AGILE: GIScience Series, 3, 16, 2022. https://doi.org/10.5194/agile-giss-3-16-2022
Proceedings of the 25th AGILE Conference on Geographic Information Science, 2022.
Editors: E. Parseliunas, A. Mansourian, P. Partsinevelos, and J. Suziedelyte-Visockiene.
This contribution underwent peer review based on a full paper submission.

GeoXTag: Relative Spatial Information Extraction and Tagging of Unstructured Text

Mehtab Alam SYED [1,3], Elena ARSEVSKA [2,3], Mathieu ROCHE [1,3], and Maguelonne TEISSEIRE [1,4]

[1]TETIS, Univ Montpellier, AgroParisTech, CIRAD, CNRS, INRAE, Montpellier, France
[2]ASTRE, Univ Montpellier, CIRAD, INRAE, Montpellier, France
[3]CIRAD, UMR TETIS, F-34398 Montpellier, France
[4]INRAE, UMR TETIS, Montpellier, France

Correspondence: mehtab-alam.syed@cirad.fr [1], elena.arsevska@cirad.fr [2], mathieu.roche@cirad.fr [1], maguelonne.teisseire@inrae.fr [1]

Abstract.

Spatial information has gained more attention in natural language processing tasks in different interdisciplinary domains. Moreover, the spatial information is available in two forms: *Absolute Spatial Information* (ASI) e.g., Paris, London, and Germany and *Relative Spatial Information* (RSI) e.g., south of Paris, north Madrid and 80 km from Rome. Therefore, it is challenging to extract RSI from textual data and compute its geotagging. This paper presents two strategies and the associated prototypes to address the following tasks: 1) extraction of relative spatial information from textual data and 2) geotagging of this relative spatial information. Experiments show promising results for RSI extraction and tagging.

Keywords. Natural Language Processing, Spatial Information, GeoTagger, GeoParser

1 Introduction

Named Entity Recognition (NER) is an important task in Natural Language Processing (NLP) that results in key information of text (Mohit, 2014). In broader context, these named entities are categorized in different types i.e., Person, Location, Organization and DateTime respectively (Nadeau and Sekine, 2007). In this work, we deal with the extraction of special cases of spatial entities in the text and how it is represented geographically. These special cases are in the form of relations with spatial entities e.g., North Milan, North-east Paris etc. However, there are a lot of challenges in the representation of these special cases of spatial entities. These challenges are the spatial semantic relations of spatial entities e.g., 20 km zone of South Korea, border of south France. These meaningful relations are important information in the perspectives of different applications to identify the correct geographical zones.

In recent years, spatial information extraction from textual data has gained more attention in the NLP field. Moreover, spatial information extraction has significance importance in different domains i.e., Healthcare, Stock Market, E-learning etc. in different unstructured textual data i.e., digital news sources, social media data. This work mainly focuses on epidemiology surveillance for identifying outbreaks and spatial information related to diseases in news. This spatial information is expressed in the textual documents in both simple and complex ways, depending on the syntax and semantic of expression. This spatial information is available in the form of absolute spatial information and relative spatial information (spatial information with relation to a location).

Geocoding is the identification of spatial information in the text and its transformation into valid spatial representation in the form of valid spatial coordinates shapes (e.g., point, line, polygon or multi-polygon) on a reference map (Avvenuti et al., 2018; Middleton et al., 2018). Geocoding involves two main steps i.e., 1) **Geoparsing** to identify location extraction or location disambiguation and 2) **Geotagging** to resolve the identified location into spatial coordinates or geographical information. Geoparsing is done through NLP techniques to extract spatial information from text prior to resolve it into geographical coordinates. Spatial information is available in the text in unstructured form e.g. “Two kilometers east of London". The Geotagging task assigns spatial coordinates to the extracted spatial information through some statistical models or algorithms. Sometimes it is also known as “Location disambiguation" which means to identify the location coordinates through a geographic database e.g., (London, UK, 51°30′26″N 0°7′39″W). Moreover, in some litera-

ture, it is also known as “location estimation" to tag spatial areas on the map (Middleton et al., 2018).

In this paper, we particularly investigate the extraction of relative spatial information (RSI) from text. Moreover, after extracting it, we identify its geographical coordinates to know the exact location on geographical map in the form of points, lines, or polygons. In our work, we mainly defined this spatial information in different categories. These categories are based on the type of relations with spatial information. These relationships of the spatial information describe the relative position to the location. In this work, we propose a rule-based entity extraction algorithm to extract those categories of relative spatial information from textual documents. The algorithm is further validated on a news dataset related to infectious disease outbreak to extract relative spatial information. Furthermore, we proposed the second algorithm for the geo-referencing of directional relation based on relative spatial information. It is used for identifying the geographical coordinates of the spatial relations of the toponym. Furthermore, a qualitative analysis is performed to validate the geographical shapes of RSI.

The remaining structure of the paper is as follows: Section 2 describes the previous work related to relative spatial information. Section 3 presents the process pipeline and the different steps of our proposal. Section 4 describes the experiments and discusses the associated results. Section 5 summarizes the software demo and the code repository. Whereas, Section 6 sums up the contribution and outlines future works.

2 Previous work in geo-referencing textual data

The purpose of this work is to identify unknown spatial entities in textual data (McDonough et al., 2019). Different researchers proposed various techniques and approaches to disambiguate spatial information in text i.e., machine learning, geographical databases, ontology-based reasoning and rule-based approaches (Kokla and Guilbert, 2020). In (Lesbegueries et al., 2006), the authors categorized the spatial information into two main categories i.e., 1) simple spatial information, known as Absolute Spatial information and 2) complex spatial information in which integration of spatial relations are involved, known as Relative Spatial information e.g., north, south, north-east, south-west.

McDonough et al. (2019) described a rule-based named-entity recognition to resolve special cases of spatial named entities in text. Chen et al. (2017) defined the spatial relations definitions and its geo-referencing. 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 spatial relation extraction based on geographical named entity recognition technology and a spatial relation-annotation corpus. Zhang et al. (2018) adopted a disambiguation method based on semantic similarity of ambiguous word using knowledge to resolve ambiguous spatial entities. Karimzadeh et al. (2019) proposed named entity recognition (NER) algorithm by adding specific heuristics and disambiguation methods for the improvement of toponym resolution in geoparsing of text. Middleton et al. (2018) came up with a geoparsing algorithm to resolve spatial information from text. Zhang et al. (2020) 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. Medad et al. (2020) used a combination of transfer learning principle and supervised learning algorithm for the identification of spatial nominal information. Zenasni et al. (2018) proposed a new methodology to identify spatial entities and their relations in in French short messages (SMS and tweets).

Geo-referencing or geotagging and extraction of relative spatial named entities are challenging to obtain that results into meaningful information for different domains (Middleton et al., 2018). Middleton et al. (2018) used OpenStreetMap database and a geotagging algorithm using a language model constructed using Geonames gazetteer approach to identify the geographical coordinates of locations. There are different types of approaches to disambiguate toponyms i.e., 1) map-based approaches (it is used to visualize toponym on a geographical map), 2) knowledge-based approach (it has external knowledge about toponyms such as gazetteers) (Buscaldi, 2011). However, there is no such approach to disambiguate the toponym having spatial relations.

After analysing the research, Relative Spatial Information (RSI) extraction from text and the geotagging of such features are not yet explored and validated. Moreover, these RSI are not geotagged with state-of-the-art geotagging application i.e. Google Maps[1], OpenStreetMap[2] or GeoNames[3] respectively.

3 GeoXTag: Processing Pipeline

In this paper, we propose a methodology to extract and geotag relative spatial information. The process pipeline is shown in Figure 1.

The process pipeline defines the process of the GeoXTag module that results in two main components i.e., 1) RSI extraction and 2) RSI tagging respectively. Relative spatial information has many challenges to address, both in its extraction and tagging. In the methodology, we address to extract (*GeoX*) different categories of relative spatial information and tag (*Tag*) these categories of relative spatial information.

[1] https://www.google.com/maps
[2] https://www.openstreetmap.org/
[3] https://www.geonames.org/

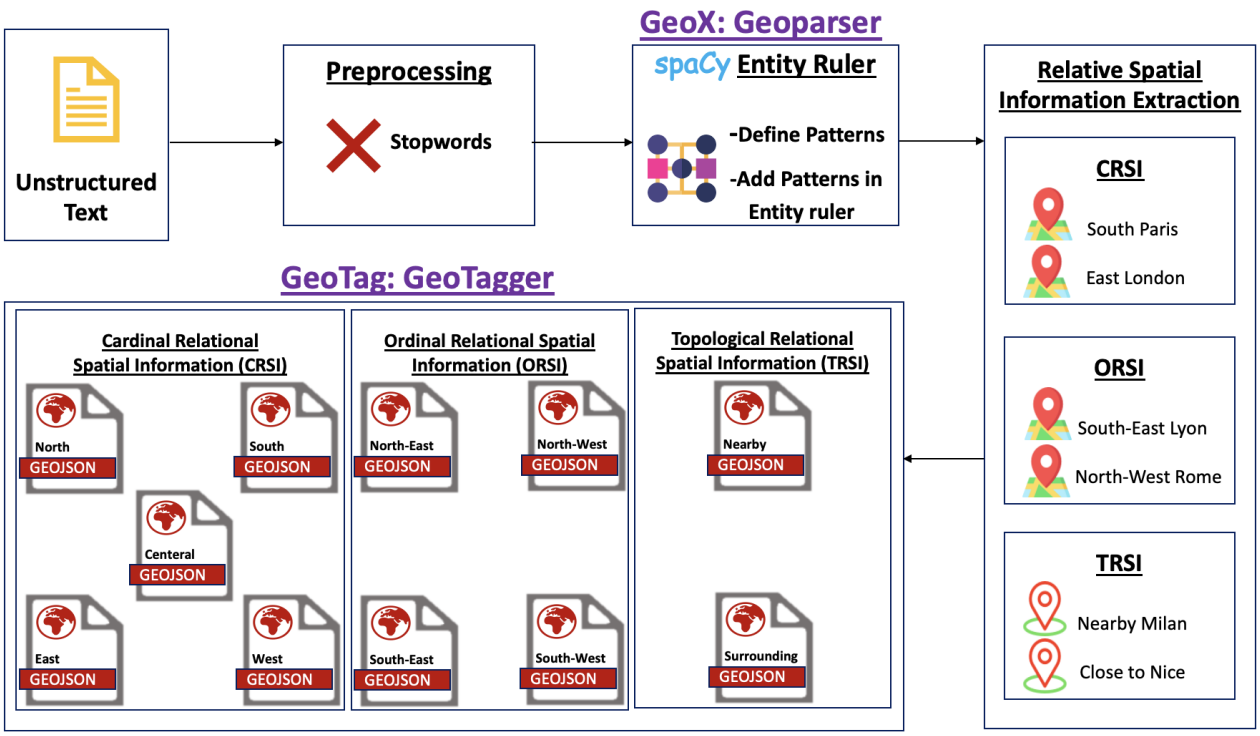

Figure 1. GeoXTag: Relative information extraction and tagging process pipeline

3.1 Textual Input

The first step of the proposed pipeline is the textual input data. It can be in the form of text documents, social media posts, tweets textual data etc. This textual data contains different types of named information including location, person, organization, date, and time etc. An example of the text data is available in the **Step 1** of Figure 2.

3.2 Spatial Dictionaries

Two spatial dictionaries extracted from GeoNames[4] are used for identifying relative spatial information in the text. These dictionaries are in the form of two CSV files i.e., `countries.csv` and `cities.csv` respectively. Country list has attributes name and code respectively. Similarly, city list have the attributes country, name, latitude and longitude respectively. These data files are used for relative spatial information extraction and tagging which are discussed in the next sections.

3.3 Preprocessing

The next step is to preprocess the textual input data. In this work, the preprocessing step is limited to remove stop words like 'am', 'is', 'are', 'was', 'were', 'be', 'been' etc.

[4]https://www.geonames.org/

3.4 GeoX: Entity Ruler

The next step is related to the extraction of relative spatial information from preprocessed textual data. ***spaCy*** (Vasiliev, 2020) NLP Python library is used for relative spatial information extraction by applying Named Entity Recognition (NER) techniques. Relative spatial information in textual data are categorized into three major categories. The terminologies for all the categories are adapted from the paper (Bateman et al., 2010). These categories are detailed in the following subsections.

3.4.1 Cardinal Relational Spatial Information

Cardinal relational spatial information (CRSI) is defined as relative spatial information that is defined with concatenation of cardinal with absolute spatial information (ASI). Cardinals are *north*, *south*, *east* and *west* respectively. Moreover, in this category, we also added the center or central of ASI. The definition and some examples of CRSI:

```
          CRSI = Cardinal + ASI
e.g.      CRSI = North + Paris
e.g.      CRSI = West + Italy
```

CRSI are identified by ***spaCy*** by adding patterns in its entity ruler. These patterns in the entity ruler are defined as follows:

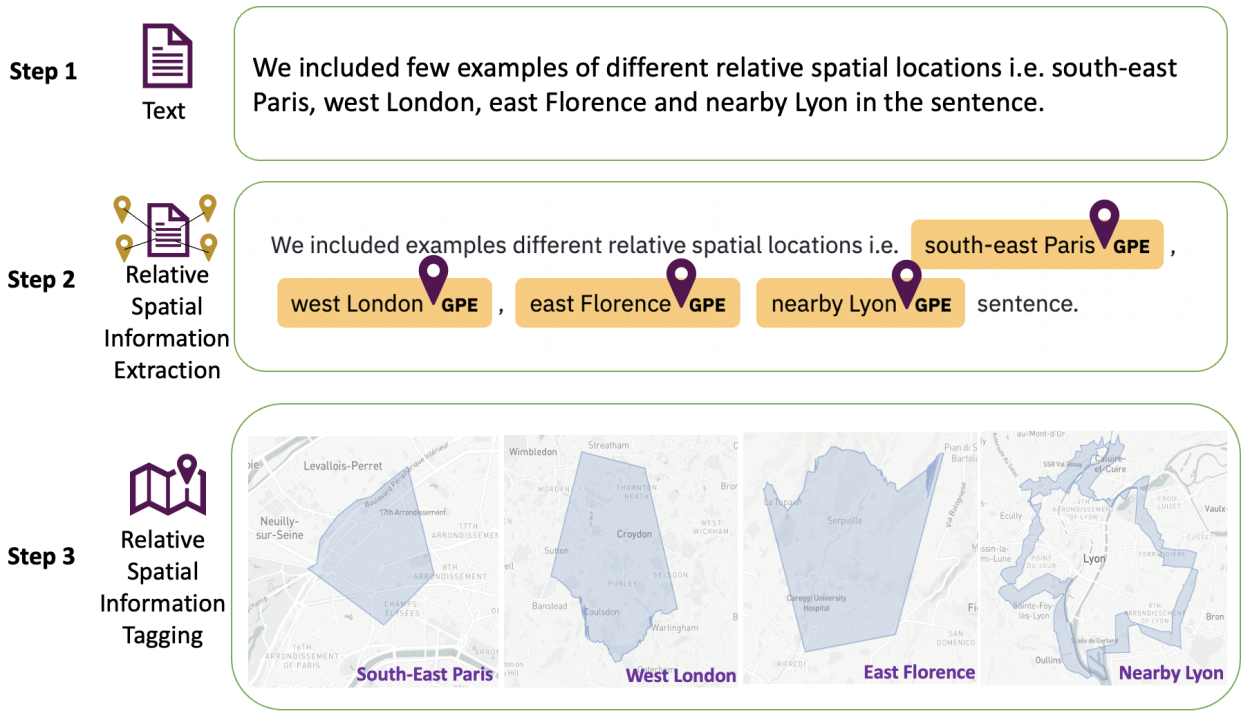

Figure 2. GeoXTag: An example of relative spatial information extraction and tagging

```
regexCardinal = r"(?i)(north|east|
        south|center|centeral|west)"
pattern = {"label": "GPE", "pattern":
    [{"LOWER":{"REGEX": regexCardinal}},
    {"LOWER":city|country}]}
```

In Figure 2, after the processing of step 1, GeoX extracted two CRSI i.e. east Florence and west London.

3.4.2 Ordinal Relational Spatial Information

Ordinal relational spatial information (ORSI) is defined as relative spatial information that is obtained by concatenation of ordinal with absolute spatial information (ASI). Ordinals are north-east, south-west, south-east, and north-west respectively. Examples of ORSI are as follows:

```
     ORSI = Ordinal + ASI
e.g. ORSI = North-east + Paris
e.g. ORSI = South-west + Italy
```

These patterns in the entity ruler are defined as follows:

```
regexSpaceDash = r"(\s+|-)"
pattern = {"label": "GPE", "pattern":
    [{"LOWER":{"REGEX": regexCardinal}},
    {"LOWER":{"REGEX": regexSpaceDash}},
    {"LOWER":{"REGEX": regexCardinal}},
    {"LOWER":row['name'].lower()}
```

In Figure 2, after the processing of step 1, GeoX extracted one ORSI i.e., south-east Paris.

3.4.3 Topological Relational Spatial Information

Topological relational spatial information (TRSI) is defined as relative spatial information obtained by concatenation of keywords with absolute spatial information (ASI). Such relations locate spatial information with respect to some specified region, but do so in a way that makes it difficult to identify the regions. These keywords are close, nearby, surrounding and 8 km respectively. The definition and examples of TRSI are as follows:

```
     TRSI = Keywords + ASI
e.g. TRSI = Nearby + Paris
e.g. TRSI = Surrounding + Spain
e.g. TRSI = 8 km from + Rome
```

These patterns in the entity ruler are defined as follows:

```
regexKeyword = r"(?i)(miles|kilometer
            |km)"
pattern = {"label": "GPE", "pattern":
    [{"LOWER":{"REGEX": regexKeyword}},
    {"LOWER":row['name'].lower()}]}
```

In Figure 2, after the processing of step 1, GeoX extracted one TRSI i.e., nearby Lyon. As there are plenty of TRSI possibilities in textual data, we extract the simple TRSI. Therefore, it is challenging to extract all the possible TRSI from the text. In particular, there are such possible keywords e.g., surrounding, nearby, far from, close to etc.

The complete pseudocode for performing RSI extraction is available in Algorithm **GeoX: Relative Spatial Information Extraction**.

Algorithm GeoX: Relative Spatial Information Extraction

Spatial Dictionaries: ListOfCities, ListOfCountries
Input: text
Output: extractedRSE

$cardinal \leftarrow$ "$(?i)(north|east|south|center|west)$"
$distance \leftarrow$ "$(?i)(miles|kilometer|km)$"
$digit \leftarrow$ "$(d+)$"
$spaceDash \leftarrow$ "$(s+|-)$"
$keyword \leftarrow$ "$(?i)(surround|near|next|close)$"
$patterns \leftarrow []$

for $city \in ListOfCities$ **do**
  $patternCardinal \leftarrow concat(cardinal, city)$
  $patternOrdinal \leftarrow concat(cardinal, spaceDash, cardinal, city)$
  $patternKeyword \leftarrow concat(keyword, city)$
  $patternDistance \leftarrow concat(digit, distance, city)$
  $patterns.append(patternCardinal)$
  $patterns.append(patternOrdinal)$
  $patterns.append(patternKeyword)$
  $patterns.append(patternDistance)$
end for

for $country \in ListOfCountries$ **do**
  $patternCardinal \leftarrow concat(cardinal, country)$
  $patternOrdinal \leftarrow concat(cardinal, spaceDash, cardinal, country)$
  $patternKeyword \leftarrow concat(keyword, country)$
  $patternDistance \leftarrow concat(digit, distance, country)$
  $patterns.append(patternCardinal)$
  $patterns.append(patternOrdinal)$
  $patterns.append(patternKeyword)$
  $patterns.append(patternDistance)$
end for
# Initialize English language processing object
$nlp \leftarrow English()$
Add Entity ruler pipeline and assign to ruler object
$ruler \leftarrow addPipeline($"$entity-ruler$"$)$
Add pattern list in ruler object
$ruler.addPatterns(patterns)$
Remove stop words e.g. is, of etc from input text
$processedText \leftarrow removeStopwords(text)$
extractedRSE have the list of relative spatial entities
$extractedRSE \leftarrow nlp(processedText).ents$

3.5 GeoTag: Relative Spatial Information Tagging

Geotagging is the process of attaching metadata that contains geographical information about a location to a digital map (Amaral, 2014). After the extraction of this RSI from text, the next step is to tag these entities on a digital map to know its exact geographical location. This RSI is identified by processing the geographical information of ASI. In the proposed methodology, we only focus on tagging of relative spatial information of CRSI and ORSI that were discussed in earlier sections. The complete algorithm for performing RSI tagging is available in Algorithm **GeoTag: Relative Spatial Information Tagging**. The detail steps to tag CRSI, ORSI and TRSI are described in the following subsections.

Algorithm GeoTag: Relative Spatial Information Tagging

Input: LocationName e.g., city, country
Output: geojson files of central, east, west, east, south, north-east, north-west, south-east, south-west

$northMin \leftarrow 337, northMax \leftarrow 22$
$eastMin \leftarrow 67, eastMax \leftarrow 112$
$southMin \leftarrow 157, southMax \leftarrow 202$
$westMin \leftarrow 247, westMax \leftarrow 292$
$neMin \leftarrow 22, neMax \leftarrow 67$
$seMin \leftarrow 112, seMax \leftarrow 157$
$nwMin \leftarrow 292, nwMax \leftarrow 337$
$swMin \leftarrow 202, swMax \leftarrow 247$
$north \leftarrow [], south \leftarrow [], east \leftarrow [], west \leftarrow []$
$northEast \leftarrow [], southEast \leftarrow [], northWest \leftarrow []$
$southWest \leftarrow [], centeral \leftarrow []$
$nearby \leftarrow [], surrounding \leftarrow []$
Call Procedure to identifiy coordinates
$identifySpatialCoordinates()$
Call Procedure to sort coordinates
$sortCardinalOrdinal()$
Call Procedure for finding minimum and maximum midpoints of cardinal/ordinal
$findMinMaxMidpoints()$
Call Procedure to set centeral coordinates
$setCenteralCoordinates()$
Call Procedure to save GeoJson files of cardinal/ordinal
$saveGeoJsonFiles()$

3.5.1 GeoTag: CRSI

CRSI are already defined in Section 3.4.1. This RSI is geotagged by following the sequence of steps:

1. The first step is to get the geographical information of ASI. e.g., To geotag "North Paris", we need to get the geographical information using **Nominatim API** of ASI "Paris".

2. The next step is to identify the possible RSI of ASI. e.g. for ASI "Paris" the possible RSI's are "North Paris", "South Paris", "East Paris" and "West Paris" respectively. Moreover, in this category we also added the center or central of ASI.

3. 'North' of an ASI is between 337° to 22° from the center of location (ASI). Therefore, in the proposed work, we collected the coordinates of ASI which have angles between 337° to 22°. These coordinates are stored in a list of coordinates of the north. Furthermore, the midpoint of minimum angle coordinate

which is closer to 337° and centroid of ASI is calculated and stored in the list of north. Similarly, the midpoint of maximum angle coordinate which is closer to 22° and centroid of ASI is calculated and stored in the list of north. Lastly, the list of coordinates of the north are saved as a GeoJson file, which is compatible with the OpenStreetMap leaflet.

4. 'East' of an ASI is between 67° to 112° from the center of location (ASI). Therefore, in the proposed work, we collected the coordinates of ASI which have angles between 67° to 112°. These coordinates are stored in a list of coordinates of the east. Furthermore, the midpoint of minimum angle coordinate which is closer to 67° and centroid of ASI is calculated and stored in the list of east. Similarly, the midpoint of maximum angle coordinate which is closer to 112° and centroid of ASI is calculated and stored in the list of east. Lastly, the list of coordinates of the east are saved as a GeoJson file, which is compatible with the OpenStreetMap leaflet.

5. Similarly, 'South' of an ASI is between 157° to 202° from the center of location (ASI). Therefore, we collected the coordinates of ASI which have angles between 157° to 202°. Likewise, the other two cardinals above the same steps are followed to save its GeoJson file.

6. Finally, 'West' of an ASI is between 247° to 292° from the center of location (ASI). Therefore, we collected the coordinates of ASI which have angles between 247° to 292°. Likewise, the other cardinals above the same steps are followed to save its GeoJson file.

Figure 2 shows two CRSI, i.e., west London and east Florence, with the geographical information in GeoJson file, that are shown in OpenStreetMap leaflets.

3.5.2 GeoTag: Ordinal relational spatial information

ORSI is already defined in Section 3.4.2. This RSI is geotagged by following the sequence of steps:

1. The output of ORSI are for each ordinal, i.e., "northeast.geojson", "northwest.geojson", "southeast.geojson" and "southwest.geojson" respectively.

2. 'North-East' of an ASI is between 22° to 57° from the center of location (ASI). Therefore, we collected the coordinates of ASI which have angles between 22° to 57°. Likewise, in Section 3.5.1 the same steps are followed to save the GeoJson file.

3. 'North-East' of an ASI is between 22° to 67° from the center of location (ASI). Therefore, we collected the coordinates of ASI which have angles between 22° to 67°. Likewise, the same steps are followed to save the "northeast.geojson" GeoJson file described in Section 3.5.1.

4. 'South-East' of an ASI is between 112° to 157° from the center of location (ASI). Therefore, we collected the coordinates of ASI which have angles between 112° to 157°. Likewise, the same steps are followed to save the "southeast.geojson" GeoJson file described in Section 3.5.1.

5. 'South-West' of an ASI is between 202° to 247° from the center of location (ASI). Therefore, we collected the coordinates of ASI which have angles between 202° to 247°. Likewise, the same steps are followed to save the "southwest.geojson" GeoJson file described in Section 3.5.1.

6. 'North-West' of an ASI is between 292° to 337° from the center of location (ASI). Therefore, we collected the coordinates of ASI which have angles between 292° to 337°. Likewise, the same steps are followed to save the "northwest.geojson" GeoJson file described in Section 3.5.1.

3.5.3 GeoTag: Topological relational spatial information

TRSI is already defined in Section 3.4.3. In this work, we geotagged only the simple TRSI, i.e., nearby, surrounding. This RSI is geotagged by following the sequence of steps:

1. The output of the two TRSI 'nearby' and 'surrounding' are saved as 'nearby.geojson' and 'surrounding.geojson' respectively.

2. Initially, we have to calculate the area of the location for which we are going to calculate its 'nearby' and 'surrounding' relations. The next is to calculate the boundary distance of the nearby and surrounding points. If the area of location is greater than 70 km^2; boundary distance of the points for 'nearby' is 0.6% of the area; whereas boundary distance of the points for 'surrounding' is 1.2% of the area. Similarly, If the area of location is less than 70 km^2; boundary distance of the points for 'nearby' is 1% of the area; whereas boundary distance of the points for 'surrounding' is 2% of the area.

3. The next step is to calculate all the points from a boundary distance from the corresponding points at similar angle for 'nearby' and 'surrounding' TRSI.

4. The next step is to save the coordinates in the corresponding geojson files i.e., "nearby.geojson" and "surrounding.geojson" files.

Algorithm Procedure for Identifying coordinates of Cardinal, Ordinal, Topological (nearby, surrounding)

procedure $identifySpatialCoordinates()$
Retrieve polygon coordinates of location

$coordinates \leftarrow getPolygonCoordinates(locationName)$
Retrieve centroid of polygon
$centroid \leftarrow getCentroid(locationName)$

for $point \in coordinates$ **do**
$angle = calculateBearing(centroid, point)$
calculate angle between centroid and point
$nearpt = getPoint(p[1], p[0], angle, distanceNear)$

$surroundingpt = getPoint(p[1], p[0], angle,$
$distanceSurrounding)$
$nearby.append(nearpt)$
$surrounding.append(surroundingpt)$
$point.append(angle)$
if $angle \geq northMin$ **or** $angle \leq northMax$ **then**
$north.append(point)$
end if
if $angle \geq southMin$ **and** $angle \leq southMax$ **then**
$south.append(point)$
end if
if $angle \geq eastMin$ **and** $angle \leq eastMax$ **then**
$east.append(point)$
end if
if $angle \geq westMin$ **and** $angle \leq westMax$ **then**
$west.append(point)$
end if
if $angle \geq neMin$ **and** $angle \leq neMax$ **then**
$northEast.append(point)$
end if
if $angle \geq nwMin$ **and** $angle \leq nwMax$ **then**
$northWest.append(point)$
end if
if $angle \geq seMin$ **and** $angle \leq seMax$ **then**
$southEast.append(point)$
end if
if $angle \geq swMin$ **and** $angle \leq swMax$ **then**
$southWest.append(point)$
end if
end for
end procedure

Algorithm Procedure for Sorting cardinal/ordinal

procedure $sortCardinalOrdinal()$
$north.sort(angle)$
$south.sort(angle)$
$east.sort(angle)$
$west.sort(angle)$
$northEast.sort(angle)$
$northWest.sort(angle)$
$southEast.sort(angle)$
$southWest.sort(angle)$
end procedure

Algorithm Procedure for finding coordinate at a particular distance at certain angle

procedure $getPoint(lat, ln, angle, distanceKm)$
$R = 6378.1$ # Radius of Earth

$brng = angle * math.pi/180$
angle in radians

$d = distanceInKm$
distance in KM

$lat1 = radians(lat)$
$lon1 = radians(ln)$
$p1 = sin(lat1) * cos(d/R)$
$p2 = cos(lat1) * sin(d/R) * cos(brng)$
$lat2 = asin(p1 + p2)$
$p3 = sin(brng) * sin(d/R) * cos(lat1)$
$p4 = cos(d/R) - sin(lat1) * sin(lat2)$
$lon2 = lon1 + atan2(p3, p4)$
$lat2 = degrees(lat2)$
$lon2 = degrees(lon2)$
return $(lon2, lat2, angle)$
end procedure

4 Results and Discussion

There are two main tasks in the proposed work to be validated: RSI extraction and RSI tagging. Concerning the extraction of RSI 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 and Pyysalo, 2019; Resnik and Lin, 2010). Precision, recall, and F-Score are defined as follows (Goutte and Gaussier, 2005):

$$Precision = \frac{Corrected\ RSI\ Recognized}{Total\ RSI\ Recognized} \tag{1}$$

$$Recall = \frac{Corrected\ RSI\ Recognized}{Total\ RSI\ in\ Corpus} \tag{2}$$

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

To evaluate the NER extraction algorithms, we need the degree of correctness of labelled entities for measuring precision and recall. It depends on how the boundaries of extracted entities are defined to count it in the entities list in the textual phrases. For instance, by considering the example of RSI in this text “10 km from south border of France". If the algorithm extracts it as “south border France" or “border France", it has to be distinct as

Algorithm Procedure for finding minimum and maximum midpoints of cardinal/ordinal

procedure $findMinMaxMidpoints()$
north[0] is the north point with minimum angle
$northMidMin \leftarrow midpoint(centroid, north[0])$
north[-1] is the north point with maximum angle
$northMidMax \leftarrow midpoint(centroid, north[-1])$
$north.append(northMidMin)$
$north.append(northMidMax)$
$southMidMin \leftarrow midpoint(centroid, south[0])$
$southMidMax \leftarrow midpoint(centroid, south[-1])$
$south.append(southMidMin)$
$south.append(southMidMax)$
$eastMidMin \leftarrow midpoint(centroid, east[0])$
$eastMidMax \leftarrow midpoint(centroid, east[-1])$
$east.append(eastMidMin)$
$east.append(eastMidMax)$
$westMidMin \leftarrow midpoint(centroid, west[0])$
$westMidMax \leftarrow midpoint(centroid, west[-1])$
$west.append(westMidMin)$
$west.append(westMidMax)$

$neMidMin \leftarrow midpoint(centroid, northEast[0])$
$neMidMax \leftarrow midpoint(centroid, northEast[-1])$

$northEast.append(neMidMin)$
$northEast.append(neMidMax)$
$nwMidMin \leftarrow midpoint(centroid, northWest[0])$
$nwMidMax \leftarrow midpoint(centroid, northWest[-1])$

$northWest.append(nwMidMin)$
$northWest.append(nwMidMax)$
$seMidMin \leftarrow midpoint(centroid, southEast[0])$
$seMidMax \leftarrow midpoint(centroid, southEast[-1])$

$southEast.append(seMidMin)$
$southEast.append(seMidMax)$
$swMidMin \leftarrow midpoint(centroid, southWest[0])$
$swMidMax \leftarrow midpoint(centroid, southWest[-1])$

$southWest.append(swMidMin)$
$southWest.append(swMidMax)$
end procedure

Algorithm Procedure for setting centeral points

procedure $setCenteralCoordinates()$
$centeral.append(northMidMax)$
$centeral.append(northMidMin)$
$centeral.append(westMidMax)$
$centeral.append(westMidMax)$
$centeral.append(southMidMax)$
$centeral.append(southMidMax)$
$centeral.append(eastMidMax)$
$centeral.append(eastMidMax)$
end procedure

Algorithm Procedure for saving GeoJson file of cardinal, ordinal and topological information

procedure $saveGeoJsonFiles()$
$saveGeoJson(north, \text{“}north.geojson\text{”})$
$saveGeoJson(south, \text{“}south.geojson\text{”})$
$saveGeoJson(east, \text{“}east.geojson\text{”})$
$saveGeoJson(north, \text{“}west.geojson\text{”})$
$saveGeoJson(northeast, \text{“}northeast.geojson\text{”})$
$saveGeoJson(northwest, \text{“}northwest.geojson\text{”})$
$saveGeoJson(southeast, \text{“}southeast.geojson\text{”})$
$saveGeoJson(southwest, \text{“}southwest.geojson\text{”})$
$saveGeoJson(centeral, \text{“}centeral.geojson\text{”})$
$saveGeoJson(nearby, \text{“}nearby.geojson\text{”})$
$saveGeoJson(surrounding, \text{“}surrounding.geojson\text{”})$

end procedure

true positive or not. The geographical location of “border France" is completely different whereas “south border France" somehow is geographically closed region which have more accuracy. In other way around, the partially detected RSI are considered as **false positive**.

The first task is to evaluate **GeoX** also known as Geoparsing. For this, we used the dataset of news articles for different infectious diseases. The dataset of the evaluation is available at https://github.com/mehtab-alam/RSI_Disease_Dataset. This dataset is composed of news articles having the outbreak information diseases,i.e. Covid-19, Avian Influenza, Antimicrobial Resistance (AMR), Tick-borne Encephalitis (TBE) and Lyme respectively. The dataset is manually labelled in which a separate column of each CSV file has the relative spatial information (RSI). The results are compared with the gold standard dataset to calculate the precision, recall and F-Score as shown in Table 1. Precision, recall and F-Score are measured for each disease dataset. Moreover, the aggregate measure of GeoX (Geoparsing) has the precision of 0.9, recall of 0.88 and F-Score of 0.88 respectively. For the evaluation, we considered some cases in RSI, e.g., ‘North America’ and ‘South Korea’. Moreover, in certain cases we considered it false positive in evaluation like spell mistakes in RSI e.g., ‘(Yonhap)South Korea’ or complex RSI e.g. ‘80 km towards the south of Yorkshire’. Currently, the dataset is manually labelled, and the evaluation is done by one expert. However, in the subsequent work, more experts will be involved in the annotation of the corpus.

The second step is to evaluate **GeoTag** also known as GeoTagger. For this, we have to perform qualitative analysis of the shapes of locations. The evaluation is performed on six sample cities with area larger, average and smaller respectively. 11 relations are geotagged in the proposed work, i.e., north, south, east, west, north-east, north-west, south-east, south-west, central, nearby and surrounding respectively. For each shape, the maximum score is 5 and the minimum score is 1. The quality score of geographical shapes are divided into five classes. ***1*** is assigned to the geographical which has less similarity concretely

Disease Name	No. of Articles	RSI Extracted	RSI 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**

Table 1. Evaluation of RSI Extraction in Diseases News Dataset

less than *20%* to the actual location shape, e.g., Shape of North Milan (see https://rb.gy/shzlix). Similarly, ***2*** is assigned based on quality of shape similarity of *20% - 40%* to the actual location shape, e.g., North Grenoble (see https://rb.gy/shzlix). Moreover, ***3*** is assigned to shape quality *40% - 60%*, in particular in these shapes we found the edges of the shape are not much smooth in few areas however it closely approximates the actual shape e.g., North Milan. Furthermore, ***4*** is assigned to a better quality shape of similarity *60% - 80%* to actual shape with a few less smooth area edges, e.g., North-East Nantes. Lastly, ***5*** is assigned to a quality shape of similarity *80% - 100%* to actual shape, e.g., East Paris, South Paris. Table 2 shows the evaluation of geotagged RSI.

For the evaluation, we categorized cities i.e., large, average, and small based on their areas. For each location (city), we identified the shapes of their 11 relations. The average score of 11 relations shapes of geographical locations (cities) out of ***5*** is ***4.35*** which is *87%* in total. Moreover, the complete evaluation of these geographical shapes are available at https://rb.gy/shzlix.

5 Data and Software Availability

The whole workflow is divided into two main components, i.e., 1) GeoX - Relative Spatial Information Extraction and 2) GeoX - Relative Spatial Information Tagging. The software demonstration, code repositories and dataset URLs are available as shown in Table 3.

Type	URL
RSI Extraction Demo	https://rb.gy/qm1inj
RSI Tagging Demo	https://rb.gy/idjgyw
RSI Extraction Code Repository	https://rb.gy/xundwa
RSI Tagging Code Repository	https://rb.gy/eaz94a
RSI Extraction Dataset	https://rb.gy/mt7qsb
RSI Tagging Evaluation	https://rb.gy/shzlix

Table 3. Software Demo, Code Repositories and Dataset

The workflow underlying this paper was partially reproduced by an independent reviewer during the AGILE reproducibility review and a reproducibility report was published at https://doi.org/10.17605/osf.io/3g9s8.

6 Conclusion and Future Work

The proposed research focuses on the relative spatial information (RSI) extraction from text and geotagging of RSI on geographical maps. We proposed a rule-based geoparser to extract RSI from the text documents. Furthermore, we proposed an algorithm to identify the geographical coordinates of the 11 spatial relations RSI based on three categories i.e., Cardinal relational RSI, Ordinal directional RSI and Topological relational RSI. The results of the RSI extraction from textual documents are validated with news dataset of infectious diseases with precision of *0.9*, recall of *0.88* and F-Score of *0.88*. Moreover, the geographical coordinates of these RSI are qualitative analysed with a qualitative score of *87%*.

This is the pioneer work that have possibility to step ahead towards the extension and tweaking of proposed work. In future work, we will deal with the identification of more possible toponym relational RSI. Furthermore, more tweaking will be performed for the geographical coordinates of cardinal relational RSI and ordinal directional RSI. A further step will be performed to find local regions inside the RSI if possible to get the exact localities inside the geographical region. We will also investigate to geotag complex toponym relational RSI, e.g., '10 km from south border of France', '10 km radius of Paris', 'small towns near to south of Paris'.

7 Acknowledgments

This study was partially funded by EU grant 874850 MOOD and is catalogued as MOOD038. The contents of this publication are the sole responsibility of the authors and do not necessarily reflect the views of the European Commission.

Quality Score (Shape)	**1 = (0%-20%), 2 = (20%-40%), 3 = (40%-60%), 4 = (60%-80%), 5 = (80%-100%)**			
Spatial Information	Area(km2)	Category	No. of Spatial Relations (Shapes)	Average Score
Milan	180	Large	11	4.1
Paris	105	Large	11	4.8
Montpellier	57	Average	11	4.6
Nantes	65	Average	11	3.9
Voiron	23	Small	11	4.2
Grenoble	17	Small	11	4.5
Average Score (Total)	**5**		**Average Score (Obtained)**	**4.35**

Table 2. Evaluation of RSI Geotagging

References

Amaral, I.: Encyclopedia of Social Media and Politics, 2014.

Avvenuti, M., Cresci, S., Nizzoli, L., and Tesconi, M.: GSP (Geo-Semantic-Parsing): geoparsing and geotagging with machine learning on top of linked data, in: European Semantic Web Conference, pp. 17–32, Springer, 2018.

Bateman, J. A., Hois, J., Ross, R., and Tenbrink, T.: A linguistic ontology of space for natural language processing, Artificial Intelligence, 174, 1027–1071, 2010.

Buscaldi, D.: Approaches to disambiguating toponyms, Sigspatial Special, 3, 16–19, 2011.

Chen, H., Vasardani, M., and Winter, S.: Geo-referencing place from everyday natural language descriptions, arXiv preprint arXiv:1710.03346, 2017.

Goutte, C. and Gaussier, E.: A probabilistic interpretation of precision, recall and F-score, with implication for evaluation, in: European conference on information retrieval, pp. 345–359, Springer, 2005.

Hakala, K. and Pyysalo, S.: Biomedical named entity recognition with multilingual BERT, in: Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, pp. 56–61, 2019.

Karimzadeh, M., Pezanowski, S., MacEachren, A. M., and Wallgrün, J. O.: GeoTxt: A scalable geoparsing system for unstructured text geolocation, Transactions in GIS, 23, 118–136, 2019.

Kokla, M. and Guilbert, E.: A review of geospatial semantic information modeling and elicitation approaches, ISPRS International Journal of Geo-Information, 9, 146, 2020.

Lesbegueries, J., Sallaberry, C., and Gaio, M.: Associating spatial patterns to text-units for summarizing geographic information, in: ACM SIGIR 2006. GIR, Geographic Information Retrieval, Workshop, pp. 40–43, 2006.

McDonough, K., Moncla, L., and van de Camp, M.: Named entity recognition goes to old regime France: geographic text analysis for early modern French corpora, International Journal of Geographical Information Science, 33, 2498–2522, 2019.

Medad, A., Gaio, M., Moncla, L., Mustière, S., and Le Nir, Y.: Comparing supervised learning algorithms for spatial nominal entity recognition, AGILE: GIScience Series, 1, 1–18, 2020.

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

Mohit, B.: Named entity recognition, in: Natural language processing of semitic languages, pp. 221–245, Springer, 2014.

Nadeau, D. and Sekine, S.: A survey of named entity recognition and classification, Lingvisticae Investigationes, 30, 3–26, 2007.

Resnik, P. and Lin, J.: 11 evaluation of NLP systems, The handbook of computational linguistics and natural language processing, 57, 2010.

Vasiliev, Y.: Natural Language Processing with Python and SpaCy: A Practical Introduction, No Starch Press, 2020.

Zenasni, S., Kergosien, E., Roche, M., and Teisseire, M.: Spatial Information Extraction from Short Messages, Expert Systems with Applications, 95, 351–367, https://doi.org/https://doi.org/10.1016/j.eswa.2017.11.025, 2018.

Zhang, C., Zhang, X., Jiang, W., Shen, Q., and Zhang, S.: Rule-Based Extraction of Spatial Relations in Natural Language Text, in: 2009 International Conference on Computational Intelligence and Software Engineering, pp. 1–4, https://doi.org/10.1109/CISE.2009.5363900, 2009.

Zhang, K., Zhu, Y., Gao, W., Xing, Y., and Zhou, J.: An Approach for Named Entity Disambiguation with Knowledge Graph, in: 2018 International Conference on Audio, Language and Image Processing (ICALIP), pp. 138–143, https://doi.org/10.1109/ICALIP.2018.8455418, 2018.

Zhang, X., Zhang, C., WU, M., and LV, G.: Spatiotemporal features based geographical knowledge graph construction, Scientia Sinica Informationis, 50, 1019–1032, 2020.