GeospatRE: extraction and geocoding of spatial relation entities in textual documents

Mehtab Alam Syed, Elena Arsevska, Mathieu Roche, Maguelonne Teisseire

Taylor & Francis
Taylor & Francis Group

Cartography and Geographic Information Science

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tcag20

GeospatRE: extraction and geocoding of spatial relation entities in textual documents

Mehtab Alam Syed, Elena Arsevska, Mathieu Roche & Maguelonne Teisseire

To cite this article: Mehtab Alam Syed, Elena Arsevska, Mathieu Roche & Maguelonne Teisseire (30 Nov 2023): GeospatRE: extraction and geocoding of spatial relation entities in textual documents, Cartography and Geographic Information Science, DOI: 10.1080/15230406.2023.2264753

To link to this article: https://doi.org/10.1080/15230406.2023.2264753

CARTOGRAPHY AND GEOGRAPHIC INFORMATION SCIENCE
https://doi.org/10.1080/15230406.2023.2264753

Taylor & Francis
Taylor & Francis Group

GeospatRE: extraction and geocoding of spatial relation entities in textual documents

Mehtab Alam Syed [a,d], Elena Arsevska [b,e], Mathieu Roche [a,d] and Maguelonne Teisseire [c]

[a]CIRAD, UMR TETIS, F-34398, Montpellier, France; [b]CIRAD, UMR ASTRE, Montpellier, France; [c]INRAE, UMR TETIS, Montpellier, France; [d]TETIS, Univ Montpellier, AgroParisTech, CIRAD, CNRS, INRAE, Montpellier, France; [e]ASTRE, Univ Montpellier, CIRAD, INRAE, Montpellier, France

ABSTRACT

Spatial information extraction from textual documents and its accurate geo-referencing are important steps in epidemiology, with many applications such as outbreak detection and disease surveillance and control. However, inaccuracy in extraction of such geospatial information will result into inaccurate location identification, which in consequence may produce erroneous information for outbreak investigation and disease surveillance. One of the problems is the extraction of geospatial relations associated with spatial entities in the text documents. In order to identify such geospatial relations, we categorized them into three major relations: 1) Level-1, e.g. center, north, south; 2) Level-2, e.g. nearby, border; 3) Level-3, e.g. distance from spatial entities e.g. 30 km, 20 miles, 100 m, etc., respectively. This work introduces a novel approach for extracting and georeferencing spatial information from textual documents for accurate identification of geospatial relations associated with spatial entities to enhance outbreak monitoring and disease surveillance. We propose a two-step methodology: (i) Extraction of geospatial relations associated with spatial entities, using a clause-based approach, and (ii) Geo-referencing of geospatial relations associated with spatial entities in order to identify the polygon regions, using a custom algorithm to slice or derive the geospatial relation regions from the place name and their geospatial relations. The first step is evaluated with a disease news article dataset consisting of event information and obtaining a precision of 0.9, recall of 0.88 and F-Score of 0.88 respectively. The second step entails using a qualitative evaluation of shapes by end-users. Promising results are obtained for the experiments in second step.

ARTICLE HISTORY
Received 23 November 2022
Accepted 7 August 2023

KEYWORDS
Natural language processing; geospatial information; geospatial relations; geo-tagger; geo-parser; geo-referencing

1. Introduction

In recent years, geospatial information (i.e. position of locations on the earth's surface, for instance "Helsinki, N 60°10'10" - E 24°56'08"") recognition from textual data has gained more attention in the natural language processing (NLP) field. The importance and relevance of the work can be strengthened by highlighting the potential impact and benefits of accurate geospatial information extraction in epidemiology, outbreak detection, and disease surveillance. For instance, an outbreak in "the center of Paris" is different from an outbreak in "southern Paris". Additionally, the potential applications of geospatial information extraction in other fields, such as health care, stock markets, and e-learning, can be briefly discussed to further emphasize the broad impact and significance of the work (Hassani et al., 2020). Furthermore, it may be useful to provide some context or background information on the limitations and challenges of existing methods for geospatial information extraction, highlighting the need for more efficient and accurate algorithms. This approach underscores the novelty and potential contribution of the proposed method. For example, a possible research question is: "Can we develop an efficient and accurate algorithm for extracting spatial relations entities from textual data and for transforming them into valid geospatial representations, specifically for the purpose of disease outbreak identification and surveillance"? The geospatial information can be expressed in the textual documents in both simple and complex ways, depending on the syntax and semantic of expression. This geospatial information is available in the form of absolute geospatial information (precise location names, e.g. Milan) and relative geospatial information which is also known as spatial relation entity, i.e. spatRE (geospatial relations associated with the location name, e.g. North Milan).

CONTACT Mathieu Roche mathieu.roche@cirad.fr
Supplemental data for this article can be accessed online at https://doi.org/10.1080/15230406.2023.2264753.

In order to extract geospatial information, Named Entity Recognition (NER) is an important task in Natural Language Processing (NLP) that results into key information extraction from text (Berragan et al., 2022; Mohit, 2014). In broader context, these named entities are categorized in different types, i.e. person, location, organization and date and time, respectively (Nadeau & Sekine, 2007). In this work, we focus on the extraction of special cases of named entities in the text are available in the form of locations associated with geospatial relations. These kinds of spatial entities are available in the form of geospatial relations with spatial entities, e.g. North Milan, North-east Paris, border of south France, etc. However, there are many challenges in the text representation of these geospatial relations associated with locations. These meaningful relations are important information in the perspectives of different applications to identify the correct geographical zones.

The process involves two key steps: extracting geospatial information from textual documents and translating that information into geographical coordinates. To extract geospatial information, we use natural language processing (NLP) techniques, such as named entity recognition (NER), combined with a set of rules. We then proposed a customized algorithm to compute the geographical coordinates of the extracted geospatial information. These geographical coordinates can be visualized on geographical maps in the form of shapes such as points, lines, polygons, or multi-polygons. In some literature, this whole process of extraction of geospatial information and its transformation into geographical coordinates are also known as "location estimation" (Middleton et al., 2018).

In this paper, we particularly investigate the extraction of spatial relations entities from text and its transformation into geographical coordinates. The geographical transformations produce in shapes/geometries that are compatible with a standard geographical information system (GIS). In our work, we mainly defined geospatial relations at different hierarchies or levels. These levels are defined depending on the type of geospatial relations associated with location information. These geospatial relations associated with location can be defined in the form of relative position to the location, either by slicing the location or deriving from location. For instance, geographical coordinates of "North Milan" are obtained by slicing the north part from Milan Polygon. Secondly, the coordinates of "Milan border" are obtained, deriving an exterior polygon from Milan polygon. To sum up, we propose a clause-based spanned entity extraction algorithm to extract geospatial relations (Level-1, Level-2, Level-3) associated with spatial entities. The methodology of the extraction of spatial relation entities (spatRE) is further evaluated with a news article dataset that is related to infectious disease outbreaks to extract spatREs. The second step is the transformation of textual geospatial information into a valid representation in the form of a shape described by geographical coordinates. To evaluate the geographical coordinates in the form of geometries/shapes, a sample of geospatial relation shapes for some cities are generated. A qualitative evaluation is performed to evaluate the shapes by groups of end-users in order to validate the methodology.

The remaining structure of the paper is as follows: Section 2 describes the previous work related to spatREs. Section 3 explains the spatial relation of entities and the related concepts. Section 4 presents the proposed methodology and the different steps involved in our process pipeline. Section 5 presents the datasets that are used to validate the methodology. Section 5 describes the experiments and the associated results. Section 6 highlights the critical discussion on results, direction to new research and limitations in the proposed work. Whereas, Section 7 summarizes the contribution and outlines future work.

2. Related work

Geospatial information is important in the context of disease surveillance, disaster management practitioners, and many other domains (Zeng et al., 2021). These geospatial information are sometimes available in the text directly and sometimes in the form of geospatial relations (spatRE) associated with locations (Haris et al., 2020). spatRE extraction and its geocoding are important and challenging in the context of such location sensitive systems. Different researchers have proposed various approaches to extract geospatial information from text, i.e. machine learning, geographical databases, ontology-based reasoning and rule-based approaches that include dictionary/lexicons and transformer-based language models (Alonso Casero, 2021; Kokla & Guilbert, 2020). However, Geocoding of such spatRE is challenging and tricky to find the regions of extracted spatREs that are ambiguous in the context.

McDonough et al. (2019) proposed a rule-based named-entity recognition to resolve special cases of spatial named entities in text and validated it with the historical corpora. However, the proposed approach did not address the complex relationship that involves other linguistic features, i.e. part-of-speech (POS), dependency parsing, word vectors, etc. Chen et al. (2017) proposed a best-matched approach to extract geospatial relations that are

referred to anchor places, gazetteered places, and non-gazetteered places. However, it is not defined in the coordinate system to be represented in geographical systems. Zhang et al. (2009) proposed a rule-based approach for geospatial relation extraction based on geographical named entity recognition technology and a spatial relation-annotation corpus. The rules are just limited to the specified corpus and syntactic patterns that described the geospatial relations. Zheng et al. (2022) proposed a knowledge-based system (GeoKG) that described geographic concepts, entities, and their relations which is used for geological problem solution and their decision-making. The solution is only limited to the geological domain that contains information about geographical events, geographical relationships and concepts.

Medad et al. (2020) proposed an approach that is the combination of transfer learning and supervised learning algorithm for the identification of spatial nominal entities. However, the scope of the work was limited to the spatial entities without proper nouns e.g. conferences, bridge at the west, summit, etc. Wu et al. (2022) proposed deep learning models i.e. CasREL and PURE in order to extract geospatial relations in the text. The proposed models were validated with two main approach's pipeline approach (spatial entities and relations were dealt separately) and joint approach. The quantitative results demonstrated that pipeline approach performed better than joint approach using deep learning models. Syed et al. (2022) proposed a rule-based approach to extract spatREs from text and a custom algorithm in order to identify the coordinates of such spatREs. The relations addressed in the proposed approach are cardinal relations, i.e. "North," "South", and geospatial keywords, i.e. "nearby," "close," etc. However, the proposed approach did not address complex and compound geospatial relations and also need improvements in identifying correct coordinates for such relation shapes.

After thorough analysis of numerous research studies, extraction of spatREs is addressed in different perspectives with different approaches. However, there are some limitations in some cases and in some studies it is associated with spatial nominal entities which doesn't belong to the geographical coordinates systems. In the above studies, there is no such approach or technique to extract spatREs from textual documents. There are two main challenges to address: 1) the extraction of spatREs from the textual documents and 2) an algorithm to compute the geometry/coordinates of the spatREs to approximate the shape that is compatible with state-of-the-art (sota) GIS systems.

3. Spatial relation entities

Geospatial relations are the spatial keywords that are associated with the place names in the text. These geospatial relations associated with place names are pointing toward region that are in referenced to that place name instead of region of place name. A few examples of spatREs are "southern Paris" and "Paris border." The state-of-the-art named-entity recognition systems are unable to identify the geospatial relations, i.e. border, south, ... that results into identification of incorrect regions. These geospatial relations associated with place names are known as **Spatial Relations Entities (spatRE)**. In linguistic, spatRE are represented in the grammar as follows:

```
spatRE <- [ADVERB][NOUN] [VERB] PROPN [NOUN]

# PROPN should be a place name, which is mandatory
# At least the left or right part of PROPN is mandatory
# NOUN and ADVERB are spatial keywords
```

Generally, *spartRE* can be defined using a set of regular expressions as follows:

$$spatRE1 = [spat\_relation\_kwd]^{+}[\texttt{place\_name}]$$
$$spatRE2 = [\texttt{place\_name}][spat\_relation\_kwd]+$$

geospatial relations can be organized into hierarchies based on the computation of geographical coordinates. For instance, the polygon of spatRE with Level-1 and Level-2 geospatial relations associated with location (North Paris border) is computed in a hierarchical way. Initially, we compute the polygon of Level-1 (North Paris) and then derive the Level-2 polygon from the Level-1 polygon. These geospatial relations are defined in the subsequent sections as follows:

3.1. Level-1 spatial relations

Level-1 spatial relations are the "*cardinal relation*" and exceptional "*center*" relation associated with *place name* in the text documents. These cardinal relations are directional relation, i.e. North, South, East, West, North-East, North-West, South-East, South-West respectively. Moreover, Level-1 spatial relations can also be available in the form of its synonyms associated with place name. A few examples of spatREs with Level-1 spatial relations are "Northern Milan", "Southern Paris", "Central London", etc. In linguistic, Level-1 spatRE is represented in the grammar as follows:

```
level1_spatRE <- [ADVERB] [Noun] [VERB] PROPN [ADVERB]

# PROPN should be a place name, which is mandatory
# At least the left or right part of PROPN is mandatory
# ADVERB and NOUN are Level-1 keywords
```

The geographical coordinates representation of *Level-1 spatRE* on geographical information systems. The geographical representation of *Level-1 spatRE* example is i.e. Paris with *Level-1 spatRE* as shown in Figure 1.

3.2. Level-2 spatial relations

Level-2 spatial relations are the spatial relation based on keywords dealing with proximity aspects, i.e. border, near, close, proximity and their synonyms associated with the place name. These spatial relations are important in the perspective of sensitive geographical information systems in order to point to the correct geographical zone. Some common examples of *Level-2 spatial relations* are "Milan border," "proximity of Lyon," "around Bordeaux," etc. In linguistic, Level-2 spatRE are represented in the grammar as follows:

```
level2_spatRE <- [NOUN] [ADVERB] [VERB] PROPN [NOUN]

# PROPN should be a place name, which is mandatory
# At least the left or right part of PROPN is mandatory
# NOUN and ADVERB are Level-2 keywords
```

The geographical coordinates of level-2 spatial relations are not as much straight forward. The graphical representation of *Level-2 spatial relations* example "Paris border" is shown in Figure 2.

3.3. Level-3 spatial relations

Level-3 spatial relations are considered as distance associated with place names. These distance keywords are compound keywords made of distance unit and number. It can be represented in different ways such as "2 km from", "2 km away", "2 km radius of", etc. Moreover, the unit of distance can be available in the text in full text form as well as in form of unit abbreviations e.g. km, mi, ft, etc. Two examples of spatREs with Level-1 spatial relations are "2 km distance from Paris", and "radius of 3 miles from Paris" etc. In linguistic, Level-3 spatRE are represented in the grammar as follows:

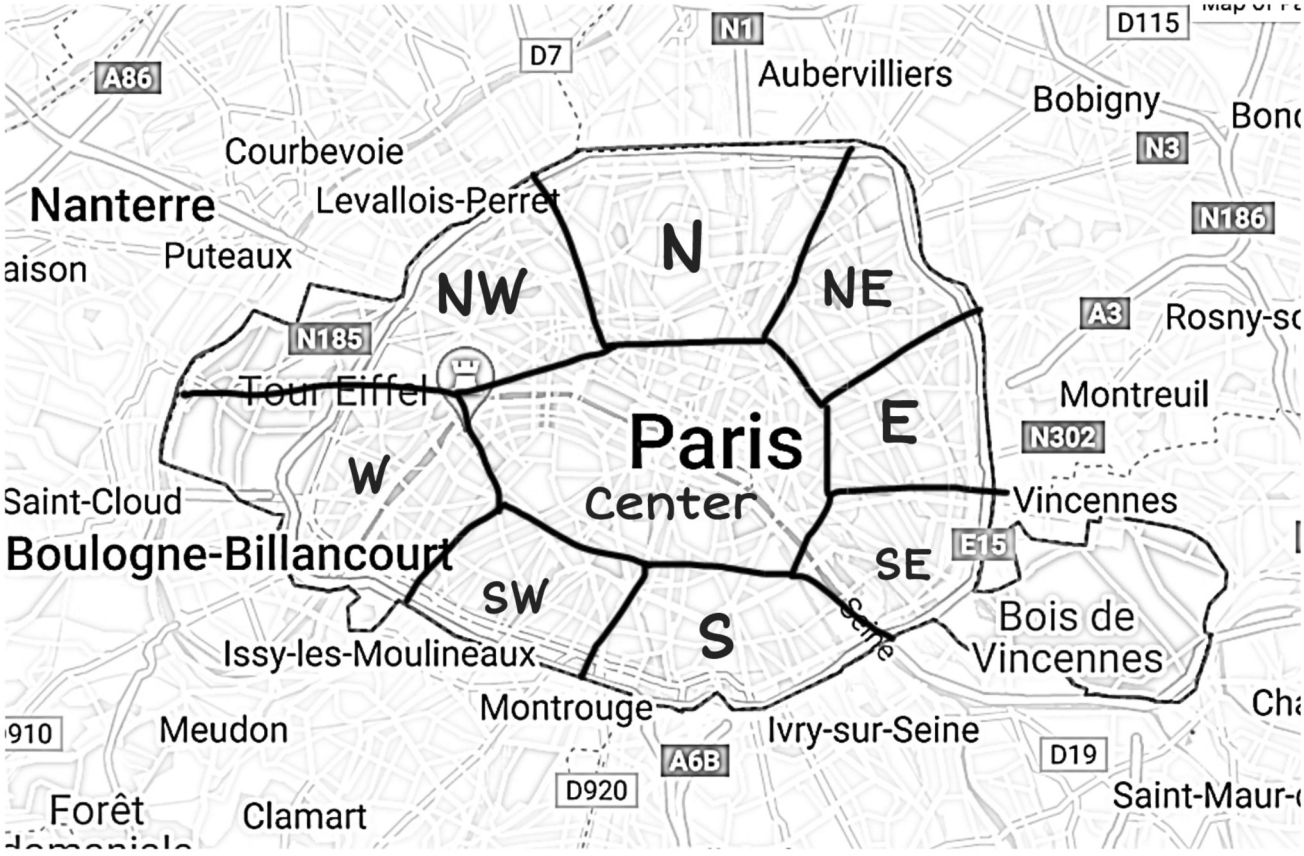

Figure 1. Level-1 spatRE.

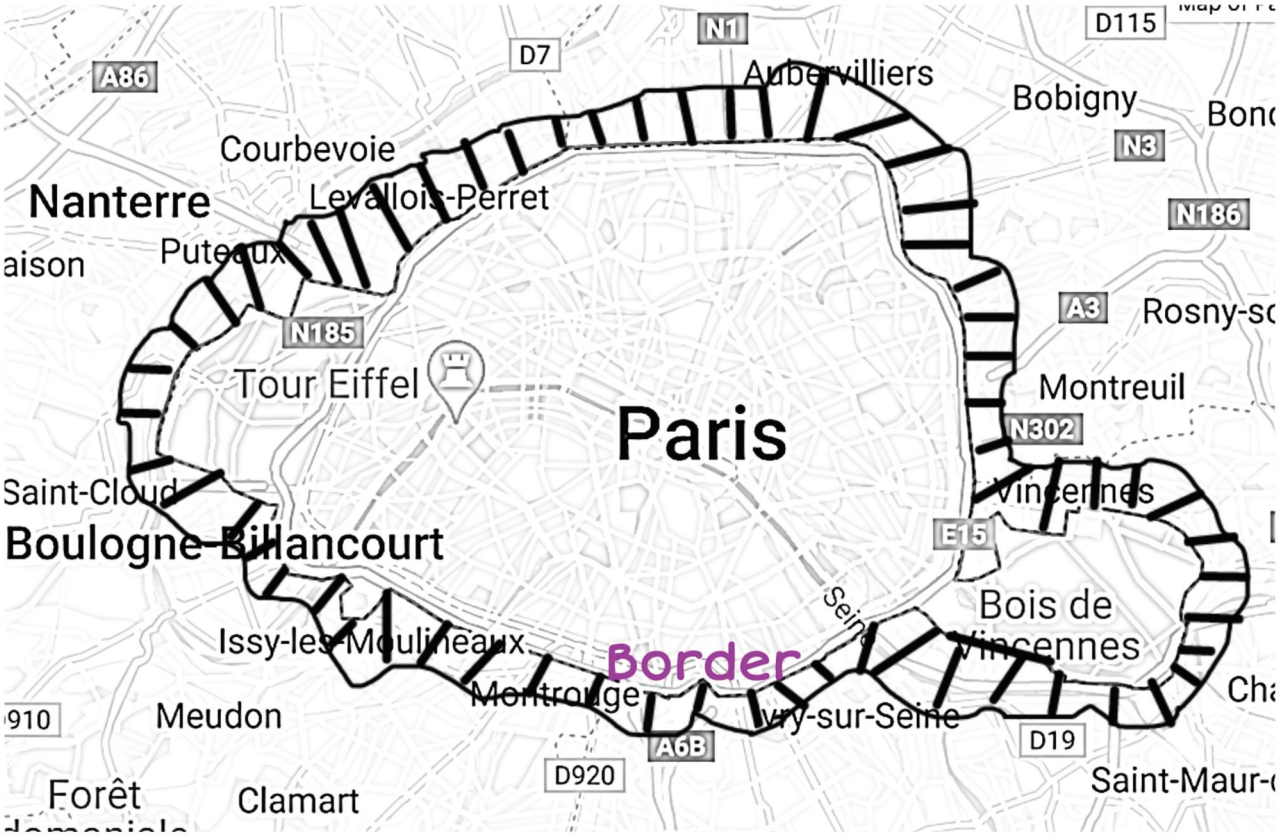

Figure 2. Level-2 spatRE.

```
level3_spatRE <- [NUM] [NOUN] [VERB] + PROPN

# PROPN should be place Name which is mandatory
# At least left of PROPN is mandatory
# NOUN is a Level-3 keywords (distance unit)
# NUM is a number
```

The geographical coordinates of Level-3 spatial relations can be available at distance from the original place names. The graphical representation of *Level-3 spatial relations* example "1 km from Paris" is shown in Figure 3.

3.4. Compound spatial relations

Compound spatial relations are also available in the text in the form of combination of Level-1, Level-2, and Level-3 spatial relations associated with the place name. These spatREs are defined with mixed spatial relations applied to place names. Some examples of such *compound spatial relations* are "1 km from north Paris border", "6 miles away from South Lyon", etc. The geographical representation of *compound spatial relations*, for example "1 km from different Level-1 relation of Paris" are shown in Figure 4.

In order to extract such geospatial relations and identify its geographical representation, a two-step methodology is proposed that is described in subsequent section.

4. Proposed methodology

In order to deal with such geospatial relations in the textual documents, our proposed methodology is divided into two main phases: 1) Extraction phase 2) Geocoding phase respectively. Extraction phase extracts spatREs from the textual documents, while Geocoding phase translates the spatREs into geographical coordinates compatible to GIS applications. The process workflow of the proposed methodology is shown in Figure 5.

The details of the two phases of the proposed methodology are explained in the subsequent sections.

4.1. Extraction phase

In the first phase, spatREs are extracted from the text data. For this, some language processing libraries or tools can be utilized to extract linguistic information. In our case, we chose state-of-the-art natural language processing library (NLP) *spaCy* (Honnibal & Montani, 2017) for python. *spaCy* is a Python library for natural

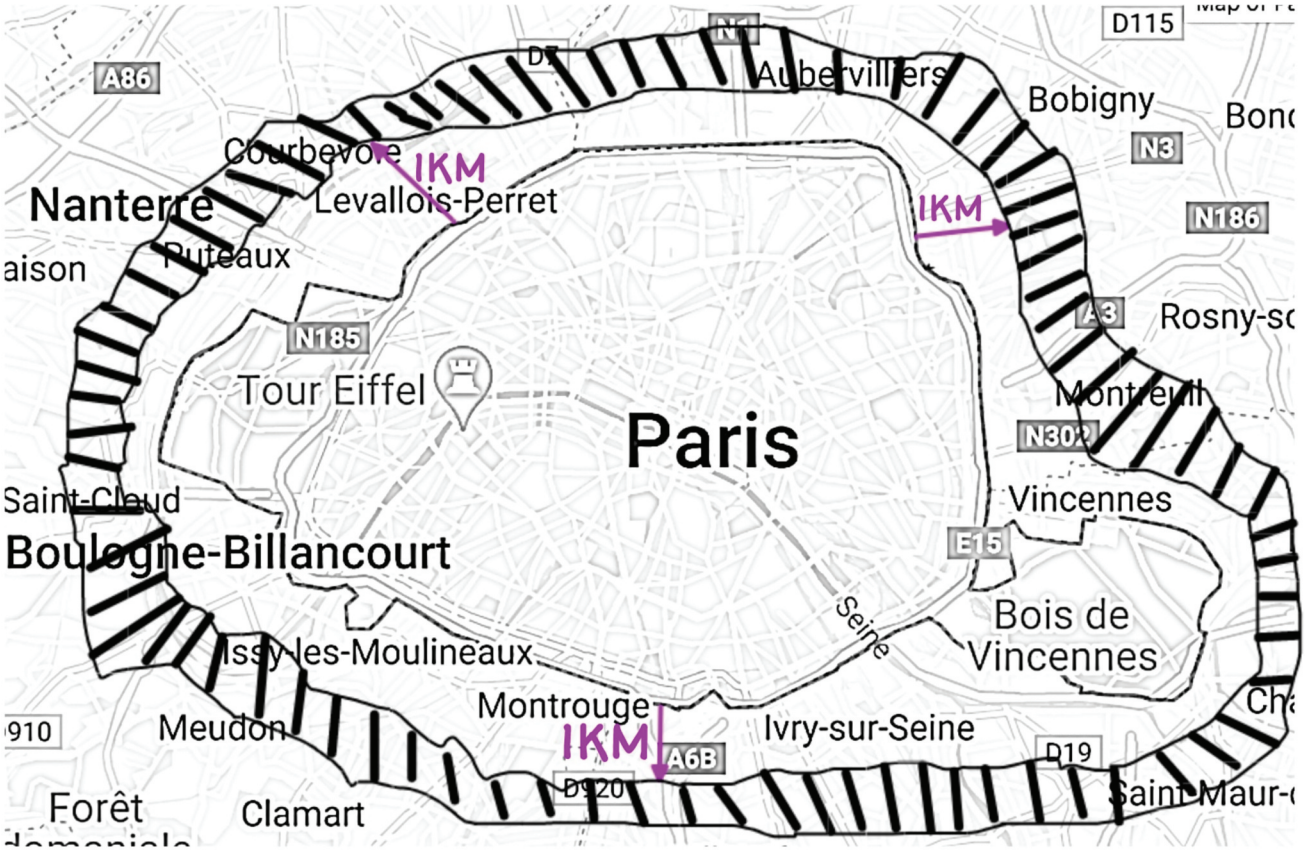

Figure 3. Level-3 spatRE.

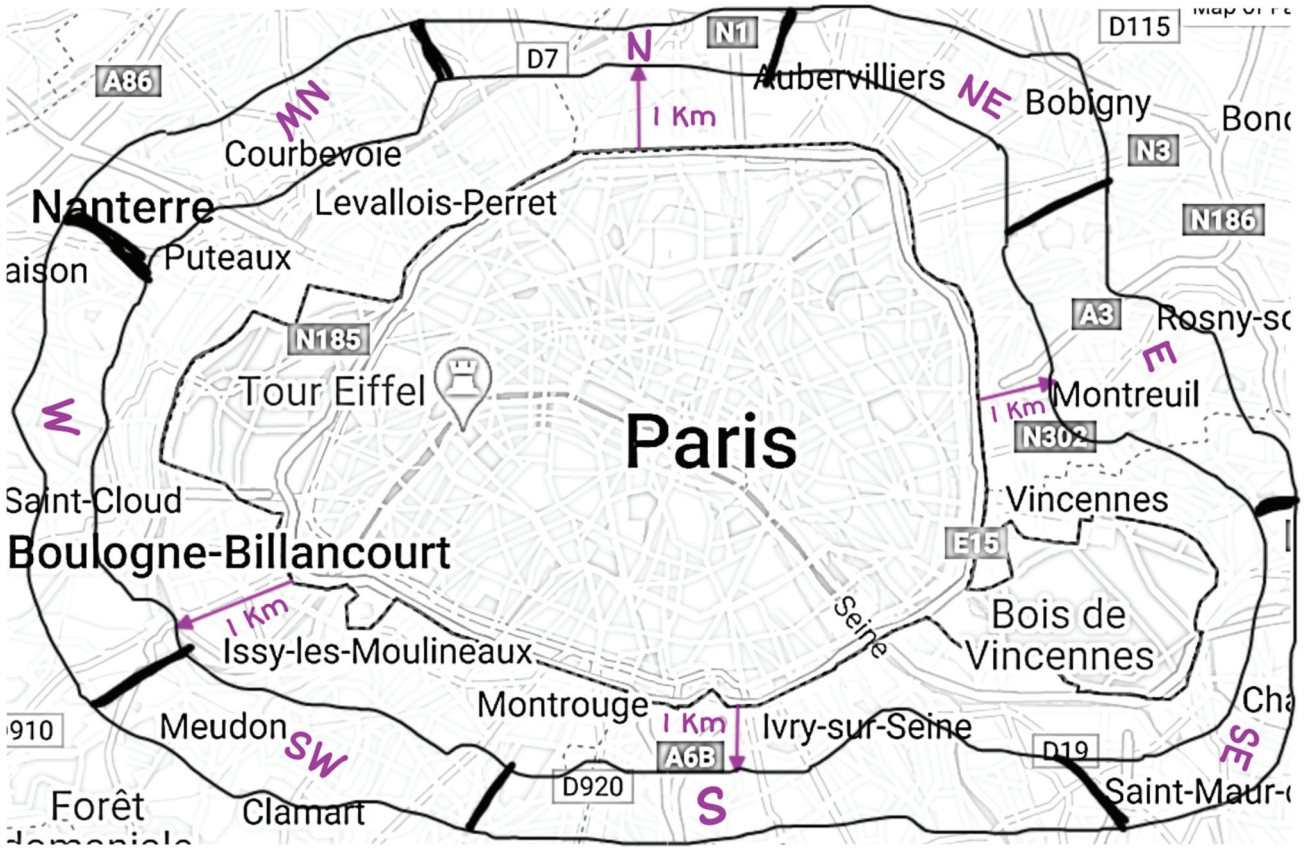

Figure 4. Compound spatRE.

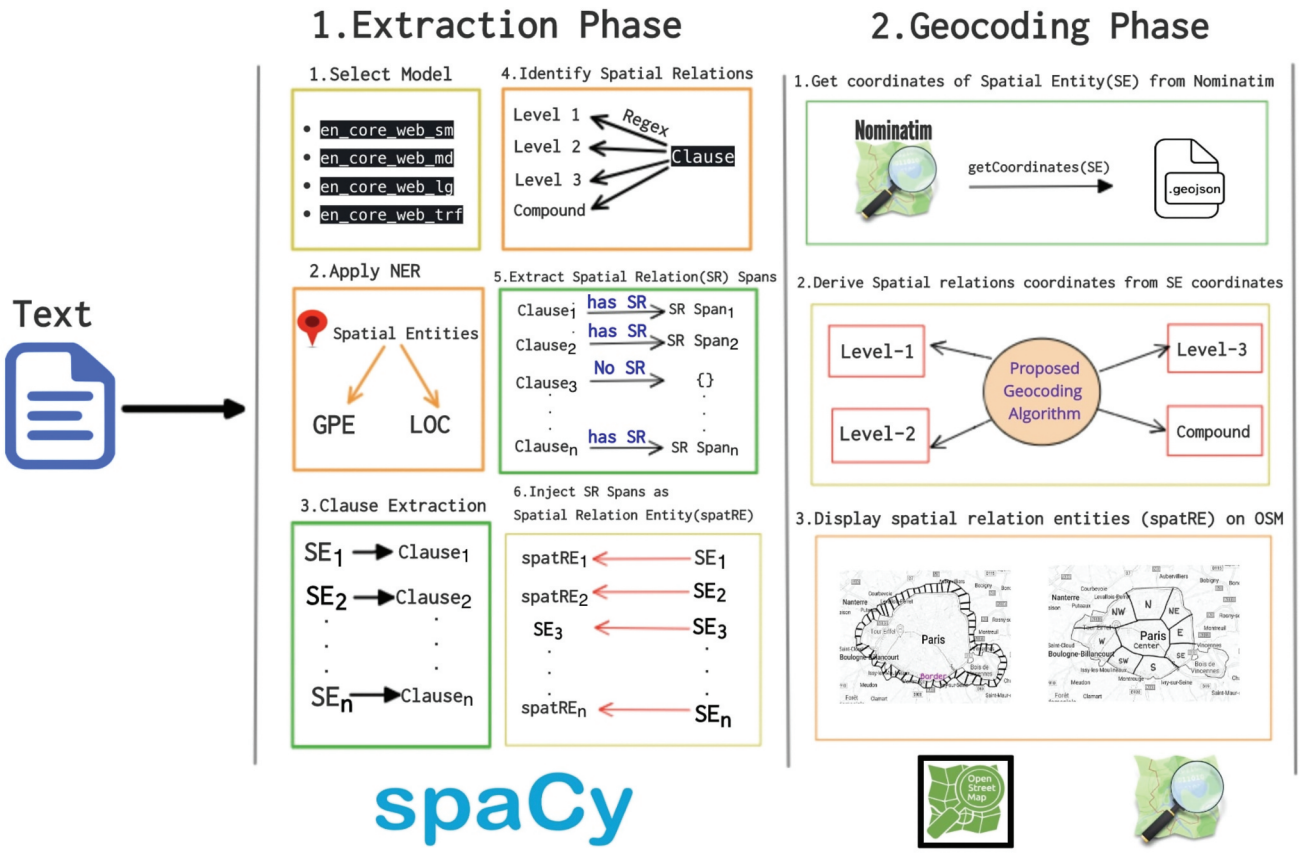

Figure 5. spatRE pipeline.

language processing tasks, i.e. named-entity recognition (NER), POS tagging, dependency parsing, word vectors, etc. This tool has a good behavior for NER tasks with other well-known tools (Vajjala & Balasubramaniam, 2022). In extraction phase, we used *spaCy* for NER task with the help of pre-built *spaCy* linguistic models.

4.1.1. Model selection

These spaCy linguistic models are en_core_web_sm, en_core_web_md, en_core_web_lg, and en_core_web_trf respectively. The en_core_web_trf model is computationally expensive compared to smaller models and requires significant computational resources to run. However, its high performance and accuracy make it a popular choice for a wide range of NLP applications. Once the model for *spaCy* is selected, the environment is setup for the NLP tasks such NER, POS tagging, etc.

4.1.2. Apply NER

The next step in extraction phase is to apply NER on the textual data. For the geospatial information, we only need to extract spatial entities from the textual data with the labels "GPE" (Geopolitical entities, e.g. Paris) and "LOC" (physical location, e.g. Safari Desert) respectively. As a result, it will identify the spatial named entity e.g. Paris, Lyon, France, Italy, etc. Once these spatial entities are extracted from the text, the next step is to identify the geospatial relations that are associated with such entities. To identify the geospatial relations associated with the spatial named entities, we need to extract the clauses that contain the spatial entities.

4.1.3. Clause extraction

With *spaCy*, for each spatial entity you can get its sentence from which the entity belongs by providing its start offset and end offset. However, in our case we want to get its clause from which it is associated. In linguistics, technically a clause can have one spatial entity at a time. In the proposed work, we applied the rules for the clauses that are separated by conjunctions known as compound clauses. These clauses are normally separated by conjunction symbols i.e. ",", ";", ":", etc., and conjunction keywords such as "and," "but," "or," "nor," "for" and "yet." In order to extract the clause of each spatial entity, every clause is separated by the conjunctions or with some keywords i.e. and, or, etc. In our algorithm, we split the sentence into clauses and

save the clause that contains the spatial entity and ignore the rest of the clauses in the sentence. Here is the example of the clause extraction as follows:

Text: Significant number of COVID-19 cases are reported close to Paris border, whereas other areas in the region are safe due to preventive safety measures.
Clause 1: Significant number of COVID-19 cases are reported close to Paris border
Clause 2: whereas other areas in the region are safe due to preventive safety measures.

The above text which is a sentence is divided into two clauses by having conjunction to separate it. The first clause is considered as it contains the spatial entity however, the second clause is ignored because of not having spatial entity. Once the clause having spatial entities is extracted from the text (news articles), the next step is to identify geospatial relations in the clauses.

4.1.4. Geospatial relations identification

In previous Section 3, we defined different hierarchies of geospatial relations. We need to extract such geospatial relations from the candidate clauses. Candidate clauses are identified in the text document as the clauses that contain spatial entities. The next step is to determine whether there are any geospatial relations associated with these entities in the candidate clauses. In order to extract geospatial relations in the clauses, we defined regular expressions for Level-1, Level-2, Level-3 spatial relations. The regular expressions of these geospatial relations are defined using *Python* regex *re* with the help of external library *quantities*. *quantities* library is used to get the different quantity units, its abbreviations and their inter-conversions. The purpose of this library is to identify the distance measurement units in the text and the interconversion of units, e.g. km to miles and miles to ft. The library also provides different representations of distance units, e.g. km, kilometers, miles and mi.

If these spatial relations are identified in the clause that contained the spatial entity, then we adjust the span offset according to the geospatial relation. The span offset is either adjusted from the end or in the start according to the occurrence of geospatial relation relative to spatial entity. In the following examples "north of Lyon", "Paris border"; "north of Lyon" has the geospatial relation before spatial entity i.e. "Lyon". Contrary to first example, "Paris border" has the geospatial relation after the spatial entity i.e. "Paris". In some cases, the occurrence of the geospatial relations exists in both ways. In the following examples, "2 km away from Paris border", "south Lyon border" have the geospatial relations before and after the spatial entities. "2 km away from Paris border" has the geospatial relations "2 km" and "border" with spatial entity i.e. "Paris", "south Lyon border" have the geospatial relations "south" and "border" with spatial entity i.e. "Lyon" respectively. The start offset or end offset of the spans depending on geospatial relations start offset or end offset in the text document. Once the offset of the spans are known, the next steps are to create the spans for geospatial relations along with spatial entities.

4.1.5. Geospatial relations spans extraction

Once the spans are identified in the whole text document. The next is to save each span into the span list. In the below example, we identified the three main clauses in the whole text document. The two of them are considered with having spatial entities and the third one is discarded. In the two clauses, we identified the geospatial relations inside it and later on identify the span having spatial_relation + spatial_entity as shown in the below example.

Text: Significant number of COVID-19 cases are reported close to Paris border, north of Lyon whereas other areas in the closer region are safe due to preventive safety measures.
Clause 1: Significant number of COVID-19 cases are reported close to Paris border
Clause 2: north of Lyon
Clause 3: whereas other areas in the region are safe due to preventive safety measures.
Spatial Entity 1: Paris **Spatial Entity 2**: Lyon
Spatial Entity Span 1: Paris border **Spatial Entity Span 2**: north of Lyon

Once the span list is obtained, then the next step is to inject these spans as entities in the default *spaCy* NER pipeline.

4.1.6. Spatial relations entities injection

After the spans identification, the next operation is to replace the default spatial entities in the "DOC" (the element that contains linguistic feature information, e.g. NER, spans, and POS.) element by spatRE which are identified in the geospatial relation spans. The label of the spatREs are injected in the "DOC" element as "spatRE." Table 1 shows the example of the default

Table 1. Spatial relation entities (spatRE).

Text	Before Extraction Phase DOC (Element)	After Extraction Phase DOC (Element)
Significant number of COVID-19 cases are reported close to Paris border, north of Lyon whereas other areas in the closer region are safe due to preventive safety measures.	Significant number of COVID-19 cases are reported close to **Paris(GPE)** border, north of **Lyon(GPE)** whereas other areas in the closer region are safe due to preventive safety measures.	Significant number of COVID-19 cases are reported close to **Paris border(spatRE)**, **north of Lyon(spatRE)** whereas other areas in the closer region are safe due to preventive safety measures.

Algorithm 1 Procedure/Pipeline to extract spatREs

Input: spaCy linguistic Element (doc)
Output: Spatial relation Entities (spatRE)

$spatEnt \leftarrow []$ ▷ To store spatial Entities
for each $ent \in doc.ents$, **if** ent is 'GPE' **or** ent is 'LOC' **do**
    $spatEnt.append(ent)$
end for
$spatRE \leftarrow []$ ▷ To store spatial relation Entities(spatRE)
for each $ent \in spatEnt$ **do**
    $relEntity = getLevel1(ent, clause(ent))$ ▷ Get Level-1 relation from clause
    $relEntity = getLevel2(ent, clause(ent))$ ▷ Get Level-2 relation from clause
    $relEntity = getLevel3(ent, clause(ent))$ ▷ Get Level-3 relation from clause
    if $relEntity$ is not None **then**
        $spatRE.append(relEntity)$
    end if
end for
$doc.ents \leftarrow spatRE$ ▷ Inject 'spatRE' in default element 'doc'
return doc

NER extraction result and after incorporation of extraction phase result.

Algorithm 2 shows the detail pseudocode of the *Extraction Phase* with details of each step explained in this Section 4.1.

4.2. Geocoding phase

After the extraction phase, the next is *geocoding phase*, which is to translate the *spatRE* into geographical coordinates. In this phase, the translation of geographical coordinates is derived either by slicing the polygon or by deriving using geospatial operations. In order to identify the coordinates of *spatRE*, we need to get the coordinates the of the spatial entity exclusive of geospatial relations.

4.2.1. Acquire coordinates from Nominatim

Nominatim API (Clemens, 2015) provides search by place name, feature description or free text search in OpenStreetMap (OpenStreetMap contributors, 2017) database and return its geographical coordinates based on search queries. The API provides the *GeoJSON* which contains the geometry along with their feature attributes. The geometry is further used to determine the geometry of the spatial relations mentioned in the spatRE as shown in Figure 4.

4.2.2. Derive/slice geospatial relation coordinates

After getting the coordinates of the place name mentioned in the spatRE, the next step is to derive the coordinates of the geospatial relations associated with the place name. This can be done depending on the type of geospatial relation. Level-1 spatial relation coordinates are acquired by slicing the main geometry of the place into **9** spatial relations geometries. For instance, The Level-1 Slicing of Paris can be sliced into 9 geographical shapes: "Northern Paris", "Southern Paris", "Eastern Paris", "Western Paris", "North-east Paris", "South-east Paris", "North-west Paris", "South-west Paris" and "Central Paris", as shown in Figure 1. In contrast to Level-1 relations, Level-2 and Level-3 relations are derived by applying geospatial operations i.e. geospatial joins, geospatial unions, intersections with the help of *GeoPandas* (Jordahl et al., 2020) and *Shapely* (Gillies et al., 2007) python libraries. Once the geospatial coordinates of the geospatial relations are extracted for the spatRE, it is converted into compatible *GeoJSON* format which can be utilized using any Geographical Information System (GIS) applications.

4.3. Visualization of spatRE

The geographical coordinates of spatRE are in the form of GeoJSON can be visualized using OpenStreetMap (OpenStreetMap contributors, 2017) leaflet. The

Algorithm 2 Algorithm for Extraction Phase

Input: text
Output: Spatial relation Entities (spatRE)

$model \leftarrow load(en\_core\_web\_trf)$ ▷ Load spaCy model
$model.add\_pipe(spatial\_relation\_pipeline)$ ▷ Calling Algorithm 1
$doc \leftarrow model(text)$
$spatRE \leftarrow doc.ents$
$displacy(spatRE)$

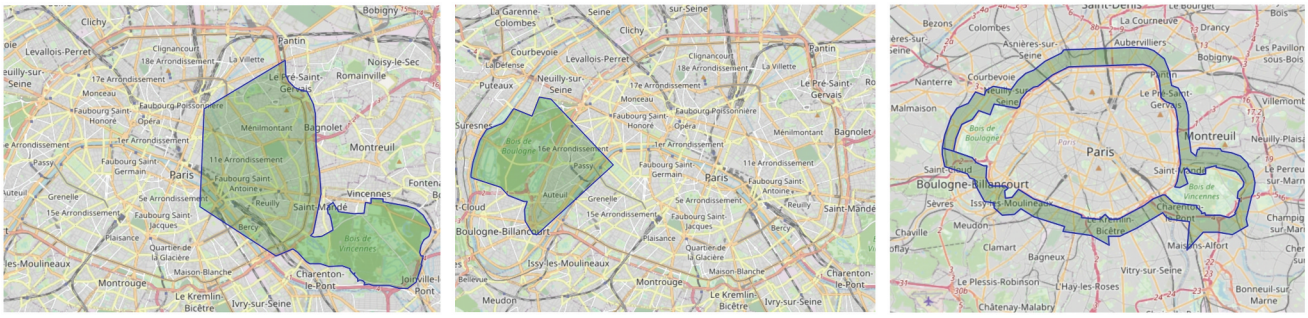

Figure 6. Level-1 & Level-2 spatial relations.

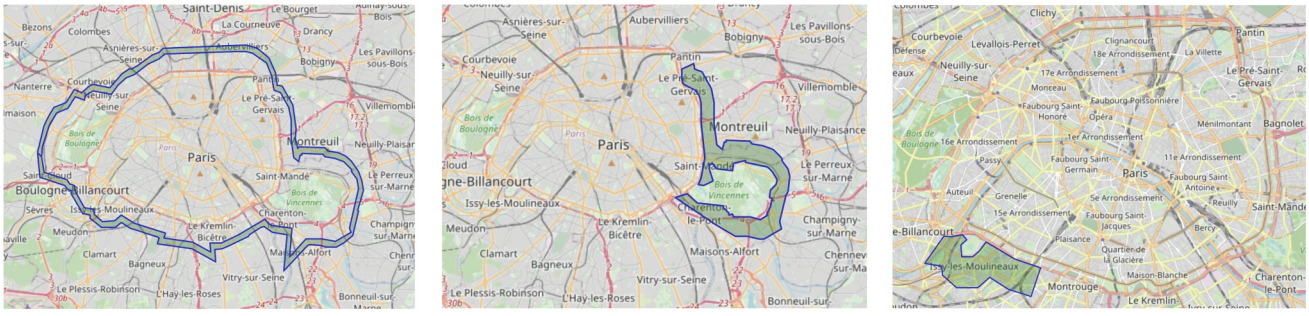

Figure 7. Level-3 & compound spatial relations.

OpenStreetMap leaflet is produced using the *Folium* Python library that is built on top of the Leaflet JavaScript library. The leaflet allows Python developers to create interactive maps using Leaflet.js maps in Python. This visualization can help you to visualize the regions for spatRE. Moreover, in the next subsequent section, it is used to evaluate the shapes of the spatRE. Some of the examples spatRE polygons visualization using OpenStreetMap (OpenStreetMap contribu- tors, 2017) Leaflet are shown in Figures 6 and 7.

The overall methodology of the proposed work is based on two phases. So, the validation of each phase must be evaluated. The evaluation of both phases is described in the following section 5 of the article. The dataset for both phases is explained in detail in the Section 5.

5. Datasets

In order to validate the methodology of the proposed work, we use some specific datasets that are detailed in the next subsections.

5.1. Extraction phase dataset

The first phase is the *extraction phase*, which extracts the spatREs from the text. The spatRE extraction is evaluated with the disease dataset available at https://tinyurl.com/2wn7ywth. We build this dataset from the news extracted by PADI-web,[1] which is an event-based surveillance system related to animal health events. The dataset contains the news articles of different diseases i.e. 1) *AMR* 2) *COVID-19*, 3) *Avian-Influenza*, 4) *Lyme* and 5) *Tick-borne Encephalitis (TBE)*. The dataset is imbalanced in terms of number of news articles for each disease. We annotated each news article in the corpus by adding spatRE. Each news article in the dataset contains the information related to the disease outbreaks or other prevention measures for the disease related to locations.

The dataset contains a *CSV* file for each disease. Each row in the CSV file contains the following columns such as "id," "title," "text," "URL," "spatRE," "source_lang," and "created_at." From the perspective of result validation, the "text" and "spatRE" are important for the evaluation. Text is used as an input for the extraction phase, and the spatRE contains the spatRE, which are used for the comparison for evaluation.

5.2. Geocoding phase dataset

For the *Geocoding Phase*, we created our own dataset for the *9* famous cities of UK and Europe, i.e. Paris, London, Milan, Madrid, Zagreb, Utrecht, Delft, Lyon,

Algorithm 3 Procedure to extract Cardinal Coordinates

Input: coordinates, centroid, direction
Output: Cardinal coordinates

```
cardinal_coordinates ← []
if direction is north then
    cardinal_coordinates ← getCoordinates(north)  ▷ Coordinates having angle with centroid between 337° - 22°
end if
if direction is east then
    cardinal_coordinates ← getCoordinates(east)  ▷ 67° - 112°
end if
if direction is south then
    cardinal_coordinates ← getCoordinates(south)  ▷ 157° - 202°
end if
if direction is west then
    cardinal_coordinates ← getCoordinates(west)  ▷ 247° - 292°
end if
return cardinal_coordinates
```

and Florence, with having Level-1, Level-2, Level-3 and compound geospatial relations associated with these cities. For each city, we identified 19 geospatial relations, which are in the form of single relations and the combination of the geospatial relations. This geographical dataset contains the shapes of 19 spatial relation shapes for each city. A further qualitative analysis is performed by end-users to evaluate these derived shapes dataset (See Table S6 for the shape datasets).

In the current context, our work generates shapes of 9 cities having a combination of 19 geographical shapes (or more) for each city just to evaluate our methodology by validation by end users. The algorithm allows users to dynamically create shapes for the discussed spatial relations of any city. However, for the experiments, we evaluated 19 different shapes for each city.

6. Results

To validate the methodology, we used a specific dataset for each phase. The two main phases have different results, i.e. 1) spatRE extraction from text and 2) Identification of geographical coordinates of spatRE. The results for each phase are described as follows in the following subsections.

6.1. Extraction phase

Concerning the extraction of spatRE from text, we can evaluate through state-of-the-art evaluation mechanism. For that, we should have a standard Named-entity recognition (NER) gold standard corpus and then evaluate it through standardized evaluation scores. More precisely, the measure for evaluating NER is the F-Score which is the harmonic mean of Precision and Recall (Hakala & Pyysalo, 2019; Resnik & Lin, 2010). Precision, recall, and F-Score are defined as follows (Goutte & Gaussier, 2005):

$$Precision = \frac{Correct\ spatRE\ Recognized}{Total\ spatRE\ Recognized} \tag{1}$$

$$Recall = \frac{Correct\ spatRE\ Recognized}{Total\ spatRE\ in\ Corpus} \tag{2}$$

$$F - Score = 2 \times \frac{Precision \times Recall}{Precision\ +\ Recall} \tag{3}$$

The results are compared with the gold standard dataset to calculate the precision, recall and F-Score as shown in Table 2. For instance, the first row of Table 2 displays the results obtained from processing 25 articles that contain AMR events. In this case, 4 spatREs were extracted. The evaluation of these spatREs yielded

Table 2. Extraction phase results (spatRE extraction).

Disease Name	No. of Articles	spatRE Extracted	spatRE Actual	Precision	Recall	F-Score
Antimicrobial resistance (AMR)	25	4	5	1	0.80	0.88
COVID-19	100	100	92	0.87	0.94	0.90
Avian-Influenza	150	57	68	0.87	0.83	0.84
Lyme	29	10	10	0.83	1	0.90
Tick-borne Encephalitis (TBE)	73	73	81	0.93	0.83	0.87
Aggregate	377	244	256	**0.9**	**0.88**	**0.88**

Algorithm 4 Procedure to extract Ordinal Coordinates

Input: coordinates, centroid, direction
Output: Ordinal coordinates

$ordinal\_coordinates \leftarrow []$
if *direction* is **northeast then**
    $ordinal\_coordinates \leftarrow getCoordinates(northeast)$ ▷ 22° - 67°
end if
if *direction* is **southeast then**
    $ordinal\_coordinates \leftarrow getCoordinates(southeast)$ ▷ 112° - 157°
end if
if *direction* is **southwest then**
    $ordinal\_coordinates \leftarrow getCoordinates(southwest)$ ▷ 202° - 247°
end if
if *direction* is **northwest then**
    $ordinal\_coordinates \leftarrow getCoordinates(northwest)$ ▷ 292° - 337°
end if
if *direction* is **central then**
    $ordinal\_coordinates \leftarrow getCoordinates(central)$ ▷ Merge all cardinal extreme coordinates
end if
return *ordinal_coordinates*

precision, recall, and F-score performance measures of 1.0, 0.8, and 0.88, respectively.

Precision, recall and F-Score are measured for each disease dataset. The overall score for all the disease's dataset is calculated with precision of ***0.9***, recall of ***0.88*** and F-Score of ***0.88***. For the evaluation, we considered some cases of spatRE, e.g. "North America" and "South Korea" are considered as spatRE. However, in certain cases, we considered it a false positive in evaluation, such as spelling mistakes in spatRE, e.g. "(Yonhap) South Korea" or some other cases in which the words are concatenated with the spatRE and in such a way that it is a part of the same word with geospatial relations.

6.2. Geocoding phase

The results are generated for the geographical shapes mentioned cities discussed in earlier Section 5.2. Each shape of Level-1, Level-2 and Level-3 are generated for each spatial relation using the Algorithm 8. To validate the methodology, the shapes are generated for the different geographically located cities. There is no state-of-the-art mechanism to evaluate the geographical geometry of coordinates. Therefore, we applied a qualitative survey to evaluate each shape. A well-defined shape concerns the boundary of the polygon/shape, if it approximately reflects the same information of the spatial relation entity (spatRE). The criteria of the shape's evaluation are 1) how well the geometry of the shapes are represented, 2) how well the geometry shows the real visualization of geospatial relations of the associated city. The shapes that are being evaluated by the end-users who are involved in the GIS and geospatial information applications.

Each relation shape of all the chosen cities are evaluated by different end-users involved in the MOOD[2] project by ranking the shapes from ***1*** to ***4***.

1 is used for ***unclear***, and ***4*** is used for ***well-defined***. Similarly, ***2*** is used for ***weak*** or if the shape is defined "not bad", and ***3*** is used if the shape is "better" defined. Each shape for the same city is evaluated by two end-users to calculate the average score of both end-users. Moreover, for the city "Florence" it was evaluated by all the end-users of the evaluation group. The total score for each city for all the geospatial relations score are *152* except "Florence" with

Table 3. Qualitative evaluation of spatial relation by city.

City	Aggregate Score	Total Score	Accuracy(100%)	Mean(4)	Remarks
Paris	136	152	**89.5**	3.6	**Excellent**
London	142	152	**93.4**	3.7	**Excellent**
Milan	106	152	69.7	2.8	Good
Madrid	77	152	50.7	2	Weak
Zagreb	116	152	**76.3**	3.1	**Excellent**
Utrecht	105	152	69.1	2.8	Good
Delft	121	152	**79.6**	3.2	**Excellent**
Lyon	114	152	75	3	Good
Florence	477	608	**78.5**	3.1	**Excellent**

Table 4. Qualitative evaluation of Level-1 spatial relations.

	N	S	E	W	NE	NW	SE	SW	Central
Score	67	61	82	73	75	78	81	78	82
Total Score	96	96	96	96	96	96	96	96	96
Accuracy(100%)	69.8	63.5	**85.4**	**76**	**78.1**	**81.3**	**84.4**	**81.3**	**85.4**
Mean(4)	2.9	2.4	3.5	3.1	3.1	3.4	3.3	3.2	3.4
Remarks	Good	Good	**Excellent**	**Excellent**	**Excellent**	**Excellent**	**Excellent**	**Excellent**	**Excellent**

Table 5. Qualitative evaluation of Level-2 and Level-3 spatial relations.

	Border	NE Border	NW Border	SE Border	SW Border	N Border	S Border	E Border	W Border	2 Miles+Border
Score	87	65	63	72	77	59	61	78	71	84
Total Score	96	96	96	96	96	96	96	96	96	96
Accuracy(100%)	**90.6**	67.7	65.6	75	**80.2**	61.5	63.5	**81.3**	74	**87.5**
Mean(4)	3.6	2.7	2.7	3	3.3	2.6	2.5	3.3	2.9	3.5
Remarks	**Excellent**	Good	Good	Good	**Excellent**	Good	Good	**Excellent**	Good	**Excellent**

having total score of *608*. The aggregated score is the score that are marked by the participants. As depicted in the first row of Table 3, the aggregate score was computed for the 19 spatial relations associated with Paris, including N, S, E, and W, using evaluations provided by 2 end users. The cumulative score amounted to 152, while the aggregated score was 136. Following the evaluation process, the accuracy for determining the shapes of the spatial relations of Paris stood at 89.5%, with an average score of 3.6 out of 4, resulting in the designation of "Excellent" in the remarks. Table 3 shows the Aggregate Score marked by the group end-users for each city with having "Accuracy" and "Mean (4)" which is the basic unit of our evaluation done by the end-users and the "Remarks" about the geometries of all geospatial relations of the city. In Table 3, the shapes of geospatial relations are defined better for 8 out of 9 cities. However, it is observed that the geospatial relations for the city "Madrid" is not up to the mark with having anomalies in different geospatial relations.

Similarly, Tables 4 and 5 described the same matrices "Score", "Total Score", "Mean (4)", and "Remarks". However, in this case it is obtained in terms of each geospatial relation for all the cities. Table 4 described the results for Level-1 spatial relations. Similarly, Table 5 shows the results of Level-2 and Level-3 spatial relation results. In both tables, in general the results are average or above the average. More precisely, the results of Level-1 spatial relations are in greater extent better than Level-2 and Level-3 spatial relations in terms of "Score", "Accuracy" and their "Mean (4)".

7. Discussion

The proposed work is aimed at extracting geospatial relation entities from text data and at determining their geographical coordinates for visualization on geographical information systems (GISs). This two-step process involves the extraction of spatial relation entities followed by the geocoding of these entities. The outcomes of this research have significant implications in various domains, such as disease surveillance, disaster identification, and event surveillance systems, and in approximating the region of interest in identifying geospatial relations with locations from informal sources of information. For instance, we consider the following text from a digital news article: "One outbreak was noted in Podkarpackie province (east of Poland), one in a farm in south of Warsaw in Mazowieckie province and one in Lubuskie province in the west of Poland" .[3] The locations of the outbreaks in the text are "east of

Algorithm 5 Procedure to extract Level-1 Coordinates

Input: coordinates, centroid, direction
Output: Level-1 coordinates

$level1\_coordinates \leftarrow []$
$level1\_coordinates \leftarrow cardinals(coordinates, centroid, direction)$ ▷ Calling Algorithm 3
$level1\_coordinates \leftarrow ordinals(coordinates, centroid, direction)$ ▷ Calling Algorithm 4
if $level1\_coordinates$ is not **None then**
    return $level1\_coordinates$
end if
return $coordinates$

Algorithm 6 Procedure to extract Level-2 Coordinates

Input: coordinates, centroid, level2_keywords
Output: Level-2 coordinates

$level2\_coordinates \leftarrow []$
if $level2\_keywords$ **then**
  $polygon1 \leftarrow Polygon(coordinates)$
  $polygon2 \leftarrow polygon1.buffer(coefficient)$ ▷ **coefficient** is buffer of external polygon
  $level2\_coordinates \leftarrow polygon2.difference(polygon1))$
  return $level2\_coordinates$
end if
return $coordinates$

Poland" and "south of Warsaw" rather than "Poland" and "Warsaw" respectively. It is crucial to identify the accurate region of the outbreak to provide appropriate information to health officials in this context. The same situation holds for other geographically sensitive alert systems. By accurately identifying spatREs and their corresponding geographical coordinates, this research contributes to improving the accuracy and efficiency of such systems, the early detection of outbreaks, timely response to disasters, and effective surveillance of events in real time. This research presents several significant improvements and extensions to the work conducted by (Syed et al., 2022). These improvements involve the proposition of a new algorithm for extracting spatREs, as well as the identification of previously unaddressed geospatial relations. However, certain limitations in the current methodology that need to be addressed. For instance, in the extraction phase, North America, South Africa, South Asia, and South Korea are spatial entities. However, they were recognized as spatREs by our approach. In the geocoding phase, we evaluated the proposed algorithm by generating geographical coordinates for geospatial relations specific to several European and UK cities. However, this approach was limited to few cities, and future work should be aimed at enriching the geocoding dataset. During the evaluation, we observed some irregularities in the shapes of geospatial relations for specific cities, including Madrid. For instance, the algorithm produced geographical coordinates for "2 miles away from Madrid border." However, these coordinates are meaningless without a polygon. Similarly, some irregular polygons were generated by the algorithm, such as "near to the south of Madrid," "west Madrid border," and "vicinity of northwest Madrid" were generated by the algorithm. These irregular polygons need to be further investigated to improve the accuracy of the algorithm. Upon closer examination of the compound spatial relation, which involves a combination of Level-1 and Level-2 spatial relations such as Border with North, it has become apparent that further investigation is necessary to improve the accuracy of the results. Specifically, we need to focus on enhancing the border with Level-1 spatial relations. This enhancement may involve making adjustments to the boundaries to refine the output. Overall, the results for the polygon shapes of spatREs are promising, except for the shapes of Madrid. To improve the accuracy of the algorithm, further investigation is needed to address the limitations and irregularities identified in this research. The proposed research offers a unique perspective on interpreting shapes for different geospatial relations. For example, the notion of the "east Paris border" is perceived differently by people. Similar discrepancies in interpretation exist for directions such as north, west, and south. Currently, there is no standardized procedure for resolving these

Algorithm 7 Procedure to extract Level-3 Coordinates

Input: coordinates, centroid, level3_keywords, distance, unit
Output: Level-3 coordinates

$level3\_coordinates \leftarrow []$
if $level3\_keywords$ **then**
  $distanceKm \leftarrow convert(distance, unit)$
  $polygon1 \leftarrow Polygon(coordinates)$
  $polygon2 \leftarrow polygon1.buffer(coefficient * distanceKm)$ ▷ Multipy coefficient with distance in KM
  $level3\_coordinates \leftarrow polygon2.difference(polygon1))$
  return $level3\_coordinates$
end if
return $coordinates$

Algorithm 8 Algorithm for Geocoding Phase

Input: Spatial relation Entity (spatRE)
Output: coordinates
$place\_name \leftarrow getSpatialEntity(spatRE)$ ▷ Get GPE or LOC part
$coordinates \leftarrow getCoordinates(place\_name)$ ▷ Get coordinates from **Nominatim**
$coordinates \leftarrow getLevel1Coordinates(coordinates)$ ▷ Call Algorithm 5
$coordinates \leftarrow getLevel2Coordinates(coordinates)$ ▷ Call Algorithm 6
$coordinates \leftarrow getLevel3Coordinates(coordinates)$ ▷ Call Algorithm 7
if *output* is **GEOJSON then**
    return $geojson(coordinates)$
else if *output* is **MAP then**
    $display(coordinates)$ ▷ Display coordinates on OSM
end if

differences in interpretation. The proposed method seeks to fill this gap and can be applied to a variety of geographically sensitive applications.

8. Conclusion and future work

The proposed research focused on the extraction of spatREs from the text and the identification of the geographical shapes that represents those spatREs. We proposed a combination of NLP techniques to extract spatREs i.e. Level-1, Level-2, Level-3 and compound spatRE from the text documents. The results of the spatRE extraction from textual documents are evaluated with a news article dataset of infectious diseases that contained information about the disease outbreaks. The extraction results have the precision of *0.9*, recall of *0.88* and F-Score of *0.88*. In the second step, the shapes are generated for such spatRE in the form of GeoJSON that are compatible to standard GIS applications. A sample of 9 cities with 19 geospatial relation shapes were generated to evaluate it from a group of GIS application end-users. The qualitative evaluation of the shapes dates with the average score of *3.07* out of 4. However, in particular, the shapes for Madrid were not well-defined with the applied methodology.

A further step ahead is the incorporation of semantic and linguistic structure application on the sentence clauses in order to extract spatRE to avoid false positives. Moreover, another step is to ignore it as spatREs that are concrete geospatial information e.g. North America, South Korea, South Asia etc. In future work, a next step is that to create a set of rules for the "shape interpretation standardization." Shape interpretation are in terms of cardinal definition and keywords definition. In cardinal definition, e.g. north could be either north that include or exclude north-east and north-east. Another future perspective of shapes definition of different keywords geospatial relation e.g. neighborhood, adjacent, proximity, etc. Moreover, we will focus on the same work for other European languages in order to extract geospatial information from text. In future work, we will also assess the effectiveness of our approach in countries across Latin America and Africa, where traditional address systems may not be commonly used or have specific characteristics.

Notes

1. https://padi-web.cirad.fr/en/
2. https://mood-h2020.eu/
3. https://www.agroberichtenbuitenland.nl/actueel/nieuws/2020/08/05/significant-increase-in-asf-outbreaks-in-pig-farms-in-poland
4. https://github.com/mehtab-alam/GeospaCy
5. https://github.com/mehtab-alam/RSI_Disease_Dataset

Acknowledgments

This study was partially funded by EU grant 874850 MOOD and is catalogued as MOOD056. The contents of this publication are the sole responsibility of the authors and do not necessarily reflect the views of the European Commission.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The work was supported by the H2020 European Institute of Innovation and Technology [874850].

ORCID

Mehtab Alam Syed http://orcid.org/0000-0003-3696-0030
Elena Arsevska http://orcid.org/0000-0002-6693-2316
Mathieu Roche http://orcid.org/0000-0003-3272-8568
Maguelonne Teisseire http://orcid.org/0000-0001-9313-6414

Data availability statement

The code and data that support the findings of this study are openly available in GitHub repositories dedicated to *GeospatRE tool*[4] and the associated *dataset*.[5]

References

Alonso Casero, Á. (2021). *Named entity recognition and normalization in biomedical literature: A practical case in sars-cov-2 literature* [Doctoral dissertation]. ETSI_Informatica. https://oa.upm.es/67933/

Berragan, C., Singleton, A., Calafiore, A., & Morley, J. (2022). Transformer based named entity recognition for place name extraction from unstructured text. *International Journal of Geographical Information Science*, *37*(4), 747–766. https://doi.org/10.1080/13658816.2022.2133125

Chen, H., Vasardani, M., & Winter, S. (2017). Geo-referencing place from everyday natural language descriptions. *arXiv preprint arXiv:1710.03346*. https://doi.org/10.48550/arXiv.1710.03346

Clemens, K. (2015). Geocoding with openstreetmap data. *GEOProcessing*, *2015*, 10. https://www.researchgate.netprofileBruno-M-Menesespublication280575974_Water_Quality_Impact_Assessment_of_Land_Use_and_Land_Cover_Changes_A_dynamic_IT_model_for_territorial_integrated_management/links/55bb739208aed621de0d9692/Water-Quality-Impact-Assessment-of-Land-Use-and-Land-Cover-Changes-A-dynamic-IT-model-for-territorial-integrated-management.pdf#page=11

Gillies, S., van der Wel, C., Van den Bossche, J., Taves, M. W., Arnott, J., Ward, B. C. (2007). Shapely: Manipulation and analysis of geometric objects. *toblerity.org*. https://github.com/Toblerity/Shapely

Goutte, C., & Gaussier, E. (2005). A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. *European conference on information retrieval* (pp. 345–359). https://doi.org/10.1007/978-3-540-31865-1_25

Hakala, K., & Pyysalo, S. (2019). Biomedical named entity recognition with multilingual bert. *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks* (pp. 56–61). https://doi.org/10.18653/v1/D19-5709

Haris, E., Gan, K. H., & Tan, T.-P. (2020). Spatial information extraction from travel narratives: Analysing the notion of co-occurrence indicating closeness of tourist places. *Journal of Information Science*, *46*(5), 581–599. https://doi.org/10.1177/0165551519837188

Hassani, H., Beneki, C., Unger, S., Mazinani, M. T., & Yeganegi, M. R. (2020). Text mining in big data analytics. *Big Data and Cognitive Computing*, *4*(1), 1. https://doi.org/10.3390/bdcc4010001

Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing*. https://doi.org/10.5281/zenodo.1212303

Jordahl, K., den Bossche, J. V., Fleischmann, M., Wasserman, J., McBride, J., Gerard, J., Tratner, J., Perry, M., Badaracco, A. G., Farmer, C., Hjelle, G. A., Snow, A. D., Cochran, M., Gillies, S., Culbertson, L., Bartos, M., Eubank, N., maxalbert, Bilogur, A. … Leblanc, F. (2020). *Geopandas/geopandas: V0.8.1* (Version v0.8.1). Zenodo. https://doi.org/10.5281/zenodo.3946761

Kokla, M., & Guilbert, E. (2020). A review of geospatial semantic information modeling and elicitation approaches. *ISPRS International Journal of Geo-Information*, *9*(3), 146. https://doi.org/10.3390/ijgi9030146

McDonough, K., Moncla, L., & van de Camp, M. (2019). Named entity recognition goes to old regime france: Geographic text analysis for early modern french corpora. *International Journal of Geographical Information Science*, *33*(12), 2498–2522. https://doi.org/10.1080/13658816.2019.1620235

Medad, A., Gaio, M., Moncla, L., Mustière, S., & Le Nir, Y. (2020). Comparing supervised learning algorithms for spatial nominal entity recognition. *AGILE: GIScience Series*, *1*, 1–18. https://doi.org/10.5194/agile-giss-1-15-2020

Middleton, S. E., Kordopatis-Zilos, G., Papadopoulos, S., & Kompatsiaris, Y. (2018). Location extraction from social media: Geoparsing, location disambiguation, and geotagging. *ACM Transactions on Information Systems (TOIS)*, *36*(4), 1–27. https://doi.org/10.1145/3202662

Mohit, B. (2014). Named entity recognition. In *Natural language processing of semitic languages* (pp. 221–245). Springer. https://doi.org/10.1007/978-3-642-45358-8_7

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30*(1), 3–26. https://doi.org/10.1075/li.30.1.03nad

OpenStreetMap contributors. (2017). Planet dump. https://www.openstreetmap.org

Resnik, P., & Lin, J. (2010). 11 evaluation of nlp systems. *The Handbook of Computational Linguistics and Natural Language Processing*, *57*. https://doi.org/10.1002/9781444324044.ch11

Syed, M. A., Arsevska, E., Roche, M., & Teisseire, M. (2022). Geoxtag: Relative spatial information extraction and tagging of unstructured text. *AGILE: GIScience Series*, *3*, 1–10. https://doi.org/10.5194/agile-giss-3-16-2022

Vajjala, S., & Balasubramaniam, R. (2022, June 20-25). What do we really know about state of the art ner? *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022* (pp. 5983–5993). Marseille, France. https://doi.org/10.48550/arXiv.2205.00034

Wu, K., Zhang, X., Dang, Y., & Ye, P. (2022). Deep learning models for spatial relation extraction in text. *Geo-Spatial Information Science*, *26*(1), 58–70. https://doi.org/10.1080/10095020.2022.2076619

Zeng, D., Cao, Z., & Neill, D. B. (2021). Artificial intelligence–enabled public health surveillance—from local detection to global epidemic monitoring and control. In *Artificial intelligence in medicine* (pp. 437–453). Elsevier. https://doi.org/10.1016/B978-0-12-821259-2.00022-3

Zhang, C., Zhang, X., Jiang, W., Shen, Q., & Zhang, S. (2009). Rule-based extraction of spatial relations in natural language text. *2009 International Conference on Computational Intelligence and Software Engineering* (pp. 1–4). https://doi.org/10.1109/CISE.2009.5363900

Zheng, K., Xie, M. H., Zhang, J. B., Xie, J., & Xia, S. H. (2022). A knowledge representation model based on the geographic spatiotemporal process. *International Journal of Geographical Information Science*, *36*(4), 674–691. https://doi.org/10.1080/13658816.2021.1962527