

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/325529280>

GSP (Geo-Semantic-Parsing): Geoparsing and Geotagging with Machine Learning on Top of Linked Data

Chapter in Lecture Notes in Computer Science · June 2018

DOI: 10.1007/978-3-319-93417-4_2

CITATIONS

0

READS

28

4 authors, including:



[Stefano Cresci](#)

Italian National Research Council

40 PUBLICATIONS 391 CITATIONS

[SEE PROFILE](#)



[Leonardo Nizzoli](#)

Italian National Research Council

4 PUBLICATIONS 2 CITATIONS

[SEE PROFILE](#)



[Maurizio Tesconi](#)

Italian National Research Council

83 PUBLICATIONS 782 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Centro di Ricerca per l'Analisi delle Informazioni Multimediali [View project](#)



SOcial Sensing [View project](#)

GSP (Geo-Semantic-Parsing): Geoparsing and Geotagging with Machine Learning on top of Linked Data

Marco Avvenuti¹, Stefano Cresci², Leonardo Nizzoli^{1,2}, and Maurizio Tesconi²

¹ Department of Information Engineering, University of Pisa, Italy
`marco.avvenuti@unipi.it`

² Institute for Informatics and Telematics, IIT-CNR, Italy
`[stefano.cresci, leonardo.nizzoli, maurizio.tesconi]@iit.cnr.it`

Abstract. Recently, user-generated content in social media opened up new alluring possibilities for understanding the geospatial aspects of many real-world phenomena. Yet, the vast majority of such content lacks explicit, structured geographic information. Here, we describe the design and implementation of a novel approach for associating geographic information to text documents. **GSP** exploits powerful machine learning algorithms on top of the rich, interconnected Linked Data in order to overcome limitations of previous state-of-the-art approaches. In detail, our technique performs semantic annotation to identify relevant tokens in the input document, traverses a sub-graph of Linked Data for extracting possible geographic information related to the identified tokens and optimizes its results by means of a Support Vector Machine classifier. We compare our results with those of 4 state-of-the-art techniques and baselines on ground-truth data from 2 evaluation datasets. Our **GSP** technique achieves excellent performances, with the best $F1 = 0.91$, sensibly outperforming benchmark techniques that achieve $F1 \leq 0.78$.

Keywords: Geoparsing; machine learning; linked data; Twitter.

1 Introduction

The ever-growing amount of user-generated content in social networking and social media platforms has recently opened up new possibilities for studying and understanding the geospatial aspects of many real-world phenomena [14]. Yet, the vast majority of user-generated content lacks explicit and structured geographic information. For instance, only 1% to 4% of all Twitter posts (henceforth *tweets*) come with latitude and longitude coordinates [4]. This lack of geospatial information drastically limits the usefulness of social data for solving many important problems [2]. Indeed, having access to geotagged content could allow journalists to identify and cross-check the location of breaking news, by corroborating multiple posts related to the same event [17]. Similarly, geotagged health-related posts could be exploited by epidemiologists to track the spread

and diffusion of epidemics [9]. Furthermore, during mass emergencies, first responders could leverage crisis maps in order to track the unfolding situation and identify stricken locations that require prioritized intervention [17,3,1].

Given the importance of geospatial information in user-generated content, a large body of research has recently tackled the tasks of geoparsing and geotagging [11,14]. However, a number of challenges make these tasks extremely difficult, thus limiting the performance of current state-of-the-art techniques. Among such challenges is the problem of toponymic polysemy. Namely, a toponym might refer to different places according to the context in which it is used [1]. For example, the word “Washington” can refer to 30 different cities in the US³. Other challenges are the variable degree of granularity with which results should be returned (i.e., country-level, city-level, or even street- and building-level, depending on the application) [17], the time-evolving nature of geospatial information (e.g., new places and points-of-interest are continuously created, moved, and removed, especially in urban environments) [8,14], and the limited amount of context information typically available for social media content (e.g., tweets are limited to 280 characters).

Meanwhile, the Semantic Web has recently seen a flourishing of new datasets published as Linked Data, thus forming a rich and interconnected network of structured information. Indeed, such data already proved valuable in a number of practical domains, comprising health, journalism, and tourism [5,17]. A few preliminary works also investigated the usefulness of Linked Data from a geospatial perspective [12,17]. However, to date no working solution has ever been proposed to perform geoparsing and geotagging of text documents by exploiting Linked Data.

Contribution. We aim at demonstrating that previous state-of-the-art geoparsing and geotagging techniques can be outperformed by leveraging powerful machine learning algorithms on top of the rich and interconnected Linked Data. Our proof-of-concept is a context-agnostic *Geo-Semantic-Parsing* (GSP) technique for automatically associating geographic coordinates to text documents. GSP receives a text document as input and returns an enriched document, where all mentions of places/locations are associated to the corresponding geographic coordinates. To achieve this goal, in a first step GSP performs semantic annotation with the aim of identifying relevant parts of the input text, and to link them to pertinent resources (e.g., DBpedia entities) in the Linked Data cloud. Then, GSP exploits the rich and structured information associated to RDF resources to identify, via machine learning, geographic resources and to extract the right geographic coordinates for each resource.

Among the advantages that GSP has over previously proposed solutions are: (i) it does not require any explicit geographic information (e.g., GPS coordinates, location information, timezones), contrarily to [9]; (ii) it only exploits text data of input documents (e.g., it does not require any user information or social network topology), contrarily to [15]; (iii) it processes only one text document at a time (e.g., it does not require all tweets from a user’s timeline, or many documents

³ <https://en.wikipedia.org/wiki/Washington>

on a given topic), contrarily to [4]; (iv) it does not require users to specify a target geographic region but, instead, it geoparses and geotags places all over the world, contrarily to [17]; (v) by leveraging Linked Data, GSP is capable of extracting fine-grained, structured geographic information (e.g., street/building \rightarrow city \rightarrow county/region \rightarrow country) similarly to [9,13].

2 Related Work

The task of associating geographic information to social media content has been carried out mainly in 3 fashions: inferring (i) users home location [4,15], (ii) posts origin location [9,8], and (iii) locations of places mentioned within posts [11,17,13,14]. In the following, we thoroughly survey relevant approaches to the latter task, since our work also falls in that category.

The majority of approaches to (iii) are either based on named entity recognition (NER), gazetteer lookup and matching, language models (LM), or on a combination of them. Among state-of-the-art systems, is the one proposed by Middleton *et al.* [17]. It is publicly accessible via the *geoparsepy* Python package⁴ and it is based on a combination of NER and gazetteer matching. Input texts undergo token expansion and tokenization, before being matched against an in-memory cache of known location n -gram tokens. The pre-loaded cache of known locations is stored in a local planet-deployment of the OpenStreetMap’s gazetteer. Possible locations matches are disambiguated via heuristics and then ranked by confidence. Highest confidence matches are then selected as the output of the algorithm. Similarly, also the *mordecai* system by Halterman [13] is distributed as the namesake Python package⁵. The system takes unstructured text as input and returns structured geographic information. It is based on *spaCy*’s NER to extract toponyms from text. Then, it leverages the Geonames gazetteer to find the potential coordinates of extracted toponyms. The final coordinates returned by *mordecai* are selected via deep learning by a neural network classifier. In [11] is described an algorithm for extracting fine-grained mentions of places (i.e., streets, buildings) from text. It is based on lexico-semantic pattern recognition to identify streets and abbreviations, lexico-semantic matching enriched with gazetteer for spell checking and toponym identification and machine learning for abbreviation disambiguation and identification of buildings.

Grounding on the assumption that NER and gazetteer approaches to geoparsing are intrinsically hard due to the informal nature of social media content, in [14] is proposed a solution based on LM. The system in [14] does not operate on top of an explicit toponym dictionary, but instead it is trained on a large corpora of geotagged images complemented with textual tags. Given the set of tags related to a non-geotagged image, the learned model provides an estimate of the likelihood that the image refers to a particular location. Although proving very effective for geotagging images, [14] cannot directly operate on text documents, differently from our proposed approach.

⁴ <https://pypi.python.org/pypi/geoparsepy>

⁵ <https://github.com/openeventdata/mordecai>

In addition to the works briefly surveyed in this section, others have tackled the tasks of geoparsing and geocoding. However, we omitted a specific discussion of those works since they are largely overlapping with, or outperformed by, those already surveyed here.

3 The Geo-Semantic-Parsing approach

We aim at developing a technique that, given a text document \mathcal{T}_i , is capable of extracting the correct set \mathcal{C}_i of geographic coordinates related to all and only the K places mentioned within \mathcal{T}_i :

$$\mathcal{T}_i \xrightarrow{?} \{\mathcal{C}_{i,1}, \mathcal{C}_{i,2}, \dots, \mathcal{C}_{i,K}\}$$

To achieve this goal, our proposed *Geo-Semantic-Parsing* (**GSP**) technique employs machine learning on top of Linked Data, and combines the strengths of several state-of-the-art approaches introduced in Section 2. In detail, we firstly perform *semantic annotation* [10] in order to identify possible toponyms within the input document \mathcal{T}_i . Semantic annotation is a process aimed at augmenting portions of a plain-text (i.e., tokens) with pertinent links to RDF resources (\mathcal{E}) contained in knowledge-bases, such as DBpedia. The result of this process is an enriched (annotated) text where mentions of knowledge-bases entities in \mathcal{T}_i have been linked to the corresponding RDF resource. This annotation process is highly informative since it enables the exploitation of the rich information associated to the RDF resources $\mathcal{E}_{i,j}$ that have been linked to the j -th annotated portion of the i -th text. The resulting text enrichment effectively mitigates the drawbacks related to the limited amount of context. Semantic annotation also has the side effect of alleviating geoparsing mistakes caused by toponymic polysemy. In fact, some tokens of a plain-text can potentially be linked to multiple knowledge-bases entities. Semantic annotators automatically perform a disambiguating operation and only return the most likely reference to a knowledge-base entity for every annotated token [22]. Notably, this disambiguation operation is much more accurate than those carried out in previous works, such as those based on simple heuristics [17].

As a result of semantic annotation, each annotated token represents a relevant portion of \mathcal{T}_i , and the entity to which it has been linked can potentially contain geographic information useful for geoparsing. Thus, after the semantic annotation step carried out by a given annotator An , we *parse* the metadata of each RDF resource $\mathcal{E}_{i,j}$ linked to \mathcal{T}_i by An and we extract any geographic information it contains. In this way, every RDF resource with geographic information is automatically associated to a geographic coordinate $\mathcal{C}_{i,j} = (\text{lat}_{i,j}, \text{lon}_{i,j})$:

$$\begin{aligned} \mathcal{T}_i &\xrightarrow[\text{annotation}]{\text{semantic}} \{\mathcal{E}_{i,1}, \mathcal{E}_{i,2}, \dots, \mathcal{E}_{i,X}\} \\ &\xrightarrow{\text{parsing}} \{\mathcal{C}_{i,1}, \mathcal{C}_{i,2}, \dots, \mathcal{C}_{i,Y}\}, \quad Y \leq X \end{aligned} \quad (\text{GSP})$$

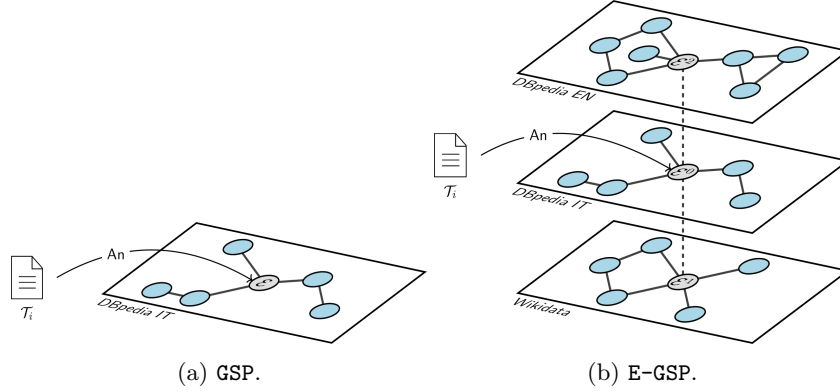


Fig. 1: The *expansion* step of E-GSP allows to extend the search for geographic information to all the $\mathcal{E}^0, \dots, \mathcal{E}^N$ resources of the \mathcal{E} sameAs graph, thus exploiting all knowledge-bases in which \mathcal{E} is described.

Given the basic GSP approach defined above, in the following we introduce two improvements that can be adopted in order to respectively (i) increase the number of geographic information retrieved (E-GSP) and (ii) optimize the output of the algorithm, thus limiting prediction errors (GSP-F). These two improvements can also be combined together, in the so-called E-GSP-F approach.

3.1 E-GSP: Extracting additional geographic information

Given a resource $\mathcal{E}_{i,j}$ linked to a portion of document \mathcal{T}_i by An , the basic GSP approach only exploits metadata of $\mathcal{E}_{i,j}$ in order to extract geographic information. However, the links between different semantic resources of the Linked Data graph open up the possibility to exploit metadata of many more nodes of the graph. Many different types of links exist between RDF resources, so as to express a broad range of different relations. Among them, `owl:sameAs` relations link the descriptions of equivalent RDF resources within and across knowledge-bases. Given an RDF resource $\mathcal{E}_{i,j}$, the graph of all $\mathcal{E}_{i,j}^0, \mathcal{E}_{i,j}^1, \dots, \mathcal{E}_{i,j}^N$ RDF resources directly or indirectly connected to $\mathcal{E}_{i,j}$ via `owl:sameAs` links is called the $\mathcal{E}_{i,j}$ *sameAs graph* [7]. Thus, in order to extract geographic information about a resource $\mathcal{E}_{i,j}$, the *Expanded Geo-Semantic-Parsing* (E-GSP) technique also exploits all semantically-equivalent resources $\mathcal{E}_{i,j}^0, \mathcal{E}_{i,j}^1, \dots, \mathcal{E}_{i,j}^N$ reachable by traversing the $\mathcal{E}_{i,j}$ sameAs graph:

$$\begin{aligned}
 \mathcal{T}_i &\xrightarrow[\text{annotation}]{\text{semantic}} \{\mathcal{E}_{i,1}, \mathcal{E}_{i,2}, \dots, \mathcal{E}_{i,X}\} \\
 &\xrightarrow{\text{expansion}} \{\{\mathcal{E}_{i,1}^0, \dots, \mathcal{E}_{i,1}^{N_1}\}, \dots, \{\mathcal{E}_{i,X}^0, \dots, \mathcal{E}_{i,X}^{N_X}\}\} \\
 &\xrightarrow{\text{parsing}} \{\mathcal{C}_{i,1}, \mathcal{C}_{i,2}, \dots, \mathcal{C}_{i,Y}\}, \quad Y \leq X \quad (\text{E-GSP})
 \end{aligned}$$

Figure 1 visually highlights the difference between the GSP and the E-GSP approaches, by leveraging the formalism of multilayer networks. The previously defined *expansion* step opens up the possibility to leverage the set of resources $\{\mathcal{E}_{i,j}^0, \mathcal{E}_{i,j}^1, \dots, \mathcal{E}_{i,j}^{N_j}\}$ for extracting geographic information, instead of the single resource used in GSP. However, only one geographic coordinate $\mathcal{C}_{i,j}$ must be associated to each resource $\mathcal{E}_{i,j}$ linked to the input document \mathcal{T}_i . Thus, the E-GSP approach also includes a *voting mechanism* used to select one coordinate, when multiple resources of the $\mathcal{E}_{i,j}$ sameAs graph contain geographic information. In detail, geographic coordinates are extracted for each resource of the $\{\mathcal{E}_{i,j}^0, \mathcal{E}_{i,j}^1, \dots, \mathcal{E}_{i,j}^{N_j}\}$ set. Then, a geospatial binning is applied in order to group and count coordinates that lay near to one another. This process acts pretty much like a geographic clustering step. The final $\mathcal{C}_{i,j}$ coordinate associated to $\mathcal{E}_{i,j}$ is the geographic centroid of the biggest cluster. When two or more clusters contain the same number of elements, the winning cluster is picked as the one containing the “best” resource $\hat{\mathcal{E}}_{i,j}$ (i.e., the most reliable one). In E-GSP, resources in $\{\mathcal{E}_{i,j}^0, \mathcal{E}_{i,j}^1, \dots, \mathcal{E}_{i,j}^{N_j}\}$ are ranked by a score $S_{i,j}^n$, and the “best” resource is the one achieving the highest score: $\hat{\mathcal{E}}_{i,j} = \mathcal{E}_{i,j}^{\bar{n}} \mid \bar{n} = \arg \max_n S_{i,j}^n$. The score $S_{i,j}^n$ quantifying the goodness of a resource can be computed in many ways – e.g., by computing its Page Rank value, by employing one of the many ranking algorithms for Linked Data or by computing simple metrics of completeness, such as the number of predicates that describe the RDF resource.

Notably, the voting mechanism introduced in E-GSP also solves possible mistakes caused by spurious wrong metadata in the description of an RDF resource. For example, extracting the coordinates for the city of Milan (Italy) exclusively from the corresponding resource in the Italian DBpedia⁶ results in a mistake, since such coordinates point (at the time of writing) to a place in Switzerland. Instead, by aggregating and counting the coordinates found in the sameAs graph of Milan, it is actually possible to infer the correct coordinates. Considering that our proposed technique grounds on data contained in collaboratively-curated knowledge-bases, the ability to automatically correct inconsistencies and mistakes represents a much desirable feature [20].

3.2 GSP-F: Filtering results to increase correctness

As with any algorithm, not all results returned by the algorithm are correct. For our task, this means that some of the $\mathcal{C}_{i,j}$ coordinates found with the GSP approach, might have been erroneously identified. In order to reduce the occurrences of wrong predictions, many machine learning algorithms include an optimization phase where candidate results are evaluated before being returned to the users. In this way, only those results for which the algorithm is reasonably confident are actually returned. In order to enhance the correctness of the coordinates identified by our GSP technique, we devised the *Geo-Semantic-Parsing with Filtering* (GSP-F) approach, in which a binary machine learning classifier

⁶ <http://it.dbpedia.org/resource/Milano>

is trained to evaluate candidate results. The classifier takes as input a number of features and outputs a binary label defining whether a candidate result $\mathcal{C}_{i,j}$ should actually be returned ($\bar{\mathcal{C}}_{i,j}$), or whether it should rather be discarded, being probably incorrect:

$$\begin{aligned} \mathcal{T}_i &\xrightarrow[\text{annotation}]{\text{semantic}} \{\mathcal{E}_{i,1}, \mathcal{E}_{i,2}, \dots, \mathcal{E}_{i,X}\} \\ &\xrightarrow{\text{parsing}} \{\mathcal{C}_{i,1}, \mathcal{C}_{i,2}, \dots, \mathcal{C}_{i,Y}\}, \quad Y \leq X \\ &\xrightarrow{\text{filtering}} \{\bar{\mathcal{C}}_{i,1}, \bar{\mathcal{C}}_{i,2}, \dots, \bar{\mathcal{C}}_{i,Z}\}, \quad Z \leq Y \end{aligned} \quad (\text{GSP-F})$$

The features used by the machine learning classifier, its training and its evaluation are described in detail in Section 4.3.

3.3 E-GSP-F: Expanded GSP with Filtering

The E-GSP and the GSP-F techniques previously described can be employed simultaneously, on top of the basic GSP approach, for optimized performances. Indeed, the E-GSP and the GSP-F improvements are orthogonal, since the former aims at increasing the set of candidate results, while the latter reduces actual results by filtering out those candidate results that are likely to be incorrect. The E-GSP-F technique resulting from the combination of E-GSP and GSP-F is defined in the following, and it is described by the pseudo-code in Algorithm 1:

$$\begin{aligned} \mathcal{T}_i &\xrightarrow[\text{annotation}]{\text{semantic}} \{\mathcal{E}_{i,1}, \mathcal{E}_{i,2}, \dots, \mathcal{E}_{i,X}\} \\ &\xrightarrow{\text{expansion}} \{\{\mathcal{E}_{i,1}^0, \dots, \mathcal{E}_{i,1}^{N_1}\}, \dots, \{\mathcal{E}_{i,X}^0, \dots, \mathcal{E}_{i,X}^{N_X}\}\} \\ &\xrightarrow{\text{parsing}} \{\mathcal{C}_{i,1}, \mathcal{C}_{i,2}, \dots, \mathcal{C}_{i,Y}\}, \quad Y \leq X \\ &\xrightarrow{\text{filtering}} \{\bar{\mathcal{C}}_{i,1}, \bar{\mathcal{C}}_{i,2}, \dots, \bar{\mathcal{C}}_{i,Z}\}, \quad Z \leq Y \end{aligned} \quad (\text{E-GSP-F})$$

4 System implementation

4.1 Semantic annotation

In recent years, the tasks of *semantic annotation*, *wikification* and *entity linking* have attracted a great interest from scholars of many disciplines [10]. This large body of work resulted in a number of readily available tools and Web APIs capable of effectively performing semantic annotation. Thus, when implementing the semantic annotation step of our system, we could rely on a number of well-known, state-of-the-art, off-the-shelf semantic annotators. In particular, we developed Python wrappers to DBpedia Spotlight [16], TagMe [10], Dexter 2.0 [21] and Dandelion⁷. All these systems provide Web applications^{8,9,10,11}

⁷ <https://dandelion.eu/>

⁸ DBpedia Spotlight: <http://demo.dbpedia-spotlight.org/>

⁹ TagMe: <https://tagme.d4science.org/tagme/>

¹⁰ Dexter 2.0: <http://dexter.isti.cnr.it/demo/>

¹¹ Dandelion: <https://dandelion.eu/semantic-text/entity-extraction-demo/>


```

input :  $\mathcal{T}$                                      // tweet to analyze
         $\text{An}$                                        // semantic annotator to use
output:  $\bar{\mathcal{C}}$                                        // coordinates extracted from  $\mathcal{T}$ 

1  $\mathcal{C} = ()$ ;
2  $\mathcal{E} = \text{semanticAnnotation}(\text{An}, \mathcal{T})$ ;
3 for  $i = 1$  to  $|\mathcal{E}|$  do
4    $\text{aliases} = \text{expansion}(\mathcal{E}_i)$ ;
5    $\text{allCoords} = ()$ ;
6   for  $n = 1$  to  $|\text{aliases}|$  do
7      $\text{coords}_{i,n} = \text{parsing}(\mathcal{E}_i^n)$ ;
8     if  $\text{coords}_{i,n} \neq \text{null}$  then
9        $\text{allCoords.append}(\text{coords}_{i,n})$ ;
10    end
11  end
12   $\text{bestCoord} = \text{votingMechanism}(\text{allCoords})$ ;
13   $\mathcal{C.append}(\text{bestCoord})$ ;
14 end
15  $\bar{\mathcal{C}} = \text{filter}(\mathcal{C})$ ;
16 return  $\bar{\mathcal{C}}$ ;

```

Algorithm 1: E-GSP-P algorithm for associating geographic coordinates $\bar{\mathcal{C}}$ to document \mathcal{T} , by exploiting the semantic annotator An .

as well as RESTful APIs for programmatic access. Each wrapper is capable of querying the Web APIs of the related semantic annotator, passing a textual document and returning URIs of the RDF resources found in the document by the semantic annotator. Despite the different inner functioning of the 4 supported semantic annotators, all wrappers expose a common interface to the rest of our system, so that the choice of the specific annotator to use is transparent to the users and to the other components of our system. Among the information returned by our wrappers for each found RDF resource, is the DBpedia URI of the resource and a confidence score ρ expressing how likely is a specific annotation (*token* \rightarrow *resource*) to be correct. Notably, our proposed technique does not depend on any specific annotator, and indeed it can be implemented with any annotator currently available or with a combination of them.

4.2 Extraction of geographic information

Supported knowledge-bases. All proposed versions of our technique involve a *parsing* step, where RDF resources are associated to geographic coordinates, whenever possible. This step is performed by looking for geographic metadata among the predicates of RDF resources. Possibly relevant metadata is fetched in JSON format via SPARQL queries, and then it is parsed. As a result of the semantic annotation step, in GSP and GSP-F all the resources to parse belong to DBpedia (either the English DBpedia or a different one, depending on the language of the input document \mathcal{T}_i). However, as a consequence of the *expansion* step employed in E-GSP and E-GSP-F, the system may be required to parse other equivalent resources belonging to different knowledge-bases (e.g., YAGO, Freebase, Geonames, etc.). In order to query a knowledge-base for metadata of a resource, our system must necessarily know the SPARQL endpoint of the

knowledge-base. However, it is not possible to know in advance all knowledge-bases that should be queried, since they depend on the resources linked to each specific input document, which is known only at runtime. Thus, in order to be able to parse the highest possible set of RDF resources, we provided support for all DBpedias deploying their SPARQL endpoint to the standard URL `http://<lang>.dbpedia.org/sparql`, as well as to 11 other well-known knowledge-bases for which we manually specified the SPARQL endpoint by means of a configuration file. Notably, our set of supported knowledge-bases leverages the results of previous studies on the distribution of geospatial information in Linked Open Data [12]. Furthermore, such set can be easily extended by adding additional SPARQL endpoints to the configuration file.

Geographic RDF predicates. In Linked Data there exist many different RDF predicates designed to store geographic information (e.g., `geo:lat` and `geo:long`, `georss:point`, etc.). The capability of our system to associate a set of geographic coordinates to an RDF resource depends on its ability to parse as many as possible of such RDF predicates. In our implementation, we provided support for as many as 45 RDF predicates. Since the geographic information conveyed by the supported predicates can be represented in different formats (e.g., decimal degrees; degrees, minutes, seconds), we then implemented a set of simple formulas for converting the different input formats into *decimal latitude and longitude* coordinates. As a result, the output of the *parsing* step is represented, wherever available, by a geographic coordinate $\mathcal{C}_{i,j} = (\text{lat}_{i,j}, \text{lon}_{i,j})$ for each provided RDF resource $\mathcal{E}_{i,j}$, thus adhering to the specifications defined in Section 3.

Voting mechanism. In the E-GSP and E-GSP-F approaches, multiple coordinates extracted from the same graph of a resource are clustered by a voting mechanism. In our implementation, such coordinates are approximated to the third decimal place before being grouped and counted. This equals to performing a geographic binning, with bin width $\simeq 0.1$ kilometers ($\simeq 0.06$ miles) at the equator. Then, the scoring metric used to resolve draws between geographic clusters containing the same number of resources, simply quantifies the number of predicates of each resource. In other words, in our implementation RDF resources described by a higher number of predicates are considered more reliable than those described by fewer predicates.

4.3 Machine learning filtering

The GSP-F and the E-GSP-F approaches involve a *filtering* step, where candidate results $\mathcal{C}_{i,j}$ are evaluated before being returned to the users. Candidate results that are likely to be incorrect ($\mathcal{C}_{i,j}$) are discarded (pruned), while those for which the system is confident ($\bar{\mathcal{C}}_{i,j}$) are outputted. Within this context, a candidate result is represented by a resource $\mathcal{E}_{i,j}$ (associated to a geographic coordinate) that has been linked to a portion of the input document \mathcal{T}_i by a semantic annotator An. This filtering operation is performed by a binary machine learning classifier, implemented with a Support Vector Machine (SVM). Once trained, the SVM classifier takes as input a set of features describing one candidate result at a

time, and it outputs the predicted class label for that candidate result (either $\underline{\mathcal{C}}_{i,j}$ or $\overline{\mathcal{C}}_{i,j}$).

Features. Features describing candidate results are divided into 3 classes, according to the information they convey. The first class of features describes the *textual* properties of the input document \mathcal{T}_i . We rely on a state-of-the-art natural language processing tool for Italian and English texts [6] for the analyses of input documents. Given a token (or a sequence of tokens) linked to a resource $\mathcal{E}_{i,j}$, we compute categorical features as its (i) coarse- and (ii) fine-grained part-of-speech (POS) tag, (iii) its morphosyntactic tag, (iv) its type of named entity (if any) and (v) a binary feature representing whether the token begins with a capital letter. Our *textual* features capture linguistic patterns correlated to the use of toponyms in texts and include information used by state-of-the-art NER approaches [11,17,13]. The second class of features evaluates the *link* established by the semantic annotator **An** between the token and the resource $\mathcal{E}_{i,j}$. Features of this class are (i) the ρ confidence of **An** for the link and the (ii) absolute and (iii) percentage edit distance between the token and the **rdfs:label** of $\mathcal{E}_{i,j}$. Notably, the confidence feature that we exploit is similar to that of [17]. The third class of features describes the properties of the $\mathcal{E}_{i,j}$ *resource*. Specifically, we define a (i) binary and an (ii) integer feature for representing if, and how many times, the token appears in the **ontology:abstract** of $\mathcal{E}_{i,j}$. Then, we devised 3 categorical features for modeling the structural properties of $\mathcal{E}_{i,j}$ as (iii) the RDF ontologies/vocabularies of the predicates of $\mathcal{E}_{i,j}$ (e.g., OWL, RDFS, FOAF, etc.), (iv) its RDF predicates and (v) its **rdf:types**. Indeed, structural properties of Linked Data have already proved useful for a number of ranking, clustering, and classification tasks [18,19].

Training pipeline. The initial data for training the SVM classifier is represented by $\simeq 7,000$ manually-annotated candidate results. This dataset is almost balanced with 57% correct geo-predictions ($\overline{\mathcal{C}}_{i,j}$) and 43% incorrect ones ($\underline{\mathcal{C}}_{i,j}$). The training pipeline begins with a stratified sampling of the whole dataset into a training (80%) and a testing (20%) dataset. Then, preprocessing steps perform imputation of missing feature values and scaling, in order to improve results of the learning algorithm. Since some of our categorical features have high dimensionality (e.g., we have over 6,800 different **rdf:types**), we perform an L1-based feature selection step, which resulted in a reduced set of only 60 features. Next in the pipeline, a hyperparameters tuning process picks the best settings for the SVM classifier (RBF kernel, $C = 100$, $\gamma = 0.001$), via cross-validation over the training dataset. We then learn our SVM models on the training dataset and evaluate their performances in classifying candidate results of the testing dataset. In detail, we learn 4 different models, one for each of the supported semantic annotators, plus one global model using all available training data.

Classification results. Table 1 reports the performances of the SVM classifiers against testing data. In order to better evaluate the difficulty of the task and the results of our SVM classifiers, we also compared results with those of 2 simple baseline classifiers that respectively (i) always predict the majority class and (ii) output random predictions. As shown, our best classifier achieves an excellent

model	evaluation metrics					
	Precision	Recall	Specificity	Accuracy	F1	MCC
Baselines						
majority class	0.569	1.000	0.000	0.569	0.725	–
random classifier	0.475	0.499	0.512	0.506	0.487	0.012
SVM classifiers						
TagMe	0.916	0.932	0.928	0.930	0.924	0.858
DBpedia Spotlight	0.968	0.949	0.971	0.960	0.959	0.921
Dexter 2.0	0.966	0.986	0.974	0.979	0.976	0.958
Dandelion	0.903	0.940	0.911	0.924	0.921	0.849
global	0.963	0.980	0.967	0.973	0.971	0.946

Table 1: Evaluation of the *filtering* step of GSP-F and E-GSP-F.

$F1 = 0.976$, way higher than the baselines and comparable with the other SVM classifiers. Since the global classifier achieved only slightly lower results with respect to the best one, in our system implementation we relied on it instead of using one different classifier for every semantic annotator.

5 Evaluation

Our evaluation metrics are those typically used in previous machine learning and entity linking tasks [14]. Specifically, we consider as a correct match a prediction by an algorithm/technique when the predicted coordinate falls within a certain distance threshold (e.g., a few kilometers/miles) from the ground-truth coordinate. This assumption is the same already made in recent machine learning challenges, such as the *MediaEval 2016 Placing Task*¹², where participants were asked to estimate the locations of multimedia items (i.e., photos or videos).

5.1 Datasets

Although capable of working with text documents of any kind, we benchmarked our proposed GSP technique with social media data, namely for geoparsing and geotagging tweets. Indeed, social media represent an environment where such techniques are most needed [4]. In addition, tweets are short documents filled with jargon and colloquial expressions, thus representing a challenging proving ground for our technique.

The first evaluation dataset (henceforth labeled **ENG-NEEL**) is composed of 9,289 English tweets. It is the official dataset of the *Named Entity rEcognition and Linking (NEEL 2016)* challenge¹³. The dataset comprises tweets extracted from a collection of over 18 million documents including event-annotated tweets covering multiple noteworthy events from 2011 and 2013, and tweets extracted

¹² <http://www.multimediaeval.org/mediaeval2016/>

¹³ <http://microposts2016.seas.upenn.edu/challenge.html>

from Twitter’s Firehose in 2014 and 2015 via a selection of hashtags. Annotated mentions of places/locations in tweets are provided by *NEEL 2016* organizers.

The second evaluation dataset (henceforth labeled **ITA-DSTR**) is composed of 1,807 Italian tweets that we collected in the aftermath of 2 major natural disasters in Italy, respectively the *Emilia 2012 earthquake* and the *Sardinia 2013 flood*. Such dataset has recently been used in a number of works related to crisis mapping and emergency management [1]. Mentions of places/locations in the dataset have been manually annotated by 2 graduate students.

The distance threshold for comparing the obtained results with the ground-truth was set equal to 50km for the **ENG-NEEL** dataset and to 20km for the **ITA-DSTR** dataset.

5.2 Benchmarks

To thoroughly evaluate our proposed technique, we compared our results to those of different state-of-the-art geoparsing and geotagging techniques, namely the geoparsing technique by Middleton *et al.* [17] and the geoparsing and geocoding technique by Halterman [13]. Such benchmark techniques are thoroughly described in Section 2. Since the technique by Middleton *et al.* only outputs location tokens extracted from the input text, we obtained coordinates for the tokens via queries to OpenStreetMap’s APIs. Instead, the technique by Halterman already outputs coordinates of found locations. Thus, no further operations are needed in order to include [13] in our evaluation.

In addition to [17,13], we also compared our results to those of 2 baselines. The “geoparser” baseline leverages the *geopy* Python package¹⁴ and employs the ArcGIS service to extract coordinates from tweets. The “NER + geocoder” baseline performs NER and then geocodes location NEs via queries to the Web APIs of Google Maps. The NER step is performed using the well-known *polyglot* natural language processing pipeline¹⁵.

5.3 Results

Table 2 reports geoparsing results for all our techniques and all benchmarks, on the two evaluation datasets. As shown, the simplest of our proposed techniques (**GSP**) already achieves results that are in line with those of the best benchmarks. In fact, it achieves the second best *F1* on the **ITA-DSTR** dataset and the best *F1* on the **ENG-NEEL** dataset, when compared to the benchmarks. Results also show the effectiveness of our 2 improvements **E-GSP** and **GSP-F**. As hypothesized in Section 3, the *expansion* step of **E-GSP** increases the number of retrieved coordinates allowing to boost *Recall* from 0.664 to 0.769 on **ITA-DSTR** and from 0.403 to 0.574 on **ENG-NEEL**, an average improvement of +37%. Similarly, the machine learning *filtering* step in **GSP-F** boosts *Precision* from 0.686

¹⁴ <https://geopy.readthedocs.io/>

¹⁵ <https://polyglot.readthedocs.io/>

technique	ENG-NEEL				ITA-DSTR			
	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>	<i>F1</i>	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>	<i>F1</i>
Benchmarks								
geoparser	0.030	0.139	0.025	0.050	0.242	0.453	0.182	0.306
NER + geocoder	0.331	0.297	0.186	0.313	0.879	0.698	0.636	0.777
Halterman (<i>mordecai</i>) [13]	0.291	0.288	0.169	0.289	0.625	0.338	0.282	0.439
Middleton (<i>geoparsepy</i>) [17]	0.173	0.373	0.134	0.236	0.567	0.754	0.478	0.647
Our contributions								
GSP	0.335	0.403	0.217	0.356	0.686	0.664	0.506	0.668
E-GSP	0.398	0.574	0.295	0.455	0.671	0.769	0.559	0.693
GSP-F	0.655	0.449	0.363	0.531	0.894	0.692	0.640	0.779
E-GSP-F	0.888	0.634	0.588	0.738	0.977	0.813	0.798	0.885

Table 2: Evaluation results of GSP-derived techniques and comparison with state-of-the-art techniques and baselines.

technique	annotator	ENG-NEEL				ITA-DSTR			
		<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>	<i>F1</i>	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>	<i>F1</i>
E-GSP-F	TagMe	0.905	0.673	0.629	0.772	0.967	0.853	0.831	0.906
	DBpedia Spotlight	0.888	0.697	0.640	0.781	0.983	0.835	0.824	0.903
	Dexter 2.0	0.860	0.513	0.474	0.643	0.992	0.716	0.711	0.829
	Dandelion	0.901	0.652	0.608	0.756	0.965	0.847	0.824	0.902

Table 3: Detailed results of the E-GSP-F technique, when using different annotators for the *semantic annotation* step.

to 0.894 on ITA-DSTR and from 0.335 to 0.655 on ENG-NEEL, with an average improvement of +63%. As a result, the E-GSP-F technique, combining both E-GSP and GSP-F, largely outperforms all other techniques and benchmarks on both datasets. Indeed on ITA-DSTR, it achieves $F1 = 0.885$ versus $F1 = 0.777$ of the best benchmark. The performance gap is even more pronounced on the much more challenging ENG-NEEL dataset, where E-GSP-F achieves $F1 = 0.738$ versus $F1 = 0.313$. Furthermore, detailed results of E-GSP-F reported in Table 3 also show that our performances are consistent when using different semantic annotators, with no annotator clearly outperforming the others. The best results are obtained with DBpedia Spotlight on the ENG-NEEL dataset and with TagMe on ITA-DSTR, respectively with $F1 = 0.781$ and $F1 = 0.906$. Instead, Dexter 2.0 achieves slightly worse results than all other annotators on both datasets, mainly because of the lower *Recall*. As such, it is not recommended to use an implementation of our techniques solely based on Dexter 2.0.

In addition to an evaluation of the correctness of our geo-predictions, we also evaluated the geospatial granularity of our results. Table 4 shows the different types of places/locations extracted from tweets by E-GSP-F. As seen, E-GSP-F geoparsed both coarse- (e.g., countries, regions) and fine-grained (e.g., buildings) locations, depending on the content of tweets. This result seems to favor the application of our proposed technique in a broad range of different situations. Finally, in Figure 2 we reported the top 5 knowledge-bases from

place types [#]
dbo:Airport
dbo:ArchitecturalStructure
dbo:Building
dbo:Castle
dbo:City
dbo:Country
dbo:Island
dbo:Mountain
dbo:Municipality
dbo:NaturalPlace
dbo:RailwayStation
dbo:Region
dbo:ReligiousBuilding
dbo:River
dbo:Settlement
dbo:Station

[#] dbo: PREFIX is <http://dbpedia.org/ontology/>

Table 4: Types of places extracted with E-GSP-F. Place types are obtained from `rdf:types`.

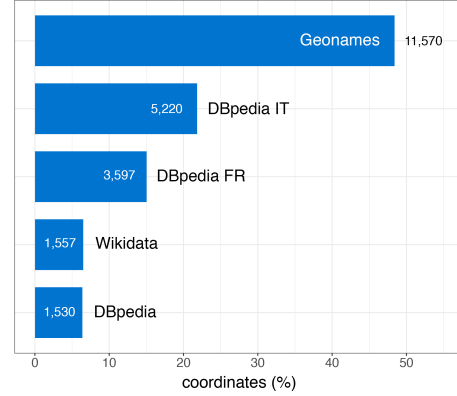


Fig.2: Number of geographic coordinates extracted from the different knowledge-bases by E-GSP-F.

which E-GSP-F extracted geographic coordinates. As shown in figure, the *expansion* step of E-GSP-F allowed to retrieve geographic information from multiple knowledge-bases, and indeed Geonames proved to be the richest source of geographic information, in our experiments.

6 Conclusions

We presented the novel *Geo-Semantic-Parsing* (GSP) technique for automatically associating geographic coordinates to text documents. Furthermore, we improved the basic GSP approach by introducing the *Expanded GSP* (E-GSP) and *GSP with Filtering* (GSP-F), which we also combined together in the so-called E-GSP-F. The excellent results obtained by E-GSP-F on 2 real-world evaluation datasets (best $F1 = 0.91$) demonstrated that previous state-of-the-art approaches can be outperformed by leveraging powerful machine learning algorithms on top of Linked Data. In particular, semantic annotation proved to be very effective in overcoming the drawbacks related to the language polysemy and the limited amount of context, being able to brilliantly perform disambiguation and enrichment tasks by leveraging on the Linked Data content and structural properties. Our results also showed that our technique is capable of extracting structured geographic information with variable degrees of granularity, ranging from country-level to building-level.

Future works along this direction should be focused on further improving the *expansion* step in E-GSP. Indeed, current results of E-GSP-F are mainly constrained by the moderate *Recall*. This calls for additional efforts aimed at increasing the set of geographic information extracted from RDF resources.

Acknowledgements. This research is supported in part by the EU H2020 Program under the schemes INFRAIA-1-2014-2015: **Research Infrastructures** grant agreement #654024 *SoBigData: Social Mining & Big Data Ecosystem*.

References

1. M. Avvenuti, S. Cresci, F. Del Vigna, and M. Tesconi. Impromptu crisis mapping to prioritize emergency response. *Computer*, 49(5):28–37, 2016.
2. M. Avvenuti, S. Cresci, A. Marchetti, C. Meletti, and M. Tesconi. Predictability or early warning: using social media in modern emergency response. *IEEE Internet Computing*, 20(6):4–6, 2016.
3. M. Avvenuti, F. Del Vigna, S. Cresci, A. Marchetti, and M. Tesconi. Pulling information from social media in the aftermath of unpredictable disasters. In *ICT-DM’15*. IEEE.
4. Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: a content-based approach to geo-locating Twitter users. In *CIKM’10*. ACM.
5. S. Cresci, A. D’Errico, D. Gazzé, A. Lo Duca, A. Marchetti, and M. Tesconi. Towards a DBpedia of tourism: the case of Tourpedia. In *ISWC’14 Demos*. Springer.
6. F. Dell’Orletta, G. Venturi, A. Cimino, and S. Montemagni. T2K²: a system for automatically extracting and organizing knowledge from texts. In *LREC’14*, 2014.
7. L. Ding, J. Shinavier, Z. Shangguan, and D. L. McGuinness. SameAs networks and beyond: analyzing deployment status and implications of owl:sameAs in linked data. In *ISWC’14*. Springer.
8. M. Dredze, M. Osborne, and P. Kambadur. Geolocation for twitter: Timing matters. In *HLT-NAACL’16*. ACL.
9. M. Dredze, M. J. Paul, S. Bergsma, and H. Tran. Carmen: A twitter geolocation system with applications to public health. In *AAAI’13 Workshops*. AAAI.
10. P. Ferragina and U. Scaiella. TagMe: on-the-fly annotation of short text fragments (by Wikipedia entities). In *CIKM’10*. ACM.
11. J. Gelernter and S. Balaji. An algorithm for local geoparsing of microtext. *GeoInformatica*, 17(4):635–667, 2013.
12. T. Gottron, J. Schmitz, and S. Middleton. Focused exploration of geospatial context on linked open data. In *ISWC’14 Workshops*. Springer.
13. A. Halterman. Mordecai: Full text geoparsing and event geocoding. *The Journal of Open Source Software*, 2(9), 2017.
14. G. Kordopatis-Zilos, S. Papadopoulos, and I. Kompatsiaris. Geotagging text content with language models and feature mining. *Proceedings of the IEEE*, 105(10):1971–1986, 2017.
15. J. McGee, J. Caverlee, and Z. Cheng. Location prediction in social media based on tie strength. In *CIKM’13*. ACM.
16. P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. DBpedia spotlight: shedding light on the web of documents. In *I-Semantics’11*. ACM.
17. S. E. Middleton, L. Middleton, and S. Modafferi. Real-time crisis mapping of natural disasters using social media. *IEEE Intelligent Systems*, 29(2):9–17, 2014.
18. H. Paulheim and J. Fürnkranz. Unsupervised feature generation from linked open data. In *WIMS’12*. ACM.
19. L. Rietveld, R. Hoekstra, S. Schlobach, and C. Guéret. Structural properties as proxy for semantic relevance in RDF graph sampling. In *ISWC’14*. Springer.
20. G. Töpper, M. Knuth, and H. Sack. DBpedia ontology enrichment for inconsistency detection. In *I-Semantics’12*. ACM.
21. S. Trani, D. Ceccarelli, C. Lucchese, S. Orlando, and R. Perego. Dexter 2.0: an open source tool for semantically enriching data. In *ISWC’14 Demos*. Springer.
22. R. Usbeck, A.-C. N. Ngomo, M. Röder, D. Gerber, S. A. Coelho, S. Auer, and A. Both. AGDISTIS – graph-based disambiguation of named entities using linked data. In *ISWC’14*. Springer.