# Evaluating the Effectiveness of Large Language Models in Representing Textual Descriptions of Geometry and Spatial Relations

**Yuhan Ji**
GeoDS Lab, Department of Geography, University of Wisconsin-Madison, WI, USA

**Song Gao**
GeoDS Lab, Department of Geography, University of Wisconsin-Madison, WI, USA

## Abstract

This research focuses on assessing the ability of large language models (LLMs) in representing geometries and their spatial relations. We utilize LLMs including GPT-2 and BERT to encode the well-known text (WKT) format of geometries and then feed their embeddings into classifiers and regressors to evaluate the effectiveness of the LLMs-generated embeddings for geometric attributes. The experiments demonstrate that while the LLMs-generated embeddings can preserve geometry types and capture some spatial relations (up to 73% accuracy), challenges remain in estimating numeric values and retrieving spatially related objects. This research highlights the need for improvement in terms of capturing the nuances and complexities of the underlying geospatial data and integrating domain knowledge to support various GeoAI applications using foundation models.

**2012 ACM Subject Classification** Computing methodologies → Artificial intelligence

**Keywords and phrases** LLMs, foundation models, GeoAI

**Digital Object Identifier** 10.4230/LIPIcs.GIScience.2023.43

**Category** Short Paper

**Funding** The authors would like to acknowledge the support from the H.I. Romnes Fellowship, National Science Foundation (No. 2112606) and Arity.

## 1 Introduction

Deep learning methods have exhibited great performance to tackle many challenging tasks in geographical sciences [16, 9]. However, the models often depend on handcrafted features for specific downstream tasks, thus being hard to be generalized into different tasks. The emergence of representation learning largely mitigated the issue by decomposing the learning process into two steps (task-agnostic data representation and downstream task) [1]. Therefore, an effective location-based representation should preserve key spatial information (e.g., distance, direction, and spatial relations) and make classifiers or other predictors easy to extract useful knowledge [13]. In geospatial artificial intelligence (GeoAI) research, although the geospatial data are usually well-formatted and can be readily understood by GIS software, not all of them can be directly integrated into a deep learning model.

The success of ChatGPT has been a milestone that attracts the general public's attention to Large Language Models (LLMs). With tons of parameters trained on a large text corpus, LLMs have learned profound knowledge across many domains. Other well-known LLMs include the Bidirectional Encoder Representations from Transformers (BERT) [5], the Generative Pre-trained Transformer (GPT) series [14, 2], etc. Despite the differences in network architectures, these LLMs can achieve state-of-the-art performance on natural language processing (NLP) benchmarks. Consequently, researchers have begun the early exploration of integrating LLMs into GIS research, such as geospatial semantic tasks [12] and

12th International Conference on Geographic Information Science (GIScience 2023).
Editors: Roger Beecham, Jed A. Long, Dianna Smith, Qunshan Zhao, and Sarah Wise; Article No. 43; pp. 43:1–43:6
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

automating spatial analysis workflows [11]. These studies have demonstrated the ability of LLMs to understand and reason about geospatial phenomena from a semantic perspective as learned from human discourse or formalized programming instructions. In contrast, accurate geometries and spatial relations in GIS are not necessarily expressed in natural languages. Therefore, it can be challenging for LLMs to reconstruct the physical world solely from the textual description of these building blocks, which is the motivation of this research.

In GIScience, spatial relations refer to the connection between spatial objects regarding their geometric properties [8] , which play an important role in spatial query, reasoning and question-answering. Using natural language to describe spatial relations is essential for humans to perceive our surroundings and navigate through space. Attempts have been made to formalize the conversion between quantitative models and qualitative human discourse [4]. For topological spatial relations, the RCC-8 (region connection calculus [15]) and the Dimensionally Extended 9-intersection (DE-9IM) model [6] are widely used. Based on the DE-9IM model, five predicates are named by [3] for complex geometries, including *crosses, disjoint, touches, overlaps, within.* On top of them, the Open Geospatial Consortium (OGC) further added the predicates *equals, contains, intersects* for computation convenience. In addition, predicates can also be used to describe the distance or direction between a subject and an object. Fuzzy logic can also be adopted to convert precise metrics into narrative predicates such as *near* and *far* [18].

However, there remains a gap between the contextual semantics of predicates in everyday language and the abovementioned formalization procedures, yielding disagreement and vagueness in the understanding. It is yet to be determined whether the LLMs can fully capture how people describe spatial objects with predicates in natural language. If so, how we can leverage such knowledge to represent geospatial contexts with LLMs.

# 2 Methodology

## 2.1 Workflow

This research focuses on assessing the ability of LLMs in representing geometries and their spatial relations through a set of downstream tasks. Figure 1 illustrates the workflow we employed, which consists of three primary modules. The first module utilizes a GIS tool to extract the attributes, such as geometry type, centroid, and area, of individual geometries and their spatial relations, including predicates and distances between pairs of geometries. The second module applies LLMs to encode the well-known text (WKT) format of geometries, e.g., LINESTRING (30 10, 10 30, 40 40), which includes the geometry type and the ordered coordinates whereas the map projection is not considered in this work. Finally, the obtained embeddings from LLMs, along with the ground-truth attributes or spatial relations, are fed into classifiers or regressors to evaluate the effectiveness of the LLMs-based embeddings.

## 2.2 Notation

The notations used in this paper are listed in Table 1.

## 2.3 Evaluation Tasks

The downstream tasks are designed for deriving the geometric attributes or identifying spatial relations, as described in Table 2. The targets of Tasks 1-5 are straightforward, that is, to train a neural network classification/regression model that can best approximate the ground-truth values computed from a GIS tool. All of these tasks use a Multilayer Perceptron

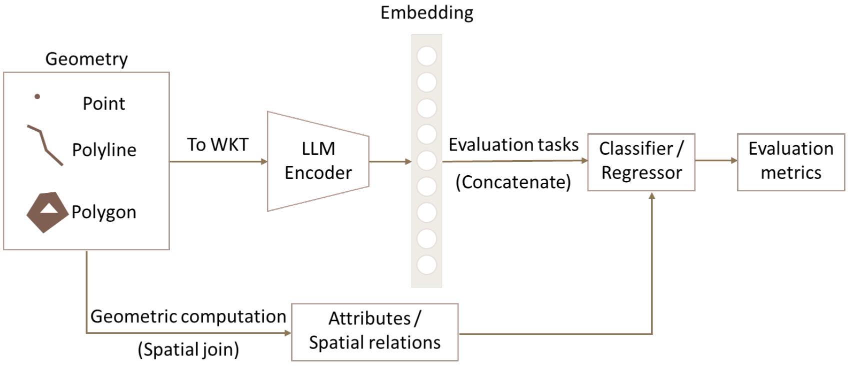

**Figure 1** The evaluation workflow of this research.

**Table 1** Notations.

| Notation | Description |
|---|---|
| $g$ | A geometry instance (e.g. Point, LineString, and Polygon) that can be processed in GIS tools |
| $WKT(g)$ | The WKT format of $g$ |
| $Enc(g)$ | The location encoding of $g$ using a LLM model to encode $WKT(g)$ |
| $Type(g)$ | The geometry type of $g$ |
| $Centroid(g)$ | The centroid of $g$ |
| $Area(g)$ | The area of $g$ |
| $rel$ | A predicate that can be used to represent the spatial relation, which is one of {equals, disjoint, intersects, crosses, touches, contains, within, overlaps}, as defined by OGC and implemented in GeoPandas. |
| $Rel(g_i, g_j)$ | The spatial relation between the subject $g_i$ and the object $g_j$ |
| $Dist(g_i, g_j)$ | The minimum euclidean distance between two objects $g_i$ and $g_j$ |
| $[Enc(g_i); Enc(g_j)]$ | The concatenation of the embeddings of $g_i$ and $g_j$ |
| $Enc(rel, g)$ | The embedding of the short phrase $rel + WKT(g)$. For example, “within Polygon ((0 0, 0 1, 1 1, 1 0, 0 0))” |

(MLP) as the classifier or regressor. Task 6 aims to investigate whether a geometry $g_i$ can be predicted based on its neighbor $g_j$ and their spatial relation $Rel(g_i, g_j)$. We employ the nearest neighbor retrieval approach to evaluate whether LLMs have learned the meaning of spatial predicates properly. During inference, given an object $g_j$ and a spatial relation $rel$, we retrieve the top-k nearest neighbors of $Enc(rel, g_j)$ and examined whether they belong to the set of subjects $\{g_i | Rel(g_i, g_j) = rel\}$. This approach assesses the ability of the LLMs to relate geographic objects through spatial predicates.

**Table 2** Evaluation Tasks.

| Task | Subtask | Model type | Input | Target |
|---|---|---|---|---|
| Geometric attributes | T1: Geometry type | Classification | $Enc(g)$ | $Type(g)$ |
| | T2: Area computation | Regression | $Enc(g)$ | $Area(g)$ |
| | T3: Centroid derivation | Regression | $Enc(g)$ | $Centroid(g)$ |
| Spatial relations | T4: Spatial predicate | Classification | $[Enc(g_i); En(g_j)]$ | $Rel(g_i, g_j)$ |
| | T5: Distance measure | Regression | $[Enc(g_i); En(g_j)]$ | $Dist(g_i, g_j)$ |
| | T6: Location prediction | Retrieval | $Enc(rel, g_j)$ | $\{g_i \| Rel(g_i, g_j) = rel\}$ |

# 3 Experiments

## 3.1 Dataset and Preprocessing

Since there is no available benchmark dataset, we constructed real-world multi-sourced geospatial datasets for our case study in Madison, Wisconsin, United States. We downloaded the OpenStreetMap road network data (including links and intersections) using *OSMnx* [1], points of interest (POIs) categorized by *SLIPO* [2], and *Microsoft Building Footprints* [3]. Our evaluation tasks focus on the spatial objects with *Point, LineString*, and *Polygon* geometry types and assessing their spatial relations, respectively. The datasets are created as follows.

1) For each geometry type, we randomly select 4,000 samples, including 2,000 road intersections and 2,000 POIs for *Point* data, 4,000 road links for *LineString* data, and 4,000 building footprints for *Polygon* data. In total 12,000 samples are used for performing the downstream tasks. The area and centroid of each polygon are also computed.

2) For the spatial predicate *disjoint*, we randomly generate pairs of geometries and check whether their spatial relation is disjoint. For other predicates, we identify spatially related objects using spatial join. Given each combination of subject/object geometry type and their spatial predicate, we keep 400 triplets (subject, predicate, object) for each category for the task of predicate prediction and distance measure. Then we compute the minimum distance between the subjects and the objects.

3) We further construct data for the task of location prediction. In addition to the subjects and objects that are spatially joined in step 2), we also relate neighboring disjoint geometries using a buffer radius of 0.003°. The predicate of "disjoint" is replaced by "disjoint but near". For each predicate except *disjoint*, we select 200 objects of each geometry type that are related to more than 5 subjects by the same predicate.

All the computations are performed by using the *GeoPandas* package in Python. We consider the predicates of *crosses, disjoint (but near), touches, overlaps, within, equals, contains* in this work but not *intersects* as it is the opposite of *disjoint.* The data for downstream tasks are further split into 80% training, 5% validation, and 15% test sets.

## 3.2 Encoding

In this work, we perform the evaluation tasks based on two LLMs: GPT-2 and BERT. Due to the computational and memory resources required to train and use the models, GPT-2 and BERT have a maximum input sequence length (i.e., 1024 and 512 tokens respectively). Therefore, a sliding window approach is employed to tackle the issue as the WKT of *LineString* and *Polygon* types can exceed the length limitation. The long input sequences are broken down into smaller segments of 512 tokens with an overlap of 256 tokens between adjacent segments. Each segment is processed by the LLMs separately. We then take the average of the token embeddings to generate the final embedding for the whole sequence of geometries.

## 3.3 Training MLPs

As we hypothesize that the learned embeddings from LLMs can be effectively utilized in downstream geometry-related tasks, we use a simple neural network architecture (i.e., MLP) across all tasks. Specifically, the input layer of the MLP is the embedding layer generated from LLMs, followed by a dropout layer for regularization purposes. Following the dropout

[1] `http://osmnx.readthedocs.io/`
[2] `http://slipo.eu/`
[3] `http://www.microsoft.com/maps/building-footprints`

layer is a single hidden layer, which employs the Rectified Linear Unit (ReLU) activation function. Finally, the MLP is concluded with the output linear layer. The number of neurons in the output layer varies depending on the specific task.

To facilitate the training process, we apply a logarithmic function to the target values for the area computation and distance measure tasks. In the centroid derivation task, we use the min-max normalization for the target values. The loss function combines the Mean Squared Error (MSE) on both the transformed and original scales. However, for reporting the performance, we only use the original scale of the target values.

## 3.4 Results

As shown in Table 3, the performance of the downstream tasks based on the embeddings generated by GPT-2 and BERT are similar, which can be understood from the similarity in their subword tokenization and transformer-based architecture.

**Table 3** LLMs Performance Comparison.

| Tasks | | Metric | GPT-2 | | BERT | |
|---|---|---|---|---|---|---|
| | | | Validation | Test | Validation | Test |
| T1: Geometry type | | Accuracy(%) | 100 | 100 | 100 | 100 |
| T2: Area computation | All geometries | MAPE(%) | 13124 | 11700 | 12251 | 10850 |
| | Polygon only | | 45.1 | 44.1 | 40.7 | 41.9 |
| T3: Centroid derivation | | RMSE | 0.037 | 0.037 | 0.029 | 0.029 |
| T4: Spatial predicate | Without geometry type | Accuracy(%) | 62.6 | 65.7 | 63.8 | 68.7 |
| | With geometry type | | 73.7 | 71.0 | 73.1 | 72.3 |
| T5: Distance measure | Disjoint only | RMSE | 0.064 | 0.063 | 0.057 | 0.075 |
| T6: Location prediction | | Precision@5 | N/A | 0.03 | N/A | 0.03 |

For T1-T3, the assessment is conducted on individual geometries. The 100% accuracy achieved on both the validation and the test dataset of T1 is expected as the geometry type are words that often occur in text documents. Considering the unit of *degree* in longitude and latitude, significant errors (measured by Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE)) are observed in area and centroid computations, and increasing or reducing the model complexity does not alleviate the issue, suggesting a potential loss of information when averaging the token embeddings or fragmentation of coordinates during tokenization. Training the regressor on all geometries for T2 does not successfully learn that *Point* and *LineString* have an area of 0. Even when training the regressor on *Polygon* separately, the results remain unsatisfactory. In T3, the centroids computed from the high-dimensional embeddings often fall outside the study area. T4-T6 evaluates the embeddings' ability to capture spatial relations. One interesting finding is that the spatial predicate can be better predicted when combined with the geometry type, with accuracy increased from 62%∼68% to 71%∼73%. This can be attributed to the imbalanced spatial relations among different combinations of geometry types. However, the distance measure task T5 still faces challenges in accurately estimating numeric values even when restricted to the "disjoint" relation only. The poor performance on T6 shows that even though the LLMs can encode the spatial relations and geometries in a consistent way, generating embeddings using an average approach alone is insufficient to support spatial reasoning and conduct geometric manipulations directly. Therefore, a different design to enhance the function of localizing spatial objects from textual descriptions [17] can improve the applications of LLMs in GeoAI.

Overall, the results indicate that the LLMs-generated embeddings have encoded the geometry types and coordinates present in the WKT format of geometries. However, it should be noted that the performance of the embeddings does not consistently meet expectations across all evaluation tasks. While the LLMs-generated embeddings can preserve geometry types and capture some spatial relations, challenges remain in estimating numeric values and retrieving spatially related objects due to the loss of magnitude during tokenization [7]. Despite the possibility of ameliorating the issue by modifying notations or applying chain-of-thought prompting [10], this research highlights the need for improvement in terms of capturing the nuances and complexities of the underlying geospatial data and integrating domain knowledge to support various GeoAI applications using LLMs.

## References

1 Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

2 Tom B. Brown et al. Language models are few-shot learners, 2020. `arXiv:2005.14165`.

3 Eliseo Clementini and Paolino Di Felice. A model for representing topological relationships between complex geometric features in spatial databases. *Information sciences*, 90(1-4):121–136, 1996.

4 Anthony G Cohn and Shyamanta M. Hazarika. Qualitative spatial representation and reasoning: An overview. *Fundamenta informaticae*, 46(1-2):1–29, 2001.

5 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

6 Max J Egenhofer. Reasoning about binary topological relations. In *Proceedings of the 2nd Symposium on Advances in Spatial Databases: SSD'91 Zurich, Switzerland, August 28–30*, pages 141–160. Springer, 1991.

7 Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Christian Petersen, Alexis Chevalier, and Julius Berner. Mathematical capabilities of chatgpt. *arXiv preprint arXiv:2301.13867*, 2023.

8 Renzhong Guo. Spatial objects and spatial relationships. *Geo-spatial Information Science*, 1(1):38–42, 1998.

9 Krzysztof Janowicz, Song Gao, Grant McKenzie, Yingjie Hu, and Budhendra Bhaduri. Geoai: spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond. *International Journal of Geographical Information Science*, 34(4):625–636, 2020.

10 Guillaume Lample and François Charton. Deep learning for symbolic mathematics. *arXiv preprint arXiv:1912.01412*, 2019.

11 Zhenlong Li and Huan Ning. Autonomous gis: the next-generation ai-powered gis. *arXiv preprint arXiv:2305.06453*, 2023.

12 Gengchen Mai, Weiming Huang, Jin Sun, Suhang Song, Deepak Mishra, Ninghao Liu, Song Gao, Tianming Liu, Gao Cong, Yingjie Hu, et al. On the opportunities and challenges of foundation models for geospatial artificial intelligence. *arXiv preprint arXiv:2304.06798*, 2023.

13 Gengchen Mai, Krzysztof Janowicz, Yingjie Hu, Song Gao, Bo Yan, Rui Zhu, Ling Cai, and Ni Lao. A review of location encoding for geoai: methods and applications. *International Journal of Geographical Information Science*, 36(4):639–673, 2022.

14 Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

15 David A Randell, Zhan Cui, and Anthony G Cohn. A spatial logic based on regions and connection. *KR*, 92:165–176, 1992.

16 Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, and Nuno Carvalhais. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019.

17 Maria Vasardani, Stephan Winter, and Kai-Florian Richter. Locating place names from place descriptions. *International Journal of Geographical Information Science*, 27(12):2509–2532, 2013.

18 Yang Wang, Huilin Peng, Yiwei Xiong, and Haitao Song. Spatial relationship recognition via heterogeneous representation: A review. *Neurocomputing*, 2023.