Group 27 Project Proposal
CS6220 Big Data Systems

Charles Liu, My Duyen Nguyen, Avinash Atluru, Sarvesh Kasture

September 27, 2024

Motivation and Objectives

To geolocate refers to the ability to accurately identify the geolocation information of an object, event, photo, or video [1]. With the rapid growth in the field of computer vision and geographical data collection, large language models (LLMs) can be an efficient and effective alternative in determining world-wide geographic locations given visual inputs [5]. This field is applicable towards a variety of domains, including urban planning, environmental monitoring, and augmented reality, but it has been widely understudied and underutilized [6].

Based on preliminary research completed during HW1 in the class, we have seen that LLM geolocation performance can be better than average humans. The highest-end models (Claude 3.5, GPT-4o) have been shown to be able to identify popular cities all over the world (i.e. Duluth, Georgia and Da Nang, Vietnam) based on features in each photo. Examples of this include identifying descriptors of Atlanta area codes, Korean writing, and combining this with demographic knowledge it knows about cities, or identifying Da Nang by recognizing Vietnamese language and coastal palm trees, letting it reason that the photo must be in Da Nang. Results are marginally improved with Chain-of-Thought prompting. While LLMs are able to achieve impressive performance because of their generic multi-domain knowledge, it could be bolstered with concrete evidence by fine-tuning with StreetView image data. This can be a redundant way to check easy cases, while allowing the LLM to potentially solve hard cases that can only be solved with few-shot learning.

To improve the accuracy of LLM geolocating capabilities, this paper outlines a two-step method by combining fine-tuning image-coordinate pairs with a novel prompt-engineering approach. The output will be a precise coordinate point that aligns with the image input.

Related Work

There are currently three popular approaches to solving geolocation. Representation models include GNN-based models, which uses human statistical data to construct graphs of geography. It also includes image-based models that take images as input that are then associated with specific coordinates. Lastly, there are NLP-based models that embed text descriptions of coordinates to get corresponding solutions [5]. Specifically, looking at image-based models, getting hands on street view image data is costly [5].

There has been some recent work in evaluating and improving geolocation tasks with LLMs. He et. al (2024) discusses a novel method for geolocation representation called LLMGeovec, which uses large language models (LLMs) and OpenStreetMap data to enhance spatio-temporal learning tasks. Traditionally, geospatial representation models require expensive data inputs like street views and mobility data, making them less prevalent compared to models in natural language processing (NLP) and computer vision [5]. However, there are Kaggle datasets with StreetView data (under NLPv3 license), with 25,000 StreetView images that we can use with associated coordinate data. There is also a 6 million general, geo-tagged dataset provided by MediaEval.

We are interested in the methodology of a recent paper, detailing a new model called LLMGeo (Wang et. al, 2024). The authors fine-tune open-source LLMs with StreetView image data. LLMGeo achieved comparable performance with closed-source LLM models. Their level of granularity came down to the country-level. In future steps, they discussed narrowing down specificity to latitude and longitude. This is the area of interest that we aim to explore with our project.

Additionally, planet-scale image geolocation is difficult because images come from anywhere in the world. Current methods have improved accuracy but only work well for specific landmarks and struggle in new locations. Haas et. al (2024) shows high accuracy in GeoGuessr against pro players, beating said players 100% of the time. Novel characteristics of their PIGEON model is using semantic geocells, or partitioning the globe into human meaningful sections, as human political boundaries not only correspond to different street signs, but also are usually on discernible geographical features like rivers or mountains that encode meaningful geographical boundaries. Vision transformer models using OpenAI's CLIP, combined with synthetic are the crux of how they tune their models on image data.

LLaVA-1.5 and Qwen-VL prove to be capable, open-source, multimodal LLMs that perform well on geographic tasks, even better than GPT-4V, in some cases like object localization [6].

Proposed Work

For this project, we will initially start with an existing, open-source multimodal LLM such as LLaMA 3.2 or LLaVA-1.5, where our group will perform benchmarking. The benchmarking will be done based on accuracy of the location as well as other geographic attributes such as climate, population, and other values. For this, we will consider metrics such as mean squared error, mean absolute error, and the coefficient of determination (R^2). Based on these values, we will get an initial baseline which can be used to see if the fine tuning done improves the model with regards to its geolocation ability.

For the fine tuning, we will use image data from StreetView. The first step in the process will be to pre-process the data to add captions about climate, population, and other factors to add context to each photo, which can be appended. LLM image fine-tuning was shown to be effective in [6], and image captioning was shown to be effective in PIGEON for adding context (Haas et. al, 2024). As for coordinates, we will either use an image dataset of over 25,000 StreetView images from Kaggle or another dataset of 6 million geo-tagged images from MediaEval [3, 4]. The variety and randomness of the images is a valuable resource for fine-tuning geolocation. The data includes less-documented locations which is an important addition to improve performances in more underrepresented areas. For this, we can use other models such as CLIP which is an image based model. These images, coordinates, and captions will be passed into these multimodal LLMs that can use visual transformers to embed these combined data points into vectors in our latent space.

In order to fine tune the models, we can use resources provided directly from the LLMs. For example, OpenAI provides an API that can be used to fine tune the model. We can also use open source libraries such as Hugging Face. We expect to use Georgia Tech's AI Maker Space as a training resource. For the novelty of this project, we will specifically explore the benefits of adapter tuning as the method of fine tuning, and this method essentially allows us to add a few adapter layers to the model in order to make it specialized over the geolocation task. This specific method is seen to be more time and resource efficient compared to standardized fine tuning. Since the process for improving geolocation tasks with adapter tuning has not been done, this presents a new field to explore.

After fine-tuning, we propose a specific prompt-engineering method to guide the LLM in its search. In order to effectively replicate the effects of the novel semantic geocells discussed in [2], we will implement a methodology that instructs the LLM to first narrow down the search itself with its foundational model. From our preliminary research, models like Claude have been promising in their ability to reason for a photo's location using a variety of corresponding data. Some methods include using OCR to read street signs, correlating telephone area codes to certain areas, and using demographic data like Korean American populations in the US to narrow down to certain regions of the world (see Motivations & Objectives). By combining both methods, we aim to develop a more reliable geolocating method that produces accurate results.

Once we have applied the fine tuning and generated the specific prompt-engineering method, we will then benchmark the performance of the model on the testing dataset and compare to see the performance change. We will also conduct benchmarks on the model on different metrics such as performance against amount of data given, training loss, number of epochs required for the model to achieve a certain level of accuracy, noisy data, and more. This is further discussed in the next section.

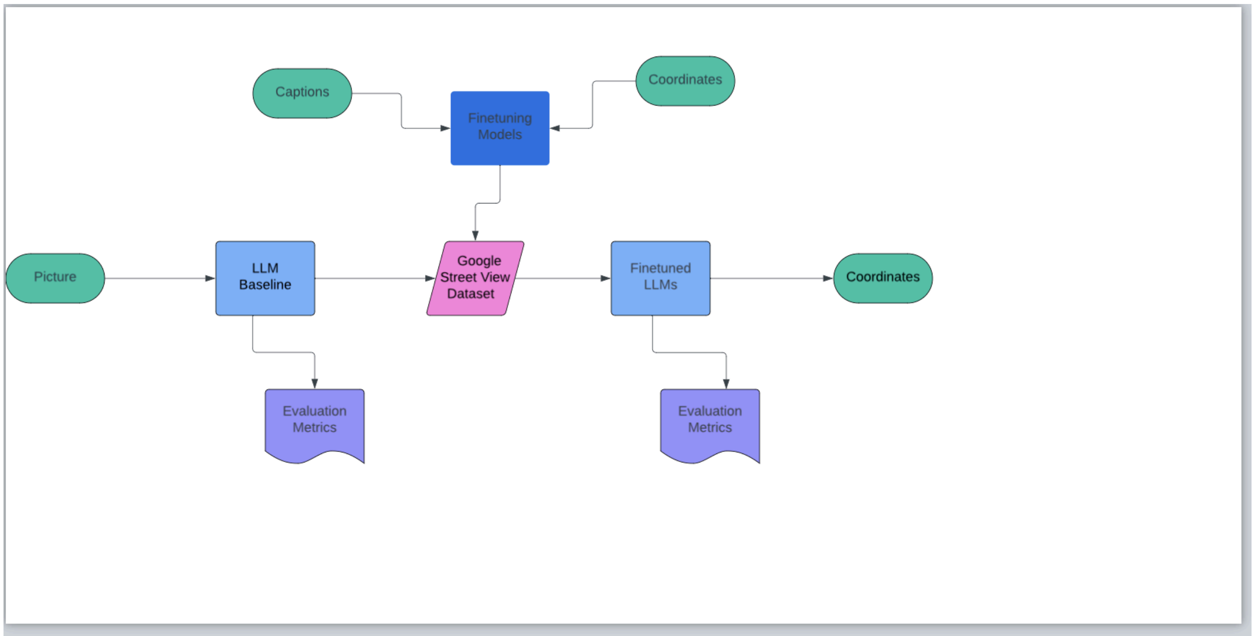

Figure 1. Architectural Diagram

Plan Of Action

Week 1: Sept 29th - Oct 5th

- Meet with Professor Liu to discuss proposal

Week 2: Oct 6th - Oct 12th

- Dive into the data source
- Find specific dataset to use for training
- Initialize process to preprocess the data
- Decide which LLMs to use

Week 3: Oct 13th - Oct 19th

- Finish Preprocessing Data
- Benchmark the LLMs before fine-tuning

Week 4: Oct 20th - Oct 26th

- Build Adapter Fine-Tuning
- Start Prompting

Week 5: Oct 27th - Nov 2nd

- Fine-tune the model
- Finish Prompting Portion

Week 6: Nov 3rd - Nov 9th

- Finish-tuning and testing the model

Week 7: Nov 10th - Nov 16th

- Conduct benchmarking on the model

Week 8: Nov 17th - Nov 23rd

- Prepare presentation and data

Evaluation and Testing Method

To begin measuring the performance of the model, we want to take a look at how existing LLMs work on identifying different types of locations at a coordinate level. This would include densely populated areas, rural areas, and racially homogenous areas to identify aspects that allow our model to improve on others. For baselines, we plan to use both standard LLMs (before fine-tuning) such as GPT-4 or Claude-3.5 and Geo-specific models such as PlaNet and DeepGeo. Using the testing dataset, we will conduct our measurements using three tests. The first one will be the MSE (mean squared error) which takes the square of the difference of prediction and actual geographic attributes. Another test we would conduct would be the Coefficient of Determination (R^2) which measures how close the model predictions are to the actual attributes and assigns a score with closer to 1 being a better model.

The testing dataset that we use will have the geotagged images meaning the location and attributes for each image are already known allowing for easy testing of the performance. Additionally for pre-existing geolocation models, we want to conduct a couple more tests such as for models that output the location, we can calculate the haversine distance which tells us the shortest path from the prediction to the actual location on a sphere. The goal for our model would be that we can output a lower haversine distance than the pre-existing models. Another metric to test for would be the top-k accuracy. This would analyze multiple possible locations for the model to determine whether the correct location was within a certain threshold. An example could be top-5 accuracy which would check that out of the best 5 predictions, did our model accurately find the location. To determine this, we would set a certain area for the prediction using the haversine distance and then if a prediction lies in that range, it would be considered a valid prediction. We would use this to find the percentage of the outputs that lie in a top-k accuracy for a k of our choice i.e. 5 or 10.

Finally for analyzing improvements, we would run the MSE and R^2 test on our model and evaluate the scores in comparison to the pre-existing models. The values for the prediction and actual would be computed using the haversine distance and help us to identify the performance just like we did for the baseline models. Additionally to evaluate the model we would discover training efficiency by looking at the number of epochs and the training loss to identify how fast our model is able to converge to the optimal solution. This would go along with looking into scalability by understanding the effect of increasing and decreasing the dataset on accuracy. One last measure would be the introduction of noisy data on our model such as blurry/low-resolution images. By going through these different measurements, our goal is to identify an approach that outperforms pre-existing solutions especially for more underrepresented areas.

Bibliography

1. *Geolocation Methods: A step by step guide — The Kit 1.0 documentation*. (n.d.). https://kit.exposingtheinvisible.org/en/geolocation.html
2. Haas, L., Skreta, M., Alberti, S., & Finn, C. (2024, May 28). *Pigeon: Predicting image geolocations*. arXiv.org. https://arxiv.org/abs/2307.05845
3. Jaeyoung Choi, Bart Thomee, Gerald Friedland, Liangliang Cao, Karl Ni, Damian Borth, Benjamin Elizalde, Luke Gottlieb, Carmen Carrano, Roger Pearce, and Doug Poland. 2014. The Placing Task: A Large-Scale Geo-Estimation Challenge for Social-Media Videos and Images. In *Proceedings of the 3rd ACM Multimedia Workshop on Geotagging and Its Applications in Multimedia (GeoMM '14)*. New York: ACM, 27-31. PDF.
4. Jagtap, Shriyash. "Geo-Location Extraction Data for LLM." Kaggle, 26 Feb. 2024, www.kaggle.com/datasets/shriyashjagtap/geo-location-extraction-data-for-llm.
5. Roberts, J., Lüddecke, T., Sheikh, R., Han, K., & Albanie, S. (2024, January 16). *Charting new territories: Exploring the geographic and geospatial capabilities of multimodal llms*. arXiv.org. https://arxiv.org/abs/2311.14656
6. Wang, Z., Xu, D., Shahroz, R. M., Khan, Y., Lin, Y., & Fan, Z. (2024, May 30). *LLMGeo: Benchmarking large language models on image geolocation in-the-wild.* Florida A. & M University. https://arxiv.org/html/2405.20363v1
7. Xiao, Zhaomin, Huang, Yan, & Blanco, Eduardo. (2024). *Analyzing Large Language Models' Capability in Location Prediction.* Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), 951–958, Torino, Italia. ELRA and ICCL. https://aclanthology.org/2024.lrec-main.85.pdf