Using Contextual Embeddings and External Geophysical Information for Toponym Resolution in Text

Ana Bárbara Cardoso
barbara.inacio@tecnico.ulisboa.pt
INESC-ID and Instituto Superior Técnico, Lisbon, Portugal

Bruno Martins
bruno.g.martins@ist.utl.pt
INESC-ID and Instituto Superior Técnico, Lisbon, Portugal

Jacinto Estima
jacinto.estima@gmail.com
INESC-ID and Instituto Politécnico de Setúbal, Portugal

ABSTRACT

Toponym resolution refers to the disambiguation of place names and other references to places present in textual documents, linking the corresponding spans of text to unambiguous geographical identifiers (e.g., geographic coordinates of latitude and longitude). One of the significant challenges in this task is the high ambiguity of place names since there are, for instance, several locations on the surface of the Earth that share the same name. In this article, we describe a novel approach for toponym resolution using a deep neural network, that does not directly involve gazetteer matching, to predict geospatial coordinates such as latitude and longitude. The proposed neural network architecture uses recurrent units with multiple inputs (e.g., the toponym to disambiguate along with the surrounding words), leveraging pre-trained contextual word embeddings (i.e., ELMo or BERT embeddings) and bi-directional Long Short-Term Memory (LSTM) units, both commonly used for textual data modeling. We tested the proposed model on three datasets that have been used to evaluate other toponym resolution systems, namely on the (1) *War of the Rebellion*, (2) *Local-Global Lexicon*, and (3) *SpatialML* corpora. Additionally, we evaluated the impact of using (1) external information in the form of raster datasets with geophysical properties, including information on land coverage, terrain elevation, among others, and (2) additional training data collected from Wikipedia articles, to guide and further help model training. The obtained results show a significantly higher quality of the proposed method, in comparison to previous approaches and particularly in the setting that involves BERT embeddings and additional data.

KEYWORDS

Geographical Text Analysis, Toponym Resolution in Text, Deep Learning for NLP, Contextual Word Embeddings, Geophysical Properties

1 INTRODUCTION

Toponymy resolution concerns the disambiguation of place names and other references to places in textual documents. The disambiguation is achieved by associating each of these place references to a unique position on the Earth's surface, e.g., through the assignment of geographic coordinates. The toponym resolution task is particularly challenging, given that place references are highly ambiguous. Three distinct types of ambiguity should be addressed when resolving toponyms in textual documents [23]: (1) geo/geo ambiguity arises when distinct locations share the same place name (e.g., the name *Dallas* can be associated with either *Dallas, Texas, United States*, or *Dallas County, Alabama, United States*); (2) geo/non-geo ambiguity corresponds to places named using common language words, i.e., when a location and a non-location share the same name. For example, the word *Charlotte* can refer to a person name or to the location of *Charlotte County, Virginia, United States* or, for instance, the word *Manhattan* that can refer to the location of *Manhattan, New York, United States*, or to the cocktail beverage; (3) reference ambiguity, which occurs when multiple names are referring to the same place (e.g., *Big Apple* is a common nickname used for referring to *New York City, New York, United States*). Problem (2) should be covered when identifying place references in textual documents, whereas Problems (1) and (3) should be addressed when attempting to associate the recognized references to physical locations unambiguously (e.g., geospatial coordinates of latitude and longitude).

Through the results emerging from toponymy resolution, it is possible to consider several applications, such as the improvement of search engine results (e.g., by geographic indexing or classification), document classification according to spatial criteria, which allows grouping documents into meaningful clusters and enables the mapping of textually encoded information [23]. Another possible application is in areas such as computational social sciences or digital humanities [34], for instance, through supporting the automatic processing and analysis of geographic data encoded over extensive collections of textual documents. Moreover, place reference resolution can be an auxiliary component for the complete geolocation of documents [22], whereas the toponyms mentioned in the text can provide indications about the overall location of the document.

Most of the previously developed systems for toponym resolution are based on the use of heuristics (e.g., population density), relying on an external knowledge database to decide which location is more likely to correspond to the reference. The place references in the text are first compared against similar entries in a gazetteer [3, 20], and highly populated places are often preferred, given that they are more likely to be used in textual documents [2, 16]. Other studies employed supervised approaches that consider these types of heuristics as features in standard machine learning techniques [8, 15, 17, 26], while later studies explore the application of language modeling approaches [4, 28] and, more recently, deep learning techniques yielding state-of-the-art results [1, 11].

This article proposes a novel method using deep learning techniques for toponym resolution by combining pre-trained contextual word embeddings, i.e. static features extracted with either the Embeddings from Language Models (ELMo) [24] or with the Bidirectional Encoder Representations from Transformers (BERT) [6],

and with bidirectional Long Short-Term Memory (LSTM) units to model the textual elements. The proposed model incorporates multiple textual inputs, such as the place name reference, the corresponding sentence, and paragraph, with multiple outputs (i.e., a primary output of geographic coordinates and a secondary output of classification into regions over the surface of the Earth corresponding to the place reference). These regions of classification are determined by applying the Hierarchical Equal Area isoLatitude Pixelisation (HEALPix) [10] method, which generates cells of equal area, obtained by recursively partitioning a spherical surface representing the surface of the Earth. The result of this classification is used to improve the prediction of geographic coordinates for each place reference, through a separate layer that directly applies the Great Circle distance as a loss function. Additionally, we consider information from geophysical properties (i.e., land coverage, terrain elevation, percentage of vegetation, and minimum distance until a water zone). This information is extracted from external raster datasets and incorporated in the proposed model to guide the prediction of the geographic coordinates. We tested the proposed model on three distinct corpora widely used in previous studies, namely the *War of the Rebellion*, the *Local-Global Lexicon*, and the *SpatialML* corpora. The obtained results exceed previously reported results over these same datasets, thus demonstrating state-of-the-art performance. Furthermore, we considered a scenario in which the model training is performed over a larger sample, to determine the impact of the training data size on the results. The instances added to the original corpora were collected from a random sample of English Wikipedia articles, leveraging the Wikipedia link structure to infer which spans of text correspond to place references, in the sense that they link to Wikipedia pages associated with geospatial coordinates.

The rest of this article follows this structure: Section 2 presents relevant related work previously developed in this field and details about the corpora used in our work. Section 3 describes the proposed model, while Section 4 details the experimental evaluation, including the evaluation methodology, as well as the obtained results. Finally, Section 5 summarizes our conclusions and presents ideas for future work.

2 RELATED WORK

The following sections present relevant studies previously developed using different techniques, namely approaches based on heuristics (Section 2.1), approaches that combine heuristics with supervised learning (Section 2.2), methods that use both geodesic grids and language models (Section 2.3), and lastly methods based on deep learning techniques (Section 2.4). Finally, Section 2.5 introduces the corpora used in the experiments reported in this article.

2.1 Heuristic Methods

Most of the previously developed toponym resolution systems rely on the use of heuristics and typically resort to external knowledge sources such as gazetteers, enabling the access to a variety of data about places on Earth (e.g., alternative names, type of places, population density, area, among others). The systems based on heuristics usually leverage this information to decide which of the possible locations refers to the place name identified in the text [2, 16].

Additionally, it is possible to consider linguistic aspects to generate heuristics. Leidner [16] considers both linguistic heuristics (i.e., rules and patterns inferred from the textual content) and extra-linguistic heuristics (i.e., based on an external knowledge source). For example, one of the linguistic heuristics used by Leidner is based on a qualifier "contained-in", that recognizes patterns such as "*toponym1* in *toponym2*" or "*toponym1* (*toponym2*)" and evaluates the spatial containment of the possible candidate locations (i.e., locations with the same name as the one under resolution) for both toponym mentions, assigning the correspondent geographic coordinates according to the spatial containment (e.g., if the pattern recognizes *London (UK)*, it assigns to *London* the coordinates of the capital of England, whereas if the pattern recognizes *London, Ontario, Canada* it assigns to the mention *London* the coordinates of London, the city of Ontario). One of the extra-linguistic heuristics that Leidner uses is the attribution of the candidate location with higher population density to the toponym mention to disambiguate. Another example is the heuristic that considers if a given toponym occurs only once in the text, and if precisely one candidate location is a capital, then it believes that the toponym mention refers to the capital (e.g., if *Madrid* occurs in the text, always assign the coordinates of Madrid, the capital of Spain, without considering other locations named Madrid). Besides the examples mentioned, it combines both types of heuristics, such as considering the textual-spatial correlation, where it assumes that textual proximity is strongly correlated with spatial proximity, assigning the locations accordingly. For example, if within a small text span (i.e., textual proximity) occurs in text the mentions *Paris* and *Versailles*, then the mention *Paris* is associated with Paris, France [16].

One of the significant disadvantages of relying on gazetteers is that often, these are outdated and incomplete, thus impacting the systems that use them and making them unable to handle new and vernacular place names [3, 20].

2.2 Combining Heuristics through Supervised Learning

Other studies use supervised approaches that consider heuristics as features in standard machine learning techniques [8, 15, 17, 26]. The work of Santos et al. [26] explores the combination of multiple features that may capture similarities between possible candidate locations, as well as other toponyms present in the text and the context of the place reference (i.e., the text surrounding the mention). Afterward, a rank is assigned to each candidate location, and the location with the highest rank is associated with the mention under resolution, this method revealed state-of-the-art results [26]. Another work that employs supervised learning techniques along with heuristics is the GeoTXT geocoder, developed by Karimzadeh et al. [15], a flexible application programming interface for extracting and disambiguating toponyms in small textual documents. This geocoder utilizes existing resources to recognize toponyms in text, focusing exclusively on disambiguating the recognized place references. When resolving toponyms, for each place reference, the system retrieves a list of candidate locations and associates a score

to each of them. The score assigned to each location is a combination of multiple scores referring to features, which include independent political entities, administrative divisions, populated places, continent, region, or type of establishment (e.g., buildings, school, airport), among others [15]. Furthermore, GeoTXT enables the incorporation of additional disambiguation mechanisms that consider the co-occurrence of toponyms in the text. Two such of these mechanisms are based on hierarchical relationships between toponyms, i.e., if two toponyms share the same geographic space containment, either applied to immediately consecutive place names (e.g., pairs consisting of city, state) or applied to toponyms that appear separately in the text. The third mechanism is based on spatial proximity, which aims to minimize the average distance between the predicted toponym location and the location for toponyms that co-occur in the text.

2.3 Methods Combining Geodesic Grids and Language Models

Besides the techniques previously mentioned, it is possible to use geodesic grids, sometimes combined with language models, to predict geographic coordinates when resolving toponyms. A geodesic grid over the surface of the Earth allows its subdivision into multiple regions of equal dimensions. Both works of Adams and McKenzie [1] and Gritta et al. [11] uses geodesic grids, respectively, to geocode textual content and to disambiguate toponyms, assigning to each toponym the corresponding region over the surface of the Earth (details about these works in Section 2.4). Wing and Baldridge [35] also explore the use of geodesic grids for document geolocation. The authors developed a model that attempts to predict the correct region of a document by applying simple supervised methods and only considering textual elements as input. The developed model records improvements over previous document geolocation studies [35].

With *TopoCluster* system, proposed by DeLozier et al. [4], the authors proposed to address and resolve the limitation brought by the recurrent need to rely on external gazetteers when resolving toponyms. *TopoCluster* considers the geographical distribution of each word, including the surrounding common language words, since there are certain words with the property of being geographically indicative. The authors use spatial statistics over multiple geo-referenced language models to create geographic clusters for each word, and derives a smoothed geographic likelihood for each word in the vocabulary and computes, which is the strongest geographic point where the toponym and context words clusters overlap. The authors show that it is possible to obtain superior results without recurring to gazetteers, noticing that the model performs well in corpora based on international news and historical texts [4].

2.4 Deep Learning Techniques

Adams and McKenzie [1] proposed a character-level convolutional neural network model for geocoding multilingual text using any character set represented by UTF-8 encoding. The model receives as input a sequence of characters encoded as a one-hot vector to which is applied a series of temporal convolution and temporal max pooling operations. Then multiple linear transformations are applied to the result. Finally, the output layer predicts the region classification using a geodesic grid. By using character-level convolutional neural networks, the approach is language independent. The authors verified that the model did not achieve the best results when diacritical characters were present, concluding that individual words are sometimes good geographical indicators [1].

Another example of a toponym resolution system that uses deep learning techniques is the *CamCoder* [11] model, which attempts to disambiguate place references by discovering lexical clues through the context words surrounding the mention. The model introduces a sparse vector representation, named *MapVec*, which encodes the prior geographic probabilities distribution of locations (i.e., based on location coordinates and population counts). The spatial data is projected onto a 2D world map, which is then reshaped into a 1D feature vector (i.e., *MapVec*), enabling the codification of additional information about spatial knowledge usually ignored in similar studies [11]. The *CamCoder* geocoder combines lexical and geographic information, thus receiving the following inputs: the context words (without the location mentions), the mentions to locations (excluding the context words), the target entity to disambiguate, and finally the feature vector *MapVec*. The textual inputs are fed into separate convolutional layers with global maximum pooling to detect words indicative of locations among context words, while the feature vector is supplied into a fully-connected dense layer. Then, the four resulting components are fed into another dense layer, followed by a concatenation of their results, which are provided to the output layer, where the model predicts a location based on classification into regions defined by a geodesic grid. *CamCoder* is a robust model that enables the consideration of geographic factors beyond lexical clues to improve the performance of the toponym resolution, presenting state-of-the-art results [11].

2.5 Corpora Used in Previous Studies and Adopted for this Study

In this section, we describe the corpora that were used during the development of this article, namely the *War of the Rebellion* [5], the *Local-Global Lexicon* [18], and *SpatialML* [21], which have been widely used in several studies in the area [2, 4, 5, 11, 12, 26]. The *War of the Rebellion* (WOTR) corpus is composed of historical texts collected from military archives of the American Civil war, among which predominate military orders, reports, and government correspondence [5]. DeLozier et al. presents the process of annotating these historical documents, as well as an evaluation of the performance of existing toponym resolution systems over the developed corpus, and additionally testing other corpora to examine the results. The authors concluded that the WOTR corpus was the most challenging corpus surveyed, with lower performance results than the *Local-Global Lexicon* corpus (i.e., considered the most challenging corpus until then). In turn, Lieberman et al. constructed the *Local-Global Lexicon* (LGL) corpus from articles retrieved from small and geographically distributed newspapers [18]. This corpus was deliberately created to present several challenges to toponym

resolution systems, given that it contains articles from small newspapers, based on near locations with highly ambiguous names. For example, *Paris* is a highly ambiguous toponym, and in this collection, there are articles from The Paris News (*Paris, Texas*), The Paris Post-Intelligencer (*Paris, Tennessee*), and The Paris Beacon-News (*Paris, Illinois*)) [18]. As mentioned before, until the appearance of the WOTR, it was considered one of the most challenging corpora for toponym resolution. The *SpatialML* corpus is provided by the Linguistic Data Consortium, which comprises documents from the ACE English, among which are broadcast conversations, broadcast news, magazine news, newsgroups, and web blogs. SpatialML is an annotation scheme, where the references of the locations identified in the text are associated with a PLACE tag, and a LATLONG attribute corresponding to the geographical coordinates.

3 THE PROPOSED MODEL

In the following sections, we present a detailed description of the techniques used to develop our model. Section 3.1 provides details about recurrent neural networks, namely the Long Short-Term Memory architecture, which we use in our model. Section 3.2 discusses distinct approaches for representing text through contextual word embeddings. Finally, Section 3.3 explains our approach to place reference resolution, which models the problem as a classification task addressed by a deep neural network and presents the architecture of the proposed model.

3.1 Recurrent Neural Networks

A recurrent neural network (RNN) architecture allows modeling over sequential structures, which is extremely useful in natural language processing (NLP) since there are sequences in any text, whether of characters, words, or phrases. An RNN enables the representation of input structures with arbitrary lengths, transforming them into a fixed-size vector while maintaining the structural properties of the input sequence. In this architecture, the connections between neurons form a graph directed over a sequence, in other words, an RNN is recursively defined through a function R that receives a state vector as input s_{j-1} (i.e., corresponding to the previous state), along with the input vector of the current state x_j, and returns a new state vector s_j. The state vector s_j is mapped by the function O into a vector that corresponds to the output vector of the current state y_j. Therefore, this structure considers the set of the history of all previous states $(x_1, x_2, ..., x_j)$ [9]. Similarly, a bi-directional RNN follows the same structure mentioned above, connecting two hidden layers of opposite directions to the same output (i.e., one that reads the sequence from the left-to-right and other that reads from right-to-left). Thus, the output layer can get information from the past (backward), and future (forward) states simultaneously.

Usually, during the training of an RNN, the gradient begins to vanish, preventing the network weights from changing and adjusting as necessary during training, which emerged as a significant obstacle to the network's performance. This problem is known as the vanishing gradient problem [13], where the gradient decreases exponentially and consequently generates a decay of information over time. Thus, gating-based approaches were developed to address the vanishing gradient problem, since these techniques help the neural network to decide when to forget the current input and when to remember it for future time steps.

3.1.1 Long Short-Term Memory. The Long Short-Term Memory (LSTM) is a concrete RNN architecture designed to address the vanishing gradient problem by using a gating mechanism. At each input state, a gate is responsible for deciding how much of the new input should be written to the memory cell and how much of the current memory cell content should be forgotten [9]. Equation 1 defines the LSTM architecture mathematically.

$$s_j = R_{LSTM}(s_{j-1}, x_j) = [c_j; h_j] \tag{1a}$$

where,

$$c_j = f \odot c_{j-1} + i \odot z \tag{1b}$$

$$h_j = o \odot \tanh(c_j) \tag{1c}$$

$$i = \sigma(x_j W^{xi} + h_{j-1} W^{hi}) \tag{1d}$$

$$f = \sigma(x_j W^{xf} + h_{j-1} W^{hf}) \tag{1e}$$

$$o = \sigma(x_j W^{xo} + h_{j-1} W^{ho}) \tag{1f}$$

$$z = \tanh(x_j W^{xz} + h_{j-1} W^{hz}) \tag{1g}$$

$$y_j = O_{LSTM}(s_j) = h_j \tag{1h}$$

The state, at time j (s_j), is composed of two vectors, c_j and h_j, which corresponds respectively to the memory component and the hidden state component (Equation 1a). There are three gating components that are responsible for controlling the amount of information, the input, forget and output gates, i.e., i, f, and o respectively (Equations 1d; 1e; 1f). The gate values are obtained from linear combinations of current input x_j and previous states h_{j-1}, to which a sigmoid activation function is applied. An update candidate, z, is obtained by a linear combination of x_j and h_{j-1} passing through a tangent activation function (Equation 1g). Afterward, the memory component c_j is updated considering the forget gate, i.e., which controls how much of the previous memory should be kept, and the input gate, i.e., which controls the amount of the proposed update that is preserved (Equation 1b). Finally, the value of h_j, which corresponds to the output y_j (Equation 1h), is computed considering the memory content c_j, then passing through a nonlinear tangent function and controlled by the output gate (Equation 1c) [9].

3.2 Contextual Word Embeddings

The text representation is an essential aspect when dealing with textual analysis tasks. Recently, novel approaches emerged to represent textual elements, capturing linguistic information, namely word embeddings. This method for representing text consists of learning to map a set of words into real number vectors (i.e., within a corpus a vector space is produced), containing a vector representing each word in the corpus. The word embedding representations capture the similarity between words. Hence, word vectors are

positioned in the vector space in a meaningful way, so that distance between words is related to their semantic similarity [9].

The contextual word embedding representations, beyond capturing similarities between words, are also able to perceive the semantic meaning of words since it considers the surrounding context words, which enables the capacity to handle polysemic properties of words. For example, if the word *wood* occurs in a text, the representation generated is distinct depending on the context, i.e., if *wood* is referring to the material made from trees, or to a geographical area with many trees. Although there are several models for generating contextual word embeddings, in this article, we decided to use only two of these models, namely the Embeddings from Language Models (ELMo) [24] and the Bidirectional Encoder Representations from Transformers (BERT) [6], described in the following sections.

3.2.1 Embeddings from Language Models.

Petters et al. presented the ELMo model for generating pre-trained contextual word embeddings. As mentioned before, this model considers the context words when creating the embedding representations handling with the semantic meaning, syntactic use, and polysemy of words [24]. To contextualize the word representations ELMo model examines the entire sentence before assigning the word embedding representation. ELMo is based on a neural language model (i.e., a model of the probability distribution over word sequences), these models are used to predict which is the most likely next word from a given sequence of words. The language model that ELMo uses relies on a multi-layer bi-directional LSTM (previously described in Section 3.1). Therefore, when generating the word embedding representation, ELMo considers both the following and the previous words. To generate a contextualized embedding representation for each word, ELMo extracts the hidden state of each layer, concatenates them, and applies a weighted sum operation.

3.2.2 Bidirectional Encoder Representations from Transformers.

Another model for generating contextual word embeddings is the BERT model, proposed by Devlin et al. [6], and inspired by previous studies including ELMo [24], UMLFiT [14], the OpenAI transformer [25], and the Transformer [29]. The architecture of the BERT model is based on a pre-trained transformer encoder stack. A transformer encoder component is composed of two sub-layers, a self-attention layer, and a feed-forward neural network. Thus, the model receives as input a set of words and each encoder layer applies a self-attention mechanism (i.e., enabling the encoder to consider other words present in the input sequence while encoding a particular word), passes the result through a feed-forward neural network, passing the output to the next encoder layer, and so on. BERT model is based on transformer encoders, whose language model considers both forward and backward words. The authors decided to adopt a "masked language model" to train the model, i.e., randomly applying masks to 15% of the input tokens, and using the output of the position of the masked word to predict which word was masked. Besides, occasionally, the words are randomly replaced by another word, and the model is asked to predict the correct word in that position. Similar to ELMo, the pre-trained BERT model can be used to generate contextual word embeddings (i.e., the word embedding representations can consist of one of the vectors or a combination of multiple vectors generated by the encoder representations).

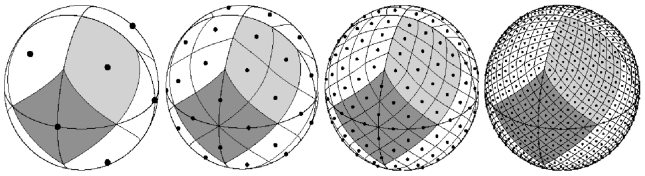


Figure 1: Orthographic view of the HEALPix partitioning.

3.3 Proposed Model Architecture

The overall idea behind our model is the following: from a textual document with previously annotated references to locations (i.e., identified as toponyms and associated with geographical coordinates of latitude and longitude), by providing textual elements as input to the model, including the context, using contextual word embeddings together with bi-directional LSTMs units to model the text sequence, predict the region classification upon a geodesic grid and use the classification probability distribution to obtain geographical coordinates (i.e., latitude and longitude) of each recognized place reference. As mentioned before, this work focuses exclusively on the toponym resolution task, intending to assign an unambiguous position over the surface of the Earth to each place name reference in textual contents. With this in mind, we choose to approach the problem as a classification task, where each place name reference is associated with a delimited region on the surface of the Earth through a geodesic grid. Therefore, we use the Hierarchical Equal Area isoLatitude Pixelization (HEALPix) scheme proposed by Gorski et al. [10], an algorithm that performs partitions on a sphere generating cells of equal area, corresponding to different regions on the Earth's surface. These partitions are obtained hierarchically from recursive divisions over a spherical surface, where the user defines the number of recursive divisions to execute over that surface (i.e., the desired resolution). These partitions are exemplified in Figure 1 that shows the grid is divided according to different resolution parameters, differing in the number of cells generated.

Moreover, our neural network model only receives textual inputs, more specifically three elements for each place reference recognized in the text: (1) the place mention itself; (2) the words around the mention (i.e., a fixed window size, to the left and right sides of the focus span of the text with the toponym, totaling 50 words); and, (3) a paragraph text, also defined by a fixed window size of larger dimensions (i.e., a total of 500 words), so it can capture the text around the sentence where the toponym occurs. Both the sentence and paragraph input consider the context around the mention in the forward and backward directions. When feeding the neural network with the paragraph of the mention, we are considering the general context of the document, and by considering a smaller textual window where the mention is present (i.e., the sentence), we are considering the closest context to the entity. Since other toponyms,

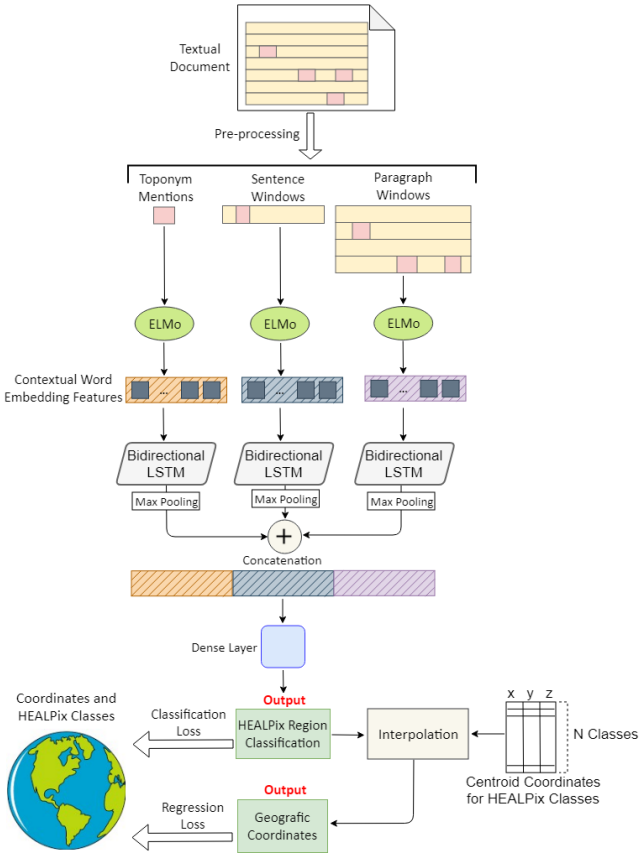


Figure 2: The proposed neural network architecture.

or even common language words appearing in the surrounding text, can be characteristic of specific regions that might provide clues about the location of the mention.

The structure of the developed model is represented in Figure 2. The model starts by pre-processing the text documents, extracting, for each annotated toponym, its geographical coordinates together with the three textual components corresponding to the inputs of the neural network model, namely (1) the mention itself, (2) the sentence and finally, (3) the paragraph text, as previously explained. To generate the contextual representation of the text elements, we use pre-trained contextual word embeddings (Section 3.2). To generate the textual representations, we apply the contextual word embedding approach to each one of the inputs, resulting in one embedding vector for each of them, which is fed into a separate bi-directional LSTM layer to model the word sequence. In Figure 2, we represent the use of contextual word embeddings with the ELMo embedding model. However, the developed architecture is versatile since it allows the usage of other word embedding models, which in our case, we also evaluated the proposed model with the BERT embeddings.

To each bi-directional LSTM layer, we apply the penalized hyperbolic tangent, an improvement of the hyperbolic tangent loss function, suggested by Eger et al. [7]. As demonstrated in Equation 2, the function penalizes the identity function in the negative region (i.e., whenever the input is negative). This loss function has proved to achieve superior results across a variety of natural language processing tasks.

$$f(x) = \begin{cases} \tanh(x), & \text{if } x > 0 \\ 0.25 \cdot \tanh(x), & \text{otherwise} \end{cases} \quad (2)$$

Afterward, we concatenate the resulting representation from the maximum pooling operation over each bi-directional LSTMs and use them to predict the HEALPix region class (the first output), through a dense layer where we apply a softmax activation function, obtaining a probability vector of equal size to the number of distinct HEALPix region classes. This HEALPix class probability vector is used to estimate the corresponding geographic coordinates (the second output) through a cubic interpolation between the class probability distribution, and the centroid coordinates matrix, i.e., a previously constructed matrix that contains the centroid coordinates of each HEALPix class, where each row corresponds to a distinct class. By applying a cubic interpolation (i.e., raising the HEALPix class probability vector to the power of three and normalizing), we accentuate the classes with higher probabilities leading to a more peaked distribution. Each of the outputs is associated with a separate layer, and during the model training, the goal is to minimize the combined loss of the outputs, thereby mutually guiding the learning process and improving the results. In the geographic coordinates output layer, we apply a regression loss based on the Great Circle distance (i.e., to calculate the distance between two points, namely the predicted point and the actual one, over the surface of the Earth), while for the region classification output layer we apply the standard cross-entropy categorical loss.

When training, we use the Adam optimization algorithm with a Cyclical Learning Rate (CLR) [27] policy, adjusting the learning rate throughout the training process, with the basis on a cycle between a lower bound of 0.00001 and an upper bound of 0.0001. Besides, we also use an early stopping strategy (i.e., a form of regularization used to avoid overfitting that interrupts the training process once the model performance stops improving). In our case, the training was stopped when the combined loss over the training data was not improving for five consecutive epochs.

4 EXPERIMENTAL EVALUATION

This section provides an overview of the experimental settings used throughout the model evaluation. Section 4.1 describes both the methodology used and the additional experiments performed with the proposed model. Section 4.2 presents the obtained results and their analysis.

4.1 Experimental Methodology

The neural network architecture described in Section 3.3 is the architecture adopted for our base model, called the ELMo model. However, additional experiments with other models were also performed, the Wikipedia model, the BERT model, and the model that integrates geophysical properties. A short description of all these models is provided below:

- **ELMo model** - The base model described in Section 3.3 and represented in Figure 2. This model uses the ELMo embeddings model (Section 3.2.1) to generate contextual word embedding representations for the text.

- **Wikipedia model** - To determine the impact of the size of the training instances, we created a new corpus with articles from the English Wikipedia dumps. From random Wikipedia articles, we verify existing hyperlinks with associated geographic coordinates, collecting the article text, the hyperlink text, and the geographic coordinates. The instances added to the train data were filtered to coincide with the HEALPix regions present in the original corpora, thus adding more instances to the training data without modifying the region classification space of each corpus.

- **BERT model** - To observe the impact of using a different contextual word embeddings model, we chose to use the BERT contextual embeddings (Section 3.2.2), instead of the ELMo contextual embeddings. Therefore, the only difference between this experiment and the ELMo model is the embedding model chosen to represent the text.

- **Integration of geophysical properties** - In this experiment, we consider additional information about geophysical properties, such as land cover, elevation, percentage of vegetation, and minimum distance to a water zone. We extract this extra information from datasets in raster format (i.e., a grid mapping properties with geographic coordinates). We incorporate this information into the model using the same interpolation technique used to estimate the prediction of geographic coordinates, described previously. For each of the properties, we create a matrix with the values corresponding to each centroid of the distinct HEALPix class and interpolate these matrices with the probability distribution of the HEALPix classes, with the purpose of using the geophysical information to guide the prediction of the geographic coordinates.

We tested both the Wikipedia model and the model that integrates geophysical properties, with both contextual word embedding models covered in this article, ELMo and BERT, originating the following models: (1) **Wikipedia+ELMo** model and **Wikipedia+BERT** model, and (2) **Geophysical+ELMo** model and **Geophysical+BERT** model, respectively.

To conduct the described experiments, we use three well-known corpora, namely the *War of the Rebellion* [5], the *Local-Global Lexicon* [18], and the *SpatialML* [21] (Section 2.5). As these corpora have different sources (i.e., historical documents, news from small places, and international news, respectively), naturally, they also have different textual structures. For example, SpatialML is based on international news documents, which tend to be more extensive and with more toponym references than the other corpora. This can be verified in Table 1, which presents a statistical characterization of the corpora. We did our best to simulate the conditions of the experiments conducted by previous systems enabling the comparison of results and model performance. In the WOTR corpus, we used precisely the same data split (i.e., division of train and test data) provided by the authors. Regarding the results presented with the LGL and SpatialML corpora, we split the data in the following proportion: 90% of the instances for train and the remaining 10% for test.

Table 1: Statistical characterization of the corpora used in our experiments.

Statistic	WOTR	LGL	SpatialML
Number of documents	1644	588	428
Number of toponyms	10377	4462	4606
Avg. toponyms per document	6.3	7.6	10.8
Avg. tokens per document	246	325	497
Avg. sentences per document	12.7	16.1	30.7
Vocabulary size	13386	16518	14489

To calculate the regions over the surface of the Earth, we used the Healpy python library[1], based on the HEALPix scheme (Section 3.3), enabling to calculate the region code knowing the latitude and longitude coordinates, specifying the resolution, and vice versa. While to evaluate the prediction of geographic coordinates of each toponym, we calculate the distance between the predicted coordinates and the real coordinates on the Earth's surface. This distance is computed using Vincenty's geodesic formulae [30] (i.e., an iterative method that calculates the shortest geographic distance between two points on the Earth's surface, with an accuracy within 0.5 millimeters). From the error distances between the two points, we can calculate the mean and median of these values, as well as the accuracy@161, i.e., a widely used measure in previous studies, that reflects the percentage of distance errors less or equal than 161 kilometers.

4.2 The Obtained Results

The developed model achieves impressive results, sometimes outperforming previous state of the art results. In Table 2, we summarize the results obtained by our base model (i.e., using ELMo embeddings) comparing them with previous systems. Overall, our model exceeds expectations, as it achieves outstanding results across all corpora, recording the lowest mean error in both the WOTR corpus and the LGL corpus, yielding a difference of minus 281 kilometers and 463 kilometers, respectively, when compared to the second-best value obtained by other systems. As for the SpatialML corpus, the system of Santos et al. records the best mean. However, our model reaches the median value of 9.08 kilometers, which represents less 19.63 kilometers. Regarding the accuracy at 161 kilometers measure, our model obtains a value of 81.5% for the WOTR and a value of 86.1% for the LGL corpus, representing an increase of 9.5% and 10.1% respectively, when compared to the previous second-best result reported. In SpatialML, we record a value of 87.4% for the accuracy@161 metric.

[1] http://pypi.org/project/healpy/

Table 2: Experimental results obtained with the proposed architecture (ELMo model).

TR system	Mean (km)	Median (km)	Acc@161 (%)
WOTR			
TopoCluster [5]	604	–	57.0
TopoClusterGaz [5]	468	–	72.0
GeoSem [2]	445	–	68.0
Our approach	**164**	11.48	**81.5**
LGL			
GeoTxt [11]	1400	–	68.0
CamCoder [11]	700	–	76.0
TopoCluster [4]	1735	274.00	45.5
Santos et al. [26]	742	**2.79**	–
Our approach	**237**	12.24	**86.1**
SpatialML			
Santos et al. [26]	**140**	28.71	–
Our approach	395	**9.08**	87.4

Table 3: Comparison of the results between the different variations of the experiences with the proposed model.

Experiment	Mean (km)	Median (km)	Acc@161 (%)
WOTR			
ELMo	164	11.48	81.5
Wikipedia+ELMo	158	11.28	82.4
Geophysical+ELMo	166	11.35	81.9
BERT	117	**10.99**	**87.3**
Wikipedia+BERT	122	11.04	86.4
Geophysical+BERT	**114**	**10.99**	**87.3**
LGL			
ELMo	237	12.24	86.1
Wikipedia+ELMo	304	12.16	87.4
Geophysical+ELMo	282	12.24	87.7
BERT	**193**	11.81	90.1
Wikipedia+BERT	226	**11.51**	**90.6**
Geophysical+BERT	216	12.24	87.9
SpatialML			
ELMo	395	**9.08**	87.4
Wikipedia+ELMo	364	**9.08**	88.5
Geophysical+ELMo	387	**9.08**	87.4
BERT	363	**9.08**	89.2
Wikipedia+BERT	**205**	**9.08**	**92.4**
Geophysical+BERT	339	**9.08**	89.4

In Table 3, we present the results obtained in the several additional experiments conducted with the proposed architecture. The results show that the selection of the textual representations has a significant impact on the results achieved. By using the BERT contextual representations instead of the ELMo, we achieved amazing results, with, on average, less 41 kilometers in the mean value, less 0.3 kilometers in the median value, and an increase of 3.9% in the accuracy@161. We observed that by increasing the size of the training data, with more training instances, resulted in a slight improvement in the results obtained previously. Both comparing the ELMo model with ELMo+Wikipedia and the BERT model with BERT+Wikipedia, we verified the same pattern in the LGL corpus and the SpatialML corpus. For example, in the LGL corpus, both with the ELMo and the BERT embeddings, we recorded an increase in the mean values, a slight decrease in the median values, and finally, an increase in the accuracy@161 values. Regarding the WOTR corpus, the results are inconclusive. Possibly due to the difference in the textual register, i.e., this corpus is composed of historical documents, mainly governmental correspondence, while the Wikipedia articles have an informative register and a modern tone. As for the experiments with geophysical information (i.e., such as land coverage, among others) both with ELMo and BERT, we recorded a slight improvement on the obtained results in either experiment when compared with the ELMo and BERT model, respectively. Thus, the model benefits from the addition of geophysical information, which also helps to guide predicting the coordinates. However, the addition of geophysical properties does not provide relevant information when applied to the LGL corpus, leading us to the conclusion that the geophysical data does not have enough spatial resolution in the case of this corpus.

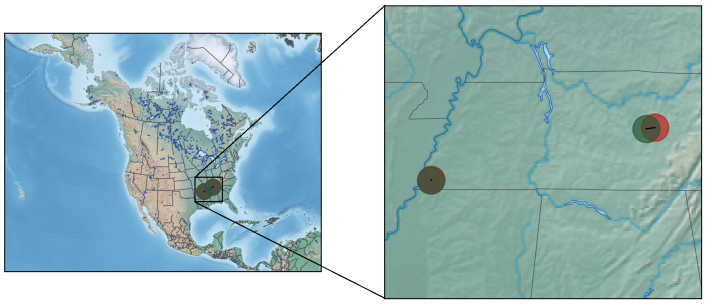
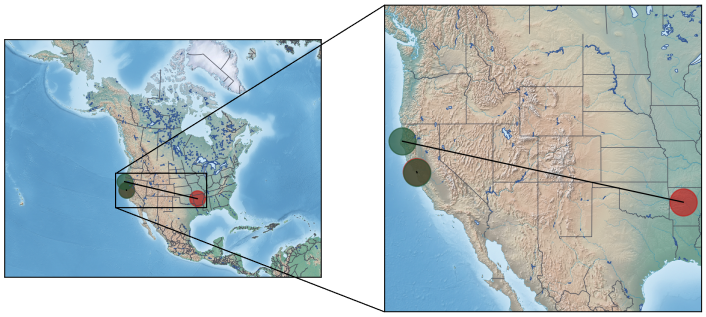
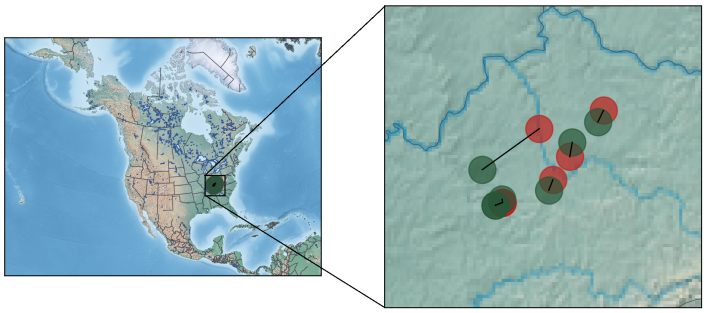
Table 4: Locations with lower and higher and distance error of prediction.

Corpus	Lowest error (km)		Highest error (km)	
WOTR	(0.63)	Mexico	(3104.59)	Fort Welles
	(1.00)	Resaca	(3141.29)	Washington
	(1.09)	Owen's Big Lake	(3682.01)	Astoria
LGL	(1.21)	W.Va.	(8854.04)	Ohioans
	(1.36)	Butler County	(9225.86)	North America
	(1.51)	Manchester	(9596.54)	Nigeria
SpatialML	(0.45)	Tokyo	(9687.43)	Capital
	(2.38)	Lusaka	(10818.50)	Omaha
	(2.44)	English	(13140.64)	Atlantic City

In Table 4, we present the locations with the lowest and highest distance error of prediction for all corpora. It is worth noting that, in all corpora, between the locations with lower prediction distance error, there are cases of demonym (e.g., *English* in the SpatialML corpus, resolved to the location *England, UK* with only an error of 2.44 kilometers), or even small places designated using vernacular names (e.g., in the case of *Owen's Big Lake* in the WOTR corpus).

We also present illustrative figures together with the document text, retrieved from the WOTR corpus in Table 5. Each of the example document text has the annotated toponyms highlighted in red, and the corresponding image, where it is shown the real location (green point) and the predicted location (red point), the distance between the two points is represented through a black line. In the examples shown, we included clear cases where the error between the predicted point and the actual point is small, and in other cases, this distance is significantly considerable. It is noteworthy that the

Table 5: Illustrative examples.

Figure	Text
	[Indorsement.] HDQRS. DETACHMENT SIXTEENTH ARMY CORPS, **Memphis**, **Tenn.**, June 12, 1864. Respectfully referred to Colonel David Moore, commanding THIRD DIVISION, SIXTEENTH Army Corps, who will send the THIRD Brigade of his command, substituting some regiment for the Forty-ninth Illinois that is not entitled to veteran furlough, making the number as near as possible to 2,000 men. They will be equipped as within directed, and will move to the railroad depot as soon as ready. You will notify these headquarters as soon as the troops are at the depot. By order of Brigadier General A. J. Smith: J. HOUGH, Assistant Adjutant-General.
	**HYDESVILLE**, October 21, 1862 SIR: I started from this place this morning, 7. 30 o'clock, en route for **Fort Baker**. The express having started an hour before, I had no escort. About two miles from Simmons' ranch I was attacked by a party of Indians. As soon as they fired they tried to surround me. I returned their fire and retreated down the hill. A portion of them cut me off and fired again. I returned their fire and killed one of them. They did not follow any farther. I will start this evening for my post as I think it will be safer to pass this portin of the country in the night. Those Indians were lurking about of rthe purpose of robbing Cooper's Mills. They could have no othe robject, and I think it would be well to have eight or ten men stationed at that place, as it will serve as an outpost for the settlement, as well as a guard for the mills. The expressmen disobeyed my orders by starting without me this morning. I have the honor to be, very respectfully, your obedient servant, H. FLYNN, Captain, Second Infantry California Volunteers. First Lieutenant JOHN HANNA, Jr., Acting Assistant Adjutant-General, Humboldt Military District.
	**LEXINGTON**, **KY.**, June 11, 1864–11 p. m. Colonel J. W. WEATHERFORD, **Lebanon**, **Ky.** Have just received dispatch from General Burbridge at **Paris**. He says direct Colonel Weatherford to closely watch in the direction of **Bardstown** and **Danville**, and if any part of the enemy's force appears in that region to attack and destroy it. J. BATES DICKSON, Captain and Assistant Adjutant-General.

presence of toponym co-occurrence consecutively, i.e., *Memphis, Tenn.*, which can give clues about both toponym locations. In the third example, we can see that all the toponyms have assigned locations with lower errors, with an error average of approximately 16.6 kilometers, among which there is a reference to the *Paris* location, a very ambiguous place name that is well resolved by the model with the help of the surrounding context.

5 CONCLUSIONS AND FUTURE WORK

In this article, we addressed the toponym resolution task. Therefore we proposed a recurrent neural network architecture with multiple textual inputs, leveraging pre-trained contextual word embeddings (ELMo or BERT) and bi-directional Long Short-Term Memory (LSTM) units, producing multiple outputs for classification and regression tasks. The proposed model incorporates classification into HEALPix regions, divisions upon the surface of the Earth, used to improve the expected results for the regression task when predicting the geographic coordinates. We conducted several additional experiments, including training data augmentation through English Wikipedia articles, application of different contextual word embeddings, and adding geophysical properties retrieved from a raster dataset to support the prediction of geographic coordinates. We chose to test our model on the following corpora: the *War of the Rebellion*, the *Local-Global Lexicon*, and the *SpatialML*. The results obtained confirm the superiority of the proposed method over previous studies that demonstrated state-of-the-art results. Using contextual word embeddings has shown to be useful for improving many NLP tasks, particularly when involving relatively small amounts of annotated training data, as in the case of the experiments reported in this article. We compare the impact of different embedding models, concluding that BERT outperforms ELMo representations when applied to the toponym resolution task. In the data augmentation scenario through a selection of the English Wikipedia, we recorded an improvement in the results obtained when compared to a scenario where exclusively the original corpora were used. Information on geophysical properties, when incorporated into the proposed model, had a beneficial impact since it contributed to the achievement of better results.

Regarding the future work, it may be interesting to explore cross-language embeddings to support the idea of training models leveraging existing data in a given language, and capable of operating on texts from a different language with less resources. It is worth noticing that approaches such as ELMo take character information to compose word representations, this way addressing the problem of out-of-vocabulary words to some degree (i.e., we can generate representations for words that are not present in the vocabulary used for learning the word embeddings, leveraging the characters that compose these words). However, individual characters are an insufficient and unnatural linguistic unit for word representation, and, similarly to other approaches such as FastText embeddings that leverage character n-grams. It would be interesting to extend the ELMo contextual approach to consider learning representations also for sub-words.

In this article we chose to use the ELMo and BERT embedding models, however, there are numerous contextual word embedding models [36], for example the RoBERTa [19], an optimized version of BERT (i.e., including the elimination of the pre-training objective concerning the prediction of the next sentence, moreover the model is trained with larger mini-batches, higher learning rates, with more training data and for a more extended amount of time than BERT) which has proven to be more efficient, producing state-of-the-art results.

Additionally, it would be interesting to test the model more intensively (e.g., using other corpora based on different sources, such as scientific documents, and even compare the performance of the proposed model against previous systems). One possibility would be to use EUPEG [32], a new benchmark platform developed by Wang and Hu, that integrates a wide range of document collections and permits the comparison between several existing systems. The EUPEG platform is a very complete and up-to-date system that includes the collection of scientific corpora used in the SemEval-2019 competition on toponym resolution [33] and features the systems that obtained the best classifications [31].

ACKNOWLEDGMENTS

This research was supported through Fundação para a Ciência e Tecnologia (FCT), through the project grants with references PTDC/EEI-SCR/1743/2014 (Saturn), T-AP HJ-253525 (DigCH), and PTDC/CCI-CIF/32607/2017 (MIMU), as well as through the INESC-ID multi-annual funding from the PIDDAC programme (UID/CEC/50021/2019). We also gratefully acknowledge the support of NVIDIA Corporation, with the donation of two Titan Xp GPUs used in the experiments reported in this article.

REFERENCES

[1] Benjamin Adams and Grant McKenzie. 2018. Crowdsourcing the character of a place: Character-level convolutional networks for multilingual geographic text classification. *Transactions in GIS* 22, 2 (2018), 394–408.
[2] Mariona Ardanuy and Caroline Sporleder. 2017. Toponym disambiguation in historical documents using semantic and geographic features. In *Proceedings of the International Conference on Digital Access to Textual Cultural Heritage*. ACM, 175–180.
[3] Merrick Berman, Ruth Mostern, and Humphrey Southall. 2016. *Placing names: Enriching and integrating gazetteers*. Indiana University Press.
[4] Grant DeLozier, Jason Baldridge, and Loretta London. 2015. Gazetteer-independent toponym resolution using geographic word profiles. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 2382–2388.
[5] Grant DeLozier, Benjamin Wing, Jason Baldridge, and Scott Nesbit. 2016. Creating a novel geolocation corpus from historical texts. In *Proceedings of the Linguistic Annotation Workshop held in conjunction with ACL*. Association for Computational Linguistics, 188–198.
[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1. Association for Computational Linguistics, 4171–4186.
[7] Steffen Eger, Paul Youssef, and Iryna Gurevych. 2018. Is it time to swish? Comparing deep learning activation functions across NLP tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 4415–4424.
[8] Nuno Freire, José Borbinha, Pável Calado, and Bruno Martins. 2011. A metadata geoparsing system for place name recognition and resolution in metadata records. In *Proceedings of the Annual International ACM/IEEE Joint Conference on Digital Libraries*. ACM, 339–348.
[9] Yoav Goldberg. 2017. *Neural network methods in natural language processing*. Morgan & Claypool Publishers.
[10] Krzysztof Górski, E. Hivon, Anthony Banday, Benjamin Wandelt, Frank Hansen, Michiel Reinecke, and M. Bartelman. 2005. HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere. *The Astrophysical Journal* 622, 2 (2005), 759–771.
[11] Milan Gritta, Mohammad Pilehvar, and Nigel Collier. 2018. Which Melbourne? Augmenting geocoding with maps. In *Proceedings of the Annual Meeting of the*

Association for Computational Linguistics, Vol. 1. Association for Computational Linguistics, 1285–1296.

[12] Milan Gritta, Mohammad Pilehvar, Nut Limsopatham, and Nigel Collier. 2018. What's missing in geographical parsing? *Language Resources and Evaluation* 52, 2 (2018), 603–623.

[13] Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6 (1998), 107–116.

[14] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *Computing Research Repository* abs/1801.06146 (2018).

[15] Morteza Karimzadeh, Scott Pezanowski, Alan MacEachren, and Jan Wallgrün. 2019. GeoTxt: A scalable geoparsing system for unstructured text geolocation. *Transactions in GIS* 23, 1 (2019), 118–136.

[16] Jochen Leidner. 2007. *Toponym resolution in text.* Ph.D. Dissertation. University of Edinburgh.

[17] Michael Lieberman and Hanan Samet. 2012. Adaptive context features for toponym resolution in streaming news. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 731–740.

[18] Michael Lieberman, Hanan Samet, and Jagan Sankaranarayanan. 2010. Geotagging with local lexicons to build indexes for textually-specified spatial data. In *Proceedings of the IEEE International Conference on Data Engineering.* IEEE, 201–212.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *Computing Research Repository* abs/1907.11692 (2019).

[20] Hugo Manguinhas, Bruno Martins, José Borbinha, and Willington Siabato. 2009. The DIGMAP geo-temporal web gazetteer service. *E-Perimetron* 4, 1 (2009), 9–24.

[21] Inderjeet Mani, Christy Doran, Dave Harris, Janet Hitzeman, Rob Quimby, Justin Richer, Ben Wellner, Scott Mardis, and Seamus Clancy. 2010. SpatialML: annotation scheme, resources, and evaluation. *Language Resources and Evaluation* 44, 3 (2010), 263–280.

[22] Fernando Melo and Bruno Martins. 2017. Automated geocoding of textual documents: A survey of current approaches. *Transactions in GIS* 21, 1 (2017), 3–38.

[23] Bruno Monteiro, Clodoveu Davis, and Frederico Fonseca. 2016. A survey on the geographic scope of textual documents. *Computers & Geosciences* 96 (2016), 23–34.

[24] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1. Association for Computational Linguistics, 2227–2237.

[25] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).

[26] João Santos, Ivo Anastácio, and Bruno Martins. 2015. Using machine learning methods for disambiguating place references in textual documents. *GeoJournal* 80, 3 (2015), 375–392.

[27] Leslie Smith. 2017. Cyclical learning rates for training neural networks. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision.* IEEE, 464–472.

[28] Michael Speriosu and Jason Baldridge. 2013. Text-driven toponym resolution using indirect supervision. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Vol. 1. Association for Computational Linguistics, 1466–1476.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the International Conference on Neural Information Processing Systems.* Curran Associates Inc., 5998–6008.

[30] Thaddeus Vincenty. 1975. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review* 23, 176 (1975), 88–93.

[31] Jimin Wang and Yingjie Hu. 2019. Are we there yet?: Evaluating state-of-the-art neural networkbased geoparsers using EUPEG as a benchmarking platform. In *Proceedings of the ACM SIGSPATIAL International Workshop on Geospatial Humanities.* ACM.

[32] Jimin Wang and Yingjie Hu. 2019. Enhancing spatial and textual analysis with EUPEG: an extensible and unified platform for evaluating geoparsers. *Transactions in GIS* 23, 6 (2019), 1393–1419.

[33] Davy Weissenbacher, Arjun Magge, Karen O'Connor, Matthew Scotch, and Graciela Gonzalez-Hernandez. 2019. SemEval-2019 task 12: Toponym resolution in scientific papers. In *Proceedings of the International Workshop on Semantic Evaluation.* Association for Computational Linguistics, 907–916.

[34] Benjamin Wing. 2015. *Text-based document geolocation and its application to the digital humanities.* Ph.D. Dissertation. University of Texas at Austin.

[35] Benjamin Wing and Jason Baldridge. 2011. Simple supervised document geolocation with geodesic grids. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.* Association for Computational Linguistics, 955–964.

[36] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Transformers: State-of-the-art natural language processing. *Computing Research Repository* abs/1910.03771 (2019).