in combination with existing techniques, effectively improving their geolocation completeness and correctness. In the surveys (Ajao, Hong, and W. Liu, 2015; X. Zheng, Han, and A. Sun, 2018), a general overview of different methods for location inference can be found.

The complete discussion is based on microblog data, more precisely on Twitter data. As we presented in Chapter 1, Twitter contains three types of information. All of them can be used to solve various location prediction problems. We follow the categorization defined in (X. Zheng, Han, and A. Sun, 2018) and discuss location prediction based on the different types of information involved: content, network and context. This is because multiple data sources can help enrich the available information. However, each different type of information is involved and used in different instances of the problem.

We identify three different instances of the location prediction problem:

1. home location prediction,
2. user location prediction,
3. tweet location prediction.

Home location prediction is the problem of inferring the location of the home of a user. For example, looking at the messages of a user, we try to understand where (s)he came from. User location prediction is the problem related to understanding where a user currently is. For example, when a user write a message like: "Lunch with my collegues", we want to understand that is user is in lunch-break more probably close to the office. Finally, tweet location prediction is the problem of inferring the location to which a tweet refers to. Twitter messages contain a lot of time implicit geographic information that reveals their position in the world. Processing tweets to find the implicit geographic information is an essential step for applications that need to gather geographic information to analyze users or events (Imran et al., 2015).

The literature is also subdivided in the aforementioned problem instances. In our opinion, there is no big difference between tweet location prediction and mentioned location prediction. In these two instances, the only difference is the precision with which we geotag. In the following, we discuss these problem instances and the proposed approaches to address them.

2.2.1 Home Location Prediction

Home location prediction is the problem related to predicting the location of the home of a user.

In case of Twitter, home locations refer to users' long-term residential addresses. The home location of a user is relevant for different applications such as local content recommendation, location-based advertisement, public health monitoring, public opinion polling, etc. As we will see also for the other problems, research is very active in this area due to the fact that the home

field on Twitter is optional and often this field is empty or noisy. We highlight that usually the home location is predicted at city-level or state/country level (Cheng, Caverlee, and Lee, 2010; Kinsella, Murdock, and O'Hare, 2011; Samet et al., 2014). In fact, the home location may be represented at different levels of granularity:

- Administrative regions, i.e., countries, states, cities.
- Geographical grids, i.e., the earth is partitioned into cells of equal or different sizes.
- Geographical coordinates, i.e., homes are represented by their geographical coordinates. In general, the coordinates are self-reported or converted from administrative regions or geographical grids by taking the center in with their fall in.

In the following, we start to discuss the problem based on Twitter information used. First, inference based on tweet content, then on Twitter network and we conclude with Twitter context.

2.2.1.1 Inference based on Tweet Content

Approaches that use tweet content to understand where a user came from can be in general subdivided into two categories: word-centric and location-centric. The involved techniques are based on probability in both categories. Word-centric methods estimate the probability of a location l given a word w in text message, i.e., $p(l|w)$. Location-centric methods estimate the probability of generating a tweet t at a given location l, i.e., $p(t|l)$.

WORD-CENTRIC METHODS Word-centric methods aim at identifying and exploiting words that help predict users' home locations. Notice that, not all words are linked to a spatial location. Only words that can be linked with a spatial location should be involved. We want to highlight that location information implied by local words should be learned from data before making predictions.

One of the most important and first word-centric approaches is by Backstrom, Kleinberg, et al., (2008). They propose a spatial variation model based on words. They assume that each word has a spatial location, a frequency in that location and a dispersion factor. They have shown that the probability to show a word in that location with a distance d to the spatial location is proportional to the frequency multiplied by the distance powered by minus of a dispersion factor, i.e.,

$$p = Cd^{-\alpha}$$

where C is a constant that represents a spatial location, d is the person's distance from that spatial location and α is the exponent that determines how quickly a frequency of toponym appearance in the data can fall off away from the spatial location. More easily, they present a model that specifies a one-peak distribution in the spatial location with exponential decay. The work

of Cheng, Caverlee, and Lee, (2010) uses a probabilistic framework to estimate the city-level coordinates of tweets based on the text message. They do not use geotags of a single tweet but use the place information field extracted from the user profile. They identify words that have a local focus and then model their geographic distribution, i.e., they build a statistical predictive model. More precisely, the distribution of a user u's home location l given his/her tweet contents $S(u)$ is decomposed as

$$P(l|u) \propto \sum_{w \in S(u)} P(l|w)P(w).$$

They present a variation of the model presented in Backstrom, Kleinberg, et al., (2008) as a smoothed distribution. In their extended work, Cheng, Caverlee, and Lee, (2013) generalized the one-peak model by wave-like smoothing to allow multi-peaks for word-distribution. Finally, also Ryoo and Moon, (2014) use both the probabilistic approach of Backstrom, Kleinberg, et al., (2008) and of Cheng, Caverlee, and Lee, (2013), showing how this approach should be used in a Korean dataset too and not only in an English one.

LOCATION-CENTRIC METHODS Location-centric methods aim to give a more central role to locations. To do so, they use classification-based approaches on location to infer the user's home location.

Hecht et al., (2011) propose a CALGARI score for words, which is similar to information based measures. They select the top 10,000 words with the highest CALGARI scores as local words. Then, they represent users as 10,000-dim term frequency vectors and feed them into a multinomial Naive Bayes classifier. Calgari is an algorithm that calculates a score for each term present in the corpus using the following formula:

$$CALGARI(t) = \begin{cases} 0 & \text{if } users(t) < MinU \\ \frac{\max(P(t|c=C))}{P(t)} & \text{if } users(t) \geq MinU \end{cases}$$

where t is the input term, *users* ia a function that calculates the number of users who have used t at least once, $MinU$ is an input parameter to filter out spam, and C is a geographic class, i.e., a state or a country. Mahmud, Nichols, and Drews, (2012) apply a series of heuristic rules to select local words. They adopt a hierarchical ensemble algorithm to train two-level classifier ensembles on the granularity of timezone-city or state-city. They use classifiers on three different terms: words, hashtags, and place names. These classifiers can be created for any level of granularity for which they have ground truth.

A similar approach can be found in Kinsella, Murdock, and O'Hare, (2011). They model locations from zip code to country level using a probability distribution of terms associated with a location.

2.2.1.2 Inference based on Twitter Network

One of the major activities of Twitter users is to establish following-type relationships and interact with friends. It is argued in the literature that social

closeness, which is based on friendship, interactions, retweeting, etc., is more reliable for estimating home location than friendship only. Therefore, it could be interesting to involve the network in order to infer home location.

In social science the concept of homophily (McPherson, Smith-Lovin, and Cook, 2001) is the tendency of individuals to associate and bond with similar others, in other words, similar people contact at a higher frequency than dissimilar ones. A considerable amount of work on home location inference is based on this concept. A quick intuition maybe that one's home location is very likely to be close to his/her friend's home location. One of the earliest approaches that used the friendship model to infer location is Backstrom, E. Sun, and Marlow, (2010). This study is conducted on Facebook data but forms the basis for some other approaches used for Twitter data. The authors analyze Facebook users with a known home location and their friends. They fit the probability of two users being friends w.r.t. their home distance with a following curve:

$$P(u_i, u_j \text{ are friends } | dist(u_i, u_j) = x) = a(b + x)^{-c},$$

i.e., the probability of friendship is inversely proportional to home distance (with $c = 1$ that is the value with which they obtain a good fit). If they start only with direct friends, Kong, Z. Liu, and Huang, (2014) find that rich indirect friendship on Twitter may be better to indicate off-line friendship between two users, and thus their home location proximity.

However, not only friendship is a part of a Twitter network information. We can have important social-closeness information from mentions, retwitting and influence. Kong, Z. Liu, and Huang, (2014) also shown that social closeness, or how familiar two users are in real life, is a better indicator of home proximity. Since mentions is another form of interaction, McGee, Caverlee, and Cheng, (2013) observe that friendship probability w.r.t. home distance on Twitter roughly satisfies a bimodal distribution (in contrast to the equation proposed in Backstrom, E. Sun, and Marlow, (2010)). By observing mentions and friendship, they find that besides mutual friendship through following, users' actions of mentioning and actively chatting with each other also indicate their home proximity. They also make another interesting observation: if the followed user has a protected account [10], the two users are geographically close and local newspaper accounts are close to their followers.

Chandra, Khan, and Muhaya, (2011), use a probabilistic framework to estimate the city-level Twitter user location. The probabilities are based on the contents of the tweet messages with the aid of reply-tweet messages generated from the interaction between different users in the Twitter social network.

Notice that influence (i.e., the popularity of a user) also impacts on social closeness. Differently for the other factors, influence has a negative impact. Kwak et al., (2010) find that users with fewer than 2000 mutual friends are more likely to be geographically close to most of them, in contrast with famous people like VIP.

[10] A protected account is an account with no public information. In general, a protected account is an ordinary person.

2.2.1.3 Inference based on Tweet Context

Tweets have different context information that can come from tweet metadata. Among them, tweet posting time and self-declared user profiles like location and timezone are mainly employed to help to predict home location.

Mahmud, Nichols, and Drews, (2014), in the extensions of their previous work, propose identifying and removing traveling people from training data to improve the accuracy of home location classifiers. A user is considered traveling if any two of his/her tweets were sent from locations with distance above 100 miles. Moreover, they take into consideration tweet posting time. Users are viewed as distributions of tweet posting times. Distribution became a feature in the classifiers. Efstathiades et al., (2015) utilize a probabilistic approach on geo-tags associated with tweets to estimate user home location. In general, they observe that during the day users spends a significant amount of time in two key locations that are work and home. These location appear always in specific time w.r.t. locations that are not so frequent in the user routine.

2.2.1.4 Summary

In order to give a good overview of the state-of-the-art related to home location prediction, we summarize related work in Table 2.1. We notice that in the state-of-the-art, content, network and context are all commonly used to solve the problem. All the existing methods are data-driven and use probabilistic and inference approaches, or in a lot of cases classification, to solve the problem. This could be considered true for all the different inputs that can be used with Twitter data. In the research paper of Jurgens et al., (2015), we find an experimental comparison of the articles shown in this section. They conducted the experiments on the same dataset using as ground-truth both self-declared home location and geo-tags. In this way, they provide a detailed overview of the methods.

The work presented in this thesis has a different aim w.r.t. home location prediction. We aim at predicting a location where a tweet originates from. This is discussed in further details in the discussion section of this chapter.

2.2.2 User Location Prediction

User location prediction aims to find the location where a user is when (s)he posts. In literature, we can also find the general terminology of tweet location prediction. Tweet location prediction, in general, means the place from where a tweet is posted. Tweet location prediction could be ambiguous because it can refer to user location prediction or mentioned location prediction. If we look for a user location prediction, we want to have a complete picture of a user's mobility (this is in contrast with a home location prediction presented in Section 2.2.1). This means that these types of algorithms are based on the geo-tags of tweets. We can interpret user location prediction as tweet location

TABLE 2.1. Research papers on home location prediction.

	Content		Network		Context
	Word-centric	Location-centric	Friendship	Social-closeness	-
Backstrom, Kleinberg, et al., (2008)	✓				
Cheng, Caverlee, and Lee, (2010, 2013)	✓				
Ryoo and Moon, (2014)	✓				
Hecht et al., (2011)		✓			
Mahmud, Nichols, and Drews, (2012)		✓			
Kinsella, Murdock, and O'Hare, (2011)		✓			
Backstrom, E. Sun, and Marlow, (2010)			✓		
Kong, Z. Liu, and Huang, (2014)			✓		
McGee, Caverlee, and Cheng, (2013)				✓	
Chandra, Khan, and Muhaya, (2011)				✓	
Kwak et al., (2010)				✓	
Mahmud, Nichols, and Drews, (2014)					✓
Efstathiades et al., (2015)					✓

prediction. In user location prediction we want take care of all the information of a user is in order to geolocate a tweet.

In this case, we do not have representations based on administrative regions or grids, but instead, the location is represented as:

- Coordinates;
- Point Of Interest (POI).

In the following, we will not discuss approaches that have as a main focus messages content. In this section, we cover approaches that are interested in understanding where a user is and not where (s)he came from or his/her messages came from. Hence, the approaches reviewed used different inputs together with the content. Not many approaches tackle this problem.

2.2.2.1 Inference based on Twitter Network

In user location prediction the location inference is, in general, finer than home location prediction. In home location prediction, we are interested only at the city where a user came from. In this specific problem, instead, researchers need to add other information to messages in order to exactly understand where a user is. The general idea is to align friends networks to a user's messages to enrich the available information. Sadilek, Kautz, and Bigham, (2012) use as input the real-time location of a user's friends and his/her historical location. They study the correlation between the trajectories of friends and auto-correlation within one's trajectory and use it in a dynamic Bayesian network trained on the location sequence of each user with his/her friend location, considering time and day as a features.

2.2.2.2 Inference based on Tweet Context

In this section, we discuss algorithms that exploit different context information to geolocate a user. They use self-declared home location, timezone, and website. Moreover, some approaches also make use for third-party applications to understand the activity of a user.

Table 2.2. Research papers on user location prediction.

	Network	Context
Sadilek, Kautz, and Bigham, (2012)	✓	
Schulz, Hadjakos, et al., (2013) and Schulz, Mencia, et al., (2014)		✓
Chong and Lim, (2017)		✓
Ikawa, Enoki, and Tatsubori, (2012)		✓

Schulz, Hadjakos, et al., (2013) and Schulz, Mencia, et al., (2014) propose a method that relies on a multi-indicator spatial approach to disambiguate toponyms. The result is an algorithm for determining the location where a tweet was generated and also the user's home at a fine-grained level. They accumulate tweet location indicators from user profiles: self-declared home location, websites, timezone and possible location names mentioned in the tweets. Using multiple databases, they resolved these indicators to polygon-shaped administrative areas that they used to produce a spatial distribution of possible tweet locations of a user.

Chong and Lim, (2017) exploit use context information in a different way. They observe that both venues' active times and users' visiting places histories could help on tweet location prediction. Estimating the probability that a location is popular in a given time, they propose a smoothed kernel density estimation method.

Ikawa, Enoki, and Tatsubori, (2012) propose a standard machine learning approach. The learn a location estimation function in order to infer the location of a new users' message. In the learning phase they classified messages in two types: locations messages that represent the current user's location and expression messages that represent the current user' situation.

2.2.2.3 Summary

In Table 2.2, we summarize the discussed approaches in user location prediction. Differently from the approaches in home location inference, classification-based methods are not commonly used. We notice that, due to the aim of user location inference, no methods only use messages content without any other information. The inference granularity is fine: the location inference is always represented as coordinates or Point of Interest. Due to this, some of these algorithms could be possible candidates for a comparison with our method but, although Chong and Lim, (2017) declare that they find the location with a very small distance error, the evaluation method is not based on standard error measurement (i.e., kilometers). This specific approach is not easy to compare on, in general, with the majority of state of the art methods.

2.2.3 Tweet Location Prediction

It is reported that less than 1% of tweets are explicitly geotagged (Graham, Hale, and Gaffney, 2014). For this reason, a lot of research has been devoted to the topic of geolocating tweets. To facilitate the discussion, we subdivide

tweet location prediction approaches into generic tweet location prediction and mentioned location prediction.

We define generic tweet location prediction as the problem of inferring the location of a tweet whatever its content. We define mentioned location prediction as the problem of finding toponyms mentioned in tweets. These location predictions can benefit applications like location recommendation and disaster and disease analysis. In this thesis, we also present an algorithm that has the aim to geolocate a microblog message if it contains mentions of locations or of geographic classes (e.g., restaurant or hotel).

As compared to home location prediction and user location prediction, in this case the inputs of the approaches are different. The methodologies, in general, are the same.

2.2.3.1 Generic Tweet Location Prediction

Generic tweet location prediction presents several similarities to home location prediction problems explained before. The critical difference is the input: generic messages instead of messages grouped by users.

INFERENCE BASED ON TWEET CONTENT Due to the similar problem definition, generic tweet location prediction and home location prediction share the same techniques. As in Section 2.2.1, we can subdivide the inference based on tweet content into:

- word-centric/location-centric (Hulden, Silfverberg, and Francom, 2015; Kinsella, Murdock, and O'Hare, 2011; Priedhorsky, Culotta, and Del Valle, 2014).
- topic model based (Eisenstein, Ahmed, and Xing, 2011; Eisenstein, O'Connor, et al., 2010; Yuan et al., 2013).

Priedhorsky, Culotta, and Del Valle, (2014) employ Gaussian mixture models. They model the spatial usage of not only words but also *n-grams*. This choice was motivated by the fact that they want to infer the coordinate of just one message at a time. Therefore they find a solution to increase the quantity and quality of the information. As an example of the location-centric prediction model, we cite Kinsella, Murdock, and O'Hare, (2011). This is an information retrieval based solution. They treat both tweets and locations as Dirichlet smoothed unigram language models. Moreover, some approaches use classification methods to geolocate a message. An interesting example is Hulden, Silfverberg, and Francom, (2015). They classify tweets' text into discretized cell grids with words as features.

Other approaches use topic models to take into account the relations between topics and classify tweets w.r.t. the topics extracted (Eisenstein, Ahmed, and Xing, 2011; Eisenstein, O'Connor, et al., 2010; Yuan et al., 2013). A topic model could integrate different aspects related to locations as latent variables into a unified model. These methods are also called geo-topic-model-based methods.

An example of these model can be found in Eisenstein, Ahmed, and Xing, (2011) and Eisenstein, O'Connor, et al., (2010). In the first work (Eisenstein, O'Connor, et al., 2010), they extend the classical topic models by changing the conventional topics and produce location variation topics. For example, "NBA" may be a representative word in "basketball" topic produced by conventional models. By sampling from a Gaussian distribution centered at the "basketball" topic vector, the changed "basketball" topic for Boston may also include "Celtics" (a Boston-based team) while slightly changing other word frequencies. In the second approach (Eisenstein, Ahmed, and Xing, 2011), they propose a Sparse Additive GEnerative model (SAGE). This model supports the idea of the changed topics presented above, also managing the sparsity problem. Since these models use not only tweets but also users, these are home location prediction algorithms. However, we decide to present them here (as also X. Zheng, Han, and A. Sun, (2018) did in their survey) since they are considered the first ones that use a topic model to geolocate tweets and are fundamental to understand other work on the topic model.

A good example of generic tweet location prediction algorithms that exploit topic modeling is Yuan et al., (2013). They add an intermediate variable to topic models called regions between users and locations. For example, a user may have a "work" region and a "home" region, which are Gaussian distributions centered at her workplace and home address, respectively. Suppose the user is at her work region and wants to eat, i.e., choosing "eating" from her common interests. She will pick a restaurant near her workplace and write a tweet about eating and the work region, tagged with the name of the restaurant.

INFERENCE BASED ON TWEET CONTEXT Tweet posting times are indicative of users' home locations, where a distribution of posting times characterizes a user. Unlike home locations, for tweet location prediction we only access a tweet's posting time rather than a distribution. We cite here again Yuan et al., (2013) since they also use time periods in their model. Paraskevopoulos and Palpanas, (2015) improve the geolocation based on the content similarities of tweets, as well as their time-evolution characteristics. Cunha, Soares, and Rodrigues, (2014) analyze tweets to find a spatiotemporal pattern. They use data mining to analyze different types of information from tweets: spatial, temporal, social, and content information.

2.2.3.2 Mentioned Location Prediction

In general, mentioned location prediction can involve two sub-tasks:

- Mentioned Location *Recognition*: extract text fragments in a tweet that potentially refer to location names.
- Mentioned Location *Disambiguation*: map recognized location names to the right entry in a gazetteer.

Notice that, for well-formatted documents (e.g., news) entity recognition, and disambiguation are extensively studied (Nadeau and Sekine, 2007). Of

course for tweets, there are more problems related to the nature of the messages. The messages contain noise and are not always written in natural language.

INFERENCE BASED ON TWEET CONTENT There are many solutions that involve Name Entity Recognition (NER) in general (X. Liu et al., 2013; Ritter, Clark, Etzioni, et al., 2011), however we are mainly interested in discussing the ones that are specific to location entity recognition. These solutions are characterized by the use of gazetteers, e.g., GeoNames[11] or Foursquare[12].

Solutions of course exist for long and well-structured text. An example is Lieberman, Samet, and Sankaranarayanan, (2010).

Moreover, Zhang and Gelernter, (2014) use similar ideas in their research work. The use of the hierarchical structure of locations they employ is interesting. They consider parent-child locations pairs, e.g., "Paris" and "France" should be capital and country if they are used in the same tweet. However, to have the strongest coherence, they also use in the hierarchy the relation that can hold between the locations that are not just parent-child-like cities in the same country, e.g., "Paris" and "Bordeaux".

Differently from Zhang and Gelernter, (2014), C. Li and A. Sun, (2014) use disambiguation coherence at user-level rather than at tweet level. They exploit the idea that mentioned locations in a user's tweets are generally inside his/her living city. They first identify the city by aggregating candidate locations for the mentions and then refine those candidates with the city.

INFERENCE BASED ON TWEET NETWORK/CONTEXT Also in the case of mentioned location prediction the use of network and contextual information is essential. We cite two research proposals in this paragraph that are relevant: Fang and Chang, (2014) and Hua, K. Zheng, and Zhou, (2015).

Hua, K. Zheng, and Zhou, (2015) use the idea that others could influence a user on mentioning a location. If many users in his/her network used to indicate a place, more probably the user will say it. They adopt an incremental disambiguation approach. They pre-process a large number of tweets to estimate friendship-based user interest for locations. In addition, they also use the timestamp information of tweets. Fang and Chang, (2014) use geo-tags and timestamp in the disambiguation phase. Both these approaches should be useful for general entities, not only for locations.

2.2.3.3 Summary

In this section we provided a review of the literature on tweets location prediction algorithms. We considered two aspects of the tweets location:

- generic tweet location: even if the tweet does not explicit mention a location, we can associate a location with it (based on content/network contest).

[11] geonames.org
[12] foursquare.com

Table 2.3. Rßesearch papers on tweets location prediction.

	Content		Network	Context
	Word/Location -centric	Topic model based	-	-
Generic Tweet location inference				
Priedhorsky, Culotta, and Del Valle, (2014)	✓			
Kinsella, Murdock, and O'Hare, (2011)	✓			
Hulden, Silfverberg, and Francom, (2015)	✓			
Eisenstein, Ahmed, and Xing, (2011) and Eisenstein, O'Connor, et al., (2010)		✓		
Yuan et al., (2013)		✓		✓
Cunha, Soares, and Rodrigues, (2014)				✓
Paraskevopoulos and Palpanas, (2015)				✓
Mentioned location prediction				
Zhang and Gelernter, (2014)	✓			
C. Li and A. Sun, (2014)	✓			
Hua, K. Zheng, and Zhou, (2015)			✓	✓
Fang and Chang, (2014)			✓	✓

- mentioned location: if in the tweet is mentioned a location, we attach that location to it.

These two aspects involved different techniques to infer a location. Moreover, the ground truth are collected differently. In generic tweet location prediction we use geolocated tweets, in mentioned location prediction we use manually annotated tweets in order to be sure which one is the toponym appearing in the tweet. However, also these approaches are data-driven. We can find some approaches that use also external knowledge (Gelernter, Ganesh, et al., 2013; Gelernter and Mushegian, 2011; Zhang and Gelernter, 2014), in general in mention location prediction, but it is a plus only.

In Table 2.3, we provide a summary of the state of the art.

2.3 *Discussion*

In this chapter we analyzed the state-of-the-art by looking at two different research directions. First, we discussed various types of geographic knowledge, in order to provide a good understanding on what types of information we can use to augment current geolocation algorithms. Second, we provided an extensive review on Twitter message location inference.

This thesis forms part of the Twitter message location research field. However, since the solution proposed in this thesis is a knowledge-based algorithm, we discuss the state of the art of geospatial ontologies.

Giving another important point of view, we not only studied geospatial ontologies based on what they conceptualize but also we studied geospatial database on the sources from what they are generated. We classify geographic sources as authoritative and crowdsourced. One of the most popular examples of a semi-authoritative source is GeoNames, and one famous example of a crowdsourced data source is OSM. GeoNames is structured with its explicit ontology level, while OSM is not associated with an explicit ontological level, however, it is possible to extract semantic information from tags associated with toponyms. In the state-of-the-art, this problem is largely investigated and it is possible to attach a semantic level to OSM. Examples of knowledge obtained by enriching OSM with this semantic level are *LinkedGeoData* (Stadler et al., 2012)(LGD) an ontology that links OSM information to DBpedia and

other ontologies, and OpenStreetMap Facet Ontology (Di Rocco, 2013, 2016) an ontology structured on top of OSM for conceptualizing a city.

In regards of geolocation algorithm, we discuss specifically Twitter location prediction algorithms. It is important to note that various methods exist (e.g., Zhang and Gelernter, (2014)), which investigate the use of explicitly mentioned location information in microblog messages (see Section 2.2.3.2). However, the extracted information is only used as part of a training pipeline in order to create labels. An important problem is that such pipelines suffer from a large pre-processing overhead. Moreover, the majority of those methods relies on data sources such as GeoNames that do not contain much information at sub-city level. Finally, such methods rely on manually labeled messages in order to be trained. When such manually pre-processed datasets are not available, data-driven methods fail, while a knowledge-based solution could still be applied.

The proposed classification of location inference algorithms is only one possible classification. During our study we also concentrated the literature considering algorithms that geolocate at city-level or algorithms that geolocate at sub-city level, also called, fine-grained level. This is another classification using which we can categorize related work. In general, the majority of the research work shown in this chapter are city-level algorithms.

However, there exist some research projects that have also tackled the issue of fine-grained microblog message geolocation. Gelernter, Ganesh, et al., (2013) and Gelernter and Mushegian, (2011) improve the level of detail in geotagging locations that occur in disaster-related social messages. Using a dataset containing messages exchanged during the Haiti earthquake of 2010 and Japan tsunami of 2011, they improve the location identification at the level of the neighborhood, street, or building.

Paraskevopoulos and Palpanas, (2015) improve the geolocalisation based on the content similarities of tweets, as well as their time-evolution characteristics.

Differently, from Gelernter and Mushegian, (2011), we want to separate geolocation from a particular event. In our work, we propose the use of semantically enriched knowledge, in order to improve the precision of geolocation. To achieve that, a detailed spatial data source can be used for georeferencing (in our case, GeoNames and OSM) by means of ontologies.

Other approaches for inferring the fine-grained location of messages include C. Li and A. Sun, (2014) and G. Li et al., (2014). However, they focus mainly on Points of Interest (and in some cases district) of a particular place. Our aim is to geolocate also messages related to streets. We highlight also that those methods rely on a training phase.

In Table 2.4, we show some research work classified as city-level geolocation algorithms or fine-grained, i.e., sub-city level algorithms.

A complete overview of the article presented in this chapter related to geolocation is shown in Table 2.5.

Table 2.4. Different classification of most representative research papers in location prediction.

	City-level location			Fine-grained location	
	Gazetteers	NER	Ontologies	Similarity of texts	Event-detection
Amitay et al., (2004)	✓				
Lieberman, Samet, Sankaranarayanan, and Sperling, (2007)	✓				
Lieberman, Samet, and Sankaranarayanan, (2010)	✓				
Grover et al., (2010)	✓				
Purves et al., (2007)		✓	✓		
Stokes et al., (2008)		✓	✓		
Cheng, Caverlee, and Lee, (2010)	✓	✓(ML techniques)			
Kinsella, Murdock, and O'Hare, (2011)	✓	✓(ML techniques)			
Chandra, Khan, and Muhaya, (2011)	✓(ML techniques)				
Ikawa, Enoki, and Tatsubori, (2012)		✓(Context of messages)			
Mahmud, Nichols, and Drews, (2014)		✓			
Cunha, Soares, and Rodrigues, (2014)	✓	✓(spatio-temporal patterns)			
Schulz, Hadjakos, et al., (2013)	✓	✓(multi-indicator spatial approach)			
Gelernter and Mushegian, (2011)					✓
Paraskevopoulos and Palpanas, (2015)				✓	

TABLE 2.5. Research papers on geolocation algorithm.

	Home Location			User Location		Tweet Location	
	Content	Network	Context	Network	Context	Content	Network/Context
Backstrom, Kleinberg, et al., (2008)	✓						
Cheng, Caverlee, and Lee, (2010, 2013)	✓						
Ryoo and Moon, (2014)	✓						
Hecht et al., (2011)	✓						
Mahmud, Nichols, and Drews, (2012)	✓						
Kinsella, Murdock, and O'Hare, (2011)	✓						
Backstrom, E. Sun, and Marlow, (2010)		✓					
Kong, Z. Liu, and Huang, (2014)		✓					
McGee, Caverlee, and Cheng, (2013)		✓					
Chandra, Khan, and Muhaya, (2011)		✓					
Kwak et al., (2010)		✓					
Mahmud, Nichols, and Drews, (2014)			✓				
Efstathiades et al., (2015)			✓				
Sadilek, Kautz, and Bigham, (2012)				✓			
Schulz, Hadjakos, et al., (2013) and Schulz, Mencia, et al., (2014)					✓		
Chong and Lim, (2017)					✓		
Ikawa, Enoki, and Tatsubori, (2012)					✓		
Generic tweet location prediction							
Priedhorsky, Culotta, and Del Valle, (2014)						✓	
Kinsella, Murdock, and O'Hare, (2011)						✓	
Hulden, Silfverberg, and Francom, (2015)						✓	
Eisenstein, Ahmed, and Xing, (2011) and Eisenstein, O'Connor, et al., (2010)						✓	
Yuan et al., (2013)						✓	✓
Cunha, Soares, and Rodrigues, (2014)							✓
Paraskevopoulos and Palpanas, (2015)							✓
Mentioned location prediction							
Zhang and Gelernter, (2014)						✓	
C. Li and A. Sun, (2014)						✓	
Hua, K. Zheng, and Zhou, (2015)							✓
Fang and Chang, (2014)							✓

3 Geographic Knowledge

The discussion in Chapter 2 pointed out that state-of-the-art methods are manly data-driven. Supported by other authors such as Kinsella, Murdock, and O'Hare, (2011), we claim that is very difficult to reach sub-city level geolocation without exploiting prior knowledge. Thus, in order to accurately geolocate a microblog message, our approach exploits terms that refer to objects in the physical world the tweet may refer to. We will refer to such words of dual (semantic and spatial) nature as *geo-terms*. In this thesis, we propose a framework that, using an external geographical knowledge, can infer the position of a microblog message. Our framework is composed of two phases: during the first, offline phase, we preprocess the external geographical knowledge exploitable by the geolocation algorithm. In this chapter, we present in detail how we model a semantic gazetteer in order to obtain the external geographical knowledge. We first introduce the general problem and the idea of the solution that we propose and then we go in deep on the structure of the external knowledge that we need.

Appendix A contains a summary of the notions employed in this chapter.

3.1 The Role of Geographic Knowledge in Geolocation Algorithms

PROBLEM STATEMENT Given (*i*) a geographical area of interest, (*ii*) some external knowledge in the form of a set of objects located in the area of interest and their semantic descriptions, (*iii*) a microblog message originating from the area of interest mentioning at least one of these objects (referred to as *localizable* message) the goal is to infer the coordinates of the location inside the target area.

To cope with this problem, we propose an algorithm that we call Sherloc [1]. In Figure 3.1 we show a graphical representation of Sherloc steps. These five steps will be discussed in detail in Chapter 4. There are two important steps that involved the geographic knowledge: `k-NN identification` and `Phisycal locations`. More precisely, given a microblog message, Sherloc identifies the nearest neighbor terms closest to the message in terms of semantics, i.e., using the geographic knowledge. After that, Sherloc extracts the physical locations of semantically similar terms always using the geographic knowledge. Sherloc is able to infer the location of a message *m* without any prior training, exploiting only an indexed geographical external knowledge that we describe here. Notice that, these steps involve the geographic knowledge that we describe in this chapter.

[1] A simple word pun between Sherlock Holmes and location.

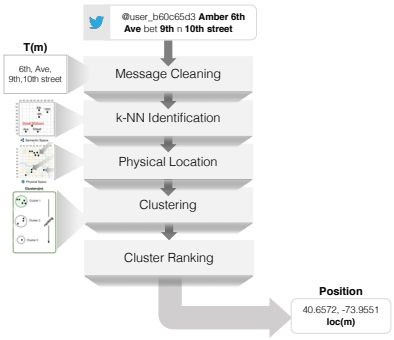

FIGURE 3.1. Sherloc schematic steps.

3.2 *Formalization of Geographic Knowledge*

In this section, we present the formalization of a semantic geographic gazetteer. Starting from the definition of a gazetteer, we discuss the representation of this type of datasets and their relation with a semantic level, represented by an ontology.

3.2.1 Gazetteers

Dictionaries of placenames are called gazetteers (Goodchild and Hill, 2008). They contain descriptive information about named places (i.e., a toponym) which can include their geographic locations (i.e., its geometries), types/categories, and other information. It has, in the most atomic representation, a simple structure as a set of pairs (top, p) where top is a toponym, and p is the location of the object denoted by top in the physical world.

Definition 3.1. *Given a toponyms top and its physical location p, defined as a pair of coordinates* $p = (c_1, c_2)$*, a gazetteer* Γ *is defined as:*

$$\Gamma = \{(top, g) | top \text{ is a toponym} \wedge g \in World\}$$

A more accurate version uses g, that is, the geometry of the object denoted by top instead of p. Gazetteers can be classified w.r.t. sources that generated them.

INSTANTIATION OF GAZETTEERS: GEONAMES AND OSM. As we presented in Section 2.1, we can classify geographic data sourced as authoritative

and crowdsourced. In order to have an analysis as complete as possible, we select GeoNames as semi-authoritative gazetteer and OSM as crowdsourced one. Both gazetteers follow the structure that we present here.

3.2.2 Semantic Gazetteers

Since we aim at exploiting more information about a geographical object than just its name, the geographic knowledge we rely on consists of a set of objects for which we have: (*i*) names in the form of textual descriptions (i.e., strings, also referred to as *toponyms*) used to refer to the object; (*ii*) a semantic description in terms of a category or class to which the object belongs; (*iii*) the geometry of the object, i.e., its representation in terms of points, lines, and polygons.

For example, *Saint Stephen Green* in Dublin, Ireland, is associated with the *park* class and the actual location of the park. Sometimes, a gazetteer organizes classes of objects into an ontology, in order to formalize the semantic relations between toponyms and classes and among classes. A gazetteer, in general, since it has no semantic level, has no inference procedure defined: given an assertion, it can only be considered to be entailed by the gazetteer if this assertion is explicitly expressed in it. We assume that, given an area, a gazetteer represents this area at the fine-grained level and an associated ontology describes the semantic relations between toponyms in a gazetteer.

More formally, let O be a geographic ontology, where objects are denoted by (*denotedBy* relation) strings (toponyms) and are related to classes by the *isInstanceOf* relation. Classes are related by the specialization/generalization *isSubClassOf* relation within the ontology. Let us now ignore place nodes (that are thus omitted in the ontology) and relate toponyms with classes directly. We obtain a graph where nodes are either strings (toponyms) or classes and edges model relationships among them. We can now define a semantic gazetteer as follows.

Definition 3.2. *Given a gazetteer Γ, a geographic ontology O, a semantic gazetteer $\Gamma(O)$ is a set of triples of the form $(top, class, g)$, where $(top, g) \in \Gamma$ and $class \in O$ is the class of which an object denoted by top is an instance.*

Since we assume a fixed geographic ontology O, consisting of a set of classes, a set of textual descriptions, and the relationships among them, in what follows when no ambiguity arises, we denote $\Gamma(O)$ simply with Γ.

To get an atomically decomposed representation of our knowledge, we split each geometry into a set of pairs of coordinates by considering a function ϕ transforming a geometry into a set of pairs of coordinates of the points that are used to represent such geometry. Thus, e.g., ϕ(WAY(-0.2578581 51.6536603,-0.2569054 51.6535858, ... ,-0.2561158 51.6522651)) = {(-0.2578581, 51.6536603), (-0.2569054, 51.6535858), ..., (-0.2561158, 51.6522651) }

Definition 3.3. *Given a geometry g, a flattering function ϕ is:*

$$\phi : Geom \rightarrow 2^{Coord}$$

$$\phi(g) = \{c_1, ..., c_n\}$$

where Geom is the set of geometry, Coord is the set of coordinates and $\{c_1, ..., c_n\}$ *is the set of coordinate of the geometry.*

Definition 3.4. *Given a semantic gazetteer* Γ *and a transforming function* ϕ*, an atomic semantic gazetteer* G *is:*

$$G = \{(top, class, p) \mid (top, class, g) \in \Gamma \wedge p \in \phi(g)\}$$

INSTANTIATION OF SEMANTIC GAZETTEERS: GEONAMES ONTOLOGY, LINKEDGEODATA AND OSM FACET ONTOLOGY. We can define the structure of *Geonames* not as a simple set of pairs but as a set of quadruples (*top*, *class*, *lat*, *lon*) where *top* is a toponym, *class* is the class of the toponyms it refers to come from its ontology, and *lat* and *lon* are the coordinates of the toponym.

Therefore, considering (*lat*, *lon*) as a point, the structure matches the semantic gazetteer representation.

We can define the structure of OSM not as a simple set of pairs but as a set of triples (*top*, *tags*, *component*) where *top* is a toponym, *tags* is a set of tags associated with *top* and *component* is the component that describes *top*.

The structure of LGD can be seen as a set of triples (*top*, *class*, *g*) where *top* is a toponym, *class* is the class of the toponym (coming from its ontology), and *g* is the geometry of the toponym. The triples are constructed using OSM data where the class associated with a toponym comes from its tags. Therefore, the structure matches the semantic gazetteer representation.

OSM facet ontology it is only a structure that works on top of OSM, therefore we create the semantic gazetteer starting from OSM.

Since *node* is the smaller unit in OSM data representation and relevant non-spatial tag information is associated with a node, we select from OSM all the data items that have the representation (*top*, *tags*, *node*). Given a mapping between tags and classes of the ontology, the tags are matched with the correct class, in order to obtain the structure of the semantic gazetteer. Notice that, given a mapping, not every tag has a representation in the ontology, therefore the ontology on OSM works also as a filter. Therefore, the structure of OSM facet ontology is a set of triples (*top*, *class*, *node*) where *top* is a toponym, *class* is a class match with at list one of the tags of *top* and *node* is the smaller unit in the dataset, and it is the geometry representation of *top* coordinates.

3.2.3 From Toponyms to One-grams: Geographic Knowledge

Toponyms are textual representations, i.e., strings that may consist of multiple terms (i.e., words). To get an atomically decomposed representation of our knowledge, we split toponyms in one-grams. There are two reasons to choose an atomically representation of the toponyms:

1. *Matching*: the idea to use toponyms as one-gram terms enable a fast and simple matching without requiring NLP techniques.

2. *Semantics*: using one-gram terms, we can unify the same terms also if they represent different geographical objects. This choice can increase ambiguity but, at the same time, it reflects the idea that in general in cities, some neighborhood share common names, e.g., in Liberty Island in New York City, different geographic objects share the term "Liberty" as Liberty Statue, Liberty museum, etc.

Therefore we define a function η transforming a string in the set of one-gram terms it contains. Thus, e.g., η(*Saint Stephen Green*)= {*Saint, Stephen, Green*}. The geographic knowledge that our approach exploits represents an association among one-gram terms, classes, and geographic points.

Definition 3.5. *Given a toponym top, a transforming function η is:*

$$\eta : Top \rightarrow 2^{Terms}$$
$$\eta(top) = \{t_1, ..., t_n\}$$

where Top is the set of toponyms, $Terms$ is the set of terms and $\{t_1, ..., t_n\}$ is the set of terms of a toponym.

Definition 3.6. *Given a atomic semantic gazetteer G, the geographical knowledge K_G associated with G is*

$$K_G = \{(t, class, p) \mid (top, class, p) \in G \wedge t \in \eta(top)\}$$

In the following, the set of all possible terms we employ for geolocating a microblog message consists of one-gram terms obtained from toponyms and names of classes, coming from the semantic gazetteer. Class information, indeed, helps in characterizing the object and are thus helpful for geolocating it.

Definition 3.7 (Geo-terms Related to a Geographic Knowledge)**.** *Let K_G be a geographic knowledge, the set of geo-terms related to K_G is $T(K_G) = \cup_{i=1}^{2} \pi_i(K_G)$ where π_i is the projection function extracting $i-th$ components.*

FROM GEONAMES, LGD, AND OSM FACET ONTOLOGY TO GEOGRAPHIC KNOWLEDGE. In order to transform the semantic gazetteers into geographic knowledge we preprocess them as follows:

- *Geonames* structure quite closely matches the semantic gazetteer definition we exploit. To obtain K_{Geonames}, we only need to apply function η.
- *LinkedGeoData* structure does not match the semantic gazetteer definition we exploit. To obtain K_{LGD}, we need to apply function ϕ on the geometry and function η on toponyms.
- *OSM facet ontology* structure quite closely matches the semantic gazetteer definition we exploit. To obtain K_{OSM}, we only need to apply function η. Since *Node* is the representation of a point in OSM, for the spatial part, we already have the correct representation.

For simplicity of notation, in the following, we indicate K_{Geonames}, K_{LGD} and K_{OSM} with the name of the related geographic data source only.

3.2.4 Semantic Embedding

From the geographic ontology O we can obtain a graph in which nodes are terms in $T(K_G)$. Edges among class nodes capture *isSubClassOf* relations among classes. Edges between one-gram term and class node correspond to the relations between one-gram terms included in a toponym and the class of the toponym, i.e., the relation obtained by composing $isIntanceOf \circ denotedBy \circ \eta$ relations. The semantic distance d^T between terms is the length of a path connecting the two corresponding nodes in such a graph (Dassereto, 2018). Since a *geo-term* could be instance of more than one class, the distance d^T between two *geo-terms* could be or not the shortest path between these two instances. To be more clear, given I_1 $isIntanceOf\, C_1 \wedge I_1$ $isIntanceOf\, C_2 \wedge I_1$ $isIntanceOf\, C_3$ and I_2 $isIntanceOf\, C_3$, the shortest path between I_1 and I_2 is 2 if they are connected from C_3, but, in specific situations we can know that I_1 corresponds to class C_1. Therefore, the distance between I_1 and I_2 is not the shortest path (see Figure 3.2).

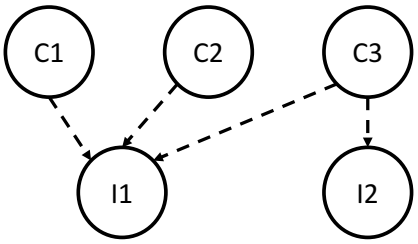

FIGURE 3.2. An example of distance between I_1 and I_2.

In order to avoid the problem to understand how to calculate d^T and to avoid the computational problem of computing every path, we propose to represent a geographic ontology by embedding it on a $n-$dimensional space.

In order to determine, given a term, its most semantically similar terms, we embed terms in a metric space with a distance function that captures the semantic distance among terms, thus maintaining the hierarchical structure of the ontology.

Definition 3.8 (Semantic Embedding Function)**.** *Let S be an n-dimensional metric space (a set of points) and d be a distance function between points in this space. A semantic embedding function emb is a function $emb : T(K_G) \rightarrow S$ such that, given $t_1, t_2, t_3, t_4 \in T(K_G), d^T(t_1, t_2) \geq d^T(t_3, t_4)$ if and only if $d(emb(t_1), emb(t_2)) \geq d(emb(t_3), emb(t_4))$. Function emb is invertible and, given a point $s \in S$, the term in $T(K_G)$ corresponding to s is denoted as $emb^{-1}(s)$.*

GEOGRAPHIC KNOWLEDGE AND EMBEDDINGS ALGORITHMS We need to process geospatial data looking at the ontology. Since an ontology is a tree, or more generally a graph, computing the distance between two objects is not always an easy problem. In order to address this problem, we represent the

semantics of geospatial data using embeddings. Specifically, in this thesis, we describe a methodology that involves the use of hierarchical structure embeddings. There are three possible cases in which we need to use embeddings to represent data:

1. when we cannot calculate the distance between two objects because it is too complicated;
2. when we know the distance between some pairs of objects but not for all pairs;
3. when we do not know the distance between objects.

In general, text data fall in the third case. In our work, we will consider textual data with a semantics, therefore our problem fall into the first case. In other words, calculating every path between two objects it too computationally intensive. Alternatively, our work could fall in the second case, i.e., we cannot be sure that the shortest path is a good distance metric between two geographical objects.

INSTANTIATION OF S: POINCARÈ EMBEDDING AND GLOVE The most important part of our approach is the representation of semantic space S related to K_G. In order to select the best embedding, we use an algorithm called Poincaré embedding (Nickel and Kiela, 2017), as well as a precomputed embedding called GloVe (described in Chapter 1). We use these two embeddings alternately since they provide two different representations of the same data, and they are learned from different training data: Poincaré is trained on K_G, GloVe is trained on Wikipedia and we filter the data using K_G.

We want to reach the following objective:

1. Precise representation of our geographic knowledge maintaining its structure,
2. Understand if an embedding can represent the geographic knowledge.

Since we require an embedding that can accurately represent distances between classes and instances[2] in the ontology, we need an algorithm that maintains the hierarchical structure of the ontology. Therefore, we embed the semantic knowledge in a Poincaré ball space, using the algorithm described in Nickel and Kiela, (2017). This non-Euclidean space is suitable for embedding data that have a hierarchical structure such as ontologies. In this way, we obtain a space in which classes and instances have a position (x, y coordinates) and, where similar semantic objects are near (in terms of distances) to each other.

GloVe is an embedding of words already described in Chapter 1. GloVe is a dictionary that for every word contains the corresponding vector in a Euclidean space. Since it is not learned on a geographic ontology instead on Wikipedia, we decide to map the terms that are in our geographic knowledge

[2] Actually, one-grams included in toponyms denoting instances.

on GloVe and obtain the specific representation in the GloVe space. In this way, we obtain an embedding space of our knowledge learned from Wikipedia. Not every geo-terms is in GloVe. Therefore we have a subset of our knowledge represented in the space. We use this representation in order to understand if a pre-computed space can represent geographic data as well since Wikipedia also contains geographic information.

4 Sherloc: a Sub-city Level Geolocation Algorithm

In this chapter, we present the Sherloc algorithm in detail. We start by giving an overview of the design of the algorithm, also presenting the pseudo-code of all its components. We then move on to describe each step of the algorithm in more detail.

Appendix A contains a summary of the notions involved in this chapter.

4.1 Overall Approach

In order to give an exaustive overview of the approach proposed in this thesis, we want to provide an analogy with Information Retrieval (IR) systems. Our aim is to give a better idea of how Sherloc works and how it is different from the other approaches in literature that are based on classification techniques.

Following the formal definition of an IR model proposed by Baeza-Yates, Ribeiro-Neto, et al., (1999), an IR model is a quadruple $(D, Q, \mathcal{F}, R(q_i, d_j))$ where:

1. D is a set of documents in the collection,
2. Q is a set of queries, i.e., what in IR are called user information needs,
3. $\mathcal{F}$ is a framework for modeling the document representation, the query, and their relationship,
4. $R(d_i, q_j)$ is a ranking function which associates a number with a query $q_j \in Q$ and a document $d_i \in D$. Such ranking defines an ordering among the documents concerning the query q_j.

Our framework, shown in Figure 4.1, has strong analogies with the IR system. More in details:

- D corresponds to $G|_A$, i.e., a pairs of geo-terms and coordinates (see Definition 3.4).
- Q corresponds to the set $T(m)$, i.e., a bag of geo-terms (see Section 4.2).
- $\mathcal{F}$ correspond to Sherloc. Sherloc models the semantic gazetteer, as described in the previous chapter, the query $T(m)$, as we will see in detail in the next section, and provides the algorithm to relate documents and queries that allow to retrieve the best documents (i.e., positions) for a query (i.e., a tweet).
- $R(d_i, q_j)$ corresponds to $rf(C)$ (see Section 4.3).

Notice that, in IR systems, the document representation can be seen as a triple $(d, [t_1, ..., t_n], [freq_1, ..., freq_n])$ where d is the document identifier, $[t_1, ..., t_n]$ is the list of terms contained in the document d and $[freq_1, ..., freq_n]$ is the frequence of the corresponding term in the document. The query $q \in Q$ is a list of terms and $R(d_i, q_j)$ ranks the documents according to their relevance for the query. In our case, the document representation is $K_G|_A$ (see Definition 3.6). However R ranks clusters and not single points. Our choice is due to the difference in information: we cannot create a set of clusters *a priori* in a query independent way since the clustering constructed on the message helps in the disambiguation phase.

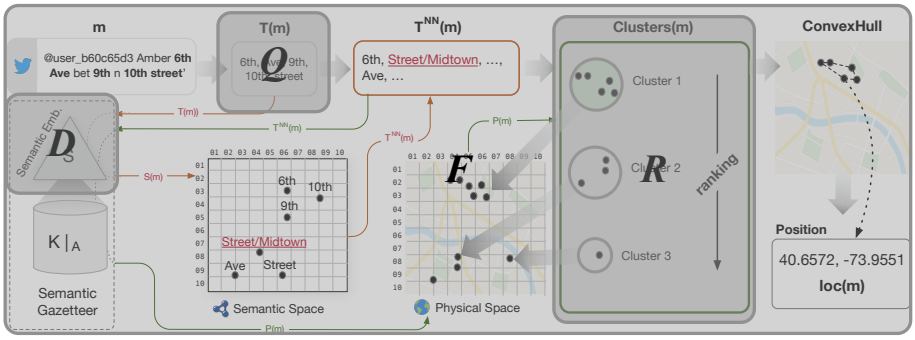

FIGURE 4.1. The framework architecture.

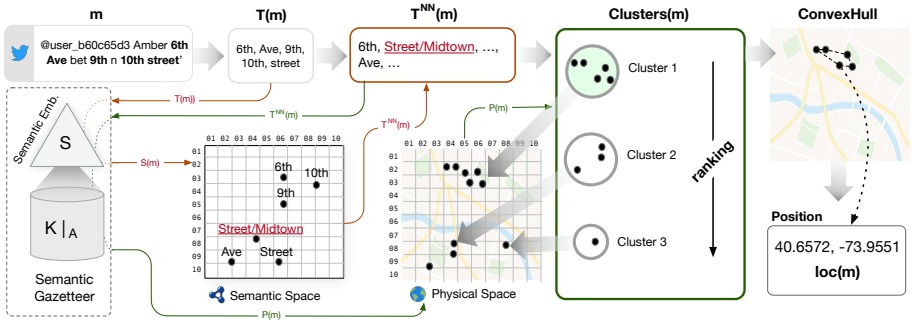

FIGURE 4.2. An example of the complete process carried out by Sherloc starting from the input message m and returning the inferred position $loc(m)$. The message is a real Twitter message from one of our evaluation datasets.

In Algorithm 1, we present how Sherloc works. As we introduced in Section 3.1, Sherloc has five steps. Comments in the pseudo-code highlight these five steps. The input is a message m. We first of all clean the message keeping only terms that exist in the geographic knowledge. With the cleaning process, we obtain a cleaned message $T(m)$ which is a set that only includes the terms present in $K_G|_A$. We call this a geo-message.

Using the geo-message, we compute n k-NN queries, one for each term, on the semantic space S constructed on $K_G|_A$. The new message is a set of points in S that has a maximum cardinality equal to δ (this is given as input to the algorithm). This part is described in Section 4.2. To infer the coordinates of our

Algorithm 1: Sherloc

Input: maximum number of similar terms δ, message m, semantic space S, geographic knowledge $K_G|_A$
Result: $[lat, lon]$ coordinates of the message m

```
// Step 1: Cleaning
```
$T(m) = \text{cleaning\_matching}(m)$;
```
// Step 2: Nearest neighbors identification
```
$k = \frac{\delta}{|T(m)|}$;
$NN(m) = \{\}$;
for $t \in T(m)$ **do**
    $NN(t) = \text{query}(t, S, k)$; // Find NN on S
    $NN(m).append(NN(t))$;
end
$T^{NN}(m) = emb^{-1}(S, NN(m))$; // Retrieve closest terms
```
// Step 3: Physical points extraction
```
$P(m) = points(T^{NN}(m), K_G|_A)$;
```
// Step 4: Clustering
```
$Clusters(m) = clustering(P(m))$;
```
// Step 5: Ranking and inferred coordinates
```
$dC(m) = \max(rf(Clusters(m)))$;
$[lat, lon] = centroid(ConvexHull(dC(m)))$;
return $[lat, lon]$

message, in step 3, we use the inverse function emb^{-1} to convert the geographic terms to their spatial coordinates. This set of points is the input of a clustering algorithm. The collected clusters are then ranked. The densest cluster is the cluster that Sherloc identifies as the cluster of the message. Finally, to geolocate the message, we compute the convex hull of this cluster. The predicted location is that of the centroid of this convex hull. This part is described in Section 4.3.

In Figure 4.2, the whole process is illustrated with an example of a tweet coming from New York City. In the rest of the chapter, we detail every step of Sherloc.

4.2 Geo-term Extraction and Semantic Similarity

In this section, we describe the message transformation: from the raw message to the set of most similar geo-terms that enrich our input information. In order to better understand the whole process, we use a real example to describe all the phases.

TARGET AREA Sherloc is an algorithm that exploits external geographic knowledge in order to infer the coordinates of unseen microblog messages. Its design goal is to be able to perform this inference with a sub-city level accuracy. As a result, it operates in a bounding box, which corresponds to the area of a given city.

We refer to this area as the target area A. It is represented as a rectangular bounding box with sides parallel to the axes of the chosen reference system. Such a rectangle is represented by the coordinates of two points, the top-left and bottom-right corners of the rectangle. Therefore, in the following we will indicate a target area A with $[c_1, c_2]$ where $c_i = (lat_i, lon_i), i = 1, 2$.

The bounding box A works as a filter on the external geographical knowledge. The geographic knowledge is defined in Definition 3.6, as:

$$K_G = \{(t, class, p) \mid (top, class, p) \in G \wedge t \in \eta(top)\}$$

Therefore, we filter K_G w.r.t. a bounding box A as follows:

$$K_G|_A = \{(t, class, p) \mid (t, class, p) \in K_G \wedge p \subset A\}$$

where t is a term that was part of a toponym, $class$ is the class which the term refers to and p is the point (i.e., coordinates) of the term.

MESSAGE CLEANING Our first step involves cleaning the message to keep only geo-terms, i.e., elements of $T(K_G)$. Therefore, the raw message, containing n terms, is transformed into a bag of geo-terms, i.e., a multiset, $T(m) = \wr t_1, ..., t_p \wr$, where $t_i \in T(K_G)$ and $p \leq n$. The geo-terms are in lower case. We use the standard cleaning technique on $T(m)$ and $K_G|A$ of deleting the stopwords. A message is then transformed into a bag of semantic points using the embedding, $S(m) = \wr s_1, ..., s_p \wr$ where $s_i = emb(t_i) \in S$ is the point corresponding to term t_i, $i = 1..p$. Notice that S is the semantic space, i.e., the embedding of our semantic gazetteer, see Definition 3.8.

NEAREST NEIGHBORS QUERYING. While Definition 3.8 defines functions emb and emb^{-1} with terms/points as arguments, we extend them to sets of terms/points as argument, respectively, in a straightforward way.

Given a term t, its most semantically similar terms are now characterized regarding nearest neighbors in S.

Definition 4.1 (Nearest Neighbors). *Let $s \in S$, $k \in \mathbb{N}$ and d be the distance on S, the k nearest neighbors of s, denoted as $NN_k(s) = [s_1, ..., s_k]$ are k points $s_1, ..., s_k \in S$, such that $d(s, s_1) \leq ... \leq d(s, s_k) \wedge \forall s' \in S, s' \notin [s_1, ..., s_k], d(s, s') \geq d(s, s_k)$.*

Given $S(m)$, we retrieve the k nearest neighbors of each point $s_i \in S(m)$. k is not set *a priori*, rather we rely on a heuristic in order to identify a meaningful value for it. Sherloc requires as input an integer δ that captures the maximum number of semantic points that can be associated with a message. This allows us to define the number of similar terms to retrieve for every message in order to obtain a set with cardinality at most δ. The value of k that we use for a given message m, such that $S(m) = [s_1, ..., s_p]$, denoted as $k(m)$ is thus computed as follows:

$$k(m) = \lfloor \frac{\delta}{p} \rfloor$$

The set $NN(m)$ of nearest neighbors for a message m such that $S(m) = [s_1, ..., s_p]$ is obtained as

$$NN(m) = \cup_{i=1}^{p} \{s \mid s \in NN_{k(m)}(s_i))\}.$$

Note that $|NN(m)| \leq \delta$. Referring to the example in Figure 4.2, $NN(m)$ is formed by the terms semantically "closest" to *"6th, Ave, 9th, 10th, street"* like *Street/Midtown.*

Example 4.1. *We receive as input the message m: "Hyde Park. Early running 7. #insandreuninlondon Hyde Park Gate. https://t.co/2dkQzvsFR8". Sherloc identifies* $T(m) = \{$*"hyde", "park", "hyde", "park", "gate"*$\}$*. The elements in* $T(m)$ *are then converted in the correspondent points on S, i.e., in Table 4.1.*

TABLE 4.1. Position in space S related to $K_G|A$ of the terms in $T(m)$.

$T(m)$	$S(m)$	
hyde	0.6469810	-0.334322
park	0.6527729	-0.314527
hyde	0.6469810	-0.334322
park	0.6527729	-0.314527
gate	0.9782389	-0.204853

At this stage, Sherloc performs a k-NN query on S for every points in $S(m)$ *with the input* δ *value that is divided by* $|S(m)|$*, returning the k value. In this example, Sherloc is running with* $\delta = 500$*. Therefore,* $k = \frac{\delta}{|S(m)|} = \frac{500}{5}$*. We obtain a new set* $NN(m)$ *with 138 entries with the identification number, i.e., the id of the term in the space S, of each term "close" to points in* $S(m)$*, i.e.,* $NN(m) = \{$ *1543, 2058, 2285, 2070, 2073, 2083, 2086, 2055, 2094, 2096, 2097, 2098, 53, 2103, 568, 948, 570, 2108, 578, 585, 1613, 1520, 2140, 2145, 2147, 2149, 1641, 1644, 1647, 1648, 2166, 631, 2169, 2172, 1666, 1814, 2182, 2185, 707, 2198, 2199, 2200, 2203, 2205, 1184, 674, 677, 681, 1197, 1710, 2230, 722, 2240, 800, 1219, 2246, 716, 2252, 1231, 721, 2258, 2260, 725, 2264, 2267, 734, 2272, 2276, 1768, 1769, 2283, 1773, 2286, 1781, 2295, 1786, 1792, 1793, 2306, 1795, 1288, 1804, 782, 790, 1819, 804, 807, 812, 816, 824, 1853, 833, 834, 1868, 1869, 846, 2389, 1878, 858, 1885, 862, 1222, 870, 871, 878, 895, 907, 911, 1774, 1942, 925, 928, 929, 1443, 1958, 1450, 941, 943, 645, 1460, 1463, 1464, 1981, 1982, 1475, 1441, 1996, 1997, 1491, 1500, 2298, 2015, 2020, 2032, 2033, 1524, 2037, 1531*$\}$.

Note that $T(m)$ is defined as a bag. The choice to maintain duplicates in the message is based on the intuition that if a user sends a message repeating many times the same information, (s)he wants to remark that information. We assume that this will retrieve more specific and relevant information.

4.3 From Semantic Space to Physical Space

In this section, we detail the Sherloc's steps related to the conversion from a semantic space to a physical space. We need to define the set of physical points related to a message. We then describe the clustering technique and consequently how Sherloc infers the location.

PHYSICAL LOCATIONS. So far, we have obtained a set of points $NN(m)$. We can now associate with each one of the points in this set a pair of physical coordinates inside the target area A. These coordinates refer to the locations associated with the terms in $T^{NN}(m) = emb^{-1}(NN(m))$ in the physical world.

Definition 4.2 (Physical points of a message)**.** *The set of physical points of a message m is defined as a set of points $p = (lat, lon)$ in A obtained as:*

$$P(m) = \cup_{t \in T^{NN}(m)} \{p | (t, class, p) \in K_G|_A\}$$

Example 4.2. *In Example 4.1, we obtained $NN(m)$. Now Sherloc converts the semantic points in $NN(m)$ to their corresponding terms, obtaing $T^{NN}(m)$. Then, we compute $P(m)$, converting every term in the corresponding physical point, i.e., coordinates. From $NN(m)$, we take back the term using the ids obtained $T^{NN}(m) = \{Arlington, Mansfield, ..., hostel, Astoria, ..., Wimbledon, Winchester\}$. Hence, we obtain $P(m)$ with 181 coordinates from GeoNames.*

Notice that $|T^{NN}(m)| \leq |P(m)|$ because a gazetteer is a dictionary and thus it is not a function. Therefore, more than one pair of coordinates can be associated with a toponym. For instance, Starbucks is a toponym corresponding to different cafeterias with different positions.

CLUSTERING. In order to identify a structure within the points in $P(m)$, we cluster them. We cluster points in $P(m)$, as shown in Figure 4.2, where a cluster (as discussed in Section 1.4.2) is a set of points that minimize distances among points inside the cluster while maximizing distance from points outside the cluster, based on a distance function (in this specific case the Euclidean distance). In this way, we highlight locations of particular interest based on the geo-terms contained in the message. Such clusters signify that there is a high concentration of semantically related elements in specific regions of our target area.

Note that clustering algorithms can produce multiple such clusters. Let $Clusters(m)$ denote the set of clusters obtained from $P(m)$.

RANKING AND INFERRED LOCATION. To select the cluster corresponding to an area which the message has the highest probability to come from, we rank the clusters in $Clusters(m)$ according to their density, i.e., the ratio between the number of points in $P(m)$ it contains and its area.

Definition 4.3 (Ranking Function)**.** *Clusters in $Clusters(m)$ are ranked through a ranking function $rf : Clusters \rightarrow \mathbb{N}$ defined as the density of the cluster, i.e., the ratio between the number of points in $P(m)$ belonging to Cl and its area.*

The ranking function calculates the density of a cluster.

Definition 4.4 (Top-1 Cluster)**.** *The "best" cluster is determined as the cluster with the highest density, i.e., $dC(m) = C \in Clusters(m)$ s.t.*

$$rf(C) = max_{C' \in Clusters(m)} rf(C')$$