remote sensing

MDPI

Article

Adaptive Geoparsing Method for Toponym Recognition and Resolution in Unstructured Text

Edwin Aldana-Bobadilla [1,*], Alejandro Molina-Villegas [2], Ivan Lopez-Arevalo [3], Shanel Reyes-Palacios [3], Victor Muñiz-Sanchez [4] and Jean Arreola-Trapala [4]

[1] Conacyt-Centro de Investigación y de Estudios Avanzados del I.P.N. (Cinvestav), Victoria 87130, Mexico
[2] Conacyt-Centro de Investigación en Ciencias de Información Geoespacial (Centrogeo), Mérida 97302, Mexico; amolina@centrogeo.edu.mx
[3] Centro de Investigación y de Estudios Avanzados del I.P.N. Unidad Tamaulipas (Cinvestav Tamaulipas), Victoria 87130, Mexico; ilopez@cinvestav.mx (I.L.-A.); shanel.reyes@cinvestav.mx (S.R.-P.)
[4] Centro de Investigación en Matemáticas (Cimat), Monterrey 66628, Mexico; victor_m@cimat.mx (V.M.-S.); jean.arreola@cimat.mx (J.A.-T.)
[*] Correspondence: edwyn.aldana@cinvestav.mx

Received: 10 July 2020; Accepted: 13 September 2020; Published: 17 September 2020

Abstract: The automatic extraction of geospatial information is an important aspect of data mining. Computer systems capable of discovering geographic information from natural language involve a complex process called geoparsing, which includes two important tasks: geographic entity recognition and toponym resolution. The first task could be approached through a machine learning approach, in which case a model is trained to recognize a sequence of characters (words) corresponding to geographic entities. The second task consists of assigning such entities to their most likely coordinates. Frequently, the latter process involves solving referential ambiguities. In this paper, we propose an extensible geoparsing approach including geographic entity recognition based on a neural network model and disambiguation based on what we have called *dynamic context disambiguation*. Once place names are recognized in an input text, they are solved using a grammar, in which a set of rules specifies how ambiguities could be solved, in a similar way to that which a person would utilize, considering the context. As a result, we have an assignment of the most likely geographic properties of the recognized places. We propose an assessment measure based on a ranking of closeness relative to the predicted and actual locations of a place name. Regarding this measure, our method outperforms OpenStreetMap Nominatim. We include other assessment measures to assess the recognition ability of place names and the prediction of what we called geographic levels (administrative jurisdiction of places).

Keywords: geoparsing; toponym resolution; geographic named entity recognition; named entity recognition in Spanish

1. Introduction

Geoparsing is a sophisticated process of natural language processing (NLP) used to detect mentions of geographical entities and to encode them in coordinates [1]. Roughly, the language is analyzed to obtain place names and to visualize their locations on maps. One way to achieve this kind of analysis is to process textual information through a pipeline in which place names are first recognized and then geolocated. Named entity recognition (NER) is an important NLP task that seeks to recognize and classify entities in predefined categories such as quantities, people, organizations, places, expressions of time, and events, among others. This research topic has become very relevant, since high-performance NER systems usually precede other complex NLP tasks including information extraction, knowledge base population, named entity linking, machine translation, and word sense disambiguation.

Particularly, the task of recognizing entities alluding to place names (also named *toponyms*) has been called geographic-named entity recognition (GNER). This is a challenging task because there is frequently a huge amount of ambiguities that make it difficult to attain the level of human performance. For instance, in the sentence *"Paris Hilton is in Paris"*, some entities could refer to a person or place name in the same text. When these ambiguities are solved, the subsequent task is to assign the geographic properties (georeferences and geographic levels) of the place names found. This problem is known as *toponym resolution* and represents another important challenge in which place name instances are linked with geographic properties, thus overcoming possible ambiguities. For example, the text *"From Paris, France to Paris, Arkansas: Why Geographic Information Science Matters"*, by the US Senator from Arkansas, John Boozman (18 November 2019), denotes a common situation that involves homonym place names of two countries: France and the USA. It would be desirable that a geoparser system relate each place in the document with its corresponding country, and more precisely with its actual geographic coordinates.

Regarding the above challenges, several works have been proposed; some of them are discussed in Section 2. We have considered some limitations of these works in designing our proposal described in Section 3. Afterwards, experiments and the corresponding results are depicted in Section 5. Finally, the main conclusions are provided in Section 6.

2. Related Work

Algorithms and methods behind geoparsing remain a very active research field, and new systems with new and better characteristics are starting to be developed. Most of the existing approaches for geoparsing are knowledge-based [2], which use external sources (vocabularies, dictionaries, gazetteers, ontologies). In general, these are heuristic-based methods (place type, place popularity, place population, place adjacency, geographic-levels, hierarchies, etc.) [3]. Both tasks toponym recognition and resolution rely strongly on the evidence contained in the used knowledge sources, among other Natural Language Processing or Machine Learning tools. Next, some of these approaches are presented.

Buscaldi and Rosso [4] described a knowledge-based word sense disambiguation method for the disambiguation of toponyms. The proposed method is based on the disambiguation of nouns by using an adaptation of the Conceptual Density approach [5] using WordNet [6] as the external knowledge resource. The Conceptual Density is an algorithm for word sense disambiguation, measures the correlation between the sense of a given word and its context taking into account WordNet subhierarchies. Authors use holonymy relationships to create subhierarchies to disambiguate locations. The method was evaluated using geographical names from the SemCor corpus available at www.sketchengine.eu/semcor-annotated-corpus (last access 26 August 2020). Michael et al. [7] presented a method for geographical scope resolution for documents. For this, the method is based on the administrative jurisdiction of places and the identificacion of named entities of type person to assign a greographic scope to documents. The method applies a set of heuristic for geographical attributes (population-based prominence, distance-based proximity, and sibling relationships in a geographic hierarchy). The method is focused on international wide scope documents within a Georgaphic Information Retrieval system. This method does not consider toponym disambiguation. In this line, Radke et al. [8] proposed an algorithm for geographical labeling of web documents considering all place names without solving possible ambiguities between them. Woodruff et al. [9] developed a method that automatically extracts the words and phrases (only in English) that contain names of geographical places and assigns their most likely coordinates. From this method, a prototype system called Gipsy (*georeferenced information processing system*) was developed.

Inkpen et al. [10] developed an algorithm that extracts expressions composed of one or more words for each place name. Authors use a conditional random fields classifier, which is based on an unguided graphical model that is used for unstructured predictions [11]. They focused on tweet location entities by defining disambiguation rules based on heuristics. The corpus contains tweets in English from the states and provinces of the United States and Canada. Middleton et al. [12] presented a comparison of location extraction algorithms, two developed by authors and three from the state-of-the-art. The author's algorithms use OpenStreetMap, and a combination of language model from social media and several gazetteers. The third-party algorithms use NER-based based on DBpedia, GeoNames and Google Geocoder API. In the OpenStreetMap approach, location entities are disambiguated using linguistic and geospatial context. To create the model in the language model approach, a corpus of geotagged Flickr posts and a term-cell map were used. To enchance the accuracy of the model, authors used some heuristics to refine it. A fine-grained quantitative evaluation was conducted on English labeled tweets taking into account streets and buildings. Karimzadeh et al. [13] described the GeoTxt system for toponym recognition and resolution from tweets in English. For place name recognition, the system can use one of six publicly available NERs: Stanford NER (nlp.stanford.edu/software/CRF-NER.html), Illinois CogComp (github.com/IllinoisCogComp/illinois-cogcomp-nlp), GATE ANNIE (gate.ac.uk/ie/annie.html), MITIE (github.com/mit-nlp/MITIE), Apache OpenNLP (opennlp.apache.org), and LingPipe (www.alias-i.com/lingpipe); the above URLs were last accessed on 26 August 2020. For place name disambiguation within the place name resolution, the system can use three disambiguation mechanisms: two hiearchical-relationship-based methods and a spatial proximity-based disambiguation method. The system uses the Geonames gazetteer. GeoTxt has global spatial scope and was optimized for English text from tweets.

An interesting work for historical documents was presented by Rupp et al. [14]. These documents were translated and transcribed from their corresponding original books. Authors use a system called VARD, for spelling correction, and a historical gazetteer. One of the drawbacks is that the names of current places in historical texts may induce ambiguities with respect to places in past centuries. In this sense, Tobin et al. [15] presented another approach wherein there are three historical corpora digitized beforehand. The authors used information extraction techniques to identify place names in the corpus. They rely on gazetteers to compare the results obtained with human annotations of the three corpora. A similar method was applied for the SpatialML corpus (catalog.ldc.upenn.edu/LDC2011T02, last access 20 August 2020), which is a geo-annotated corpus of newspapers [16] which is made up of two main modules: a geotagger and a georesolver. The first one processes input text and identifies the strings that denote place names. The second one takes the set of place names recognized as input, searches for them in one of the different gazetteers and determines the most likely georeference for each one. A disadvantage is that it does not find place names if they are misspelled, even slightly, and so it relies only on exact matches. Also, Ardanuy and Sporleder [17] presented a supervised toponym disambiguation method based on geographic and semantic features for historical documents. The method consists of two parts: toponym resolution (using information from GeoNames) and toponym linking (using information from Wikipedia). Geographic features include latitude and logitude, population, the target country, and inlinks to the Wikipedia article. Semantic features include the title of the location's Wikipedia article, the name of the country and the target region, and the context words referring to History. The method was tested on five data sets in Dutch and German.

Some approaches has been proposed for non-English documents. This is a research niche, since each language has specific language patterns that provide extra information in order to identify toponyms. Nes et al. [18] presented a system called GeLo, which extracts addresses and geographic coordinates of commercial companies, institutions and other organizations from their web domains. This is based on part-of-speech (POS) tagging, pattern recognition and annotations. The tests included web domains of organizations located in the region of Tuscany in Italy. The platform was developed in Italian and English, though authors proposed an independent language option. This system is composed of two modules: (1) a tracking tool for indexing documents, and (2) a linguistic analyzer that takes as input the web documents retrieved from the previous module. Martins and Silva [19] presented an algorithm based on PageRank [20]. Here, geographical references are extracted from the text and two geographical ontologies. The first one is based on global multilingual texts, and the second is based only in the region of Portugal. One of the limitations is that, similar to the original PageRank algorithm, they assign the same weight to all edges (references), which causes dense nodes (with many references to other sites) to tend to produce higher scores, whether or not they are unimportant. In another work, Martins and Silva [21] presented what they called *geographic scope* of web pages using a graphic classification algorithm. The geographic scope is specified as a relationship between an entity in the web domain (web page) and an entity in the geographic domain (such as an administrative location or region). The geographic scope of a web entity has the same footprint as the associated geographic entity. The scope assigned to a document is granted due to the frequency of occurrence of a term and by considering the similarity to other documents. The work was focused on feature extraction, recognition and disambiguation of geographical references. The method makes extensive use of an ontology of geographical concepts and includes an architecture system to extract geographic information from large collections of web documents. The method was tested on English, Spanish, Portuguese, and German. Gelernter and Zhang [22] presented a geoparser for Spanish translations from English. This method is an ensemble from four parsers: a lexico-semantic Named Location Parser, a rule-based building parser, a rule-based street parser, and a trained Named Entity Parser. The method was capable to recognize location words from Twitter hash tags, website addresses and names. Authors developed a parser for both languages; the NER parser is trained by using the GeoNames gazetteer and the Conditional Random Fields algorithm. The method was capable to deal with street and building names. The method was evaluated on an set of 4488 tweets collected by the authors and used the kappa statistic for human aggrement. Moncla et al. [23] proposed a geocoding method for fine-grain toponyms. The approach consists of two main parts: geoparsing based on POS tagging and syntactic-semantic rules; and a disambiguation method based on clustering of spatial density. This approach can deal with toponyms not in gazetteer. The proposal was tested on a hiking corpus obtained from websites in French, Spanish, and Italian. These documents describe displacements using toponyms, spatial relations, and natural features or landscapes.

There is a lack of NER annoted corpus for Spanish variants. In this sense, Molina-Villegas et al. [24] presented a promising project to compile a corpus of Mexican news and train a GNER model with dense vectors obtained from the corpus. This is important because, in general, the training corpora somehow restricts the outcome of GNER models. For instance, a GNER model trained exclusively with news from Spain could recognize places or local names from Spain, but not from Mexico. Though this is an inescapable fact, it is possible to obtain GNER models through a processing pipeline in which the only change is at the input corpora; that is, without making structural changes in the training process *per se*. Guided by achieving this ability, in this paper, we present a GNER framework in which the training corpus could be changed to extend its capability to recognize places in a wide set of geographic contexts. Another complex issue around geographic entities is the spatial ambiguity that arises from the fact that there are many places with the same name. In this regard, we propose what we have called *dynamic context disambiguation*, an approach based on a gazetteer and knowledge base that mimics the human process with respect to how a place mentioned in a text must be solved.

3. Geoparsing Approach

Our proposal includes two main modules: geographic-named entity recognition (GNER) and dynamic context disambiguation. The GNER module is used to detect entities alluding to place names in an unstructured text. The geographic disambiguation module solves and determines the most likely *geographic properties* (which we will discuss later) of these places. These modules, as well as their most important elements, are illustrated in Figure 1 and detailed throughout this section.

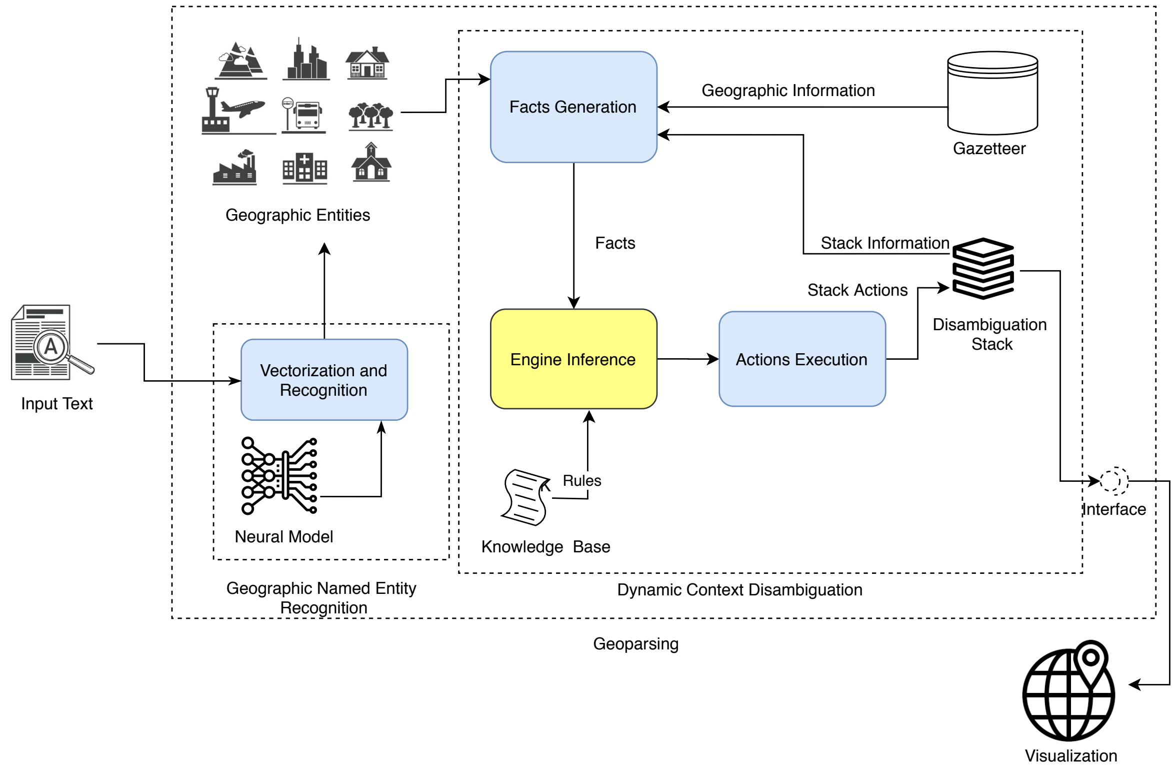


Figure 1. Proposal overview.

3.1. Geographic-Named Entity Recognition

The geographic-named entity recognition (GNER) module is based on a trained model (GNER model) whose inputs are vector representations of words, also referred to as embeddings in a semantic space. Basically, the input text is transformed into *dense vectors*, and then the GNER model determines when a specific word or n-gram is a geographic entity. A dense vector, opposite to a sparse vector, is one that has not zero entries. In the field of Natural Language Processing, representations of words in a vector space help learning algorithms to achieve better performance in specific tasks like named entity recognition. This mapping from the vocabulary to vectors of real numbers conceptually involves a mathematical embedding from a space with many dimensions (one per word which causes sparse vectors based on frequencies) to a continuous vector space with a much lower dimension. This lower dimension vector representation of a word is known as *word embedding* or *dense vectors*. It is worth mentioning that, given the lack of NLP resources for Mexican-Spanish, we deployed our own GNER module based on a fusion model of lexical and semantic features alongside a neural network classifier.

3.1.1. Preprocessing and Vectorization

First, texts are preprocessed with standard tokenization, where we preserve capital letters. This is because capitalized words are actually part of the standard lexical features for GNER; most of the time, location entities appear capitalized. According to Cucerzan and Yarowsky [25], even disambiguation can be learned from a set of unambiguous named entities. Their semi-supervised learning technique illustrates that we can greatly disentangle the problem by using a combination of lexical heuristics and classical binary classification. In summary, including lexical boolean features like *Starts with a capital letter* or *has internal period* (e.g., St.) improves the results. In consequence, not including this valuable information as part of the features must decrease the global accuracy. Other lexical features are also included as binary variables. For instance, for each token, we check whether its individual characters are numeric, the number of characters, if the token belongs to specified entities, if it is a stopword, and its part-of-speech (POS) tag, among other features.

The semantic features of this module are based on word embeddings. For this, we used the context encoder (ConEc), described in Section 5.1, to transform the original words into dense vectors.

3.1.2. Neural Network GNER Classifier

Once we obtain lexical and semantic characteristics of words, we fuse them by concatenating all features in what we call a *bag of features* and then we use a one-layer perceptron neural network for a binary classification (geographic entity or not). Although state of the art models for general NER are based on modern deep encoder-decoder architectures such as convolutional neural networks with bidirectional long-short-term-memory networks (CNN-BiLSTM) and attention mechanisms (Li, P.-H. et al. [26]), and more recently, by using language representations with transformers (BERT, Luo et al. [27], Li et al. [28]), it is well known that they require a huge amount of data in order to train the models. In our case, this is a drawback, given the lack of data for Mexican-Spanish language. For this reason, we prefer to use a simpler model, a one-layer perceptron neural network, which inputs are semantic and lexical characteristics of our corpus.

We choose our model by using a popular method for selecting models, Cross-Validation, which help us to choose the appropriate complexity of the neural network which achieves a better performance. In our case, and after testing a wide range of architectures, the best model was a neural network with one hidden layer, 3 hidden units and a sigmoid activation function with weight decay. The fact that we are using a simple model for classification, means we are obtaining a good representation of text based on the features we are using, i.e., semantic and lexical characteristics, and it is not necessary to train a complex model in order to obtain good results. At the end of the neural network GNER classifier, the last layer determines, for each token, one of the two possible classes `{<location>,<word>}`. Finally, for entities composed of two or more tokens, we use a heuristic to reconstruct the whole entity by using the class of the tokens in the original text. That is, two or more consecutive `<location>` labels will be considered as one single location entity. The labeled version of the original text is sent to the dynamic context disambiguation module.

3.2. Dynamic Context Disambiguation

Our disambiguation approach derives decisions based on rules and facts. The rules specify how ambiguities could be solved by considering the context in a similar manner as that of a person. Given that the rules are activated as needed in execution time, we call it *dynamic context disambiguation*. This involves a set of elements that are illustrated in Figure 1 and described in what follows:

- Knowledge base (KB): A set of rules which mimic human knowledge about how a place mentioned in a text must be solved. A rule is an *IF-THEN* clause with an antecedent and a consequent. The IF part of the clause is made up of a series of clauses connected by a logic AND operator that causes the rule to be applicable, while the THEN part is made up of a set of actions to be executed when the rule is applicable.
- Gazetteer: Reservoir of places and geographic information.
- Facts generation (FG): Creates instances of facts defined in the rules. The instances are deduced from a grammar and the Gazetteer on the fly during the execution of the toponym resolution algorithm.
- Inference engine (IE): Determines what rules should be triggered. Then, when a rule is triggered, other facts could be generated; these, in turn, would trigger other rules which make the IE context dynamic.
- Action execution (AE): Executes the consequents of the rules that have been activated. This execution involves read and write operations on a disambiguation stack.
- Disambiguation stack (DS): Data structure in which the entities that have been solved are stored. By "solved," we mean that their most likely geographic locations have been determined.

- Visualization: Once all place names have been solved, this module allows us to visualize, on a map, the resulting place names in DS.

3.2.1. Notation

The rules in KB specify how ambiguities could be solved by considering the context, in a similar manner as that of a person. To describe them and show the way in which these are activated, we will use the following notation:

$\mathbb{G}$	A gazetteer containing tuples of the form: $(place\ name, location, geographic\ level, parent\ geographic\ level)$.
R_i	The ith rule defined as a Horn clause of the form $P_1 \land P_2 \land \ldots \land P_k \implies Q$, where P_i is a fact and Q is an action or set of actions to be executed when the antecedent is true.
$\mathbb{D}$	Input text document containing the entities to be georeferenced.
A, B	Geographic entities in $\mathbb{D}$.
$\mathbb{S}$	Stack data structure, in what follows *disambiguation stack*, wherein the entities will be stored as soon as the rules are activated and executed.
$\mathbb{C}$	Auxiliary stack used in situations, in which we lack information to solve the ambiguities relative to a place name.

The properties *location, geographic level and parent level* are denoted in what follows *geographic properties*. Location is defined in terms of latitude and longitude. Geographic level and parent geographic level are nominal values corresponding to an administrative division (provinces, states, counties, etc.), such as those shown in Appendix A.

3.2.2. Base Facts and Rules

As we pointed out, KB is a set of rules representing situations of how a place or entity mentioned in a text should be solved. Rules are Horn clauses that consist of *facts* and *goals*. A *fact* is a clause P_i representing a condition, while a *goal* is a clause Q_i representing an action to be executed. For any rule, the sets of facts and goals correspond to its *antecedent* and *consequent*, respectively. For our purposes, we have defined the following set of base facts, from which the rules in KB are defined.

P_1 A matches a location name in $\mathbb{G}$.

P_2 The disambiguation stack is empty.

P_3 A has a more specific administrative level than B (but A is not necessarily contained in B).

P_4 There is a relationship between A and B, meaning that there is a path between A and B in the administrative hierarchy.

P_5 There are no entities in $\mathbb{D}$ to be processed.

P_6 There are elements in $\mathbb{C}$ that must be processed.

The set of rules that guides the stages of the disambiguation process are defined in Table 1.

Table 1. Set of rules for dynamic context disambiguation.

RULE	DESCRIPTION	ANTECEDENT	CONSEQUENT
R_0	The entity to be processed (A) has a match in $\mathbb{G}$ and $\mathbb{S}$ is empty.	$P_1 \wedge P_2 \implies$	Q_1 Assign A with the geographic properties of the matched location in $\mathbb{G}$ that has the highest hierarchy (relative to the administrative level). Q_2 Push A on $\mathbb{S}$.
R_1	The entity to be processed (A) has a match in $\mathbb{G}$, $\mathbb{S}$ is not empty, A is lower than the entity (T) at the top of $\mathbb{S}$ and there is a relationship between A and T.	$P_1 \wedge \neg P_2 \wedge P_3 \wedge P_4 \implies$	Q_1 Assign A with the geographic properties of the matched location in $\mathbb{G}$ whose parent code is equal to the parent code of $\mathbb{T}$'. Q_2 Push A on $\mathbb{S}$.
R_2	The entity to be processed (A) has a match in $\mathbb{G}$, $\mathbb{S}$ is not empty, A is lower than the entity (T) at the top of $\mathbb{S}$ and there is no relationship between A and T.	$P_1 \wedge \neg P_2 \wedge P_3 \wedge \neg P_4 \implies$	Q_1 Assign A with the geographic properties of the matched location in $\mathbb{G}$ that has the highest hierarchy (relative to the administrative level). Q_2 Push A on $\mathbb{S}$.
R_3	The entity to be processed (A) has a match in $\mathbb{G}$, $\mathbb{S}$ is not empty, A is not lower than the entity (T) at the top of $\mathbb{S}$ and there is a relationship between A and T.	$P_1 \wedge \neg P_2 \wedge \neg P_3 \wedge P_4 \implies$	Q_1 Assign a new entity B with T. Q_2 Pop $\mathbb{S}$. Q_3 Assign A with the geographic properties of the matched location in $\mathbb{G}$ that has the highest hierarchy. Q_4 Push A on $\mathbb{S}$. Q_5 Push B on $\mathbb{S}$.
R_4	The entity to be processed (A) does not have a match in $\mathbb{G}$, and $\mathbb{S}$ is not empty.	$\neg P_1 \wedge \neg P_2 \implies$	Q_1 Assign A with the geographic properties of T (relationship creation). Q_2 Push A on $\mathbb{S}$.
R_5	The entity to be processed (A) does not have a match in $\mathbb{G}$, and $\mathbb{S}$ is empty.	$\neg P_1 \wedge P_2 \implies$	Q_1 Push A on $\mathbb{C}$ (conflicts stack).
R_6	The entity to be processed (A) has a match in $\mathbb{G}$, $\mathbb{S}$ is not empty, A is not lower than the entity (T) at the top of $\mathbb{S}$ and there is no relationship between A and T.	$P_1 \wedge \neg P_2 \wedge \neg P_3 \wedge \neg P_4 \implies$	Q_1 Assign A with the geographic properties of the matched location in $\mathbb{G}$ that has the highest hierarchy. Q_2 Push A on $\mathbb{S}$.
R_7	There are no more entities from the text to be processed, but there are still entities in the conflicts stack $\mathbb{C}$.	$P_5 \wedge \neg P_6 \implies$	Q_1 Dump stack $\mathbb{S}$ into a new stack ($\mathbb{S}'$) so that the bottom of $\mathbb{S}$ becomes the top of $\mathbb{S}'$. Q_2 Pop T_s from $\mathbb{S}'$. Q_3 Obtain the child of T_s with the lowest level and assign its geographic properties to all entities in the conflicts stack $\mathbb{C}$.

3.3. Geoparsing Process

After a set of place names have been found in $\mathbb{D}$ by the GNER module, dynamic context disambiguation must assign their most likely geographic properties. This involves the execution of the KB and FG modules, orchestrated by the IE (see Figure 1), so this last one determines which rules must be activated and applied through the AE module.

The above corresponds to the whole process of *geoparsing*, which we have formalized in Algorithm 1, where the function ***entity_recognition*** encompasses the process of GNER from which a list $\mathbb{L}$ containing geographic entities is obtained. The function ***rules_inference*** identifies the most likely geographic properties of each entity $e \in \mathbb{L}$, based on the information in $\mathbb{G}$ and the current state of $\mathbb{S}$. This information generates instances of base facts (via FG) that are considered by IE to determine the rules that must be activated and executed by AE. As a result of this execution, the state of $\mathbb{S}$ is changed, containing at this point solved entities. If a conflict is found as a consequence of the above process, e is pushed on $\mathbb{C}$, where by the term conflict we mean that there is no information to assign the geographic properties of e. Finally, when there are no more entities to be processed, the function ***solve_conflicts*** is called, which assigns the most suitable geographic properties to those entities belonging to $\mathbb{C}$.

```
Algorithm 1: Geoparsing algorithm
  Data:
  $\mathbb{D}$: Document,
  $\mathbb{G}$: Gazetteer,
  Result: Toponym Resolution over $\mathbb{D}$
  /* Initialization of Disambiguation stack and Conflicts stack */
  $\mathbb{S} \leftarrow \emptyset$;
  $\mathbb{C} \leftarrow \emptyset$;
  /* Geographic-Named Entity Recognition */
  $\mathbb{L}$=entity_recognition($\mathbb{D}$);
  /* Geographic-Named Entity Disambiguation */
  foreach $e \in \mathbb{L}$ do
      rules_inference($e$,$\mathbb{S}$,$\mathbb{G}$);
      if there is a conflict then
          $\mathbb{C}$.push($e$)
      end
  end
  solve_conflicts($\mathbb{C}$,$\mathbb{S}$);
  return $\mathbb{S}$
```

Geoparsing Example

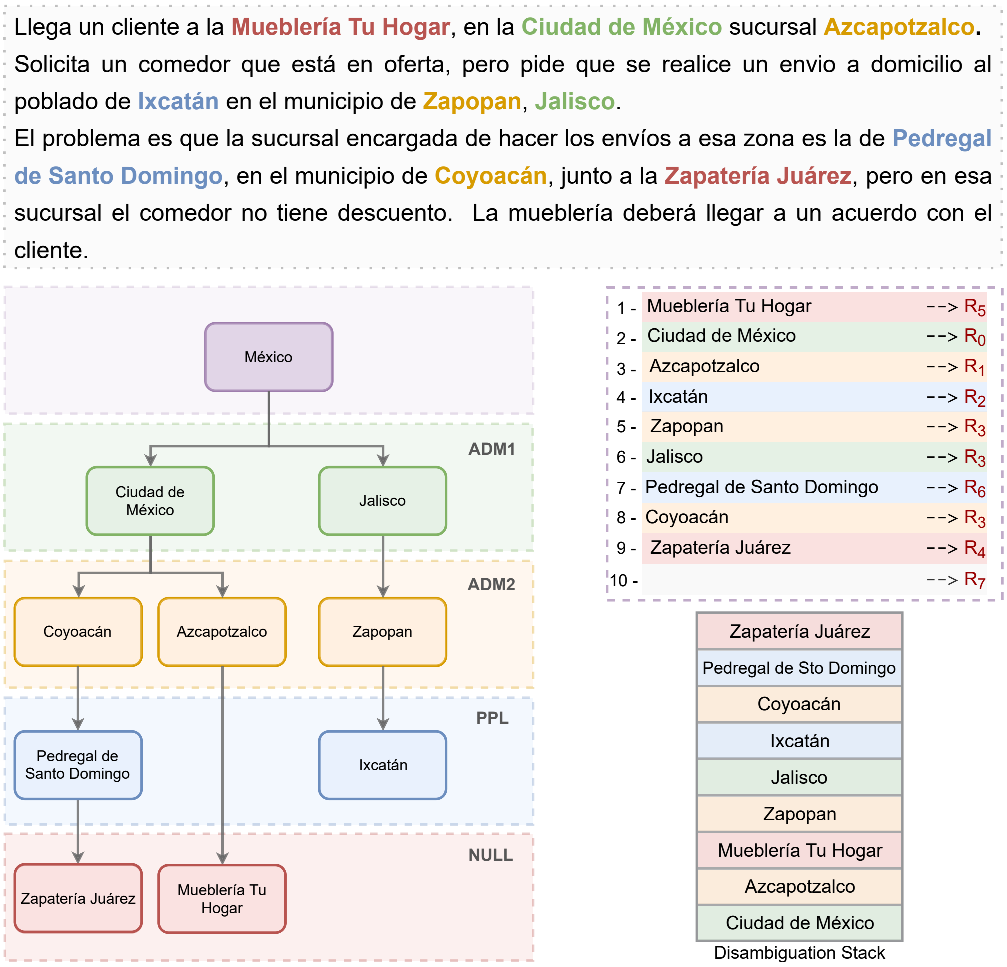
To illustrate the execution of Algorithm 1, we provide the following example. The input text is presented at the top of Figure 2, emphasizing the geographic entities. An approximate translation reads thus: *A customer arrives at **Tu Hogar furniture store**, branch **Azcapotzalco** in **Mexico City**. He requests a table that is on sale and he asks the delivery to be at the town of **Ixcatán** in the municipality of **Zapopan**, **Jalisco**. The problem is that the branch in charge of delivering to that area is the **Pedregal de Santo Domingo** one, in the municipality of **Coyoacán**, next to the **Zapatería Juárez**, but in that branch the table is not on sale. The furniture store should reach an agreement with the client*. On the left side of Figure 2, we have included the inherent hierarchy derived from prior classification based on the geographic levels defined in Appendix A. The top right presents the list of rules activated and applied during the execution of Algorithm 1. Finally, the bottom right shows the final state of $\mathbb{S}$.

Algorithm 1 begins by determining the place names contained in the text, in which case the list $\mathbb{L}$ is initialized via the function ***entity_recognition***. For each entity in $\mathbb{L}$, the function ***rules_inference*** is iteratively called. The following actions are executed at each stage.

1. The entity in $\mathbb{L}$ to be processed (*Mueblería Tu Hogar*) does not have a match in $\mathbb{G}$ and, at this point, $\mathbb{S}$ is empty. This means that there is not enough information to determine the geographic properties of the entity, in which case the rule R_5 is activated and applied. As a consequence, the conflict condition is met, and so the entity is pushed onto $\mathbb{C}$.
2. The next entity in $\mathbb{L}$ (*Ciudad de México*) has several matches in $\mathbb{G}$, from which the one with the highest geographic level is taken and pushed onto $\mathbb{S}$ according to R_0.
3. At this point, the entity in $\mathbb{L}$ to be processed (*Azcapotzalco*) has a match in $\mathbb{G}$, with *Ciudad de México* being the top of $\mathbb{S}$, so the rule R_1 is activated, in which case, *Azcapotzalco* is pushed onto $\mathbb{S}$ as a child entity of *Ciudad de México*.
4. The entity to be processed (*Ixcatán*) has two matches in $\mathbb{G}$. In this case, these names have the same geographic level, in which case, either one can be selected and pushed onto $\mathbb{S}$, according to R_2.
5. At this point, the top of $\mathbb{S}$ is *Ixcatán* and the entity in $\mathbb{L}$ to be processed is *Zapopan*. This has two matches in $\mathbb{G}$, and thus the place with the highest feature level is selected to be added to $\mathbb{S}$. However, this place (*Zapopan*) is not lower than the top of $\mathbb{S}$ (*Ixcatán*), though there is a relationship between them. This situation causes rule R_3 to be triggered.

6. Similarly, the entity in $\mathbb{L}$ to be processed (*Jalisco*) is not the predecessor of the current top (*Ixcatán*), and there is a relationship between them, so the rule R_3 is applied again.
7. Then, given that the entity in $\mathbb{L}$ to be processed (*Pedregal de Santo Domingo*) does not have any relationship with the top of $\mathbb{S}$ (*Ixcatán*), the rule R_6 proceeds.
8. At this point, the top of $\mathbb{S}$ is *Pedregal de Santo Domingo*. When the entity in $\mathbb{L}$ to be processed (*Coyoacán*) is evaluated, it happens that the top of $\mathbb{S}$ is its predecessor, and so R_3 must be applied.
9. The entity in $\mathbb{L}$ to be processed (*Zapatería Juarez*) appears in the text, but not in $\mathbb{G}$, implying R_4.

Finally, the function ***solve_conflicts*** is invoked in order to solve the remaining entities in $\mathbb{C}$ through the rule R_7.



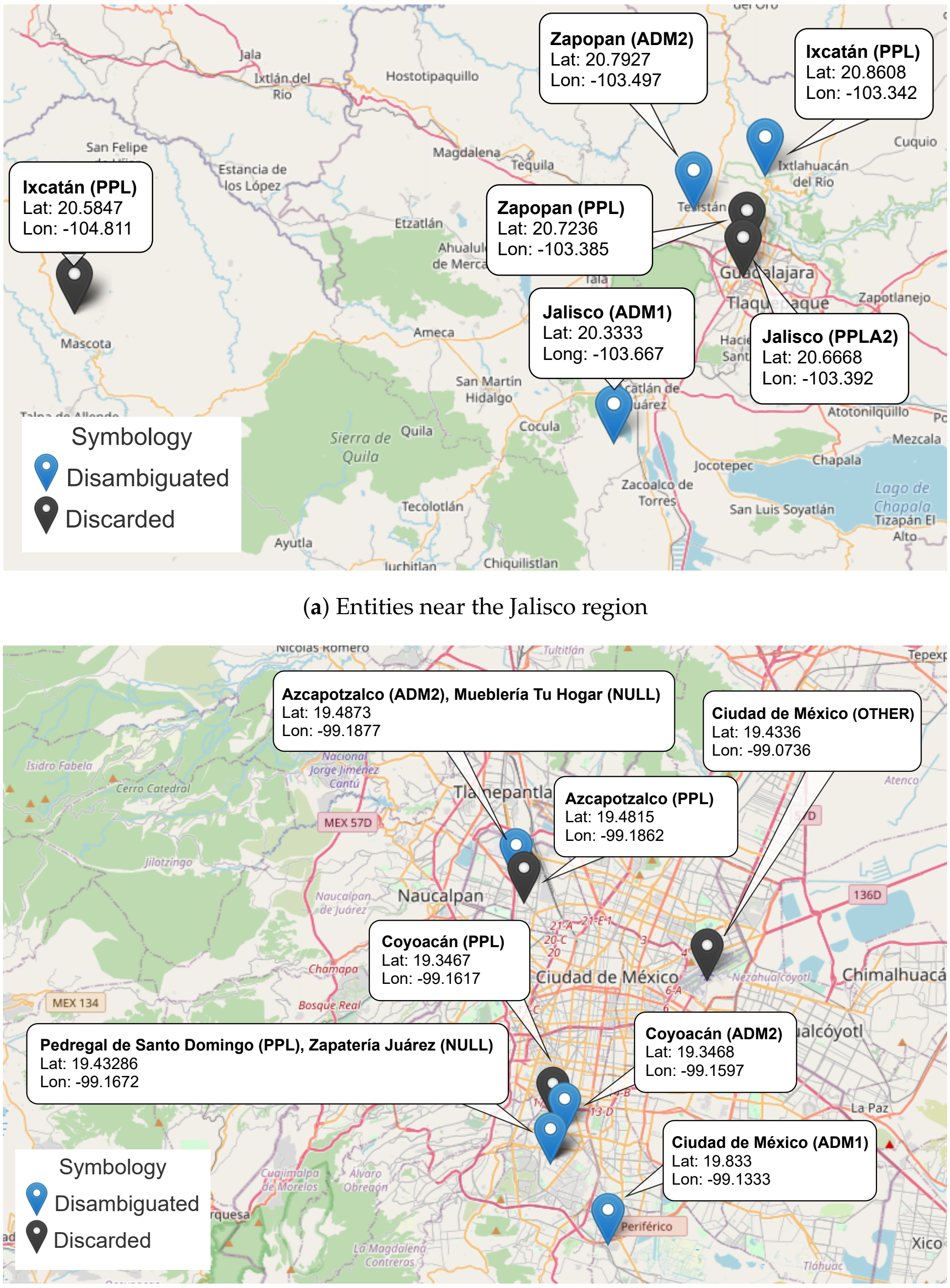
Llega un cliente a la **Mueblería Tu Hogar**, en la **Ciudad de México** sucursal **Azcapotzalco**. Solicita un comedor que está en oferta, pero pide que se realice un envio a domicilio al poblado de **Ixcatán** en el municipio de **Zapopan**, **Jalisco**.
El problema es que la sucursal encargada de hacer los envíos a esa zona es la de **Pedregal de Santo Domingo**, en el municipio de **Coyoacán**, junto a la **Zapatería Juárez**, pero en esa sucursal el comedor no tiene descuento. La mueblería deberá llegar a un acuerdo con el cliente.

1 -	Mueblería Tu Hogar	--> R_5
2 -	Ciudad de México	--> R_0
3 -	Azcapotzalco	--> R_1
4 -	Ixcatán	--> R_2
5 -	Zapopan	--> R_3
6 -	Jalisco	--> R_3
7 -	Pedregal de Santo Domingo	--> R_6
8 -	Coyoacán	--> R_3
9 -	Zapatería Juárez	--> R_4
10 -		--> R_7

Figure 2. Example of toponym resolution processes based on dynamic context identification for a text in Spanish. The left diagram represents the inherent hierarchy derived from the text. The top right lists the rules applied to every entity, and the bottom right shows the final state of $\mathbb{S}$.

The map presented in Figure 3 shows all entities with their corresponding geographic properties (location and geographic levels). For simplicity, the Figure 3 is presented in two parts. This is because the two regions involved are so far apart that it is convenient to focus on them separately. In Figure 3a, all of the entities near the *Jalisco* region are mapped. Note that there are two toponyms of *Jalisco* in the gazetteer, and so the algorithm has disambiguated *Jalisco* (ADM1) as the correct one. Another case is when there are two entities with the same name and geographic level; this is the case for *Zapopan* and *Ixcatán*. Algorithm 1 would determine the most likely entity according to the rule activation; this situation is illustrated by the relationship between the instances of *Zapopan* (ADM2) and *Ixcatán* (PPL) that are indeed the nearest. Similar issues appear in Figure 3b for *Ciudad de México* and *Coyoacán*.

In addition to the mentioned issues, in Figure 3b, we observe that, for those entities that are not included in the gazetteer (*Mueblería Tu Hogar* and *Zapatería Juarez*), Algorithm 1 deduces the most likely location. This is just an example to exhibit the feasibility of our proposal. In Section 5, we will show the results of applying this proposal to a massive volume of news.



(**a**) Entities near the Jalisco region

(**b**) Entities near Ciudad de México (Mexico City)

Figure 3. Geolocated entities found after disambiguation algorithm. The disambiguated entities preserve the inherent hierarchy after dynamic activation of rules. Entities tagged as discarded correspond to alternative locations detected by Algorithm 1.

4. Data and Annotation Criteria

The assessment of our method included three different corpora. The first corpus C1 was used to produce word embeddings. As we will see in Section 5.1, this is necessary to feed a neural network in order to create the GNER model.

The second corpus C2 is the Corpus of Georeferenced Entities of Mexico (CEGEOMEX) (http://geoparsing.geoint.mx/mx/info/, last access 20 August 2020) reported in the project [24]. The corpus was annotated manually with geographic-named entities and is, as far as we know, the only existing data source in Mexican-Spanish for GNER. CEGEOMEX was labeled according to the following criteria:

- Named entities are considered geo-entities only when it is possible to assign them geographic coordinates.
- A place name composed of many words, expressions or other entities must be spanned by a single label. For example, `<loc>Hermosillo, Sonora</loc>` must be a single geo-entity.
- Imprecise references such as *"at the south of"* or *"on the outskirts of"* must not be included as part of the geo-entity. For instance, in *"on the outskirts of Mexico"*, only `<loc>Mexico</loc>` must be labeled.
- Place names that are part of other entity types such as *"the president of Mexico"* or *"Bishop of Puebla"* or the band name *"Los tucanes de Tijuana"* must not be labeled.
- The names of sports teams accompanied by names of places such as countries, states or municipalities, such as *"Atletico de San Luis"*, must not be considered place names. Instead, the names of the stadiums, sports centers or gyms, such as `<loc>Estadio Azteca</loc>`, must be considered place names.
- Country names must be considered as place names when they point to the territory, but they must not be labeled when they refer to the government. For example, in *'California sends help to those affected by the earthquake in Mexico City"*, only `<loc>Mexico City</loc>` must be labeled.

Finally, a corpus C3 containing news not included in the training and validation process of GNER was used to test and validate the disambiguation process. Unlike previous corpora, the tagging process included the geographic properties (location and geographic level) of each tagged entity. As a summary, a description of the above corpora is shown in Table 2.

Table 2. Corpora description.

Data	Description	Num. of Docs	Tags	Num. of Entities	Purpose
C1	News documents from the main digital media in Mexico.	165,354	None	Unknown	Word Embeddings
C2	News documents from the main digital media in Mexico.	1233	GNER	5870	GNER Model
C3	Documents from a news agency (El gráfico) which is not included in the above corpora.	500	GNER Actual Coordinates Geographic levels	1956	Disambiguation

5. Experiments and Results

The experiment consisted of assessing the performance of our geoparsing approach in terms of three assessments: (1) the recognition ability of the GNER module in terms of standard evaluation metrics (accuracy, precision, recall and F-measure), (2) the accuracy of geographic level predictions by the toponym resolution module; and (3) a ranking consisting of six categories of closeness, where by the term closeness we mean the distance from the predicted location to the actual location.

5.1. Geographic-Named Entity Recognition

We have obtained the semantic features based on word embeddings obtained with `word2vec` [29]. Although `word2vec` offer computational advantages compared to other methods such as `Glove` [30] or `fastText` [31], there are two main drawbacks: it is not possible to obtain embeddings for out of vocabulary (OOV) words, and it cannot adequately represent words with several meanings.

OOV words can be solved with `fastText`, but in order to tackle both problems, we used an extension of `word2vec` based on the context encoder (ConEc) [32]. The ConEc training is identical to that of CBOW-`word2vec`, with a difference in the calculations of the final embeddings after the training is completed. In the case of `word2vec`, the embedding of a word w is simply its corresponding row of the weight matrix W_0 obtained in the training process, while with *ConEc* the embedding is obtained by multiplying W_0 with the mean context vector x_w of the word, such that the *local* and *global* context of a word are distinguished. The global vector is obtained with the mean of all binary context vectors x_{w_i} corresponding to the M_w occurrences of w in the training corpus, according to $x_{w_{global}} = \frac{1}{M_w} \sum_{i=1}^{M_w} x_{w_i}$, while the local context vector is computed according to $x_{w_{local}} = \frac{1}{m_w} \sum_{i=1}^{m_w} x_{w_i}$, where m_w corresponds to occurrences of w in a single document. The final embedding y_w of a word is obtained with a weighted sum of both contexts, as defined in Equation (1):

$$y_w = (\alpha \cdot x_{w_{g}lobal} + (1 - \alpha) \cdot x_{w_{l}ocal})^T W_0, \tag{1}$$

where $\alpha \in [0, 1]$ is a weight that determines the emphasis on the local context of a word. For OOV words, their embeddings are computed solely based on the local context, i.e., setting $\alpha = 0$.

After an extensive search for classifiers and their parameters with cross validation, we decided to use a simple neural network classifier. The GNER neural network classifier has 1 hidden layer with 3 hidden units and a sigmoid activation function with weight decay. The reason to utilize this type of classifier (instead of a complex one) is that we obtained a good representation of the texts based on the bag of features we used, in such a way that it is possible to discriminate geographic entities. It is sufficient to use a simple classifier, such as the one we used, or even a support vector machine with a linear kernel. For entities composed of two or more tokens, we used a heuristic to reconstruct the whole entity by using the class of the tokens in the original text. Details can be found in [33].

The results of three different *encoders* are presented in Table 3. The best performance for GNER was obtained using the global and local context encoders. However, the local context encoder was useful to obtain embeddings for words outside of the vocabulary. The full procedure and results have been documented in [33].

Table 3. Results of three different encoders for geographic-named entities recognition.

	Accuracy	Precision	Recall	F-Measure
word2vec	0.9454	0.3953	0.07545	0.5348
Global Encoder	0.9633	0.7085	0.5663	0.8071
Global & Local Encoder	0.9626	0.7055	0.5761	0.8093

5.2. Geographic Level Assignment

To evaluate the geographic level assignment accuracy of our method, we compared 1956 entity levels from the corpus C3. The resulting confusion matrix is presented in Table 4. Each column of the matrix represents the instances in the predicted levels (using the levels in the gazetteer of Table A1), while rows represent the actual geographic level. Note that in all cases the maximum is found in the main diagonal, suggesting that, in general, the algorithm makes the correct assignment of the geographic level to the entities. This was corroborated using the metrics presented in Table 5, where a remarkable performance of the algorithm regarding the geographic level assignment is observed, with a global accuracy of 0.9089.

Table 4. Confusion matrix for geo-entity class assignment after dynamic disambiguation. The geographic levels follow the GeoNames convention (www.geonames.org/export/codes.html, last access 20 August 2020).

	ADM1	**PPLA**	**ADM2**	**PPLA2**	**PPL**	**LCTY**	**OTHER**	**NULL**
ADM1	222	1	2	0	8	0	1	2
PPLA	3	49	2	0	0	0	1	0
ADM2	6	5	655	0	16	0	1	3
PPLA2	2	3	0	73	0	0	0	1
PPL	2	0	6	1	265	0	1	11
LCTY	0	0	0	0	0	1	0	0
OTHER	14	0	4	1	4	0	26	5
NULL	17	5	17	2	28	0	2	488

Table 5. Results of geographic category disambiguation over a news corpus with 1956 entities.

	Accuracy	**Recall**		**F-Measure**		**Precision**	
Unseen		Micro	Macro	Micro	Macro	Micro	Macro
News corpus	0.9089	0.9089	0.7489	0.9089	0.7493	0.9089	0.7634

5.3. Ranking of Closeness

As we have seen in Section 5.2, our geographic entity disambiguation method is very accurate with respect to the assignment of the geographic level of entities. Nevertheless, there is a natural question that arises when looking at those results: What if the geographic level is correct, but the real coordinates are not? To address this issue, we propose to carry out the evaluation one step further by measuring the distance from the coordinates determined by the proposed algorithm to the actual point.

It is difficult to define, in general, what is the correct location of a place. Locations could have a variety of shapes and sizes. Specific points, lines, multi-lines, polygons, among other shapes cannot be treated with the same criteria for evaluation of geoparsing. This is still an open issue. In this regards, we used ranges of distances (in km) between the point provided by our method and a manually annotated point (our actual coordinate) to decide if a location is correct or not.

As a baseline, we used the Nominatim importance score (https://nominatim.org/, last access 20 August 2020) to compare its results against our disambiguation approach. Nominatim is a search engine to search OpenStreetMap (OSM) data for locations by name or by address. Nominatim uses some heuristics in order to determine the priority of each response that could match with the query. The heuristics used by Nominatim include lexical similarity (between the query and OSM data), the bounding box of the current map (when used in a web interface) and a location importance estimation called the *importance score*. The importance score is used to rank the results in a Nominatim query according to their prominence in Wikipedia.

To contrast our approach against Nominatim, we just obtain the coordinates of the best guess for both methods. In the case of Nominatim, this corresponds to the coordinate values (latitude, longitude) of the first entity in the ranking, if any. Our method assigns the coordinates values after the algorithm described in Section 3 is executed. Note that a key feature of our approach is that it always obtains coordinates, even when the entities are not found in the gazetteer.

In Table 6, we contrast the coordinates of Nominatim vs. our method in terms of a standard distance (haversine distance) given in kilometers. Each row corresponds to a range from 0 to less than 5 kilometers to 100 kilometers or more. The columns are divided into two sections. In the first section (entitled All Entities), the distribution of ranges includes the not georeferenced category, which corresponds to the case when Nominatim does not assign any coordinates. The same information is plotted in Figure 4, where we can observe that our method is able to assign very close coordinates (approximately 5 km) in more than half of the cases. Furthermore, in 75% of cases, the assigned point will be within 25 km of the actual point. Assuming as a *correctly located* place the one whose distance range among the actual and predicted coordinates is less than 1 Km, we found that the proportion of correctly located places is 51% compared to 23% obtained by Nominatim.

In the second section (entitled Only Georeferenced Entities) of Table 6, entities not found in the gazetteer are left outside of the distribution. However, as is presented in Figure 5, while this represents a slight improvement of the baseline, the same precision of our proposal prediction accuracy for coordinates holds.

Table 6. Distance from the coordinates of the geographic entities determined by the proposed disambiguation method to the actual point. The results are compared with Nominatim importance score (baseline).

	All Entities				**Only Georeferenced Entities**			
	Proposed Method		**Nominatim**		**Proposed Method**		**Nominatim**	
Distance Range to the Actual Coordinates (km)	No. of Entities		No. of Entities		No. of Entities		No. of Entities	
[0–5)	1100	56.24%	446	22.80%	1100	56.24%	446	32.65%
[5–25)	367	18.76%	313	16.00%	367	18.76%	313	22.91%
[25–50)	75	3.83%	60	3.06%	75	3.83%	60	4.39%
[50–100)	136	6.95%	101	5.16%	136	6.95%	101	7.39%
[100–2000)	278	14.21%	446	22.80%	278	14.21%	446	32.65%
Not georeferenced	0	0%	594	30.36%	-	-	-	-

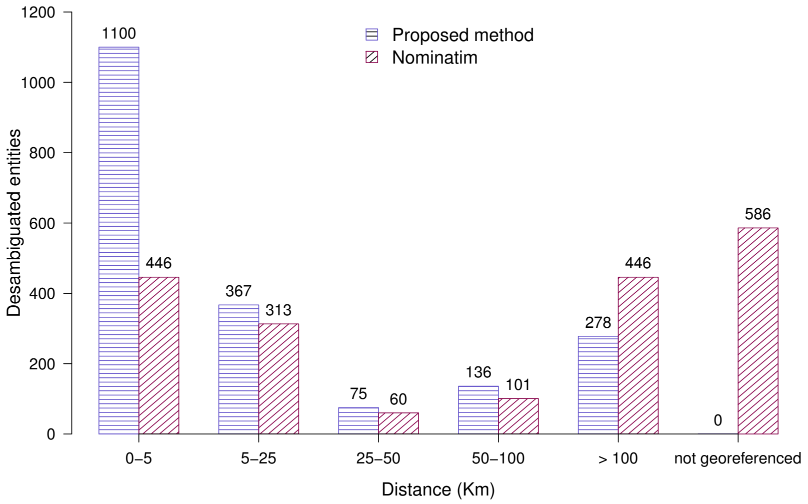


Figure 4. Frequencies of toponyms in ranges according to their distances to the actual points for all entities.

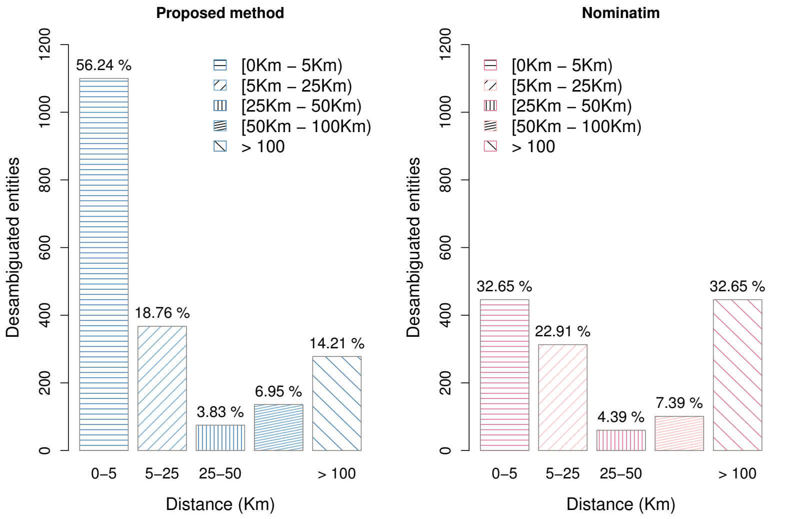


Figure 5. Frequencies of toponyms in ranges according to their distances to the actual points only for entities found in the gazetteer.

6. Conclusions and Future Work

The assignment of geographic coordinates to place names in text is a challenge in NLP and data mining, which is desirable to solve before other tasks such as information retrieval, information extraction, and summarization, among others. In this paper, we presented a novel geoparsing approach based on word embeddings for toponym recognition and dynamic context identification for toponym resolution. The approach is composed of the geographic-named entity recognition and dynamic context disambiguation modules. In the first module, a neural network model is trained by means of dense vectors to recognize geographically named entities. In the second module, a set of rules and facts is applied to take advantage of context to assign the most suitable geographic level to place names, and then to identify the correct locations.

Experimentation was carried out to determine the performance improvement of our method over a well-known baseline (OpenStreetMap Nominatim). This experimentation included a annotated corpus in which the geographic properties (georeferences and geographic levels) of the entities are known beforehand. In the experiments, our approach led to promising results, outperforming the baseline. The performance was given in terms of two metrics: (1) accuracy relative to the geographic level and (2) distance between the georeferences assigned by our method and prior georeferences. Our method manages to georeference 75% of entities in a range of 0 to 25 km and 50% within less than 5 km. In real-time applications, this type of result can be highly relevant; for example, for linking emergency medical systems [34], event detection on social media [12], and place detection in location-based services [35], among others.

We have proposed a geoparsing method that allows us to recognize locations from a text and assign their most likely geographical properties. For location recognition we have trained a machine learning model which predicts those words in the input corresponding to locations. Once the locations have been recognized, our method attempts to find their geographical properties through an inference process based on a set of facts and rules, and a gazetteer. Though our method was successfully tested on Mexican-Spanish, this can be adapted to other languages taking into account the following remarks: (a) A GNER model must be used or trained for a different language (or Spanish variant); (b) a gazetteer containing an appropriate set of places should be included (OpenStreetMap is suitable for English and Spanish); (c) the disambiguation process does not require modifications because the facts and rules do not depend on the target language. Finally is worth mentioning that, our proposal allows us to assign the geographical properties of specific locations not contained within the gazetteer which is very helpful in applications like gazetteer enrichment and could also be beneficial for the study of historical texts where existent gazetteers are limited.

The application use of the created system is around the phenomenon of clandestine graves in Mexico which has been scarcely addressed from the scientific point of view. In the project, border research questions are raised that will facilitate the creation of a clandestine grave search protocol that takes advantage of the scientific knowledge generated. The proposal tackles two large complementary problems around the search for clandestine graves. On the one hand, the development of potential distribution models generated from a machine learning approach is proposed. A relevant contribution in this area with respect to previously developed approaches is that includes the development of a Geoparser specialized in extracting information on the location of clandestine graves from journalistic reports and official documents of public access. Then other techniques and concepts of geospatial modeling will be incorporated such as proximity analysis and network analysis.

Author Contributions: I.L.-A., E.A.-B., V.M.-S. and A.M.-V.; Methodology, I.L.-A., E.A.-B., V.M.-S. and A.M.-V.; Software, E.A.-B., S.R.-P., V.M.-S. and J.A.-T.; Validation, E.A.-B., V.M.-S., S.R.-P. and J.A.-T.; Investigation, I.L., E.A.-B., V.M.-S. and A.M.-V.; Resources, A.M.-V., V.M.-S.; Data curation, A.M.-V., E.A.-B. and V.M.-S.; Writing—original draft preparation, I.L.-A., E.A.-B., V.M.-S., A.M.-V., J.A.-T. and S.R.-P.; Writing—review and editing, I.L.-A., E.A.-B., V.M.-S. and A.M.-V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Geographic Levels Based on GeoNames

Table A1. Geographic levels based on GeoNames (www.geonames.org/export/codes.html, last access 20 August 2020). We have included the level OTHER to agglomerate those specific places that do not lie within the administrative levels of first, second and third order; we have included NULL to group those places that are not found in the gazetteer.

Geographic Level	Description	Examples
ADM1	first-order administrative division	States Departments Provinces
PPLA	seat of a first-order administratie division	Capital city in a country Capital city in a state Capital city in a province
ADM2	second-order administrative division	Counties Districts Municipalities
PPLA2	seat of a second-order administrative division	City council Village council Community council
PPL	small populated place	Village Town
LCTY	a minor area of unspecified character and indefinite boundaries	Basin of Hudson river Valley of Flowers El Zapote
OTHER	specific places	Volcán Popocatepetl Túnel de Tenecoxco Arroyo el Zapote
NULL	places not found in the gazetteer	Centro de Justicia Penal Federal de Puebla Autopista Puebla – Orizaba Hospital Regional de Apatzingán

References

1. Aguirre, E.; Alegria, I.; Artetxe, M.; Aranberri, N.; Barrena, A.; Branco, A.; Popel, M.; Burchardt, A.; Labaka, G.; Osenova, P.; et al. *Report on the State of the Art of Named Entity and Word Sense Disambiguation*; Technical Report 4; Faculdade de Ciências da Universidade de Lisboa on behalf of QTLeap: Lisboa, Portugal, 2015.
2. Andogah, G.; Bouma, G.; Nerbonne, J. Every document has a geographical scope. *Data Knowl. Eng.* **2012**, *81–82*, 1–20. [CrossRef]
3. Gritta, M.; Pilehvar, M.; Collier, N. A pragmatic guide to geoparsing evaluation. *Lang. Resour. Eval.* **2020**, *54*, 683–712. [CrossRef] [PubMed]
4. Buscaldi, D.; Rosso, P. A conceptual density-based approach for the disambiguation of toponyms. *Int. J. Geogr. Inf. Sci.* **2008**, *22*, 301–313. [CrossRef]
5. Agirre, E.; Rigau, G. Word sense disambiguation using conceptual density. In Proceedings of the 16th Conference on Computational Linguistics, Copenhagen, Denmark, 5–9 August 1996; pp. 16–22.
6. Miller, G. WordNet: A lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]
7. Michael, H.; Lieberman, D.; Sankaranayananan, J. Geotagging: Using proximity, sibling, and prominence clues to understand comma groups. In Proceedings of the 6th Workshop on Geographic Information Retrieval. ACM, Zurich, Switzerland, 18–19 February 2010; pp. 1–8.
8. Radke, M.; Gautam, N.; Tambi, A.; Deshpande, U.; Syed, Z. Geotagging Text Data on the Web A Geometrical Approach. *IEEE Access* **2018**, *06*, 30086–30099. [CrossRef]
9. Woodruff, A.; Plaunt, C. GIPSY: Automated Geographic Indexing of Text Documents. *J. Am. Soc. Inf. Sci.* **1996**, *45*. [CrossRef]

10. Inkpen, D.; Liu, J.; Farzindar, A.; Kazemi, F.; Ghazi, D. Location detection and disambiguation from twitter messages. *J. Intell. Inf. Syst.* **2017**, *49*, 237–253. [CrossRef]
11. Gupta, R. Conditional Random Fields. In *Computer Vision: A Reference Guide*; Springer: Boston, MA, USA, 2014; p 146.
12. Middleton, S.E.; Kordopatis-Zilos, G.; Papadopoulos, S.; Kompatsiaris, Y. Location Extraction from Social Media: Geoparsing, Location Disambiguation and Geotagging. *ACM Trans. Inf. Syst.* **2018**, *36*, Article 40. [CrossRef]
13. Karimzadeh, M.; Pezanowski, S.; MacEachren, A.; Wallgrun, J. GeoTxt: A scalable geoparsing system for unstructured text geolocation. *Trans. GIS* **2019**, *23*, 118–136. [CrossRef]
14. Rupp, C.; Rayson, P.; Baron, A.; Donaldson, C.; Gregory, I.; Hardie, A.; Murrieta-Flores, P. Customising geoparsing and georeferencing for historical texts. In Proceedings of the IEEE International Conference on Big Data, Big Data, Silicon Valley, CA, USA, 6–9 October 2013; pp. 59–62.
15. Tobin, R.; Grover, C.; Byrne, K.; Reid, J.; Walsh, J. Evaluation of Georeferencing. In *Proceedings of the 6th Workshop on Geographic Information Retrieval*; ACM: New York, NY, USA, 2010; pp. 1–8.
16. Mani, I.; Hitzeman, J.; Richer, J.; Harris, D.; Quimby, R.; Wellner, B. SpatialML: Annotation Scheme, Corpora, and Tools. In Proceedings of the Sixth International Conference on Language Resources and Evaluation, LREC 2008, Marrakech, Morocco, 28–30 May 2008.
17. Ardanuy, M.C.; Sporleder, C. Toponym disambiguation in historical documents using semantic and geographic features. In Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage, Göttingen, Germany, 1–2 June 2017; pp. 175–180.
18. Pantaleo, G.; Nesi, P. Ge(o)Lo(cator): Geographic Information Extraction from Unstructured Text Data and Web Documents. In Proceedings of the 2014 9th International Workshop on Semantic and Social Media Adaptation and Personalization, Corfu, Greece, 6–7 November 2014; pp. 60–65.
19. Martins, B.; Silva, M. A Graph-Ranking Algorithm for Geo-Referencing Documents. In Proceedings of the Fifth IEEE International Conference on Data Mining, Houston, TX, USA, 27–30 November 2005; pp. 741–744.
20. Page, L.; Brin, S.; Motwani, R.; Winograd, T. *The PageRank Citation Ranking: Bringing Order to the Web*; Technical Report 1999-66; Stanford InfoLab: Stanford, CA, USA, 1999.
21. Silva, M.J.; Martins, B.; Chaves, M.; Afonso, A.P.; Nuno, C. Adding geographic scopes to web resources. *Comput. Environ. Urban Syst.* **2006**, *30*, 378–399. [CrossRef]
22. Gelernter, J.; Zhang, W. Cross-lingual geo-parsing for non-structured data. In Proceedings of the 7th Workshop on Geographic Information Retrieval, Orlando, FL, USA, 5 November 2013; pp. 64–71.
23. Moncla, L.; Renteria-Agualimpia, W.; Nogueras-Iso, J.; Gaio, M. Geocoding for texts with fine-grain toponyms: An experiment on a geoparsed hiking descriptions corpus. In Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Dallas/Fort Worth, TX, USA, 4–7 November 2014; pp. 183–192.
24. Molina-Villegas, A.; Siordia, O.S.; Aldana-Bobadilla, E.; Aguilar, C.A.; Acosta, O. Extracción automática de referencias geoespaciales en discurso libre usando técnicas de procesamiento de lenguaje natural y teoría de la accesibilidad. *J. Nat. Lang. Process.* **2019**, *63*, 143–146.
25. Cucerzan, S.; Yarowsky, D. Language independent named entity recognition combining morphological and contextual evidence. In Proceedings of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, College Park, MD, USA, 21–22 June 1999.
26. Li, P.; Fu, T.; Ma, W. Why Attention? Analyze BiLSTM Deficiency and Its Remedies in the Case of NER. In Proceedings of the The Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; AAAI Press: 2020; pp. 8236–8244.
27. Luo, Y.; Xiao, F.; Zhao, H. Hierarchical Contextualized Representation for Named Entity Recognition. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2020), New York, NY, USA, 7–12 February 2020.
28. Li, X.; Feng, J.; Meng, Y.; Han, Q.; Wu, F.; Li, J. A Unified MRC Framework for Named Entity Recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 5849–5859. doi:10.18653/v1/2020.acl-main.519. [CrossRef]
29. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.

30. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
31. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
32. Horn, F. Context encoders as a simple but powerful extension of word2vec. *arXiv* **2017**, arXiv:1706.02496.
33. Trapala, J.A. Reconocimiento de Entidades Nombradas Georeferenciables con Word Embeddings. Master's Thesis, Centro de Investigación en Matemáticas, Monterrey, Mexico, 2019.
34. Amorim, M.; Antunes, F.; Ferreira, S.; Couto, A. An integrated approach for strategic and tactical decisions for the emergency medical service: Exploring optimization and metamodel-based simulation for vehicle location. *Comput. Ind. Eng.* **2019**, *137*, 106057. [CrossRef]
35. Hsiao, Y.H.; Chen, M.C.; Liao, W.C. Logistics service design for cross-border E-commerce using Kansei engineering with text-mining-based online content analysis. *Telemat. Inform.* **2017**, *34*, 284–302. [CrossRef]