Decision Support Systems 136 (2020) 113346

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Geo-semantic-parsing: AI-powered geoparsing by traversing semantic knowledge graphs

Leonardo Nizzoli[a,b], Marco Avvenuti[a], Maurizio Tesconi[b], Stefano Cresci[b,*]

[a] Dept. of Information Engineering, University of Pisa, Italy
[b] Institute of Informatics and Telematics, National Research Council (IIT-CNR), Italy

ARTICLE INFO

Keywords:
Geoparsing
Geotagging
Artificial intelligence
Knowledge graphs
Twitter

ABSTRACT

Online social networks convey rich information about geospatial facets of reality. However in most cases, geographic information is not explicit and structured, thus preventing its exploitation in real-time applications. We address this limitation by introducing a novel geoparsing and geotagging technique called Geo-Semantic-Parsing (GSP). GSP identifies location references in free text and extracts the corresponding geographic coordinates. To reach this goal, we employ a semantic annotator to identify relevant portions of the input text and to link them to the corresponding entity in a knowledge graph. Then, we devise and experiment with several efficient strategies for traversing the knowledge graph, thus expanding the available set of information for the geoparsing task. Finally, we exploit all available information for learning a regression model that selects the best entity with which to geotag the input text. We evaluate GSP on a well-known reference dataset including almost 10 *k* event-related tweets, achieving $F1 = 0.66$. We extensively compare our results with those of 2 baselines and 3 state-of-the-art geoparsing techniques, achieving the best performance. On the same dataset, competitors obtain $F1 \leq 0.55$. We conclude by providing in-depth analyses of our results, showing that the overall superior performance of GSP is mainly due to a large improvement in recall, with respect to existing techniques.

1. Introduction

Online Social Networks (OSN) are privileged observation channels for understanding the geospatial facets of many real-world phenomena [1]. Unfortunately, in most cases OSN content lacks explicit and structured geographic information, as in the case of Twitter, where only a minimal fraction (1% to 4%) of messages are natively geotagged [2]. This shortage of explicit geographic information drastically limits the exploitation of OSN data in geospatial Decision Support Systems (DSS) [3]. Conversely, the prompt availability of geotagged content would empower existing systems and would open up the possibility to develop new and better geospatial services and applications [4,5]. As a practical example of this kind, several social media-based systems have been proposed in recent years for mapping and visualizing situational information in the aftermath of mass disasters – a task dubbed as *crisis mapping* – in an effort to augment emergency response [6,7]. These systems, however, demand geotagged data to be placed on crisis maps, which in turn imposes to perform the geoparsing task on the majority of social media content. Explicit geographic information is not only needed in early warning [8,9] and emergency response systems [10–14], but also in systems and applications for improving event promotion [15,16], touristic planning [17–19], healthcare accessibility [20], news aggregation [21] and verification [22]. In addition, also other important tasks such as the monitoring of epidemics [23] and crime prevention [24–26] would benefit from the availability of additional geotagged OSN content, let alone those situations in which geographic information is relevant per se, such as in demographic studies [27].

Given the great importance of geotagged data for DSS, much effort has been recently devoted to tasks such as geotagging and geoparsing [28,29]. In detail, *geotagging* is defined as the generic task of associating geographic coordinates to a given document or to a portion of a document (e.g., a token). Instead, *geoparsing* is a more complex task that can be used to perform geotagging and that involves parsing a text, identifying location mentions and complementing them with their corresponding geographic coordinates [30]. There exists also other approaches to geotagging that are not necessarily based on free text analysis, such as those based on OSN account information [31] or on social relationships [32].

In this work, we focus on the geoparsing task, and we propose a novel technique called Geo-Semantic-Parsing (GSP). GSP is able to achieve state-of-the-art results by adopting machine learning and

* Corresponding author.
E-mail addresses: l.nizzoli@iit.cnr.it (L. Nizzoli), marco.avvenuti@unipi.it (M. Avvenuti), m.tesconi@iit.cnr.it (M. Tesconi), s.cresci@iit.cnr.it (S. Cresci).

https://doi.org/10.1016/j.dss.2020.113346
Received 20 March 2020; Received in revised form 19 June 2020; Accepted 22 June 2020
Available online 07 July 2020

artificial intelligence (AI) techniques to extract geographic information from the rich data contained in semantic knowledge graphs, such as *DBpedia* and *GeoNames*. In particular, in a first step GSP leverages a semantic annotator to identify relevant portions of the input text (i.e., the document to geoparse) and to link them to pertinent entities in a reference knowledge graph. Then, it exploits several different strategies to traverse the knowledge graph by navigating links between entities. The result of this second step is an expanded set of candidate entities, that are likely to contain relevant geographic information for the task. Finally, among the geographic information of all retrieved entities, we select those with which to geotag the input document by means of a regression model, that we trained on labeled data. The combination of powerful AI techniques and the rich, structured, interconnected data contained in multiple knowledge graphs allows GSP to achieve $F1 = 0.66$, whereas other state-of-the-art techniques and baselines obtain $F1 \leq 0.55$.

More in detail, one of the reasons why our solution achieves unprecedented results is because it mitigates the problem of *toponymic polysemy* – that is, the fact that the same toponym can refer to different places according to the context in which it appears.[1] The majority of traditional geoparsing techniques adopt heuristics to disambiguate toponyms matched in a gazetteer, a solution that might prove ineffective, especially at world-scale. As a consequence, the application of such techniques is often constrained to geographically-limited areas, in order to achieve satisfactory performance [33]. Instead, GSP mitigates this issue by performing semantic annotation of the input text – an operation that intrinsically performs disambiguation of tokens based on their context. In addition, our experimental results demonstrate that the expansion and selection steps of GSP also allow to correct some of the errors made by the semantic annotator. A second reason for the increased performance of GSP is related to the simplicity of previous approaches to this task. In fact, traditional approaches simply match geographic named entities found in the input text with entries in a gazetteer. Contrarily, our solution is based on powerful techniques (e.g., semantic annotation, regression via gradient boosting decision trees, word and graph embeddings) and information-rich semantic knowledge graphs. Finally, GSP also has a number of additional advantages over previous techniques: it does not require any explicit geographic information (e.g., GPS coordinates, timezones), contrarily to [34]; it only exploits text data of input documents (e.g., it does not require any user information or social network topology), contrarily to [13,32]; 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 on a given topic), contrarily to [35,36]; it does not require users to specify a target geographic region, but instead it geoparses places all over the world, contrarily to [33]; by leveraging knowledge graphs, GSP is capable of extracting fine-grained, structured geographic information (e.g., at the level of buildings, cities, counties and regions, countries) similarly to [34,37].

Contributions of this work. Our main contributions can be summarized as follows:

- we propose a novel geoparsing technique (GSP), capable of significantly improving state-of-the-art performance at this task;
- to reach our goal, we propose and experiment with several different expansion strategies to efficiently traverse a knowledge graph, thus expanding the set of entities to scan for geographic information;
- we learn a regression model to assign a confidence score to all retrieved entities, in order to select only those providing pertinent geographic information;
- we experimentally demonstrate the practical advantage of the design choices on which GSP is rooted. The main improvement brought by GSP is a large boost to the *recall* metric, which we attribute to the optimized expansion strategies previously introduced.

Adding to the previous scientific contributions, our solution is also based on state-of-the-art technologies and implementations, for all the necessary steps. In particular, we leverage TagMe [38] for semantic annotation and entity linking, and Microsoft's gradient boosting framework LightGBM [39] for learning our regression model, which are currently considered the state-of-the-art for the respective tasks. We also design and experiment with a large number of regressors, some obtained via a process of feature engineering resulting from textual analyses of our documents with FLAIR [40], while others directly learned from our data via the use of BERT contextual word embeddings [41] and *rdf2vec* graph node embeddings [42].

Roadmap. The remainder of this paper is organized as follows. In Section 2, we survey existing works for geoparsing and geotagging of social media content. In Section 3, we provide background information, and we introduce the GSP technique. In Sections 4 and 5, we respectively delve into the details of the *expansion* and *selection* steps of GSP, also providing experimental results to support our choices. Then, in Section 6 we describe our dataset, and we report experimental results of GSP and other techniques for the geoparsing task. We conclude with Section 7 discussing our results, and with Section 8 summarizing our work and highlighting directions for future research and experimentation.

2. Related Wworks

A recent survey on location prediction on Twitter [29] proposed a taxonomy of geotagging and geoparsing techniques according to different tasks. These can be: the prediction of (i) the locations mentioned in tweets, (ii) the tweet origin location, or (iii) the user home location. The remainder of this section adopts the same structure, with a particular focus on works dealing with the prediction of mentioned locations, since our contribution also falls in this category.

2.1. Mentioned location prediction

The goal of this task is to identify locations mentioned in a text and link them to the corresponding geographic coordinates. It has been investigated for a long time on well formatted documents – like news articles, and researchers identified entity mention variability and ambiguity as the two main challenges of this task. However, mentioned location prediction is even more challenging in OSNs, due to the noisy and short user-generated posts.

The most similar work to our present contribution is our previous attempt at this task, where we proposed a preliminary version of the GSP technique [2]. Similarly to this work, the core idea of [2] is to first exploit a semantic annotator to identify relevant portions of the input text, and then to parse the corresponding semantic resources in search for possible geographic information. In our previous work, however, we only leveraged a very limited number of semantic resources, actually disregarding many nodes of the available knowledge graphs that could bring useful information for the geoparsing task, as outlined in Fig. 2. Moreover, the selection of the geographic information to geoparse the input document was carried out by means of a binary classifier, based on a Support Vector Machine (SVM). The AI-driven exploration of the semantic knowledge graphs, together with the accurate selection of informative graph nodes via gradient-boosted regression, thus represent the main differences between our previous and current works. In turn, the profitable exploitation of this additional information significantly improves geoparsing performance, as demonstrated by our experimental results, reporting $F1 = 0.66$ vs $F1 = 0.55$ of our previous work.

Apart from our previous contribution, the task of mentioned location prediction was traditionally tackled in two steps: (i) mentioned location recognition, and (ii) their subsequent disambiguation. Mentioned location recognition is generally considered as a special type

[1] As an example, at the time of writing *GeoNames* returns 5331 records for "Rome", distributed across all six continents: https://www.geonames.org/search.html?q=rome.

of Named Entity Recognition (NER) task, and treated accordingly [43]. A few recent works also proposed other specific techniques for identifying location names in texts, outperforming traditional approaches based on NER. As an example, LNEx [44] learns a statistical language model by applying a skip-gram model to token sequences extracted from gazetteers, which contributes to the accurate and rapid detection of location mentions in texts. The work discussed in [12] applies a classification approach based on word embeddings and on a convolutional neural network for detecting location mentions. The focus of [45] is instead posed on the exploitation of spatial relations for location estimation. Authors first employ information extraction methods to identify toponyms and spatial relations in a text. Then, they use expectation maximization to learn models based on spatial probability density functions, and they use these models to infer the location of unknown objects.

Location disambiguation is usually performed by matching the detected location NEs with the entries of a geographic gazetteer (e.g., *GeoNames,*[2] *OpenStreetMap*[3]). Gazetteers also contain the association between toponyms and their geographic coordinates, thus making it trivial to return the geographic coordinates. An example of this type of approach is *geoparsepy,*[4] which combines NER techniques for the detection of text chunks, containing location mentions, with heuristics to disambiguate and match the detected NEs with *OpenStreetMap* entries [33,46]. A similar approach was proposed by Halterman with the *mordecai*[5] system [37]. The main difference between *geoparsepy* and *mordecai* is that the latter leverages a deep learning neural network classifier to select best candidate coordinates from the matching entries of the *GeoNames* gazetteer. Given that both *geoparsepy* and *mordecai* currently represent state-of-the-art and widely-used geoparsing systems, in our evaluation section we compare our geoparsing results with those benchmarks.

The aforementioned works, as well as our present contribution, propose general-purpose geoparsing techniques that are suitable for application to any standalone textual document (e.g., news articles, tweets, emails, etc.). Other geoparsing systems are instead specifically developed for social media, and they exploit some of the features available on specific social networking platforms. For example, the systems discussed in [35,36] leverage a whole user's timeline to collectively disambiguate toponyms, while [32] exploits user friendship networks to improve geoparsing accuracy. Similarly, TAGGS [13] enhances location disambiguation in Twitter by employing tweet and user metadata together with other contextual spatial information extracted from additional tweets.

2.2. Post origin location prediction

This task targets the prediction of the location from which a post is shared, given its textual content and possible additional information. The simplest approaches to this task scan the textual content of the post and user's profile information in search for geographic clues, and they combine the retrieved geographic information to infer the post origin location. For example, [47] applies scoring and ranking algorithms to data extracted from textual content, users' profile location and place labels for predicting locations at the finest possible granularity. Instead, [48] significantly extends the possible sources of geographic information, adding user's network, external resources and knowledge-bases, and posts related to similar topics. The obtained results are combined to provide the final prediction, following an unsupervised approach.

However, most of the approaches tackle the more challenging scenario of predicting a post origin location in the absence of explicit geographic information. For example, [49] employs several off-the-shelf machine learning classification algorithms (e.g., SVMs, Random Forests, etc.) fed with eight tweet- and metadata-derived features to classify global tweets at the country level. More sophisticated approaches look for content similarities between geotagged and non-geotagged posts. The system proposed in [50] estimates the location from which a post was generated by exploiting the similarities in the content between the post and a set of geotagged tweets, as well as their time-evolution characteristics. Similarly, [51] leverages a ranking model to relate a non-geotagged tweet to the most similar geotagged ones, based on the content. Then, it predicts the non-geotagged tweet location by combining the locations of the geotagged tweets, using a weighted majority voting algorithm. Contrarily to [49], both [50,51] are capable of providing accurate, fine-grained (i.e., within a city) location estimates. However, their application is typically restricted to geographically-limited areas, whereas [49] can be conveniently applied to classify world tweets. Finally, [52] tackles post origin location prediction under the interesting perspective of user privacy. Authors present a deep learning approach to violate user geolocation privacy on Twitter, by predicting her last post origin. To do so, the model leverages previous geotag leakages related to the user itself, and her network neighbors. Notably, the model proves able to violate the privacy of the 60% of the analyzed users. However, authors also propose defensive strategies, based on data perturbation techniques, to reduce prediction accuracy.

2.3. User home location prediction

This task aims to identify users home locations – a goal that is typically achieved by leveraging a portion of the user's posting history. As a striking example of this kind, in [14] user home locations are predicted based on historical locations of the same users, extracted using a Markov model. Similarly to [49], also the technique recently proposed in [31] focuses on country-level predictions. The latter method is based on a comparison of frequent word distributions from user timelines with country-based lists of popular Web searches from Google Trends. Given a user, [31] computes a ranked list of possible home countries, weighted by means of a confidence score obtained via statistical and machine learning methods.

The previously described approaches are solely based on a user's posting history (i.e., its timeline). Instead, another large body of work also leveraged the correlation between strong connectivity patterns in the social graph and geographic proximity in the real world [53]. Authors of [54] define as “locatable” those users that present geographic information allowing their geolocation. Then, given a non-locatable user, their proposed technique iteratively considers reciprocal (i.e., bidirectional) social relationships to infer the user geolocation from its locatable network neighbors. This simple and preliminary approach is similar to those described in [47,48] for predicting post origin locations, in that it only combines readily and explicitly available geographic information. A similar approach is proposed in [53] for efficiently detecting users belonging to a given city. The technique exploits both textual tweet content and Twitter social graph. Instead, a more powerful technique is proposed in [55], leveraging a probabilistic approach that jointly models geographic labels and Twitter texts of users, organized in the form of a graph representing the friendship network. In detail, authors use a Markov random field probability model to represent the network, and they ground the learning step on a Markov Chain Monte Carlo simulation, that approximates the posterior probability distribution of the missing geographic user labels. Finally, [56] presents an integrated geolocation prediction framework and uses it to investigate what factors impact the prediction accuracy. Authors evaluate a range of feature selection methods to obtain “location indicative words”, and they investigate the impact of non-geotagged tweets, language and user-declared metadata on user location prediction. For additional references and in-depth discussions of other user home

[2] https://www.geonames.org/
[3] https://www.openstreetmap.org/
[4] https://pypi.org/project/geoparsepy/
[5] https://github.com/openeventdata/mordecai

location methods, we point interested readers to the survey by Ajao et al. [28], and references therein.

3. Geoparsing documents with GSP

In this section, we first formally define the geoparsing task and how geoparsing techniques are evaluated. Then, we provide the conceptual overview of the GSP technique and the rationale for our design choices.

3.1. Problem definition

Geoparsing involves analyzing a textual document, identifying mentions of known locations, and associating the corresponding geographic coordinates to each mentioned location. Given this formulation, a geoparsing technique is defined as a model $\mathscr{GP}$, such that $\mathscr{GP}(t_i) = \mathbf{p}_i$, where t_i is the i-th document in a collection, and $\mathbf{p}_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,N}\}$ is a set of predicted geographic coordinates $p_{i,k}$, each corresponding to a toponym detected by $\mathscr{GP}$ in t_i. Ideally, we would want the set of predicted coordinates $\mathbf{p}_i$ to be equal to the set of ground truth coordinates $\mathbf{g}_i$ for t_i:

$$\mathbf{p}_i \equiv \mathbf{g}_i \quad \Leftrightarrow \quad p_{i,k} = g_{i,j} \quad \forall \quad k, j = 1, \ldots, N$$

Hence, when evaluating the performance of a geoparsing technique, each $p_{i,k} = g_{i,j}$ is considered as a *true positive* prediction. In practice, two coordinates are considered to be equal if their geographic distance is lower than a certain threshold $\mathscr{T}$. For geoparsing tasks, the most common choice is $\mathscr{T} = 100$ miles ($\simeq$161 km) [29]. However, in our work we adopt a more severe $\mathscr{T} = 50$ km ($\simeq$31 miles), which is more suitable for practical applications. In addition to correct predictions, geoparsing techniques can also predict coordinates that do not correspond to any ground truth coordinate: $p_{i,k} \notin \mathbf{g}_i$, thus yielding a type I error (or *false positive*). Similarly, they can fail to predict a ground truth coordinate: $g_{i,j} \notin \mathbf{p}_i$, thus yielding a type II error (or *false negative*). Finally, similarly to information retrieval and entity linking tasks, *true negatives* are typically not considered for the evaluation of geoparsing techniques [29].

Given the above definitions, we can count true positives (*TP*), false positives (*FP*), and false negatives (*FN*), summarizing the results of the application of a geoparsing technique to a collection of documents:

$$TP = \sum_i |\mathbf{p}_i \cap \mathbf{g}_i|$$

$$FP = \sum_i |\mathbf{p}_i \backslash \mathbf{g}_i| = \sum_i |\{x \mid x \in \mathbf{p}_i \text{ and } x \notin \mathbf{g}_i\}|$$

$$FN = \sum_i |\mathbf{g}_i \backslash \mathbf{p}_i| = \sum_i |\{x \mid x \in \mathbf{g}_i \text{ and } x \notin \mathbf{p}_i\}|$$

where $\mathbf{a}\backslash\mathbf{b}$ is the set difference between $\mathbf{a}$ and $\mathbf{b}$, and $|\mathbf{a}|$ is the cardinality of set $\mathbf{a}$. In the remainder of this work, geoparsing results are assessed by means of standard evaluation metrics based on *TP*, *FP* and *FN*, such as *precision*, *recall*, and *F1*-score (*F1*).

3.2. Overview of GSP

In Fig. 1, we provide a schema of the GSP system. As introduced in Section 1, the main idea behind GSP is to leverage the rich, structured and linked information exposed by a knowledge graph to identify, disambiguate and geotag mentioned locations. To do so, GSP processes a single document t_i at a time, through three sequential steps:

- **step 1:** *semantic annotation* (Fig. 1a), which identifies a relevant token (*anchor*) in the input text t_i (yellow-colored) and links it to a pertinent entity (red-colored) in a knowledge graph. The purpose of this first step is to augment the input text with the information exposed by the knowledge graph;
- **step 2:** *expansion* (Fig. 1b), which traverses the information-rich, structured knowledge graph, retrieving entities (blue-colored) related to the starting one and likely to convey further useful geographic information. The purpose of this intermediate step is to take full advantage of the knowledge graph structure for enriching the available information, thus potentially increasing the model *recall*;
- **step 3:** *selection* (Fig. 1c), which analyses the entities retrieved by the expansion step to pick the best candidate (green-colored) for geotagging the anchor. In particular, GSP parses the selected entity to extract the geographic coordinates, returned as the final result of the geoparsing process. The purpose of this final step is to deal with the information overload, introduced by the expansion step, thus improving the model *precision*.

In the remainder of this Section, we describe and motivate the three steps of GSP. Furthermore, our core scientific contributions – that is, the *expansion* and *selection* steps – are also thoroughly discussed and evaluated in Sections 4 and 5. For simplicity, throughout this work we use the terms *knowledge graph* and *knowledge-base* interchangeably, since we always leverage the Linked Data representation of all mentioned knowledge-bases. Similarly, we use the terms *entity* and *resource* interchangeably, when referring to a node of a knowledge graph.

3.2.1. Step 1: Semantic annotation

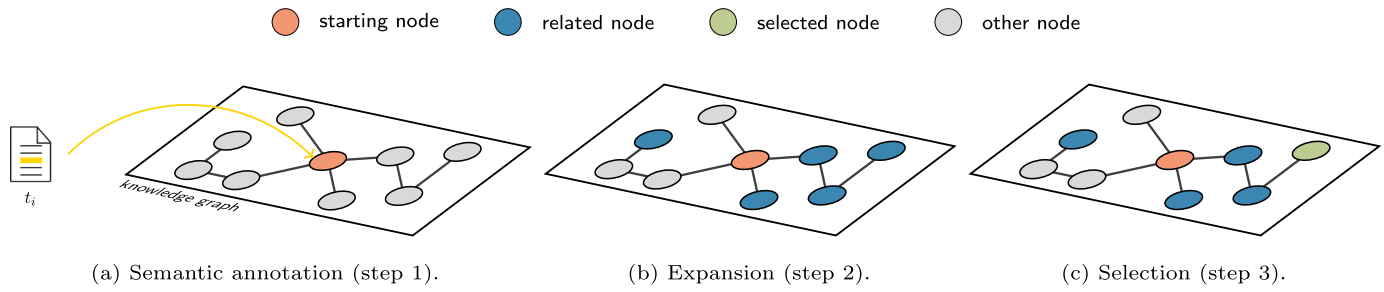
In GSP, we delegate the identification and disambiguation of location mentions to semantic annotation, which thus constitutes the starting point of our procedure (step 1 in Fig. 1). Semantic annotation is a long-studied task for augmenting documents, so that mentions of relevant entities in a text (e.g., persons, places, organizations) are linked to the corresponding entity in a reference knowledge-base [38]. This annotation process is highly informative, since it enables the exploitation of the rich information contained in the knowledge-base. By giving access to a wealth of structured and interconnected information, semantic annotation also effectively mitigates the drawbacks related to the sparsity of short social media texts. In addition, it also has the side effect of alleviating possible geoparsing mistakes caused by toponymic polysemy, since semantic annotators automatically carry out disambiguation and only return the most likely reference to a knowledge-base entity for every annotated token. Notably, this disambiguation operation is more accurate than those carried out in previous works, such as those based on simple heuristics as [33]. Semantic annotation is also more powerful than traditional NER for identifying relevant portions of a text, since it gives access to the information of a knowledge-base. Downstream of the annotation step, each relevant anchor of each input text t_i is linked to the most pertinent entity of a reference knowledge graph, providing the information needed to identify and geotag the mentioned locations. Those entities constitute the input of the subsequent expansion step.

Implementation notes. We perform semantic annotation with TagMe[6] – one of the most popular and best-performing off-the-shelf annotators currently available. In [2], we provided geoparsing results comparing the performance of 4 different semantic annotators, with TagMe achieving the best results. Moreover, TagMe is particularly suitable for our work, since it is specifically designed for short and poorly written texts, such as social media messages [38]. In order to have complete access to all its functionalities and to allow fast queries, we leverage a local deployment [57]. By default, TagMe annotates documents with *Wikipedia* entities. However, for each such entity we refer to its equivalent on *DBpedia*, in order to exploit Linked Data properties and relations. Thus, (the English) *DBpedia* is our reference knowledge graph.

3.2.2. Step 2: Expansion

As a result of the semantic annotation step, we have documents where relevant tokens are identified and linked to entities in a knowledge graph. By parsing the structured information associated with these entities, we would thus be able to retrieve the coordinates of geographic

[6] https://tagme.d4science.org/tagme/



(a) Semantic annotation (step 1). (b) Expansion (step 2). (c) Selection (step 3).

Fig. 1. Logical overview of the 3 main steps applied by GSP to the input document t_i. Semantic annotation (step 1) links a relevant token (anchor) to an entity (red-colored node) within a reference knowledge graph. Expansion (step 2) identifies related entities (blue-colored nodes) that possibly convey useful geographic information. Selection (step 3) picks the best entity (green-colored node) to geotag the anchor. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

entities, whenever available, thus solving the geoparsing task. However, this naïve approach would only exploit information of a single node in a graph, thus negating the advantage of knowledge graphs in the first place and ignoring many other potentially informative nodes. Instead, being able to combine information from multiple entities would allow to correct wrong or missing information on a node.[7] Furthermore, it would also allow to correct at least part of the errors resulting from the semantic annotation step.

Because of these reasons, both our earlier [2] and our present geoparsing technique perform an *expansion* step. Given a starting node (i.e., the one linked to a token in t_i by the semantic annotator), the goal of this second step is to find other nodes that are related to the starting one, within which to look for relevant information for the geoparsing task, as sketched in Fig. 1b. To perform expansion, in [2] we exploited relations of semantic equivalence (i.e., `owl:sameAs` links) between entities. These links connect representations of the same entity across different knowledge-bases. Thus, in order to extract geographic information about an entity, in [2] we also exploited information about all semantically-equivalent entities that are reachable by iteratively navigating equivalence links. As shown in Fig. 2a via the formalism of multilayer networks, this expansion unfolds in a visually *vertical* fashion. On the one hand, it allows to leverage information coming from more than one node. On the other hand, however, the total number of nodes reachable via this expansion strategy ($\approx 10^1$) is still underwhelming, when compared to the total number of nodes available in a knowledge graph ($\approx 10^6 - 10^7$). Moreover, it only exploits relations of semantic equivalence, disregarding all the other types of relations between entities.

In our new GSP technique, we greatly increase the number of nodes that we leverage for geoparsing. We reach this goal by devising and experimenting with several different expansion strategies that, given a starting node, are capable of retrieving a large (i.e., with potentially hundreds of entities) ordered vector of related – but not equivalent – nodes from within the same knowledge graph. The common idea to all these expansion strategies is to explore nodes *horizontally* rather than vertically (i.e., within a knowledge graph vs across knowledge graphs), as shown in Fig. 2b. This approach has the advantage of retrieving many more nodes (together with all their associated information) with respect to that of [2]. In turn, this boosts GSP's *recall* – that is, the capacity of extracting geographic coordinates for toponyms in the input document. However, retrieving too many, possibly unrelated nodes can impair *precision*. Because of this trade-off, it is important to evaluate different horizontal expansion strategies, as extensively investigated in Section 4.

Implementation notes. All our horizontal expansion strategies accept a configurable size parameter L, representing the number of nodes to retrieve. In fact, contrarily to the vertical expansion used in [2], horizontal expansions are "unconstrained" and can potentially return all nodes in a graph. Therefore, in the remainder we refer to $\texttt{strategy}|_L$, meaning a specific expansion strategy and the number L of nodes it returns. Moreover, all expansion strategies only return nodes with geographic information. Nodes that are not complemented with any geographic information are not considered during expansion, even if related or similar to the starting node, since they could not be used for geoparsing anyway. Finally, we constrain expansion strategies to depend solely on the starting node returned by the semantic annotator. In this way, the expansions can be pre-computed once and for all for the entire knowledge graph, thus avoiding to perform this demanding operation at runtime.

3.2.3. Step 3: Selection

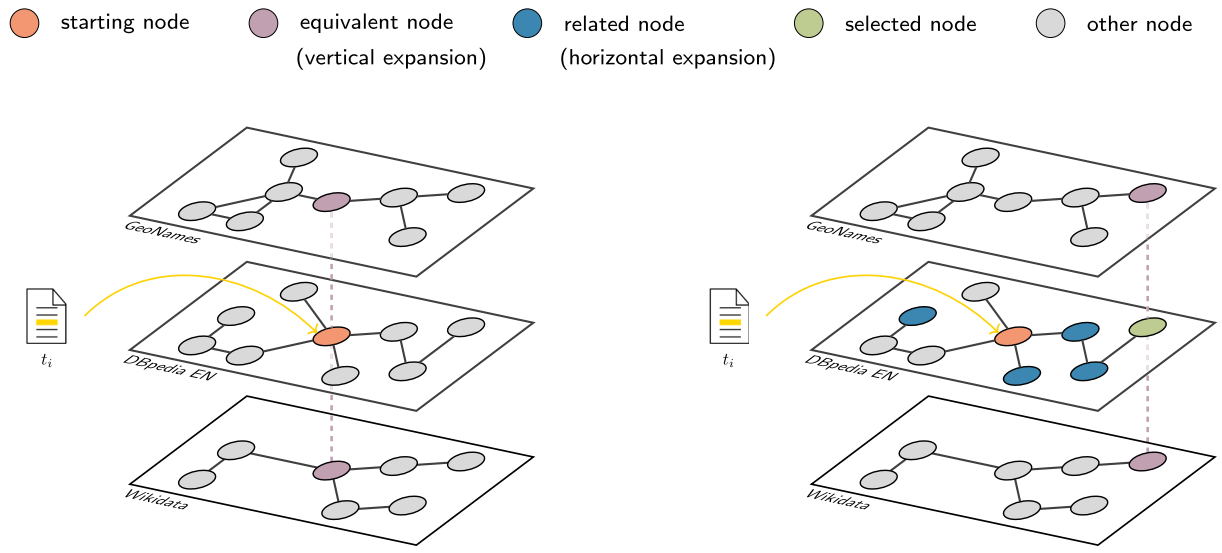
After the expansion step, we have access to a potentially large set of candidate nodes. Intuitively, the better is the expansion, the easier it is the selection of the node from which to extract geographic information. In fact, if the expansion only returned entities that are strictly geographically-related to the starting one, then any of those entities would provide pertinent geographic information for the geoparsing task, thus rendering the selection step trivial. However, in the majority of cases, the expansion step also provides some unrelated entities that should not be considered for geoparsing. Thus, the goal of this third step is to select the best node for geoparsing among all the candidates returned by the expansion step, as sketched in Fig. 1c.

In [2], results of the vertical expansion were filtered by a binary SVM classifier. This simple solution was successful because of the limited number of candidates yielded by the vertical expansion. In our present work however, the novel horizontal expansion potentially yields several orders of magnitude more candidates, thus making a binary classification task extremely unbalanced, hence impractical. For this reason, here we cast the selection problem as a regression task, where we aim to predict a confidence score for each candidate node. After assigning a confidence to each candidate returned by the expansion step, GSP simply selects the node with the highest confidence and geotags the input document with the geographic coordinates of that node. This step is conceptually similar to a filtering/pruning step – also adopted in many machine learning algorithms for improving the accuracy of predictions – where GSP selects the entity for which it is more confident. Notably, the efficacy of the *expansion&selection* approach has already been demonstrated in [2], with the selection/filtering step significantly boosting the model's *precision*. In Section 5 we further elaborate on this step, discussing how we frame the regression task, how we train our model, and which features we leverage for the regression.

3.2.4. Concluding steps

The core of the proposed GSP technique was outlined in the two

[7] We recall that Linked Data knowledge graphs are collaboratively curated. As with all user-generated content, mistakes and inconsistencies are indeed possible and should be accounted for.



(a) Vertical expansion (only considering semantically equivalent entities).

(b) Horizontal expansion (considering all related entities).

Fig. 2. Difference between vertical and horizontal expansion. Vertical expansion traverses semantically equivalent entities (purple-colored) across different knowledge graphs, whereas horizontal expansion considers those nodes that are most related (blue-colored) to the starting one, within the reference knowledge graph. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

previous sections. However, GSP performs two additional simple operations before outputting its predictions, described in the following for completeness. The vertical and horizontal expansion strategies are orthogonal with respect to one another, meaning that they give access to different nodes and, potentially, to different information. Being orthogonal, they can also be used simultaneously. Because of this, after selecting a node with the approach described in the previous section, GSP also applies the vertical expansion. In other words, our technique actually leverages both the horizontal and vertical expansions, as shown in Fig. 2.

The extraction of the geographic coordinates from a node in a knowledge graph, occurs by means of a *parsing* step. In this step, GSP scans all predicates of the semantic resource, looking for geographic information. In particular, in Linked Data there exist many different predicates designed to store geographic information (e.g., `geo:lat` and `geo:long`, `georss:point`, etc.). In GSP, we support as many as 45 geographic predicates. Moreover, since the geographic information contained in such predicates can be stored in different formats (e.g., decimal degrees; degrees, minutes, seconds), we then implemented a set of simple formulas for converting the different geographic formats into *decimal latitude and longitude* coordinates. As a result, the output of the *parsing* step and of the whole GSP technique is represented by a decimal latitude and longitude geographic coordinate (wherever available) that complements the input document.

4. Information expansion strategies

In this section, we describe the different strategies leveraged by GSP as part of its *expansion* step. Each strategy follows a different intuition, with the goal of retrieving the largest set of geographic entities related to the starting one. In the last part of this section we compare the effectiveness of the different strategies, both when applied individually and jointly. To better clarify each strategy and the differences between them, we make use of a fictitious toy experiment. Let us assume to have the following short text to geoparse:

"I'll spend a couple of hours in Bath, visiting its Roman heritage."

The ground truth for this text is represented by the geographic coordinates of the British town of *Bath*, renowned for its thermal baths dating back to the Roman Empire. However, let us assume for this example that the semantic annotator fails in linking the token *Bath* to the correct entity.[8] Instead, it erroneously links *Bath* to the entity bath[9] (in the sense of bathtub), which is the starting node in our toy example. In the following, we apply the different geographic expansion strategies with size $L = 2$ to the small toy knowledge graph repeated once per strategy in Fig. 3.

4.1. Spelling-based expansion

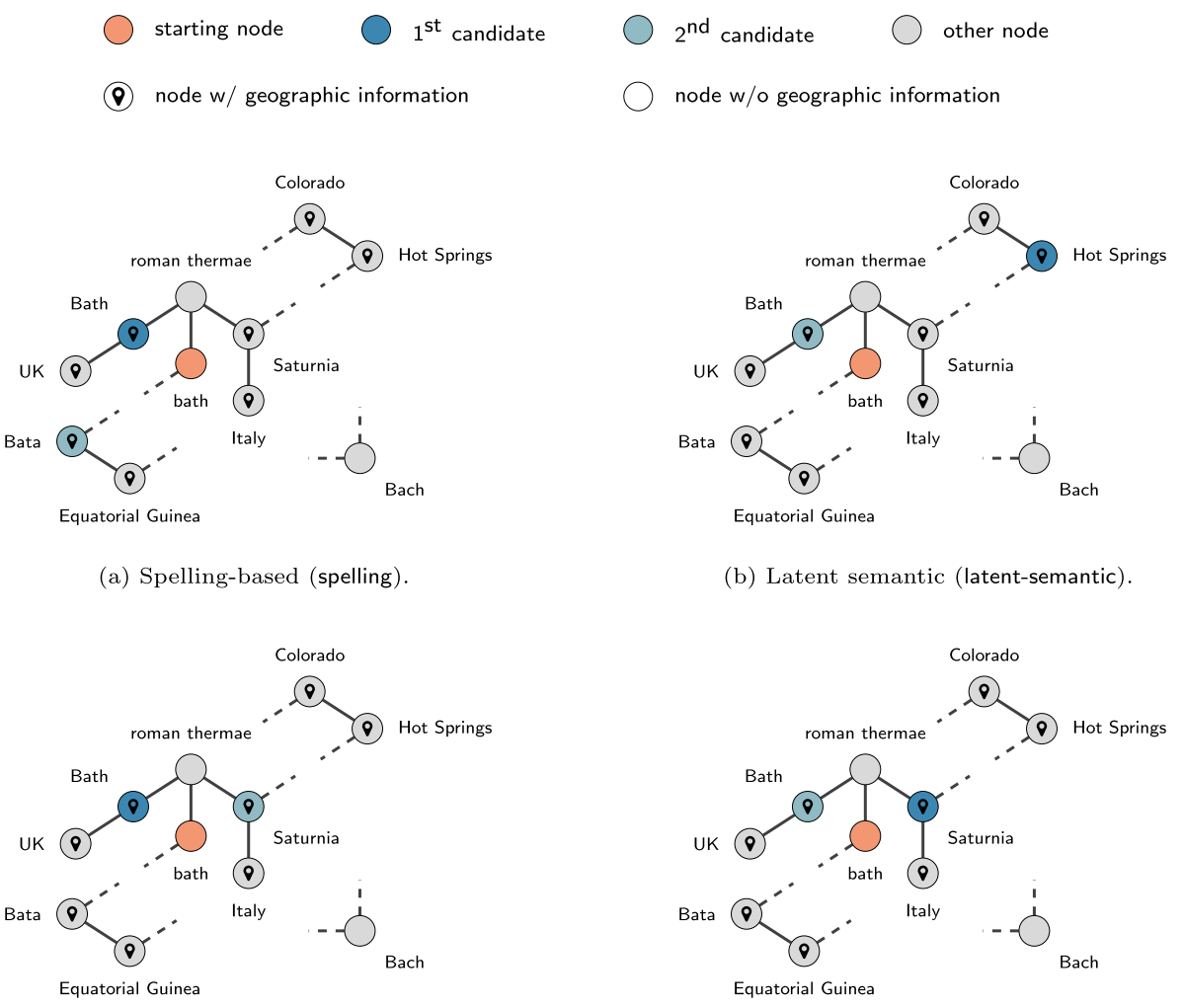
Although semantic annotators are specifically designed to disambiguate entities, polysemy, typos and jargon still pose a challenge. The majority of disambiguation errors occurs between entities with very similar names. A natural choice for expanding our set of nodes and for correcting possible errors, is therefore to consider entities whose names are similar to that of the starting node.

For any given starting node, the spelling-based expansion (henceforth `spelling`) retrieves and sorts the top-L geographic entities having closest names. The similarity between entities names is computed as the case-sensitive Levenshtein (edit) distance. Fig. 3a shows the results of `spelling` when applied to our toy experiment. After sorting the knowledge graph entities, `spelling` yields the sorted vector [Bath, Bach, Bata,...]. The entity `Bach`, corresponding to the famous musician, is not complemented with geographic information and so it is discarded. Finally, $\texttt{spelling}|_{L=2}$ returns the geographic entities corresponding to the British town Bath and to the Equatorial Guinea city `Bata`. By traversing the graph based on entities names, this strategy can retrieve nodes that are not directly linked nor topologically near to the starting one.

4.2. Latent semantic expansion

Node embeddings refer to a set of techniques for unsupervised feature extraction from large graphs, inspired by the usefulness that word and document embeddings recently demonstrated for many text

[8] http://dbpedia.org/page/Bath,_Somerset
[9] http://dbpedia.org/page/Bathtub



(a) Spelling-based (spelling).

(b) Latent semantic (latent-semantic).

(c) Topological, sorted by spelling similarity (topological-spe).

(d) Topological, sorted by latent semantic similarity (topological-lat).

Fig. 3. Toy example showing the nodes retrieved by the different expansion strategies on a small knowledge graph, for expansion size $L = 2$.

mining tasks [41]. In this representation, each node in a graph is described by a high-dimensional feature vector capable of encoding the latent structural information of the node within the graph. As such, nodes that play a similar role in the topology of a graph end up having similar node embeddings representations. Similarly to word embeddings, several different techniques have been proposed for computing the node embeddings of a graph. Among these, the *rdf2vec* technique [42] was recently proposed and specifically designed for semantic knowledge graphs, such as the ones leveraged by GSP. In particular, *rdf2vec* extends previous generic node embeddings techniques by also considering semantic node properties and the many different types of edges that represent the semantic relations between entities in knowledge graphs. As such, it is particularly suitable for graph mining tasks on knowledge graphs. In our experiments, we used the readily-available *rdf2vec* embeddings,[10] pre-trained for the English *DBpedia*.

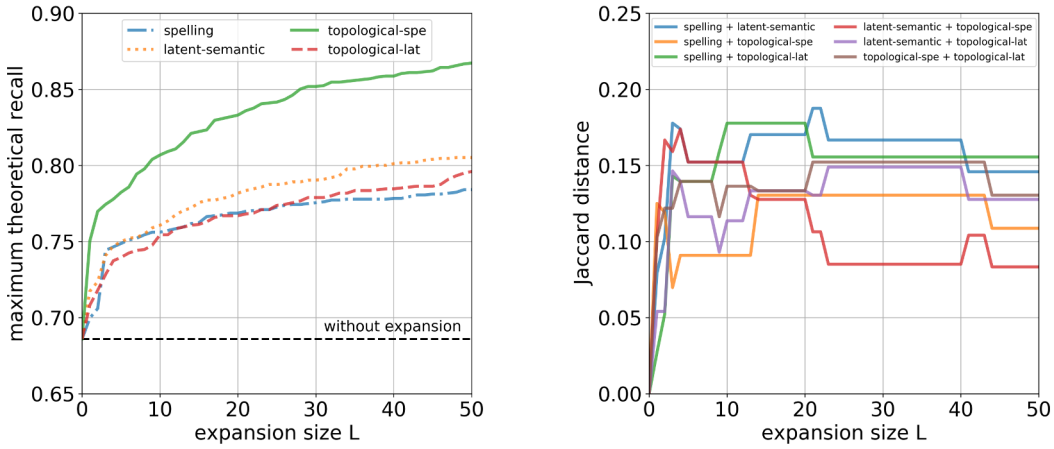
Latent semantic expansion (henceforth `latent-semantic`) retrieves and sorts the top-L nodes having largest cosine similarity between their *rdf2vec* representation and that of the starting node. In other words, this expansion strategy leverages powerful semantic node embeddings techniques to retrieve the nodes that are most similar to the starting one, in the latent semantic vector space. Similarly to the `spelling` expansion, also latent-semantic potentially retrieves nodes that are topologically far from the starting one. When applied to our toy example, we imagine $\texttt{latent-semantic}|_{L=2}$ to retrieve geographic entities that are semantically related to the concept of bathing (e.g., having hot springs), such as [Hot Springs, Bath,...], as shown in Fig. 3b.

4.3. Topological expansion

The previously introduced expansions do not explicitly consider the topology of the knowledge graph. However, when the semantic annotator fails to point to the correct starting node, or when the starting node is not complemented by geographic information, relevant geographic information is nonetheless likely to be found in a topologically near node, with respect to the starting one. In other words, we are confident that the annotator pointed us at least in the vicinity of the correct node. Thus, the following expansion strategies leverage this hypothesis and traverse existing links between nodes.

In practice, to implement topological expansion of size L, we retrieve the L nearest nodes with respect to the starting one. We begin by retrieving 1-hop geographic nearest neighbors to the starting node, then 2-hops geographic neighbors and so on, until we retrieve L nodes. Differently from previous strategies, all n-hop neighbors of the starting node share the same "similarity", which requires a mean to break ties

[10] http://data.dws.informatik.uni-mannheim.de/rdf2vec/



(a) Maximum theoretical recall produced by the different expansion strategies, when applied individually.

(b) Combined effectiveness of the expansion strategies, measured as the Jaccard distance d_J between the sets of retrieved entities.

Fig. 4. Performance evaluation of the proposed expansion strategies, when applied individually and jointly, as a function of the expansion size L.

since we want our expansion strategies to yield sorted vectors. To sort nodes that have the same distance with respect to the starting one, we leverage the similarity measures used in the `spelling` and `latent-semantic` strategies. In detail, we have the (i) `topological-spe` strategy, when the sorting criterion is based on spelling (i.e., the edit distance between entities names), and the (ii) `topological-lat` strategy, when the sorting criterion is based on latent semantic similarity (i.e., the cosine similarity between *rdf2vec* vectors). Interestingly, these two strategies can be seen as the combination of two orthogonal sorting criteria – namely, *topology* and either *spelling* or *latent semantic* similarity. Fig. 3c and d show the results of our topological expansions, when applied to the toy example.

4.4. Evaluation

In the remainder of this section, we evaluate the effectiveness of the different expansion strategies by measuring the extent to which they increase the number of geotagged toponyms (i.e, toponyms for which we retrieved geographic coordinates).

Experimental setup and evaluation metrics. As anticipated in Section 3.2.2, increasing geotagged toponyms contributes to raise geoparsing *recall*. Because of this, we introduce an ad hoc metric to evaluate the expansion strategies, called *maximum theoretical recall*, defined as the recall we would obtain by always selecting the correct geographic coordinates among all those retrieved by a given expansion strategy. In practice, the real recall also depends on the selection step, that might erroneously discard correct geographic information retrieved via an expansion strategy. We evaluate this facet in the next section, while here we only focus on measuring geographic information retrieved via expansion, that would not have been collected otherwise. The effectiveness of the various strategies is evaluated as a function of the expansion size L. By definition, the maximum theoretical recall is a monotonic non-decreasing function of L. Strategies are compared between one another, and with the baseline scenario where no expansion is performed. In Fig. 4, the latter scenario corresponds to expansions of size $L = 0$. Instead, we remind that expansions of size $L = 1$, despite returning only 1 entity as in the no-expansion scenario (where only the starting node is considered), produce improved results since expanding forces to select geographic entities, independently of L. The following results are obtained on the training split of our dataset, thoroughly described in Section 6.1.

Results. Fig. 4a shows the results of the different expansion strategies, when applied individually. Topological expansion with spelling-based sorting (`topological-spe`) largely outperforms all other strategies. With respect to the baseline scenario where no expansion is applied, `topological-spe` produces most of the recall gain within the first expansion steps: +9.4% at $L = 1$ and +12.0% at $L = 2$. This means that, assuming a flawless selection step, the sole application of $\texttt{topological-spe}|_{L=2}$ would boost the overall geoparsing recall by 12%. This important finding proves our hypotheses correct and motivates our experimentation on expansion strategies. Largely boosting recall at small expansion sizes is a desirable feature, since the complexity of the subsequent selection task dramatically increases with L. For $L \geq 3$ the incremental gain at each step is $< 1\%$. Nevertheless, `topological-spe` keeps retrieving relevant geographic information also for larger values of L, as demonstrated by its steadily rising curve in Fig. 4a, up to a recall gain of +26.3% at $L = 50$. All other expansion strategies achieve significantly worse results, demonstrating a lower capacity of retrieving relevant geographic information from their exploration of the knowledge graph. The `latent-semantic` strategy, based on *rdf2vec* node embeddings, achieves slightly better results with respect to the remaining strategies for $L > 10$. Notably, even the worst-performing strategy (i.e., `spelling`) yields a recall improvement of +2.0% at $L = 1$ and +2.9% at $L = 2$, further supporting the usefulness of this approach.

Complementarity of expansion strategies. The proposed strategies are diverse and largely orthogonal. As such, they potentially return very different sets of entities for the same input. We are thus interested in evaluating the extent to which two different strategies can complement each other by retrieving complementary information. In other words, two strategies could be individually weak, but when used simultaneously, they could nonetheless provide large amounts of relevant information. To evaluate this facet, we consider the output of all expansion strategies as sets (instead of ordered vectors) of entities. Then, for each combination of two different strategies, we compute the Jaccard distance d_J between the sets of retrieved entities. The higher is d_J, which is defined in the $[0,1]$ range, the more diverse are the entities retrieved by the two strategies. In turn, a large d_J would support their combined application. Fig. 4b shows results of this experiment, as a function of the expansion size L. As shown, no combination of strategies achieves a large Jaccard distance, with all combinations laying in the region of $d_J < 0.2$. The best results are achieved by `spelling` +`topological-lat` and by `spelling`+`latent-semantic`. However, the overall results of this experiment suggest that naïvely

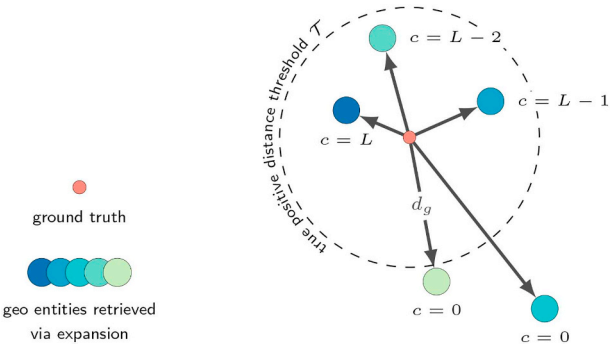


Fig. 5. Assignment of regression labels to candidates retrieved by an expansion strategy of size $L = 5$.

combining different strategies does not produce outright better results.

Given the results of the experiments with the different expansion strategies, in the remainder of our work we perform the expansion step of GSP by applying the topological expansion with spelling-based sorting (topological-spe), which achieved remarkable results also for low L values.

5. Best candidate selection

Given a set of L candidate geographic entities retrieved by an expansion strategy, the goal of the *selection* step is to choose the best candidate for geotagging the input document. Based on our problem definition, good candidates are those whose geographic distance d_g from the ground truth coordinates is $<\mathscr{T}$. In fact, all candidates that verify this assumption yield a *true positive* prediction at evaluation time. Among these, the best candidate is arguably the one with the lowest geographic distance from the ground truth.

5.1. Candidates labeling

As anticipated, we cast the selection problem as a regression task where we estimate a confidence score for each candidate, and we choose the candidate with the highest score. For any given candidate retrieved by an expansion strategy, its ground truth confidence score c should reflect its quality with respect to the ground truth coordinates. In this way, a regression model trained to estimate c scores would in fact produce a geographic ranking of the candidates. We leverage these observations by assigning a ground truth confidence score $c = 0$ to candidates whose distance from the ground truth coordinates is $d_g \geq \mathscr{T}$, as shown in the example of Fig. 5. Instead, all candidates whose $d_g < \mathscr{T}$ are assigned a positive score. In particular, the nearest candidate to the ground truth coordinates has $c = L$, the second nearest has $c = L - 1$, and so on.

5.2. Feature engineering

In the previous section we described how ground truth labels for the regression task are assigned. Now, we list and describe the features for our regression model (i.e., our regressors). Table 1 lists and briefly describes the features that we compute for each candidate. Each feature aims at measuring the relations between the candidate and (i) the starting node given by the semantic annotator, (ii) the token(s) of the input document that were linked to the starting node by the annotator (henceforth called the *anchor*), and (iii) the overall context (including the anchor's context in the input document and the nodes context in the knowledge graph). Notably, our features do not leverage any geographic information, implying that we aim to estimate the geographic quality of candidates based on other characteristics (i.e., their relations with the starting node, the anchor and their context). In the following and in Table 1, we group features according to the type of information they convey.

Annotation & expansion features (A&E). These features are based on the confidence with which the semantic annotator linked the anchor to the starting node in the knowledge graph, and on the distance traversed by the expansion strategy to retrieve the candidate. A large annotation confidence usually implies that the starting node already represents a good candidate for geoparsing. Conversely, a poor confidence is a proxy for disambiguation errors. In that case, considering other entities may be advantageous. Similarly, the further the expansion moves away from the starting node, the less the retrieved candidate is semantically related to it. As a consequence, large distances may discriminate non pertinent resources.

Spelling features (SPE). Spelling features are designed to capture the spelling characteristics of the anchors and of the entity names, both separately and compared. The rationale behind this class of features is that it is unlikely that the name of a good candidate is completely unrelated to the anchor. At the same time, a high similarity between the candidate and the starting node reflects a possible disambiguation error of the annotator. Finally, location mentions often start with uppercase letters. Hence, we consider their presence in the anchor as possible proxies of location mentions. All these features can be computed solely from the anchor string and the candidate URL, both of which are returned by the semantic annotator.

DBpedia features (DBP). This class of features considers the content and the semantic characteristics of the candidate in the reference knowledge graph (i.e., the English *DBpedia*). In particular, we leverage the *DBpedia* ontology to understand the resource type of the candidate (e.g., place, event, activity, agent), with a specific focus on places. Moreover, we measure the centrality of the corresponding node within the graph. Finally, we estimate how much its descriptive abstracts are related to the anchor in the input document.

Syntactic features (SYN). We leverage natural language processing (NLP) techniques to carry out a syntactic analysis of the input text, assigning tags to the anchors according to their syntactic role (e.g., nouns, verbs, etc.). In fact, valid location anchors are more likely to correspond to specific syntactic tags such as nouns, with respect to other tags (e.g., verbs, conjunctions). In detail, we perform two types of syntactic analysis: (i) part-of-speech (POS) tagging and (ii) text chunking. POS tagging analyses the text word by word, considering a fine-grained set of 50 possible tags. Instead, text chunking identifies and tags syntactically correlated groups of words, with a coarser-grained set of 10 possible tags. We obtain POS and chunking tags, together with their respective confidence scores, by adopting the state-of-the-art NLP framework FLAIR[11] [40].

Named Entity features (NER). Named Entity Recognition (NER) is a task that aims to locate NE mentions in unstructured text and to tag them with predefined categories (e.g., persons, organizations, locations). Similarly to syntactic features, we follow the idea that NER tags corresponding to the category *location* are good proxies for valid anchors, whereas anchors tagged differently should be discarded for geoparsing tasks. Again, we leverage FLAIR to perform the NER task on our input texts, obtaining NER tags with the respective confidence. Notably, many state-of-the-art geoparsing techniques are based on NER to identify toponyms in texts [33,37,46].

Latent features (LAT). The last group of features leverages state-of-the-art embeddings techniques designed for texts and graphs, in order to estimate the latent similarity between the candidate and, respectively, the anchor and the starting node. In Section 4.2 we introduced *rdf2vec* node embeddings [42] as a profitable mean to retrieve and sort candidate entities for expansion. Here, we leverage the same idea for estimating the similarity between the candidate and the starting node. To do so, we simply compute the cosine similarity between the

[11] https://github.com/flairNLP/flair

respective *rdf2vec* vectors. Then, in order to assess the semantic similarity between the candidate and the anchor, we leverage contextual word embeddings, and in particular their state-of-the-art implementation BERT[12] [41]. Word embeddings are dense, continuous representations of words, designed to capture their distributional and semantic properties. While traditional approaches assign a unique vector to each word, *contextual* word embeddings address the problem of polysemy by computing word vectors depending on the specific context in which words appear. Here, we are interested in assessing how much the concept addressed by the anchor in the input text is related to the candidate entity. Hence, we apply BERT to the input text, extracting the embeddings vector corresponding to the anchor. This vector captures the semantic properties of the anchor with respect to the specific context in which it appears. Similarly, we also apply BERT to the candidate's short abstract, extracting the vector corresponding to the first occurrence of the resource name in the abstract. We obtain an embeddings vector representing the candidate within its abstract. Finally, we compute the cosine similarity between the two aforementioned vectors, thus assessing the semantic similarity between the candidate and the anchor. This feature can help our regression model to address polysemy, by providing different representations of the same tokens appearing in different contexts and by estimating latent semantic similarities between candidates and their corresponding anchors.

5.3. Regression algorithms

To train our regression models we resort to state-of-the-art machine learning algorithms based on decision trees. In detail, we experiment with the 3 following algorithms: (i) Random Forest (RF), (ii) Gradient Boosting Decision Trees (GBDT), and (iii) Dropouts meet multiple Additive Regression Trees (DART). RF is a well-known ensemble learning technique based on multiple decision trees. The different trees are trained in parallel, each one receiving as input a random sample of the training instances and of the available features, implementing the so-called *bagging* approach. The outputs of the trees are aggregated by a suitable ensemble method. The bagging approach is able to mitigate the model variance and possible overfitting, making RF an efficient and accurate technique. GBDT implements the ensemble learning paradigm in a completely different way. During the training phase, GBDT grows a sequence of *weak* learners (i.e., shallow trees), in which each weak learner focuses on correcting the residual errors of the current model approximation. By aggregating the weak learner outputs, GBDT generates a *strong* learner. GBDT often outperforms RF in terms of prediction performance, but it is more computationally expensive and more prone to overfitting. The last algorithm with which we experiment is DART. It modifies an ensemble learning approach similar to GBDT by introducing *dropout*, a feature borrowed from deep learning. In this context, dropout consists in randomly dropping trees. This strategy proves useful to prevent trivial trees and to mitigate overfitting, but it has a negative impact on computational efficiency [58].

We implemented all the aforementioned algorithms with the Microsoft's LightGBM[13] framework, representing the current state-of-the-art for tree-based classification, regression and ranking [39].

5.4. Evaluation

Experimental setup. We train our regression models using the training set described in Section 6.1. For each algorithm, we adopt Randomized Search Cross Validation (RSCV) to explore the hyper-parameters space and optimize algorithms settings. We resort to RSCV since it is more efficient than grid search or manual search, especially in the presence of hyper-parameters sampled from continuous distributions [59], as in our case. Tree-based algorithms are also known for variable, non-deterministic behaviors, resulting in a certain amount of variance in their performance across different runs. To account for this behavior, we repeat each experiment 10 times, reporting mean and standard deviation of the results, to gain a more reliable estimation of their performance.

At the beginning of this section we explained that the best candidate is simply chosen as the one with the highest estimated confidence score $\hat{c}$. However, we enforce an additional constraint in order to provide more accurate results. In particular, we discard those candidates whose score $\hat{c} < c_{th}$. The value c_{th} represents a confidence threshold that allows to prune predictions generated with a very low confidence. We calibrate c_{th} to the value that maximizes the model's *F1* on the validation set, specifically created for this purpose. Finally, the obtained models are evaluated on the test set with a fully blind approach. Messages t_i belonging to validation and test sets are never used to train the regression models, thus mitigating the risk of overfitting when we calibrate the confidence threshold c_{th}, and when we evaluate the model performance.

Evaluation metrics. We evaluate each selection model as a combination of the related regression and filtering steps. Since the selection concludes the GSP's analysis pipeline, the most natural choice is to evaluate it according to the same metrics used for the overall geoparsing task. Hence, we adopt the *precision*, *recall* and *F1* metrics, as defined in Section 3.1.

Results. In Fig. 6, we present a performance comparison of the different regression algorithms for the selection task, as a function of the expansion size L. In particular, Fig. 6a, b and c report the trends of the evaluation metrics for $0 \leq L \leq 20$. Instead, Fig. 6d, e and f provide the percentage gain/loss attributable to the expansion step (thus, for $L > 0$) with respect to the case without any expansion ($L = 0$). Each point in the figures represents the mean scores obtained on the 10 runs, while error bars represent the corresponding standard deviation.

A first consideration regards the whole GSP approach. As expected, the expansion and selection steps provide complementary contributions. In fact, as the expansion size L grows, so does the overall geoparsing *recall*, as visible in Fig. 6b and e. Depending on the algorithm and the expansion size, the improvement in recall ranges from $+21\%$ to $+42\%$. This result confirms our starting motivation of using expansion for enriching the set of candidate entities from which to extract pertinent geographic information. It also confirms our previous findings related to the *maximum theoretical recall* of the different expansion strategies. However, as shown in Fig. 6a and d, expanding also hinders *precision*, given the higher likelihood of false positive predictions. Hence the need for an accurate selection step to mitigate losses in precision. In our experiments, such losses range from -2% to -12%. As a consequence of these results, the overall balance achieved by GSP is very positive, meaning that the striking recall gain largely outweighs the relatively small precision loss. This is demonstrated by the *F1* trends, presented in Fig. 6c and f. Although the previous considerations hold for all regression algorithms, GBDT achieves consistently better results with respect to RF and DART. In particular, GBDT achieves the best global result with $F1 = 0.665$ (resulting in a $+19\%$ *F1* gain) at the expansion size $L = 14$. This result corresponds to *precision* $= 0.737$ (-7%) and *recall* $= 0.606$ ($+40\%$), confirming our previous point.

Finally, Fig. 6c highlights the existence of a performance plateaux for $L \geq 4$, revealing a sort of "saturation" in the learning process at larger expansion sizes. This effect may be due to the shortage of training examples for which good candidates are retrieved only at large expansion sizes. In turn, this shortage of training examples prevents our models from effectively learning how to recognize them. Larger annotated datasets may allow to delay this plateaux, resulting in additional performance improvements for large expansion sizes.

Building on our results so far, from now on we consider GSP with its best configuration resulting from the adoption of topological expansion

[12] https://github.com/google-research/bert
[13] https://github.com/microsoft/LightGBM

Table 1
Grouping and brief description of the features used for the regression task. We specify categorical features, together with their cardinality, and case insensitive features.

A&E	1	`confidence`: semantic annotator confidence score	
	2	`hop`: topological distance between the starting node and the candidate	
	3	`expansion_rank`: number of entities traversed by the expansion strategy to reach the candidate	
	4	`expansion_rank_onlygeo`: number of geographic entities traversed by the expansion strategy to reach the candidate	
SPE	5	`num_tokens_candidate_label`: number of tokens in the candidate entity name	
	6	`len_candidate_label`: number of characters in the candidate entity name	
	7	`edit_from_original_label`: edit distance between the names of the starting node and of the candidate	
	8	`num_tokens_anchor`: number of tokens in the anchor	
	9	`len_anchor`: number of characters in the anchor	
	10	`uppercase_in_anchor`: number of uppercase characters in the anchor	
	11	`edit_from_anchor`: edit distance between the candidate entity name and the anchor	
	12	`edit_ratio_from_anchor`: ratio between `edit_from_anchor` and `len_anchor`	
	13	`num_tokens_ratio`: ratio between `num_tokens_candidate_label` and `num_tokens_anchor`	
	14	`len_ratio`: ratio between `len_candidate_label` and `len_anchor`	
DBP	15	`superclass`: *DBpedia* ontology class of the candidate entity, derived from `owl:Thing` subclasses	[`categorical, 5`]
	16	`num_of_superclasses`: number of superclasses of the *DBpedia* candidate entity	
	17	`num_of_classes`: number of classes of the *DBpedia* candidate entity	
	18	`page_degree`: node degree of the *DBpedia* candidate entity	
	19	`page_length`: number of characters contained in the corresponding *Wikipedia* article source	
	20	`anchor_in_short_abstract`: num. Occurrences of the anchor in the short abstract of the candidate entity	
	21	`anchor_in_short_abstract_ci`: num. Occurrences of the anchor in the short abstract of the candidate entity	[`case insensitive`]
	22	`anchor_in_long_abstract`: num. Occurrences of the anchor in the long abstract of the candidate entity	
	23	`anchor_in_long_abstract_ci`: num. Occurrences of the anchor in the long abstract of the candidate entity	[`case insensitive`]
SUN	24	`pos_tag`: part-of-speech (POS) tag of the anchor	[`categorical, 50`]
	25	`chunk_tag`: chunking tag of the anchor	[`categorical, 10`]
	26	`pos_confidence`: POS-tagging confidence	
	27	`chunk_confidence`: chunking confidence	
NER	28	`ner_tag`: named-entity recognition (NER) tag of the anchor	[`categorical, 5`]
	29	`ner_confidence`: NER-tagging confidence	
LAT	30	`rdf2vec_similarity`: cosine similarity between the starting node's and the candidate's *rdf2vec* vectors	
	31	`bert_similarity`: cosine similarity between the anchor's and the candidate's name BERT vectors	

with spelling-based sorting and the GBDT algorithm for the selection step. Expansion size is set to $L = 14$.

6. Geoparsing results

In this section we advance our thorough analysis of GSP's geoparsing results. Firstly, we present the details of our dataset. Then, we compare the best configuration of GSP with several baselines, with 2 state-of-the-art geoparsing algorithms, and with our previous technique. Finally, we provide additional results on GSP's predictions, both in terms of identifying the most informative features for the selection task and of assessing the spatial granularity of our predictions.

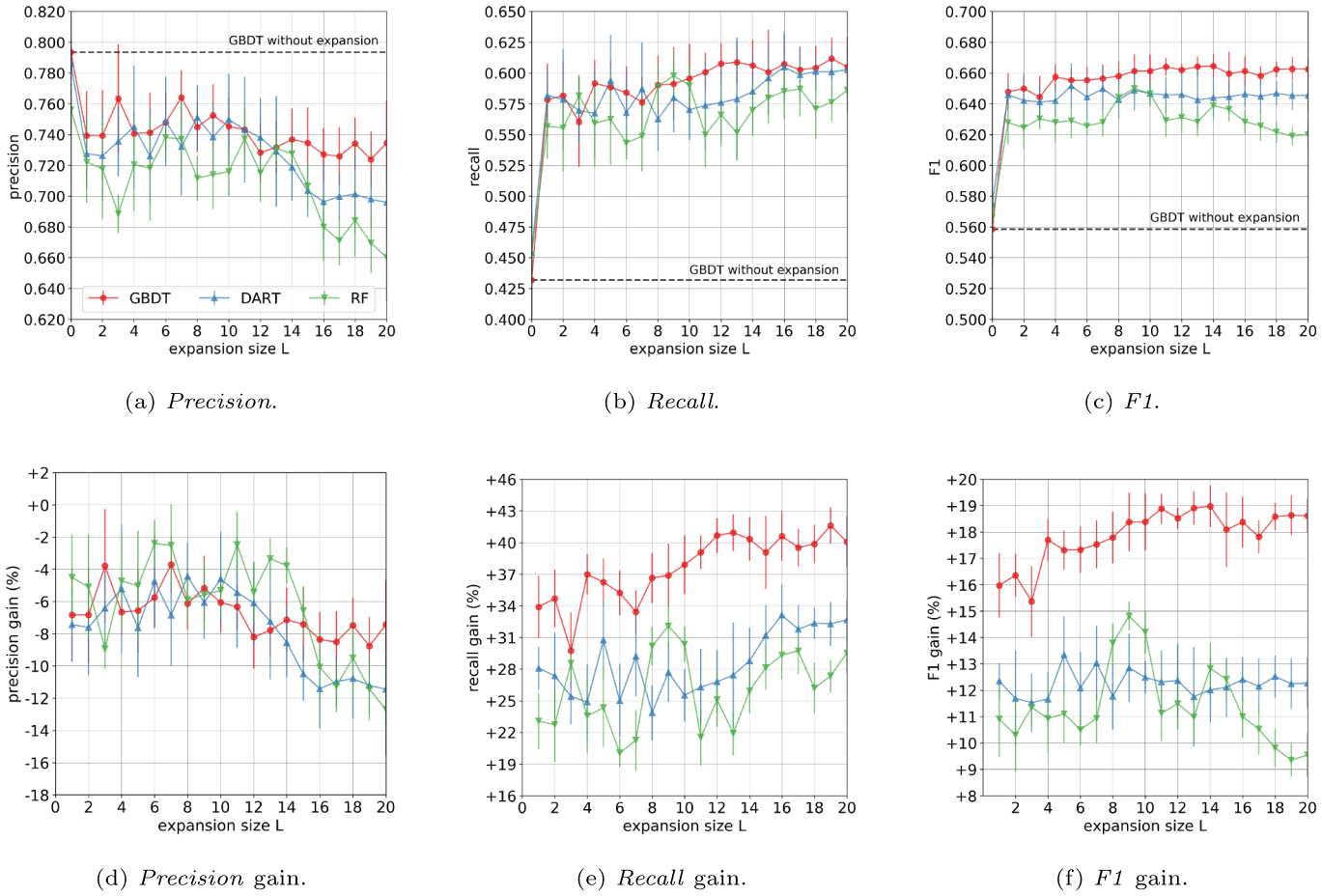
6.1. Dataset

In Section 1, we underlined the importance of the geoparsing task to properly integrate OSN data in decision support systems. Although geoparsing techniques can process any kind of texts (e.g., news articles, emails), we remarked the challenges posed by OSN user-generated content, which is characterized by short texts, poor context and use of jargon and colloquial expressions. For these reasons, we train and evaluate our technique on a dataset composed of OSN user-generated posts.

In particular, we use the official dataset of the *2016 Named entity recognition and linking* challenge (*NEEL16*).[14] This well-known, reference dataset includes 9289 English tweets, extracted from a corpus of over 18 M documents, covering several noteworthy events from 2011 to 13, and a set of hashtags from 2014 to 15. Notably, *NEEL16* challenge organizers provide annotations about mentions of places/locations, enabling the usage of the dataset for geoparsing purposes. In particular, each ground truth location is complemented with the corresponding geographic coordinates as well as with the link to the corresponding resource on the English *DBpedia*. The total number of locations is 5348. Their distribution across tweets is very skewed, as shown in Fig. 7a. Namely, 85.4% of the tweets do not mention any location, 10.1% of the tweets mentions one location, and only 4.5% of the tweets mentions multiple locations. Notably, locations are scattered all over the world, as shown in Fig. 7b. Working at the world scale poses a severe challenge to geoparsing methods, but at the same time it is a requirement for many real-world applications. As a result, the *NEEL16* dataset provides a suitable playground for training and evaluating geoparsing techniques.

Starting from the complete *NEEL16* dataset, we obtain training (64% of tweets), validation (16%) and test (20%) sets by performing a stratified sampling over the number of locations per tweet. We also include tweets without any mention of locations, because we want to ensure that all evaluated techniques do not return false positive predictions for them. As a result of this sampling strategy, the total number of locations is balanced across the obtained dataset splits. Moreover, performing the split at the message level – and not at the single location instance level – ensures that instances in the test set cannot affect the model training and validation. In this way, we evaluate our models with a fully blind test, correctly assessing possible overfitting. Our evaluation for this work is more severe than the one used in our previous work [2], thus explaining the differences in the reported performance.

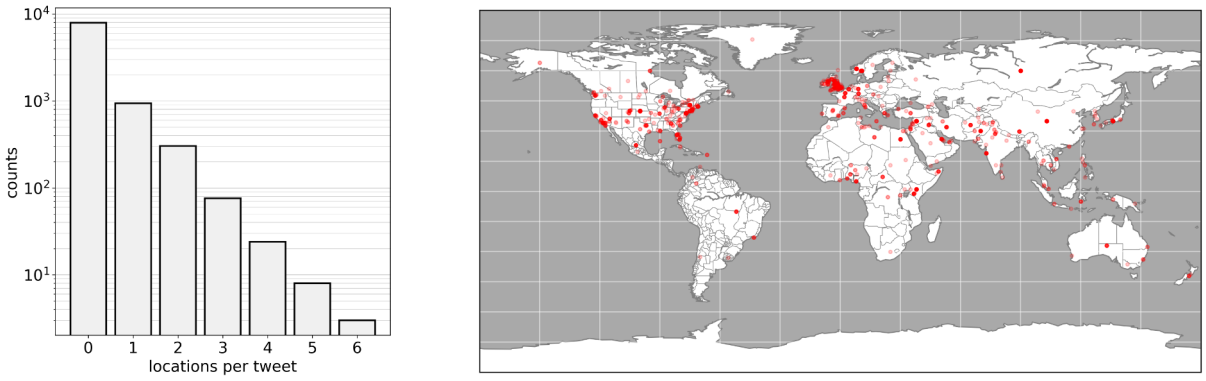
[14] https://aclweb.org/portal/content/named-entity-recognition-and-linking-challenge



(a) *Precision*. (b) *Recall*. (c) *F1*.

(d) *Precision* gain. (e) *Recall* gain. (f) *F1* gain.

Fig. 6. Performance comparison of different regression algorithms for the selection task, when varying the expansion size *L*.



(a) Distribution of mentioned locations per tweet.

(b) Geographic distribution of ground truth locations.

Fig. 7. *NEEL16* dataset profiling.

6.2. Performance comparisons

Benchmarks. Performance comparisons are aimed at evaluating the performance of our proposed GSP technique, with reference to those of baselines and other advanced geoparsing systems. The first baseline that we implemented (labeled naïve geoparser), leverages the *geopy* Python package[15] as an interface to the online *ArcGIS* geocoding service.[16] This service is designed to geocode well-formatted addresses, but it also includes a simple free-text processing feature. As such, it represents a rather simplistic approach for geoparsing short and noisy documents. The second baseline (labeled NER + geocoder) is a basic implementation of the NER + gazetteer lookup approach. We implement the NER step using the well-known *polyglot* natural language processing framework.[17] Then, we perform the gazetteer lookup step by

[15] https://geopy.readthedocs.io/

[16] https://developers.arcgis.com/features/geocoding/

Table 2
Geoparsing performance comparison between GSP, 2 baselines, and 3 state-of-the-art geoparsing techniques. Best results for each metric are shown in **bold** font. Beside each metric, we report in parentheses the % gain/loss of each technique with respect to GSP. All differences between GSP and the other benchmarks are statistically significant, except for the elapsed time with respect to Avvenuti et al.

Technique	Evaluation metrics							
	Precision (± %)		*Recall (± %)*		*F1 (± %)*		*Elapsed time (± %)*	
Benchmarks								
Naïve geoparser	0.033	(−2133.33)	0.157	(−285.99)	0.054	(−1131.48)	0.811 s	(+ 60.30)
NER + geocoder	0.300	(−145.67)	0.267	(−126.97)	0.282	(−135.82)	0.113 s	(− 184.96)
Middleton et al. [33]	0.123	(−499.19)	0.333	(−81.98)	0.180	(−269.44)	**0.011 s**	(− 2827.27)
Halterman [37]	0.320	(−130.31)	0.299	(−102.68)	0.309	(−115.21)	0.049 s	(− 557.14)
Avvenuti et al. [2]	**0.818**	(+9.90)	0.417	(−45.32)	0.553	(−20.25)	0.321 s	(− 0.31)
Our contribution								
GSP	0.737		**0.606**		**0.665**		0.322 s	

means of the *Google Maps* geocoder. Adding to the 2 simple baselines, we also compare our results against 2 state-of-the-art geoparsers. As anticipated in Section 2, we include as benchmarks the techniques proposed by Middleton et al. [33] and by Halterman [37]. Both techniques leverage the common approach to geoparsing based on Named-Entity Recognition and geographic gazetteer lookup. Middleton et al.'s technique returns the location tokens extracted from the input text, delegating the user to query the *OpenStreetMap* gazetteer by means of a dedicated API service. Instead, Halterman's technique directly performs also the disambiguation step, returning the location coordinates. In fact, this model is more sophisticated than that by Middleton et al., leveraging a deep learning neural network classifier to select the proper instances in the *GeoNames* gazetteer. In both cases, once the location named entities are linked to the proper gazetteer entries, the system returns the related coordinates. Finally, we also include our earlier geoparsing technique [2] in the comparison. Here, our goal is that of evaluating the effectiveness of the novel expansion and selection steps, with respect to the simpler approach developed in [2].

Evaluation metrics and experimental setup. We evaluate geoparsing techniques according to the *precision*, *recall* and *F1* metrics, described in Section 3.1. Moreover, we introduce also the *elapsed time*, defined as the average time that a geoparsing technique needs to process a single tweet belonging to the test set. All the compared techniques are implemented with the Python language and deployed on a machine equipped with an 8-core *CPU* featuring 50Gb of *RAM*, and a Nvidia Tesla K80 GPU.

Results. Table 2 reports a thorough comparison of geoparsing results. As shown, GSP outperforms all competitors. Both baselines performed rather poorly, as expected for a challenging task such as geoparsing. However, surprisingly the NER + geocoder baseline managed to beat the system in [33]. An analysis of the results reveals that this is mainly due to the need to perform geoparsing at world-level, which made it challenging for [33] to correctly disambiguate detected toponyms. As we discussed in Section 2, many geoparsing systems based on gazetteer lookup need to be constrained to operate in a geographically limited area, for maintaining satisfactory performance. In our case, this could not be done, since the *NEEL16* dataset is geographically unconstrained, including locations from all over the world, as shown in Fig. 7b.

When compared to previous geoparsing techniques, the *F*1 gain of GSP ranges from +269.44% with respect to Middleton et al. [33] to +20.25% with respect to our previous technique [2]. The improvement from our previous attempt at the geoparsing task and GSP is determined by our higher recall. Indeed, a relatively low recall was the limiting factor in [2]. Having improved on the recall, our proposed GSP technique managed to obtain overall better results. In turn, this motivates our design choices related to the expansion step. Anyway, the large recall gain is partly counterbalanced by a slightly reduced precision (−9.90%), demonstrating the difficulty at correctly selecting the best entity among those retrieved during the expansion step.

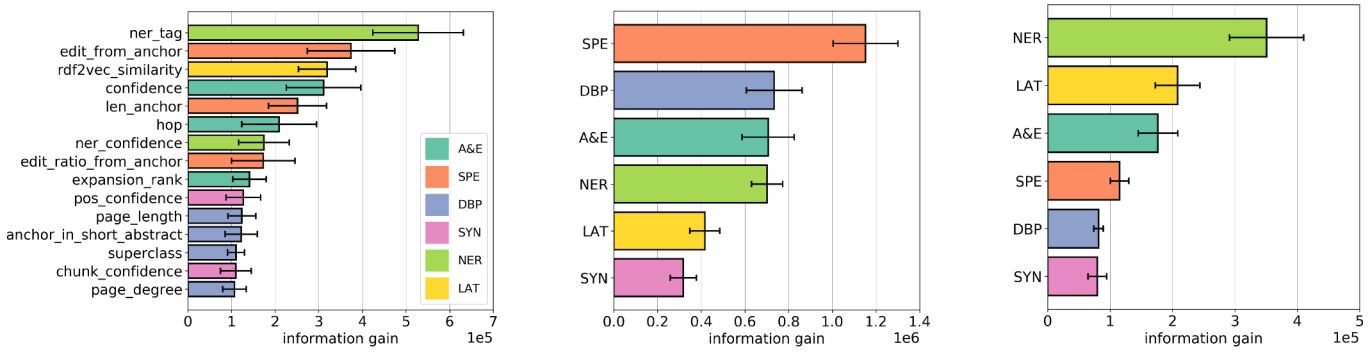
The price for more accurate predictions is slightly paid in terms of time efficiency. The last column of Table 2 shows that GSP (both in its present and earlier version) needs around 0.3 s to geoparse a single tweet, compared to 0.05 of Halterman and 0.01 of Middleton et al. Despite being computationally more demanding, GSP is nonetheless suitable for real-time applications, also considering the possibility to deploy multiple parallel instances in those cases requiring high throughput.

6.3. Feature importance analysis

In Fig. 8 we report the results of the feature importance analysis for the best configuration of GSP (i.e., the one using GBDT for the selection step). Adopting a standard approach, we compute the feature importance as the sum of the *information gain* provided by a feature each time it triggers a split in one of the trees composing the model. In particular, Fig. 8a shows the importance of the top-15 individual features. Interestingly, all feature groups make it to the top-15, meaning that each group conveys useful information for the model. The named-entity recognition (NER) tag is the most important feature in our model. This further confirms the validity of previous approaches to geoparsing, based on NER tagging and gazetteer lookup. As expected, also features measuring the difference between the anchor and the entity name play an important role (`edit_from_anchor` and `edit_ratio_from_anchor`). The third most-informative feature is a latent semantic feature – namely, the one computed out of *rdf2vec* node embeddings. This feature measures the semantic difference between the starting node returned by the semantic annotator and the candidate obtained via expansion.

By leveraging information gain's *additivity*, we also collectively evaluate feature importance by group. Fig. 8b reports the results of this analysis, showing that *spelling* (SPE) and *DBpedia* (DBP) features account for the majority of contributions. This result confirms the importance of modeling the intrinsic properties and the similarities of the anchors and the entities. In order to avoid possible biases due to the numerosity of certain feature groups with respect to others, we also normalize the feature importance of each group by its cardinality. After normalization we observe a different ranking, dominated by the more sophisticated, information-rich NER and by latent semantic features (LAT). Instead, *syntactic* (SYN) features seems to provide the smallest contribution throughout all experiments. We underline that these observations are intended to shed light on the overall learning process. In fact, a rigorous assessment of the feature importance should account for possible correlations between features, that is beyond the scope of this analysis.

[17] https://polyglot.readthedocs.io/



(a) Individual feature importance.

(b) Feature importance by group.

(c) Feature importance by group, normalized by its cardinality.

Fig. 8. Feature importance analysis for the best configuration of GSP, measured as the information gain provided by each feature in the regression model. We refer to Table 1 for details about the features and the related groups and acronyms.

6.4. Place granularity analysis

A favorable feature of GSP is the possibility to leverage knowledge graphs and the Linked Data ontology to infer the granularity of the predicted locations. Following previous works [33], we define 4 granularity levels: (i) points of interest (POIs), roughly corresponding to buildings and other notable landmarks; (ii) cities; (iii) regions and counties, and (iv) countries. To infer the granularity level of a given prediction by GSP, we analyze the *DBpedia* ontology of the geographic entity providing the coordinates. For example, if we geotagged a token with the entity `Bath`, our prediction would be at the city-level, since the *DBpedia* resource for Bath has `rdf:Type=dbo:City`.[18] By following a similar approach, we are able to assess the granularity of each distinct ground truth instance in the *NEEL16* dataset, since ground truth annotations are complemented with *DBpedia* URLs. We can thus perform a granularity-aware evaluation of GSP, by considering as true positives only those predictions matching both the ground truth coordinates and the corresponding granularity level.

We show the results of this more severe evaluation in Fig. 9a. The overall results are still satisfactory, with good performance at the country ($F1 = 0.670$) and city ($F1 = 0.600$) granularity levels. Conversely, the performance slightly drops for regions ($F1 = 0.458$) and POIs ($F1 = 0.491$). To explain this result, in Fig. 9b we provide the distribution of the ground truth granularity levels in the dataset. As shown, regions and POIs are significantly underrepresented. This possibly explains why our model struggled to learn instances at these granularity levels. In summary, GSP shows suitable performance in location granularity prediction, although there is room to improve it by enriching and balancing the training dataset.

7. Discussion

The geoparsing results of the proposed GSP technique, and the comparison with baselines and other state-of-the-art techniques, demonstrated the effectiveness of our design choices, and the favorable compelling performance of GSP. Building on these results, in this section we discuss some additional features of our technique, with a specific focus on its *robustness*, *generalizability*, *extensibility* and *applicability*.

7.1. Robustness and generalizability

Although evaluated on tweets, our technique does not make any assumption on the input text, and it does not exploit any peculiar feature of Twitter nor of OSNs in general. Because of this, it is suitable to geoparse any textual document, including longer texts such as news articles, emails and blog posts. The choice of evaluating our technique on the OSN-derived *NEEL16* dataset stems from the will to test GSP on challenging texts. In fact, OSN user-generated content – and *tweets* specifically – are known for their shortness, lexical sparsity, and for the use of abbreviations, jargon and colloquial expression. As such, they represent a proving ground for any text mining technique. Given this scenario, our already promising results are likely to further improve, should GSP be applied to geoparse longer and well-written texts. In addition to the challenges related to the analysis of short OSN texts, the *NEEL16* dataset also presents other pitfalls. Indeed, it considers multiple different events and topics, spread across a large geographic area (as shown in Fig. 7b), and encompassing several years.

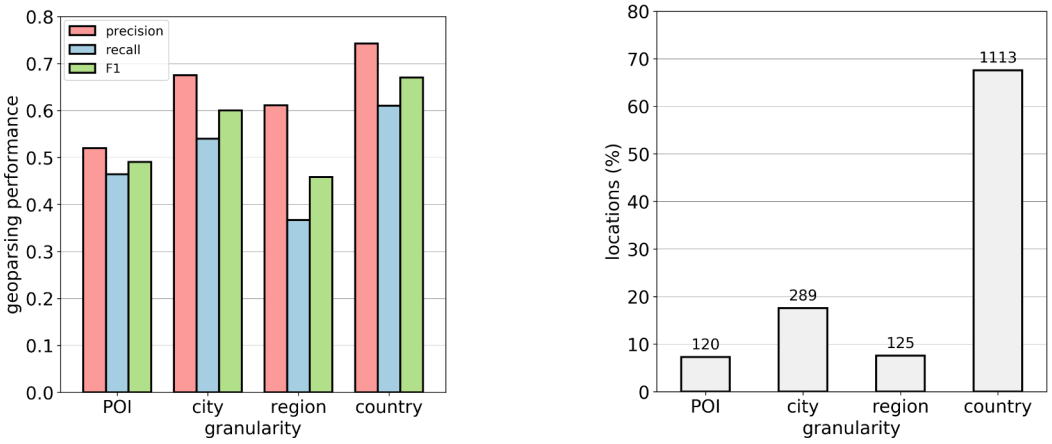
In conclusion, the positive results obtained on this challenging and diverse evaluation dataset guarantee that GSP can generalize well also to other texts and topics, thus proving its robustness and generalizability. Results of the application of GSP in-the-wild are thus likely to remain very positive. As a final remark on robustness and generalizability, we report that both GSP and its ancestor [2] correctly geoparse the challenging toy example of Section 4, whereas only the NER + geocoder succeeds among all other benchmarks.

7.2. Extensibility

Contrarily to other geoparsing techniques, GSP does not suffer from the restriction to be applied to geographically-limited or predefined areas. In fact, it can easily predict locations worldwide, as demonstrated in our experiments. However, the model proposed in this work was developed for processing English texts. This limitation does not derive from our design choices, that are totally language-independent, but it only depends on the availability of the language-specific components used by GSP. In other words, our technique could be applied to documents in any language, without any modification, provided that certain resources for that language exist. In particular, the first language-dependent component used by GSP is the semantic annotator. At the time of writing, the one used in our work (i.e., TagMe) processes English, German and Italian texts. However, other well-known annotators, such as DBpedia Spotlight,[19] support as much as 12 languages with the possibility to extend it to additional ones. Moreover, our technique needs language-specific models for NER, chunking and part-of-speech tagging, as well as for BERT and *rdf2vec* embeddings. These requirements are similar to those of many other text mining techniques, and

[18] http://dbpedia.org/page/Bath,_Somerset

[19] https://www.dbpedia-spotlight.org/



(a) Geoparsing results of GSP at different granularity levels.

(b) Distribution of distinct ground truth locations per granularity level.

Fig. 9. Breakdown of GSP geoparsing results at different spatial granularity levels.

they can be easily met. In fact, many natural language processing (NLP) tools are available for the most widespread languages. As an example, the NLP library *polyglot* supports from 16 to 196 languages, depending on the task. Similarly, BERT features a multi-lingual model, presently including 104 languages. Finally, an open-source library[20] allows training *rdf2vec* models for all existing *DBpedia*s. This discussion highlighted which tools are needed to extend GSP to other languages, and where to start for building a deployment in a specific language.

7.3. Applicability

Thanks to its previously discussed robustness, generalizability and extensibility, the GSP technique proves suitable for integration in geo-spatial decision support systems based on OSN data, allowing very general and flexible settings. When empowered with GSP, those systems benefit from a significantly increased amount of accurate and structured geographic data, provided at the most specific granularity available. The comparison with state-of-the-art benchmarks, provided in Section 6.2, demonstrated remarkably higher performance of GSP. In particular, methods based on NER + gazetteer lookup proved unable of working at the world scale, mainly due to their rough disambiguation approaches. This implies that such simple approaches can not be profitably used for tasks such as monitoring epidemic spreading or international tourism flows, contrarily to GSP. Furthermore, although the higher complexity of GSP increases the time elapsed to process a message with respect to the other benchmarks, it still allows for practical, real-time applications. Notably, the suitability of a simpler version of our geo-semantic-parsing technique for decision support systems was already proved in [6]. When integrated in the referenced crisis mapping system, GSP contributed to increase the fraction of georeferenced tweets from a poor 5% to a remarkable 39%, thus significantly extending the coverage and improving the accuracy of crisis maps.

8. Conclusions and future works

Motivated by current limitations of existing geoparsing techniques, we proposed Geo-Semantic-Parsing (GSP) – a novel technique for enriching text documents with structured geographic information. GSP leverages semantic annotation to identify relevant portions of the input text and to link them to pertinent entities in knowledge graphs, such as *DBpedia*. Then, it exploits the information-rich and interconnected nature of the knowledge graph to retrieve additional entities from which to extract geographic information. To reach this goal, we devised an *expansion* step that allows GSP to efficiently traverse the knowledge graph. Finally, in a dedicated *selection* step, GSP selects the best entity with which to geotag the input document by solving a regression task. Extensive experimental results demonstrated the viability of our solution. In particular, GSP outperformed all state-of-the-art competitors achieving $F1 = 0.66$ versus $F1 \leq 0.55$ of other techniques. Due to its robustness, generalizability and extensibility, GSP can be integrated in geo-spatial decision support systems based on OSN data, empowering them with accurate and structured geographic data, available in real time. Notably, this kind of approach was able to increase the fraction of georeferenced tweets from a poor 5% to a remarkable 39% in a real-world crisis mapping setting.

Future works on geoparsing could investigate more sophisticated methods to effectively combine different, mutually-orthogonal expansion strategies, and possibly even multiple semantic annotators. As an alternative approach, interested stakeholders could also develop semantic annotators that are specifically designed to return pertinent geographic entities, given the importance of geographic information for many downstream tasks. Finally, an alluring line of research could involve investigating end-to-end geoparsing techniques, developed on top of state-of-the-art contextual word embeddings. In fact, word embeddings vector spaces are known to reflect spatial relationships between words in their topology. As a consequence, it could be possible to map the *N*-dimensional word embeddings vector space directly on the geographic space by learning a suitable projection function. Semantic knowledge graphs provide the ideal playground to learn such models, since they represent large corpora of textual documents. Moreover, such documents could be considered as already annotated, thanks to the hyperlink structure and the geographic information contained in the semantic resources.

Acknowledgements

The authors would like to thank Franco Maria Nardini for his insightful suggestions. This research is supported in part by the EU H2020 Program under the scheme INFRAIA-01-2018-2019: Research and Innovation action grant agreement #871042 SoBigData++: European Integrated Infrastructure for Social Mining and Big Data Analytics.

[20] https://github.com/IBCNServices/pyRDF2Vec

References

[1] G. Kordopatis-Zilos, S. Papadopoulos, I. Kompatsiaris, Geotagging text content with language models and feature mining, Proc. IEEE 105 (10) (2017) 1971–1986.
[2] M. Avvenuti, S. Cresci, L. Nizzoli, M. Tesconi, GSP (Geo-Semantic-Parsing): Geoparsing and geotagging with machine learning on top of linked data, Proc. of ESWC'18, 2018, pp. 17–32.
[3] P.B. Keenan, P. Jankowski, Spatial decision support systems: three decades on, Decis. Support. Syst. 116 (2019) 64–76.
[4] G. Andrienko, N. Andrienko, C. Boldrini, G. Caldarelli, P. Cintia, S. Cresci, A. Facchini, F. Giannotti, A. Gionis, R. Guidotti, M. Mathioudakis, C.I. Muntean, L. Pappalardo, D. Pedreschi, E. Pournaras, F. Pratesi, M. Tesconi, R. Trasarti, (So) Big Data and the transformation of the city, Int. J. Data Sci. Anal. (2020), https://doi.org/10.1007/s41060-020-00207-3.
[5] L. Divyaa, N. Pervin, Towards generating scalable personalized recommendations: integrating social trust, social bias, and geo-spatial clustering, Decis. Support. Syst. 122 (2019) 113066.
[6] M. Avvenuti, S. Cresci, F. Del Vigna, M. Tesconi, Impromptu crisis mapping to prioritize emergency response, Computer 49 (5) (2016) 28–37.
[7] M. Avvenuti, S. Cresci, F. Del Vigna, T. Fagni, M. Tesconi, CrisMap: a big data crisis mapping system based on damage detection and geoparsing, Inf. Syst. Front. (2018) 1–19.
[8] D. Wu, Y. Cui, Disaster early warning and damage assessment analysis using social media data and geo-location information, Decis. Support. Syst. 111 (2018) 48–59.
[9] M. Avvenuti, S. Cresci, A. Marchetti, C. Meletti, M. Tesconi, Predictability or early warning: using social media in modern emergency response, IEEE Internet Comput. 20 (6) (2016) 4–6.
[10] M. Avvenuti, S. Cresci, A. Marchetti, C. Meletti, M. Tesconi, EARS (Earthquake Alert and Report System): A real time decision support system for earthquake crisis management, Proc. of SIGKDD'14, ACM, 2014, pp. 1749–1758.
[11] M. Avvenuti, S. Cresci, M.N. La Polla, C. Meletti, M. Tesconi, Nowcasting of earthquake consequences using big social data, IEEE Internet Comput. 21 (6) (2017) 37–45.
[12] A. Kumar, J.P. Singh, Location reference identification from tweets during emergencies: a deep learning approach, Int. J. Disaster Risk Reduction 33 (2019) 365–375.
[13] J.A. de Bruijn, H. de Moel, B. Jongman, J. Wagemaker, J.C. Aerts, TAGGS: grouping tweets to improve global geoparsing for disaster response, J. Geovisualization Spat. Anal. 2 (1) (2018) 2.
[14] J.P. Singh, Y.K. Dwivedi, N.P. Rana, A. Kumar, K.K. Kapoor, et al., Event classification and location prediction from tweets during disasters, Ann. Oper. Res. 283 (1) (2019) 737–757.
[15] Q. Yuan, G. Cong, Z. Ma, A. Sun, N.M. Thalmann, Who, where, when and what: Discover spatio-temporal topics for Twitter users, Proc. of SIGKDD'13, ACM, 2013, pp. 605–613.
[16] F.U. Rehman, I. Afyouni, A. Lbath, S. Khan, S. Basalamah, Building socially-enabled event-enriched maps, GeoInformatica 24 (2020) 371–409.
[17] A.F. Colladon, B. Guardabascio, R. Innarella, Using social network and semantic analysis to analyze online travel forums and forecast tourism demand, Decis. Support. Syst. 123 (2019) 113075https://www.sciencedirect.com/science/article/pii/S0167923619301046.
[18] S. Cresci, A. D'Errico, D. Gazzè, A. Lo Duca, A. Marchetti, M. Tesconi, Towards a DBpedia of tourism: the case of Tourpedia, in: Proc. of ISWC'14, 2014, pp. 129–132.
[19] I.R. Brilhante, J.A. Macedo, F.M. Nardini, R. Perego, C. Renso, On planning sightseeing tours with TripBuilder, Inf. Process. Manag. 51 (2) (2015) 1–15.
[20] Y. Li, A. Vo, M. Randhawa, G. Fick, Designing utilization-based spatial healthcare accessibility decision support systems: a case of a regional health plan, Decis. Support. Syst. 99 (2017) 51–63.
[21] J. Mahmud, J. Nichols, C. Drews, Home location identification of Twitter users, ACM Transac. Intell. Syst. Technol. 5 (3) (2014) 47.
[22] S.E. Middleton, V. Krivcovs, Geoparsing and geosemantics for social media: spatiotemporal grounding of content propagating rumors to support trust and veracity analysis during breaking news, ACM Transac. Inform. Syst. 34 (3) (2016) 1–26.
[23] V. Lampos, N. Cristianini, Nowcasting events from the social web with statistical learning, ACM Transac. Intell. Syst. Technol. 3 (4) (2012) 1–22.
[24] C. Kadar, R. Maculan, S. Feuerriegel, Public decision support for low population density areas: an imbalance-aware hyper-ensemble for spatio-temporal crime prediction, Decis. Support. Syst. 119 (2019) 107–117.
[25] L. Vomfell, W.K. Härdle, S. Lessmann, Improving crime count forecasts using Twitter and taxi data, Decis. Support. Syst. 113 (2018) 73–85.
[26] R.P. Curiel, S. Cresci, C.I. Muntean, S.R. Bishop, Crime and its fear in social media, Palgrave Commun. 6 (1) (2020) 1–12.
[27] M. La Morgia, A. Mei, E. Nemmi, S. Raponi, J. Stefa, Nationality and Geolocation-Based Profiling in the Dark (Web), IEEE Transactions on Services Computing, (2019) (to appear).
[28] O. Ajao, J. Hong, W. Liu, A survey of location inference techniques on Twitter, J. Inf. Sci. 41 (6) (2015) 855–864.
[29] X. Zheng, J. Han, A. Sun, A survey of location prediction on Twitter, IEEE Trans. Knowl. Data Eng. 30 (9) (2018) 1652–1671.
[30] Y. Hu, B. Adams, Harvesting big geospatial data from natural language texts, in: M. Werner, Y.-Y. Chiang (Eds.), Handbook of Big Geospatial Data, Springer, 2020.
[31] P. Zola, P. Cortez, M. Carpita, Twitter user geolocation using web country noun searches, Decis. Support. Syst. 120 (2019) 50–59.
[32] W. Hua, K. Zheng, X. Zhou, Microblog entity linking with social temporal context, Proc. of SIGMOD'15, ACM, 2015, pp. 1761–1775.
[33] S.E. Middleton, L. Middleton, S. Modafferi, Real-time crisis mapping of natural disasters using social media, IEEE Intell. Syst. 29 (2) (2014) 9–17.
[34] M. Dredze, M. J. Paul, S. Bergsma, H. Tran, Carmen: A twitter geolocation system with applications to public health, in: Proc. of AAAI'13, AAAI.
[35] W. Shen, J. Wang, P. Luo, M. Wang, Linking named entities in tweets with knowledge base via user interest modeling, Proc. of SIGKDD'13, ACM, 2013, pp. 68–76.
[36] G. Li, J. Hu, J. Feng, K.-L. Tan, Effective location identification from microblogs, in: Proc. of ICDE'14, IEEE, 2014, pp. 880–891.
[37] A. Halterman, Mordecai: full text geoparsing and event geocoding, J. Open Source Softw. 2 (9) (2017) 91.
[38] P. Ferragina, U. Scaiella, Fast and accurate annotation of short texts with Wikipedia pages, IEEE Softw. 29 (1) (2011) 70–75.
[39] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: A highly efficient gradient boosting decision tree, Proc. of NeurIPS'17, 2017, pp. 3146–3154.
[40] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, R. Vollgraf, FLAIR: An easy-to-use framework for state-of-the-art NLP, Proc. of NAACL'19, 2019, pp. 54–59.
[41] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, Proc. of NAACL-HLT'19, 2019, pp. 4171–4186.
[42] P. Ristoski, J. Rosati, T. Di Noia, R. De Leone, H. Paulheim, RDF2Vec: RDF graph embeddings and their applications, Semantic Web 10 (4) (2019) 721–752.
[43] W. Zhang, J. Gelernter, Geocoding location expressions in Twitter messages: a preference learning method, J. Spatial Inform. Sci. 2014 (9) (2014) 37–70.
[44] H. Al-Olimat, K. Thirunarayan, V. Shalin, A. Sheth, Location name extraction from targeted text streams using gazetteer-based statistical language models, Proc. of ACL'18, 2018, pp. 1986–1997.
[45] G. Skoumas, D. Pfoser, A. Kyrillidis, T. Sellis, Location estimation using crowdsourced spatial relations, ACM Trans. Spatial Algorithms Syst. 2 (2) (2016) 1–23.
[46] S.E. Middleton, G. Kordopatis-Zilos, S. Papadopoulos, Y. Kompatsiaris, Location extraction from social media: Geoparsing, location disambiguation, and geotagging, ACM Transac. Inform. Syst. 36 (4) (2018) 1–27.
[47] F. Laylavi, A. Rajabifard, M. Kalantari, A multi-element approach to location inference of Twitter: a case for emergency response, ISPRS Int. J. Geo Inf. 5 (5) (2016) 56.
[48] E. Williams, J. Gray, B. Dixon, Improving geolocation of social media posts, Pervasive Mobile Comput. 36 (2017) 68–79.
[49] A. Zubiaga, A. Voss, R. Procter, M. Liakata, B. Wang, A. Tsakalidis, Towards real-time, country-level location classification of worldwide tweets, IEEE Trans. Knowl. Data Eng. 29 (9) (2017) 2053–2066.
[50] P. Paraskevopoulos, T. Palpanas, Where has this tweet come from? Fast and fine-grained geolocalization of non-geotagged tweets, Soc. Netw. Anal. Min. 6 (1) (2016) 89.
[51] J.D.G. Paule, Y. Sun, Y. Moshfeghi, On fine-grained geolocalisation of tweets and real-time traffic incident detection, Inf. Process. Manag. 56 (3) (2019) 1119–1132.
[52] L. Luceri, D. Andreoletti, M. Tornatore, T. Braun, S. Giordano, Measurement and control of geo-location privacy on Twitter, Online Soc. Netw. Media 17 (2020) 100078.
[53] D. Kotzias, T. Lappas, D. Gunopulos, Home is where your friends are: utilizing the social graph to locate Twitter users in a city, Inf. Syst. 57 (2016) 77–87.
[54] C.A. Davis Jr., G.L. Pappa, D.R.R. de Oliveira, F. de L. Arcanjo, Inferring the location of Twitter messages based on user relationships, Trans. GIS 15 (6) (2011) 735–751.
[55] E. Rodrigues, R. Assunção, G.L. Pappa, D. Renno, W. Meira Jr, Exploring multiple evidence to infer users' location in Twitter, Neurocomputing 171 (2016) 30–38.
[56] B. Han, P. Cook, T. Baldwin, Text-based Twitter user geolocation prediction, J. Artif. Intell. Res. 49 (2014) 451–500.
[57] F. Hasibi, K. Balog, S.E. Bratsberg, On the reproducibility of the TagMe entity linking system, Proc. of ECIR'16, Springer, 2016, pp. 436–449.
[58] K.V. Rashmi, R. Gilad-Bachrach, DART: dropouts meet multiple additive regression trees, J. Mach. Learn. Res. 38 (2015) 489–497.
[59] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305.

Leonardo Nizzoli is a PhD student in the Department of Information Engineering at the University of Pisa, and a Research Fellow at IIT-CNR, Italy. His interests include Social Media Intelligence and Artificial Intelligence. Leonardo received an MSc in Physics and an MSc in Big Data Analytics & Social Mining from the University of Pisa. Email: l.nizzoli@iit.cnr.it

Marco Avvenuti (PhD) is Full Professor of computer systems with the Department of Information Engineering at the University of Pisa. His research interests include human-centric sensing and social media analysis. Marco received a PhD in Information Engineering from the University of Pisa. He is member of the IEEE. Email: marco.avvenuti@unipi.it

Maurizio Tesconi (PhD) is a Researcher in Computer Science and Head of the Web Applications for the Future Internet Lab at IIT-CNR, Italy. His research interests include big data, web mining, social network analysis and visual analytics within the context of Open Source Intelligence. He is a member of the permanent team of the European Laboratory on Big Data Analytics and Social Mining, performing advanced research and analyses on the emerging challenges posed by Big Data. Email: m.tesconi@iit.cnr.it

Stefano Cresci (PhD) is a Researcher at IIT-CNR, Italy. His interests broadly fall at the intersection of Web Science and Data Science, with a focus on information disorder, coordinated inauthentic behavior, online social networks security, and crisis informatics. Stefano received a PhD in Information Engineering from the University of Pisa. In 2018, he was selected among the winners of a SAGE Ocean Concept Grant. In 2019, he won the IEEE Computer Society Italy Section Chapter 2018 PhD Thesis Award, and the IEEE Next-Generation Data Scientist Award. He is member of the IEEE and ACM. Email: s.cresci@iit.cnr.it