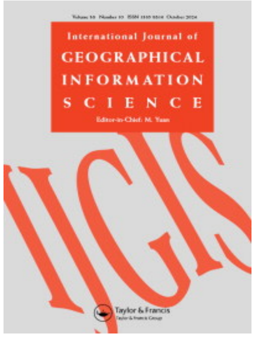

Taylor & Francis
Taylor & Francis Group

International Journal of Geographical Information Science

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tgis20

Enhancing the accessibility of regionalization techniques through large language models: a case study in conversational agent guidance

Xin Feng & Yuanpei Cao

INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE
https://doi.org/10.1080/13658816.2024.2415439

Taylor & Francis
Taylor & Francis Group

Check for updates

RESEARCH ARTICLE

Enhancing the accessibility of regionalization techniques through large language models: a case study in conversational agent guidance

Xin Feng[a] and Yuanpei Cao[b]

[a]Department of Geography and Environmental Sustainability, University of Oklahoma, Norman, OK, USA; [b]Airbnb, Inc, San Francisco, CA, USA

ABSTRACT
The concept of regions has long been crucial for understanding and managing Earth's phenomena, leading to regionalization, aggregating smaller areas into larger, contiguous, and homogeneous regions to achieve specific goals. Open-source regionalization is gaining traction because it reduces dependence on commercial software and fosters wider adoption in analysis and decision-making. However, these packages, often designed by experts for specialized tasks, can be challenging to understand and utilize due to domain-specific jargon and functionalities, especially for unfamiliar users. A prevalent disconnect must be addressed: How can we make a complex optimization approach available to a broad audience with various backgrounds? This study introduces RegionDefiner, a Large Language Modeling (LLM)-powered conversational agent, to comprehensively understand the functionality, inputs, outputs, and potential applications of regionalization problems. We selected it as an illustrative example due to its wide-ranging potential for delineating study regions in various applications. RegionDefiner is designed to guide users in framing their problems, collecting necessary data, and implementing solution approaches in a straightforward and user-centric manner. The experiments demonstrate that RegionDefiner interprets and presents the results in an understandable way for all audiences, thus bridging the gap between intricate computations and practical problem-solving needs.

ARTICLE HISTORY
Received 16 February 2024
Accepted 8 October 2024

KEYWORDS
Large language models; conversational virtual agent; spatial optimization; regionalization

Introduction

In geography and other Earth-related disciplines, the concept of regions has always held significance (Kostbade 1968, Richardson 1992). Defining, characterizing, and explaining regions has been a crucial part of study endeavors to comprehend and oversee the Earth and its diverse phenomena (Openshaw 1996, Montello 2003, Duque *et al.* 2007). Various disciplines emphasize different types of regions, such as environmental management zones (Fovell and Fovell 1993), socio-economic units (Assunção *et al.* 2006), school

CONTACT Xin Feng selena.feng@ou.edu

districts (Wei *et al.* 2022), political districts (Macmillan and Pierce 1994), and areas susceptible to a particular disease type (Martins-Bedê *et al.* 2009). Regionalization, known as region building, has grown into a significant field of study, involving the aggregation of numerous smaller areas into a reduced amount of larger, contiguous, and/or homogeneous regions. From a mathematical standpoint, regionalization can be regarded as a process of solving a constrained optimization problem. The problem attempts to determine the boundaries of regions in a way that pursues specific goals, subject to certain geospatial restrictions. Specifically, regions are delineated to maximize homogeneity within regions while maximizing heterogeneity between regions, considering regional continuity and/or compactness (Li *et al.* 2014, Wei *et al.* 2021, Feng *et al.* 2022a).

The importance of region building cannot be overstated, but the complexity associated with the regionalization models and their solution approaches continues to inhibit the full realization of their potential. The relevant application always requires a high level of mathematical background and programming skills that create a barrier to entry for those outside domains. As a result, a significant proportion of the target audience, including non-technical stakeholders, policymakers, and some researchers in fields other than GIS, have restricted access to existing regionalization algorithms and tools. This limited accessibility not only limits the utilization of algorithms and tools but also hinders relevant innovation and informed decision-making.

Open-source approaches to regionalization have received attention because their implementation promotes the reproducibility and replicability of regionalization and other GIScience methods to support analysis and decision-making. Transparency in determining regions will enhance the quality control of results, enable other researchers to verify the findings and conduct further research, encourage research collaboration, and facilitate rapid innovation. Existing open-source packages that allow the implementation of regionalization methods emphasize some specific fields, including climate (e.g. HiClimR (Badr *et al.* 2015), synoptReg (Lemus-Canovas *et al.* 2019)), biology (e.g. Phyloregion (Daru *et al.* 2020), regioneR (Gel *et al.* 2016)), hydrology (e.g. nsRFA (Viglione 2009)), social-economic systems (e.g. maxcut (Rahman 2019)). While most open-source packages have significantly reduced the learning curve through comprehensive documentation, a notable limitation is that they utilize terminologies, functionalities, or workflows that remain obscure to individuals outside that realm. This domain-specific design requires users, especially non-technical users, to invest a lot of time and effort to leverage these tools entirely.

Can these complex approaches be accessible to all users by interacting with a conversational agent? The answer is both affirmative and negative. The conversational agent is skilled at explaining specialized topics and helping users complete activities, bridging the gap between specialized approaches and a broad, diverse audience. Conversational agents, commonly known as chatbots, have been widely used commercially, but their ability to engage in comprehensive, domain-specific, solution-oriented conversations was minimal. For example, the Airbnb Bot (Airbnb 2020, 2021) can classify the user's query into predefined customer service issue categories and then match relevant actionable resolutions. However, the conversation usually ends after one or two rounds of dialogue and is promptly handed over to human personnel. Additionally, more sophisticated platforms exist, such as Amazon Alexa, Siri, and Google Assistant, which can provide more seamless conversational interactions. Nevertheless, their strategies focus on generating educational content that is obtained using information retrieval methods rather than directly solving problems.

The advancement of Large Language Models (LLMs) has fed a spectacular growth of conversational agents. LLMs, such as GPT-3 (Brown *et al.* 2020), PaLM (Chowdhery *et al.* 2023), LLaMA (Touvron *et al.* 2023), Microsoft Copilot (Microsoft 2023), ChatGPT (OpenAI 2022), and the latest GPT-4 (OpenAI 2023), has led to significant breakthroughs in Natural Language Processing (NLP). These advanced LLMs have shown remarkable proficiency in comprehending human queries and performing various NLP tasks (Feng *et al.* 2020, Chen *et al.* 2021, Dong *et al.* 2022, Zhao *et al.* 2023). This opens the potential to develop a highly efficient conversational agent tailored to specific domains. In particular, several studies have been conducted on augmenting LLMs with external tools like web search engines, HuggingFace neural network models, code interpreters, answer set programming, spatial-temporal information retrieval, and others (Radford et al. 2018, 2019, Nakano *et al.* 2021, Dai *et al.* 2023, Qian *et al.* 2023, Qin *et al*. 2023, Shen *et al.* 2023, Zeng *et al.* 2024). They aim to enhance the efficiency and accuracy of LLMs in addressing real-world problems. Furthermore, platforms such as AutoGPT (Richards 2023), LangChain, and OpenAI Assistants provide accessible resources for developers to effortlessly utilize sophisticated LLM, and it streamlines the creation of a specialized conversational agent enhanced by customized logical reasoning and external tools.

Inspired by these new LLM-based technologies, in this paper, we developed RegionDefiner, an innovative conversational agent crafted using the OpenAI Assistants API and powered by GPT-4, designed explicitly for the regionalization domain. As illustrated in Figure 1, contrasting with conventional processes where practitioners engage in exhaustive resource searches, acquire domain-specific knowledge, and delve into coding and algorithm execution, RegionDefiner is designed to serve users with interactive and instructive guidance on regionalization concepts. Utilizing OpenAI Assistants' advanced capabilities, including persistent message threading, comprehensive conversation history storage, and sophisticated tools like File Search and Function Calling, RegionDefiner adeptly navigates the complexities associated with regionalization inquiries.

Key contributions of this work to the GIS field include the following three items. Firstly, it highlights the responsible use of LLMs for geographic applications, including geographic knowledge extraction, GIS automation, and the development of autonomous GIS bots. Creating a customized GPT-4-based conversational bot integrated with GIS-specific open-source package enables more intuitive and accessible interactions with complex GIS algorithms. RegionDefiner's ability to execute code and access local files enhances its utility in practical applications, a capability typically limited in classic conversational agents. Secondly, this work involves understanding LLMs through a geographic lens. By incorporating regionalization-relevant documents and predefined wrapper functions, we enhanced RegionDefiner's ability to explain complex GIS concepts and algorithms, making them more accessible to non-technical users compared to general-purpose bots like ChatGPT-3.5, GPT-4, and Microsoft Copilot. Lastly, this work exemplifies the democratization of access to sophisticated GIS tools and knowledge. Specifically, RegionDefiner supports users by providing interactive and instructive guidance on regionalization concepts,

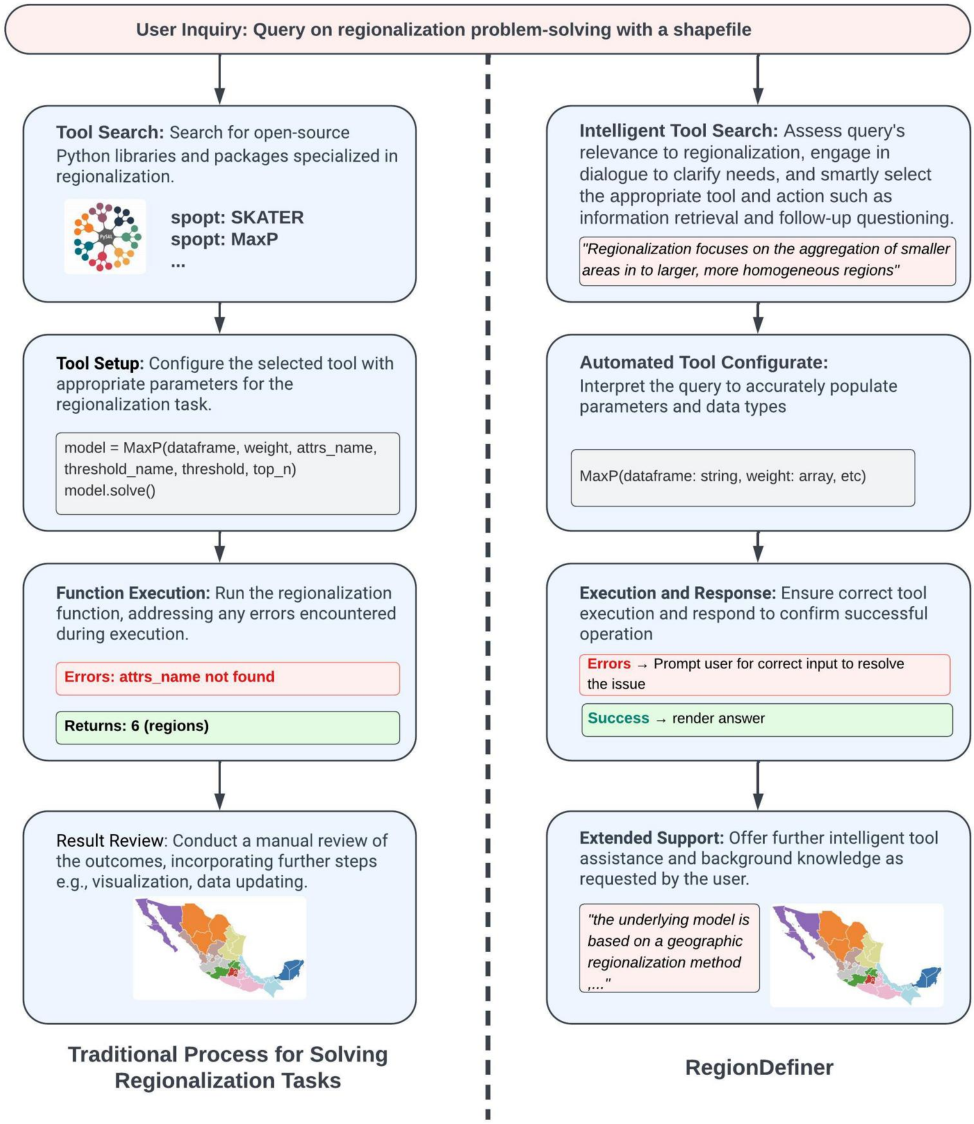

Figure 1. Comparison between (left) the traditional process of tackling regionalization tasks by open-source Python libraries and (right) the simplified approach offered by RegionDefiner, an interactive conversational agent that aids in regionalization problem-solving.

effectively enhancing code accessibility and user engagement. The development of RegionDefiner demonstrates how LLMs can facilitate coding, which is of great importance and significance to GIScience and spatial analysis more broadly.

Methodology

We introduce RegionDefiner (see the overall framework in Figure 2), an LLM-powered interactive conversational agent designed to facilitate users in addressing and solving regionalization-related tasks. Built upon the OpenAI Assistants platform, RegionDefiner

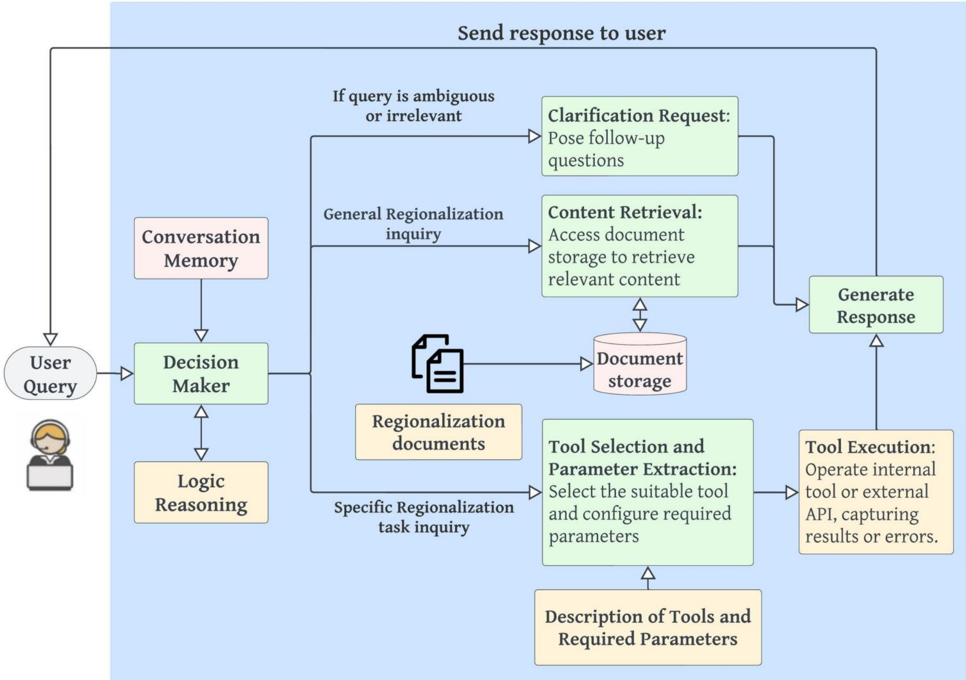

Figure 2. Architecture of RegionDefiner (in blue background): Boxes in green are managed by GPT, those in orange are pre-established by regionalization tool developers or automatically executed locally, and the pink boxes represent cache memory or database storage.

establishes an autonomous dialogue system characterized by a sophisticated chain-of-thought operational mode, facilitating seamless integration with various GPT models and the ChatGPT API. This advanced conversational structure detects users' intentions, retrieves documents, and deploys functions to address users' tasks. Specifically, the core of RegionDefiner is encapsulated in the Decision Maker component (described in Figure 3), which orchestrates the logical reasoning process to decide subsequent actionable steps by identifying the intent of user queries. In addition to the Decision Maker component, RegionDefiner has access to a collection of regionalization tools and a repository of domain-specific documents. This access enables the agent to extract domain knowledge and operationalize the requisite tools to comprehend and address users' problems efficiently. A pivotal feature of RegionDefiner is its ability to identify suitable tools and accurately extract requisite function parameters directly from user queries. These tools resemble the semantic parsing methods commonly employed in conventional NLP question-answering frameworks (which necessitate reference insertion), but they can manage more intricate tasks. In the following subsections, we will describe the overall structure and foundational components of RegionDefiner, highlighting the agent's role in effectively addressing users' regionalization challenges.

Overall structure

We developed a chain-of-thought structure for the RegionDefiner conversational agent utilizing the OpenAI Assistants platform and cutting-edge LLMs such as GPT-4. This

```
As a Regionalization Expert Virtual Assistant, you specialize in regionalization tasks using specific documents, tools, and definitions. Construct your responses as follows:

1. Identify Inquiry Type:
  - Non-Regionalization Topics: Inform users that your focus is on regionalization and you do not handle unrelated topics such as facility location, resource allocation, route planning, site selection, and urban planning, etc.
  - Regionalization Queries:
    - General Questions: Use provided documents to share principles and methods related to regionalization.
    - Specific Analysis: Execute analyses using designated tools:
      - Max-p Problems: Conduct analyses on region count or allocation using 'get_maxp_response' tool tailored for Max-P region problems, considering specified thresholds.
      - P-Region Problems: Designate regions using 'get_p_regions_response' tool equipped with algorithms like Skater, AZP, Regional K-means, Spenc, Ward, and Random Region.

2. Conduct Specific Analyses:
  - Outline Workflow: First, outline the workflow to address the query, ensuring understanding of the problem and identifying the most suitable analytical tool.
  - Apply Tool: Precisely apply the appropriate tool, requesting more information if needed for clarity.

3. Offer General Informational Support:
  - Utilize Documents: For broader questions, use the provided documents to offer clear and concise insights into regionalization.

Foundational Knowledge:
- Regionalization Overview: Aggregating smaller areas into larger, contiguous, and homogeneous regions, optimizing for internal homogeneity and external heterogeneity under geospatial constraints. Application examples include spatial unit identification, zonal object design, etc.
- Max-P Region Problem: Clusters areas into the maximum number of regions where a specific attribute exceeds a threshold.
- P-Region Problem Algorithms:
   - Skater: Groups areas into contiguous, homogeneous regions based on a connectivity graph.
   - AZP: Forms regions maximizing internal homogeneity.
   - Regional K-Means: Clusters areas based on attribute similarities with spatial contiguity constraints.
   - Spenc: Ideal for non-convex clusters using graph cuts weighted by spatial proximity.
   - Ward: Merges areas into clusters based on minimum variance with spatial connectivity.
   - Random Region: Randomly groups areas into regions for comparative analysis.

Response Output Guidelines:
- Clarity: Use simple language and define technical terms clearly. Break complex concepts into understandable parts.
- Relevance: Ensure responses directly address the user's questions or explain how the information provided relates to their inquiry.
- Expert Guidance: For complex issues beyond the assistant's scope, recommend contacting SPOPT developers.
- Conciseness: Ensure responses are concise, clear, and directly relevant to the user's inquiry.
```

Figure 3. Overview of RegionDefiner's Instruction Prompt in the Decision Maker component.

structure divides complex regionalization-related tasks into a series of less complicated, intermediate steps. The multi-step reasoning methodology, which deviates from the practice of single-step response generation in classical LLMs, closely resembles the human way of addressing intricate issues. Furthermore, OpenAI Assistants' intrinsic features for conversation history management and document storage eliminate the need for separate systems to manage conversational flow and documents in the development of RegionDefiner. This streamlined approach enables us to customize the conversational agent for domain-specific needs. As shown in Figure 2, RegionDefiner includes three specialized components:

- **Decision Maker:** This component acts as the 'brain' of RegionDefiner. Supported by an instructional prompt, Decision Maker employs a sophisticated logical reasoning process to address the user's inquiry systematically. The instructional prompt establishes the overall behavior of the RegionDefiner conversational agent, understanding requests, initiating actions, and providing educational insights through a structured, multi-step reasoning approach.
- **Document Retrieval:** This component leverages the File Search tool in the OpenAI platform to dynamically retrieve relevant content from a curated collection of academic papers and guides on regionalization in response to user queries.

- **Function Calling:** This component includes (1) Tool Selection and Parameters Extraction and (2) Tool Execution. Leveraging the Function Calling tool in the OpenAI platform, RegionDefiner selects the regionalization function most relevant to the user's task from open-source packages such as PySAL's **spopt**. Once the appropriate regionalization algorithm and its parameters are determined, the wrapper functions we developed will autonomously execute the corresponding function and provide structured outputs on the local device.

RegionDefiner activates GPT-4 through prompts, which integrate the static system prompt from the Decision Maker component with the static and dynamic prompts obtained from the Document Retrieval and Function Calling components. Typically, GPT-4 will be triggered multiple times for regionalization-related queries. Initially, the static Decision Maker prompt decides whether to engage in Document Retrieval or Function Calling or to respond directly. If the File Search and/or Function Calling tools are triggered, their outputs will be combined with the static prompt for the following GPT-4 trigger. Specifically, the content fetched in the Document Retrieval process will be integrated into a dynamic prompt, ensuring that relevant information from selected documents is included. This methodical selection guarantees the relevance and accuracy of the information, enriching the prompt with contextually pertinent data. If Function Calling is needed, a static prompt predefined will trigger GPT-4 to determine which function to call and how to trigger it by evaluating function parameter descriptions and algorithm descriptions, ensuring intelligent tool selection and parameter extraction. After the selected function is executed, its outputs, including result summaries and error diagnostics predefined in wrapper functions, will be encapsulated and incorporated into a dynamic response prompt.

In short, RegionDefiner synthesizes the outcomes from Document Retrieval and Tool Execution to construct responses. These responses offer educational insights or highlight issues, guiding users toward appropriate next steps. The dynamic enrichment of the final input prompt, courtesy of outputs from both Document Retrieval and Tool Execution, ensures that the LLM receives a comprehensive and context-rich prompt. This structured approach facilitates precise responses to user queries, making advanced regionalization techniques accessible to users with varying expertise. This dialogue progresses until RegionDefiner fully addresses the user's concern, which is indicated by the cessation of further queries.

Decision Maker

The Decision Maker component utilizes a structured prompt to conduct logical reasoning. The prompt (see Figure 3) serves four purposes: Intent Detection, Action Guidance, Knowledge Incorporation, and Response Guidelines. Each purpose plays a crucial role in forming a sequential, chain-of-thought reasoning pattern, like how people do problem-solving.

- **Intent Detection:** This initial phase categorizes the user's inquiry into non-regionalization topics, general inquiries, and targeted analysis requests. RegionDefiner effectively navigates non-regionalization queries by redirecting users, thereby maintaining a focused discourse within its area of expertise. This categorization is pivotal in steering the conversational trajectory toward answering queries particular to regionalization.

- **Action Guidance:** It's for eliciting further information, accessing relevant content for general queries, and executing analytical tools. In response to regionalization-associated questions, RegionDefiner adopts a dual approach:
  - For general questions, RegionDefiner utilizes the 'Document Retrieval' capability to retrieve the most relevant answers from pre-uploaded documents.
  - For targeted analyses, RegionDefiner employs specialized tools designed for these specific tasks.
- **Knowledge Incorporation:** RegionDefiner is fed with a comprehensive definition of regionalization, explaining its mathematical basis as aggregating smaller areas into larger, more homogeneous regions under specific geospatial constraints. It also includes descriptions of pivotal algorithms such as the SKATER for creating a pre-known number of contiguous, homogeneous regions. This foundational knowledge enables RegionDefiner to effectively identify intention, access relevant content, and choose appropriate tools.
- **Response Guideline:** We program RegionDefiner to adhere to stringent guidelines in crafting responses, ensuring they are insightful, contextually relevant, and informed by the foundational knowledge embedded within the provided documents. Notably, RegionDefiner offers a textually described workflow along with corresponding operations, enabling users to understand the problem-solving process and supporting them in replicating and customizing the analysis.

This structured architecture enables RegionDefiner to emulate a logical and sequential reasoning process that involves breaking down intricate issues into more manageable and practical phases. The technique not only mimics the human problem-solving process but also significantly improves its effectiveness in assisting users with tasks linked to regionalization.

Document Retrieval

RegionDefiner is equipped with a 'Document Retrieval' component that leverages OpenAI's 'File Search' tool. This tool can handle up to 10,000 files, parsing, chunking, and embedding documents to ensure efficient and accurate retrieval. Parallel and multi-threaded search operations enable quick access to relevant information, and advanced query rewriting and reranking techniques optimize the relevance of search results. We carefully selected a collection of academic papers and guides on regionalization, focusing on critical algorithms, including the Max-P Region-Problem (Duque *et al.* 2012, Wei *et al.* 2021), the P-Region Problem (Duque *et al.* 2011), the SKATER algorithm (Assunção *et al.* 2006), etc. It is worth noticing that even though we only consider these algorithms in this proof-of-concept paper, the Document Retrieval component can involve more regionalization algorithms. RegionDefiner distinguishes itself from conventional LLM conversational agents by directly accessing stored papers to include the most recent academic advancements in regionalization. This allows RegionDefiner to generate accurate and well-informed replies to user inquiries instead of relying simply on pre-trained general knowledge bases. The integration of Retrieval tools increases the utility of this conversational agent, making it a valuable asset for anyone seeking expert-level assistance with regionalization tasks and theory.

Tool selection and parameters extraction

OpenAI's Function Calling feature enables advanced LLMs like GPT-4o, GPT-4-turbo, and GPT-4 to select tools and extract parameters intelligently. To leverage this capability, we developed wrapper functions that enable the execution of specific spatial optimization algorithms from open-sourced Python packages (PySAL: **spopt**, Feng *et al.* 2022b) and the generation of interpretable responses by LLMs. For example, two wrapper functions in Table 1 are dedicated to executing Max-P-region and P-region heuristic algorithms. These wrapper functions can provide informative outputs, such as a summary message like '*The regionalization process was successful, resulting in* $\{n\_cluster\}$ *regions*' or error diagnostics like '*Error:* $\{e\}$. *Please provide a valid list of attribute names*'. The wrapper functions in this process are crucial as they encapsulate and incorporate the outputs of the executed functions in a dynamic prompt, efficiently informing subsequent LLM decisions and guiding proper user interactions.

To ensure accurate tool selection and parameter extraction by the underlying LLMs, these functions are further meticulously defined within the OpenAI Assistants' Function Calling tool, adhering to a prescribed schema shown in Tables 1 and 2. This schema includes:

Table 1. Overview of wrapper functions utilized by RegionDefiner, configured within OpenAI Assistants' Function Calling.

Function Name	Description	List of Required Parameters
get_maxp_response	Cluster a set of geographic areas into the maximum number of homogeneous and spatially contiguous regions such that the value of a spatially extensive regional attribute is above a predefined threshold.	['*threshold*', '*threshold_name*', '*attrs_name*', '*input_file_path*']
get_p_regions_response	Group a set of geographic areas into spatially contiguous regions according to the predefined number of regions and attributes that quantify regional similarity.	['*n_clusters*', '*attrs_name*', '*algorithm*', '*input_file_path*']

Table 2. Detailed specification of parameters for wrapper functions in RegionDefiner, configured in Function Calling of OpenAI Assistants.

Parameter Name	Description	Type
threshold	The predefined threshold value required for each region.	Integer
threshold_name	Attribute name used for thresholding—the name of the spatially extensive attribute variable.	String
attrs_name	The variables (attributes) in the data frame (attributes table) will be used to measure regional homogeneity.	Array of String
n_clusters	The desired number of contiguous regions into which spatial units are aggregated.	Integer
algorithm	The selected p-region algorithm.	String
input_file_path	Directory for loading shapefile. It should end with '.shp'. This must be explicitly provided by the user. DO NOT Infer a file path and it is not the file already uploaded.	String
plot_results	Set to True to visualize the regionalization results. Defaults to False if not specified.	Boolean

- **Function name and description**: It aids LLMs in selecting the appropriate function from the available library.
- **Parameter Specifications:** Parameter information, including their types and other detailed descriptions, equips LLMs with the context needed for extracting and formatting parameters, as outlined in Table 2.
- **List of Required Parameters**: This list provides crucial parameters to run the algorithms correctly. It highlights the need for precise user instructions to enable the LLMs to extract parameters accurately. RegionDefiner uses this checklist to prompt users for more details, ensuring all essential parameters are gathered from the user's input before proceeding.

Tool execution

It's noteworthy that OpenAI itself doesn't directly execute the function. RegionDefiner autonomously executes these functions on a local device by using our built-in logic after choosing the appropriate wrapper function and extracting the required parameters. The outcomes of this specific execution are transmitted back to the LLM that has been set up in RegionDefiner. LLMs thereafter generate customized replies to assist users in navigating the subsequent stages of the inquiry procedure.

Experiments and results

This section highlights interactions between users and RegionDefiner powered by the GPT-4-turbo-preview model, an upgraded version of GPT-4. Below, we present two sample conversations for your reference.

Conversation 1: understanding regionalization

Figure 4 illustrates how RegionDefiner specifically tackles inquiries related to habitat protection posed by users. The user is seeking guidance on strategies for preserving habitats to mitigate the risk of species extinction, specifically emphasizing optimal placement and connectivity of these habitats. RegionDefiner explicitly emphasizes regionalization, explaining its methodology of grouping areas according to similar habitat types or species distributions while not directly addressing the placement of facilities or allocation of resources. Further in the discussion, RegionDefiner clarifies the principles of regionalization and differentiates it from generic clustering techniques, drawing upon the information provided in the prompt of Figure 3 and preloaded regionalization papers. This dialogue demonstrates RegionDefiner's proficiency in guiding users through their inquiries and educating them on regionalization and associated applications.

Conversation 2: regionalization problem solving

Figures 5 and 6 document a dialogue where RegionDefiner skillfully navigates the user to group states into regions to analyze regional development in Mexico (a case study in Rey and Sastré-Gutiérrez 2010). This sample data is from PySAL, but

User: We want to set aside appropriate amounts and types of habitats to protect species from becoming extinct. Where should we place them? How to connect these selected habitats?

RegionDefiner: Your question involves aspects of conservation planning, including habitat selection and connectivity, which are crucial for biodiversity preservation. However, these topics fall outside my specialization in regionalization, which focuses on aggregating smaller areas into larger, contiguous, and homogeneous regions based on specific criteria and constraints. My expertise is more aligned with tasks such as spatial unit identification and zonal object design, rather than specific conservation strategies like habitat placement and connectivity.

For guidance on habitat selection and connectivity, I recommend consulting with experts in conservation biology, ecology, or environmental planning. These professionals can provide insights based on ecological principles, species-specific needs, and landscape connectivity analysis to ensure effective conservation outcomes.

User: What is the regionalization problem?

RegionDefiner: The regionalization problem involves the process of aggregating smaller geographic areas into larger, contiguous, and homogeneous regions. This process is guided by specific criteria and constraints with the aim of optimizing certain objectives. The primary goal is to ensure that the resulting regions are internally homogeneous in terms of certain attributes (e.g., demographic, economic, environmental characteristics) while being distinct from each other. This is achieved under the constraint of spatial contiguity, meaning that areas within a region must be geographically adjacent to each other.

There are two main types of regionalization problems:

1. **Max-P Region Problem**: This involves clustering areas into the maximum number of regions where a specific attribute or set of attributes exceeds a predefined threshold. The objective is to create as many regions as possible, each meeting or surpassing a certain criterion, while ensuring that the regions are spatially contiguous.

2. **P-Region Problem**: This focuses on grouping spatial units into a predefined number of contiguous regions based on attributes to measure regional homogeneity. The aim is to form regions that are as homogeneous as possible regarding the selected attributes, again under the constraint of spatial contiguity.

Regionalization is applied in various fields, including geography, urban planning, environmental management, and public health, to address issues like service delivery optimization, conservation planning, and regional development. The process involves complex decision-making and computational techniques to analyze spatial data and generate meaningful regional divisions.

User: Is it the same as clustering?

RegionDefiner: Regionalization and clustering are related but distinct concepts, each with its own objectives and methodologies. Here's a breakdown of the differences and similarities:

Clustering:
- **Objective**: Clustering aims to group a set of objects (which can be anything from data points to geographic areas) in such a way that objects in the same group (called a cluster) are more similar to each other than to those in other groups. The primary focus is on the similarity of attributes, regardless of spatial proximity.
- **Spatial Consideration**: Traditional clustering algorithms (like k-means, hierarchical clustering) do not necessarily consider spatial contiguity. That is, the resulting clusters may not be geographically contiguous.
- **Applications**: Clustering is widely used in data analysis to identify patterns or groups within datasets, such as customer segmentation, image recognition, and anomaly detection.

Regionalization:
- **Objective**: Regionalization specifically focuses on aggregating smaller geographic areas into larger, contiguous, and homogeneous regions based on certain criteria. The goal is to optimize for internal homogeneity within regions and external heterogeneity between regions, under the constraint of spatial contiguity.
- **Spatial Consideration**: A key aspect of regionalization is the requirement for spatial contiguity. This means that areas within a region must be geographically adjacent, making regionalization particularly relevant for geographic and spatial data analysis.
- **Applications**: Regionalization is applied in fields such as urban planning, environmental management, and public health, where the spatial arrangement of regions is crucial. Examples include designing service delivery regions, conservation area planning, and regional economic development.

Similarities:
Both clustering and regionalization aim to group entities based on similarity or homogeneity. They are used to simplify complex data sets or landscapes into manageable units for analysis, decision-making, or policy development.

Key Difference:
The main difference lies in the consideration of spatial contiguity. Regionalization requires the resulting regions to be spatially contiguous, which is not a prerequisite for most clustering algorithms. This makes regionalization particularly suited for spatial data analysis where geographic context and continuity are important.

Figure 4. Sample dialogues demonstrating RegionDefiner's ability to determine relevance to regionalization and address general Regionalization-related inquiries.

RegionDefiner is capable of operating on any user-specified data on a local device. The conversation reveals the following strengths of RegionDefiner in managing regionalization tasks:

User: I want to delineate several groups of states to analyze regional development in Mexico.

RegionDefiner: To assist you with delineating several groups of states for analyzing regional development in Mexico, I will need to conduct a specific analysis using the appropriate regionalization tool. Before proceeding, could you please provide the following details:

1. **Number of Regions**: How many groups or regions do you want to delineate?
2. **Attributes for Homogeneity**: What attributes (e.g., economic indicators, population density) should be considered to ensure the regions are homogeneous?
3. **Algorithm Preference**: Do you have a preference for the regionalization algorithm (e.g., Skater, AZP, Regional K-means, Spenc, Ward, or Random Region)?
4. **Visualization**: Would you like a visualization of the regionalization results?

Once you provide this information, I can outline the workflow and apply the most suitable tool for your analysis.

User: I want to create five groups.

RegionDefiner: To proceed with creating five groups of states in Mexico for analyzing regional development and to provide a visualization of the results, I still need a bit more information:

1. **Attributes for Homogeneity**: Could you specify which attributes (e.g., GDP, population density, education level) should be considered to ensure the regions are homogeneous? This is crucial for the analysis as it will guide the grouping based on similarities within these attributes.

2. **Algorithm Preference**: Although you haven't specified a preference for the regionalization algorithm, given the task, algorithms like Skater, AZP, or Regional K-means could be suitable. If you have no preference, I can select the most appropriate one based on the attributes you're interested in.

Please provide the required details so I can proceed with the analysis.

User: Can I set the state name as an attribute describing regional homogeneity? The shapefile has been uploaded in 'mexicojoin.shp' file path.

RegionDefiner: Using the state name as an attribute for describing regional homogeneity in a regionalization analysis is not typical. Regional homogeneity usually refers to quantitative or qualitative attributes that can vary across different areas, such as economic indicators (GDP, unemployment rate), demographic characteristics (population density, education level), or other measurable factors that can be used to assess similarity or dissimilarity between regions.

The state name, being a categorical and unique identifier for each state, does not provide a basis for measuring homogeneity or heterogeneity among regions. Instead, it's used to identify or label the spatial units in the analysis.

For a meaningful regionalization analysis aimed at analyzing regional development in Mexico, it would be more appropriate to use attributes that can reflect the development status or potential of the states. Examples of such attributes could include:

- Economic indicators (e.g., GDP per capita, unemployment rate)
- Social indicators (e.g., average education level, health access)
- Demographic indicators (e.g., population density, urbanization rate)

Could you specify any such attributes available in your shapefile that you would like to use for the analysis? If you're unsure about the attributes contained in your shapefile, I can perform a search within the uploaded file to identify the available attributes for you. Would you like me to do that?

Figure 5. An example of dialogues showcasing RegionDefiner's proficiency in identifying the task and selecting the appropriate function.

- **Domain Comprehension and Issue Identification**: RegionDefiner initially understands the user's request to examine regional development by delineating states into groups. It skillfully inquires about specific questions, such as the criteria for grouping states and the desired quantity of groups, to guarantee a clear understanding of the user's goals.
- **Tool Selection and Parameter Extraction**: RegionDefiner guides users to provide specific details, enabling the extraction of essential parameters. For example, based on the user's input, RegionDefiner chose the SKATER algorithm among various regionalization problems. It also delved into details involving spatial data availability

User: Alright, I'd like to use GDP data from 1980–1990 as the attributes. In addition, Can you explain how the various algorithms would handle this differently?

RegionDefiner: Certainly! Using GDP data from 1980–1990 as attributes for regional homogeneity will allow for an analysis that considers economic performance over a decade, which is a solid basis for understanding regional development patterns in Mexico. Now, let's briefly explain how different regionalization algorithms would handle this task differently:

1. **Skater**: The Skater algorithm groups areas into contiguous, homogeneous regions based on a connectivity graph. It aims to minimize the variance within regions while considering the spatial contiguity of units. For your case, Skater would create regions where the GDP data from 1980–1990 is as similar as possible within each region, ensuring that all states in a region are spatially adjacent.

2. **AZP (Automatic Zoning Procedure)**: AZP also forms regions by maximizing internal homogeneity but does so through an iterative process where initial regions are adjusted by moving units between regions to improve the overall homogeneity. AZP would look at the GDP data across the decade and iteratively adjust the composition of regions to find a configuration where the similarity of GDP performance within regions is maximized.

3. **Regional K-means**: This algorithm is a spatially constrained version of the traditional K-means clustering, which groups areas based on the similarity of their attributes, in this case, GDP data. The spatial constraint ensures that the resulting clusters are contiguous. Unlike Skater and AZP, K-means starts with a random initial partition and iteratively refines it, which can lead to different results on different runs.

4. **Spenc**: The Spatially Encouraged Spectral Clustering (Spenc) algorithm is designed for non-convex clusters and uses graph cuts weighted by spatial proximity. It would consider the GDP data while also heavily weighting the spatial configuration of the states, potentially leading to regions that are more geographically compact.

5. **Ward**: The Ward algorithm merges areas into clusters based on the principle of minimizing the variance within clusters. When applied to your data, it would start with each state as its own region and iteratively merge them into larger regions, ensuring that each merger results in the smallest possible increase in total within-cluster variance of GDP data.

6. **Random Region**: This algorithm randomly groups areas into regions, serving as a comparative baseline rather than a method driven by data homogeneity or spatial contiguity. It's not suitable for your analysis if the goal is to understand genuine regional development patterns based on GDP data.

Given your focus on GDP data from 1980–1990 for analyzing regional development, algorithms like Skater, AZP, or Ward might be particularly suitable as they are designed to optimize for internal homogeneity while respecting spatial contiguity. The choice between them could depend on whether you prioritize the compactness of regions (Ward), the iterative improvement of homogeneity (AZP), or the construction of regions based on a connectivity graph (Skater).

Would you like to proceed with one of these algorithms for your analysis? If so, which one?

User: Cool, let's try Skater for this problem.

RegionDefiner: The regionalization analysis using the Skater algorithm with GDP data from 1980 and 1990 has been successfully completed. We have delineated Mexico into five distinct regions based on economic performance over that decade, ensuring that each region is homogeneous in terms of GDP growth and development.

The results have been saved in the 'mexicojoin_output.shp' file, with the regions indicated in the 'region' column. Each state has been assigned to one of the five regions, allowing for a detailed analysis of regional development patterns across Mexico.

If you need further analysis or have additional questions, feel free to ask!

Figure 6. (continued) An example of dialogues illustrating RegionDefiner's ability to extract parameters and effectively navigate error messages.

and regional homogeneity description, demonstrating its capacity to extract critical parameters from a natural language dialogue.

- **Function Execution**: As shown in Figure 6, RegionDefiner executed the selected regionalization algorithm locally. During the execution, RegionDefiner detected potential data discrepancies about the absence of the specified attribute 'household income 1980-1990', illustrated by our built-in error message. Users are further suggested with other attributes like 'PCGDP1980' and 'PCGDP1990' as proxies for household income. Upon successful execution of the SKATER algorithm, it

Figure 7. Visualization of five regions created by RegionDefiner, presented to the user during the conversation.

presented users with a visual representation of the regionalization results (shown in Figure 7) and saved the data in distinct shapefiles.

- **Reasoning and Educational Summarization**: RegionDefiner transparently communicated its reasoning process, detailing the implementation of the SKATER algorithm and emphasizing its consideration of spatial contiguity and regional homogeneity (Figure 6). Notably, RegionDefiner's insight into the SKATER algorithm is enriched by the pre-uploaded paper Assunção *et al*. (2006). This example emphasizes the significance of the Document Retrieval component in enhancing the conversational agent's subject knowledge, particularly in situations where explicit knowledge is absent from our instruction prompt.

In short, conversation 2 exemplifies an end-to-end conversational instructional experience. RegionDefiner showcases an interactive dialogue-based approach to execute functions and analyze results. This will eliminate the necessity for users to peruse papers and conduct scripts, greatly simplifying the acquisition, learning, and implementation of regionalization techniques and tools.

Evaluation

To provide more insight into the performance of RegionDefiner, we present various dialogue examples and compare our generated responses with those obtained from leading pre-trained LLM, including Microsoft Copilot (accessed on 2024/2/4 via https://copilot.microsoft.com/), as well as ChatGPT-3.5 and ChatGPT-4 (both accessed on 2024/2/4 via https://chatgpt.com/). These models serve as benchmarks for comparison against RegionDefiner with the model version 'gpt-4-turbo-preview'.

Evaluation of relevance recognition

We selected a collection of 10 representative queries encompassing various subjects about spatial optimization to assess the accuracy of identifying regionalization-relevant queries. Five of these queries refer to case studies from existing regionalization articles, while the others were about different issues such as facility location, resource allocation, and route planning. Table 3 compares RegionDefiner with other conversational bots in detecting regional dependencies, and Table 4 details their accuracy, false positive, and false negative rates.

In this comparison, RegionDefiner outperformed Microsoft Copilot, ChatGPT-3.5, and ChatGPT-4, especially in minimizing false positive rates in identifying non-regionalization topics. Table 5 delved deeper into their decision-making process by examining their responses to the last query in Table 3 regarding the selection of green spaces in Phoenix. Microsoft Copilot, ChatGPT-3.5, and ChatGPT-4 misclassified this as a regionalization issue, associating it with the analysis of social-environmental factors and the aggregation of regions. Instead, RegionDefiner correctly recognized that although it may seem similar to regionalization, the query falls under urban planning for site selection.

Evaluation of response precision

We posed a typical question, '*What is the difference between Regionalization and Clustering?*', to RegionDefiner and other conversational bots to assess their level of accuracy in providing answers (Table 6). Considering the similarities between clustering and regionalization, this question can be challenging to answer, especially for people unfamiliar with regionalization. While all conversational bots can distinguish the two notions, RegionDefiner's response was particularly notable for its:

- **Emphasis on spatial contiguity:** RegionDefiner distinctly emphasizes spatial contiguity, a key characteristic of regionalization essential for differentiating it from clusters. RegionDefiner emphasizes the necessity of spatial continuity in regionalization, as opposed to attribute-based clustering methods that lack geographical components.
- **Practical Implications:** RegionDefiner emphasizes the distinctions between regionalization and clustering in terms of objectives, applications, and constraints. The information provided was not only comprehensive but also tailored to practical use.

Through this comparative evaluation, RegionDefiner demonstrated a superior ability to convey complex domain-specific knowledge with clarity and applicability.

Evaluation for multi-round dialogues

The evaluation above focused on single-reply interactions, which might not fully capture the dynamics of multi-round dialogues inherent in conversational agents involving human-like interactions. Given the lack of comprehensive metrics for evaluating task-specific conversational agents, we introduce a qualitative framework with four critical criteria to compare RegionDefiner against other agents more accurately:

Table 3. Regionalization relevancy recognition by RegionDefiner, Microsoft Copilot, ChatGPT-3.5, and ChatGPT-4. 'True' indicates relevance to regionalization, while 'False' denotes irrelevance.

Query	Ground Truth	Region-Definer	Microsoft Copilot	ChatGPT-3.5	ChatGPT-4
1. How do we divide the contiguous United States into four major climate zones, considering variations in temperature and precipitation? (Fovell and Fovell 1993)	True	True	False (incorrect)	True	True
2. How do we identify the most appropriate spatial unit to delineate neighborhoods in urban social science research? (Wei *et al.* 2021)	True	True	True	True	True
3. I want to define ecological areas in the contiguous United States and see how the areas identified by algorithms are compared to ecoregions defined by experts. (Aydin *et al.* 2021)	True	True	True	True	True
4. I would like to explicitly design zonal objects as meaningful entities related to a specific goal. For example, I'd like to decide which kind of aerial objects would be most useful for studying housing characteristics or unemployment rates. (Openshaw and Alvanides 2001)	True	True	True	True	True
5. I intend to aggregate 4000 transportation analysis zones from six counties in Southern California, USA, into around 100 model zones, the maximal number considered viable for an urban economic model. What should I do? (Li *et al.* 2014)	True	True	True	True	True
6. The Tennessee Valley Authority (TVA) operates several coal-fueled power plants in the Tennessee Valley region. How do we determine the largest amount of additional generation capacity that can be added to existing power plant sites?	False	False	False	False	False
7. We want to set aside appropriate amounts and types of habitats to protect species from becoming extinct. Where should we place them? How do we connect these selected habitats?	False	False	True (incorrect)	True (incorrect)	True (incorrect)
8. Drones are dispatched with valuable and time-sensitive medical supplies to respond to emergency cases. To optimize the drone emergency delivery system, which stations should be selected to equip the drones? Which drone should serve which patient?	False	False	True (incorrect)	True (incorrect)	False
9. We drive from different work locations and would like to find an intermediate meetup location to hand over important documents. Where is the ideal meet-up location? How should we route people?	False	False	True (incorrect)	True (incorrect)	False
10. Phoenix is the capital of the state of Arizona. Its subtropical desert climate has particularly hot summers and mild winters. I want to identify suitable areas for the new green space.	False	False	True (incorrect)	True (incorrect)	True (incorrect)

- **Proactivity**: Measures the agent's initiative in maintaining focus on regionalization tasks and its ability to prevent divergences into unrelated topics.
- **Explainability:** Assesses the clarity and relevance of the explanations provided by the agent, specifically regarding regionalization domain knowledge.
- **Efficacy:** Evaluates the agent's success in performing practical regionalization tasks accurately and thoroughly.

Table 4. Comparison of Accuracy, False Positive Rate (incorrect relevance attributed to non-regionalization topics), and False Negative Rate (incorrect irrelevance attributed to Regionalization topics) across RegionDefiner, Microsoft Copilot, ChatGPT-3.5, and ChatGPT-4 over the set of 10 queries.

	RegionDefiner	Microsoft Copilot	ChatGPT-3.5	ChatGPT-4
Accuracy	100%	50%	60%	80%
False Positive Rate	0%	80%	80%	40%
False Negative Rate	0%	20%	20%	0%

Table 5. Logical reasoning process on relevancy recognition of green space area identification issue (the 10th query from Table 3) over RegionDefiner, Microsoft Copilot, ChatGPT-3.5, and ChatGPT-4.

Conversational agent	Logical reasoning
RegionDefiner	This question revolves around identifying specific areas within a city suitable for creating new green spaces based on climatic and possibly other environmental or urban criteria. **While it does involve the analysis of spatial data and consideration of geographic attributes, the focus here is more on site selection for urban planning purposes, rather than on aggregating smaller areas into fewer, larger, and more homogeneous regions.** Regionalization concerns creating larger regions from smaller units based on specific attributes to achieve homogeneity within those regions, which is distinct from selecting sites within a city for specific urban development projects like the creation of green spaces.
Microsoft Copilot	Identifying suitable areas for new green spaces in the City of Phoenix involves dividing the city into smaller regions based on certain criteria. These criteria could include factors such as proximity to residential areas, existing land use, population density, and environmental considerations.
ChatGPT-3.5	This question is related to regionalization because it involves identifying suitable areas for green space within a specific geographical region (the City of Phoenix) based on climate characteristics and other factors, which requires delineating and analyzing different spatial regions within the city.
ChatGPT-4	Identifying suitable areas for new green space in the City of Phoenix, which has a subtropical desert climate, is a Regionalization-related issue. This task involves analyzing various geographic and environmental factors to determine areas within the city that could benefit most from the addition of green spaces, such as parks or urban forests. Factors to consider include current land use, urban heat island effects, population density, accessibility for residents, and the potential for irrigation and sustainable maintenance of vegetation in a desert climate. The process of selecting these areas involves regional planning and environmental management principles, aiming to enhance urban livability, biodiversity, and ecological resilience. This task aligns with regionalization in that it seeks to categorize parts of the city based on their suitability for green space development, taking into account the unique climatic and geographic characteristics of Phoenix.

- **Efficiency:** Considers the time the agent takes to respond, ensuring that it is quick enough to maintain user engagement without sacrificing the quality of the information provided.

We conducted a comparative analysis involving RegionDefiner and similar conversational agents to apply these criteria. The results, presented in Table 7, indicate that:

- **Proactivity**: RegionDefiner is designed with a focused intent to prioritize discussions related to regionalization. This is achieved through a domain-specific dialogue management system that guides conversations back to regionalization topics, even if the user's input deviates. In contrast, general conversational agents like ChatGPT are not dedicated solely to regionalization problems and are programmed to

Table 6. The answer to '*What's the difference between Regionalization and Clustering?*' from RegionDefiner, Microsoft Copilot, ChatGPT-3.5, and ChatGPT-4.

Conversational Agent	Answer
RegionDefiner	The concepts of regionalization and clustering are closely related but distinct in their objectives, applications, and constraints. Here's a brief comparison to delineate the differences: Regionalization 1. Objective: ***Aggregates smaller areas into fewer, larger regions that are contiguous and/or homogeneous, with an emphasis on spatial contiguity***. The goal is to optimize for homogeneity within these regions while maximizing heterogeneity between regions, under specific geospatial constraints. 2. Applications: Used in geographical information systems (GIS), urban planning, environmental management, and political districting. Regionalization is often adopted to create areas that have similar characteristics for more effective management or representation. 3. Constraints: ***Must ensure that the created regions are spatially contiguous, meaning that they share borders or points of connection. This constraint is pivotal in maintaining the geographic coherence of the regions.*** Clustering 1. Objective: Groups data points into clusters where items in the same cluster are more similar to each other than to those in other clusters. The focus is on discovering the inherent groupings in the data based on various attributes, without necessarily considering spatial contiguity. 2. Applications: Widely used across fields like machine learning, data mining, statistics, and biology. Examples include customer segmentation, pattern recognition, image analysis, and gene expression analysis. 3. Constraints: ***Does not require spatial contiguity, meaning elements within a cluster can be disparate or globally distributed as long as they share similar attributes***. The primary focus is on attribute similarity rather than geographic proximity. In essence, while clustering is concerned with the inherent groupings in a dataset based on attribute similarity, regionalization places a critical emphasis on spatial contiguity alongside homogeneity in attributes, making it particularly significant in geographic and spatial analyses.
Microsoft Copilot	Clustering and regionalization are both methods used to group similar observations, but they differ in their application and constraints. Clustering is a fundamental method of geographical analysis that draws insights from large, complex multivariate processes. It works by finding similarities among the many dimensions in a multivariate process, condensing them down into a simpler representation. Clustering involves sorting observations into groups without any prior idea about what the groups are. These groups are delineated so that members of a group should be more similar to one another than they are to members of a different group. Each group is referred to as a cluster. On the other hand, regionalization is a special kind of clustering where the objective is to group observations which are similar in their statistical attributes, but also in their spatial location. In this sense, regionalization embeds the same logic as standard clustering techniques, but also it applies a series of geographical constraints. A region is similar to a cluster, in the sense that all members of a region have been grouped together, and the region should provide a shorthand for the original data within the region. However, regions are more complex than clusters because they combine this similarity in profile with additional information about the location of their members: they should also describe a clear geographic area.
ChatGPT-3.5	Regionalization and clustering are both techniques used in data analysis to group similar entities together. However, there are some differences between the two: 1. Purpose: • Regionalization: Primarily used in geographical contexts, regionalization focuses on dividing geographic areas into regions based on certain criteria such as proximity, administrative boundaries, or socio-economic characteristics. The goal is often to understand spatial patterns or to facilitate decision-making related to resource allocation, planning, or policy-making. • Clustering: Clustering, on the other hand, is a more general technique used in various fields including data mining, machine learning, and statistics. It aims

(*continued*)

Table 6. Continued.

Conversational Agent	Answer
	to partition a dataset into groups or clusters such that data points within the same cluster are more similar to each other compared to those in other clusters. Clustering can be applied to diverse types of data, including spatial data, but it's not limited to geographical contexts. 2. Data Type: ● Regionalization: Typically deals with geographic data, such as maps, spatial datasets, or georeferenced data, where the proximity and spatial relationships between entities are crucial for defining regions. ● Clustering: Can be applied to various types of data, including numerical, categorical, or mixed data. It's not inherently tied to spatial or geographic contexts. 3. Methodology: ● Regionalization: Often involves algorithms or methods specifically designed for geographical analysis, considering factors such as spatial contiguity, administrative boundaries, or spatial autocorrelation. ● Clustering: Utilizes a wide range of algorithms and techniques, including k-means, hierarchical clustering, density-based clustering, and spectral clustering, among others. The choice of clustering method depends on the nature of the data and the desired outcomes. 4. Outcome: ● Regionalization: Typically results in the delineation of distinct geographic regions or zones, each characterized by certain commonalities or attributes. ● Clustering: Produces clusters or groups of data points, where each cluster represents a subset of similar entities within the dataset. The number of clusters and their characteristics can vary depending on the algorithm and parameters used. In summary, while both regionalization and clustering involve grouping similar entities together, regionalization is specific to geographic contexts and focuses on dividing geographical areas into meaningful regions, whereas clustering is a more general technique applicable to various types of data for discovering inherent patterns or structures within the data.
ChatGPT-4	Regionalization and clustering are both methods used to group data, but they are applied in different contexts and for different purposes: Regionalization 1. Context: Regionalization is primarily used in geography, spatial analysis, and related fields. It involves grouping adjacent areas into larger regions based on certain criteria such as physical geography, cultural characteristics, economic activities, or other relevant factors. 2. Purpose: The main goal of regionalization is to simplify spatial complexity by creating larger, manageable units that share common characteristics. It helps in understanding and managing geographical areas more effectively for planning, resource management, policy-making, etc. 3. Characteristics: In regionalization, the continuity of space is a key factor. It ensures that the grouped regions are contiguous, meaning each part of a region is physically connected to another part of the same region. Clustering 1. Context: Clustering is a statistical method used in data analysis across various disciplines, including machine learning, statistics, and computer science. It groups a set of objects in such a way that objects in the same group (or cluster) are more similar to each other than to those in other groups. 2. Purpose: The aim of clustering is to discover underlying patterns in data by organizing similar data points into groups. It is used for exploratory data analysis to find natural groupings, anomalies, and structures in data. 3. Characteristics: Clustering does not necessarily consider the spatial or geographical continuity of the data points. It is more about the similarity in attributes or features of the data. Clusters can be formed based on various criteria like distance, density, or a predefined function that measures similarity.In summary, regionalization is focused on grouping contiguous geographic areas based on shared characteristics, often for spatial analysis and planning, while clustering is a broader statistical technique for grouping data points based on similarity, regardless of their spatial arrangement, and is widely used in various data analysis applications.

Table 7. Comparison of the capabilities of RegionDefiner, Microsoft Copilot, ChatGPT-3.5, and ChatGPT-4.

Conversational Agent	Proactivity	Explainability	Efficacy	Efficiency
RegionDefiner	✓	✓	✓	✓
Microsoft Copilot	✗	–	✗	✓
ChatGPT-3.5	✗	–	✗	✓
ChatGPT-4	✗	–	✗	✓

handle various questions. For instance, if a user asks a non-related question such as 'How are you?' general agents are likely to respond in a friendly manner without steering the conversation back to the domain of regionalization. This lack of focus can lead to extended dialogues on topics irrelevant to the primary task, diminishing the productivity of the interaction in terms of achieving specific regionalization objectives.

- **Explainability:** RegionDefiner excels in providing detailed and accurate explanations within the regionalization domain, as evidenced by conversation example 2 in our data. This capability is due to the integration of pre-uploaded regionalization-relevant documents in OpenAI's 'File Search' tool. These documents provide a rich knowledge base that RegionDefiner can draw upon to deliver precise and contextually relevant explanations. Additionally, the agent utilizes predefined wrapper functions that explain the logic behind the execution of **spopt** functions and the meanings of their variables. These wrappers ensure users receive clear and comprehensible insights into how regionalization algorithms are applied, enhancing their understanding and trust in the agent's recommendations.
- **Efficacy:** RegionDefiner effectively selects and applies appropriate **spopt**-based regionalization tools, performing tasks such as code execution and local file access. These capabilities enable RegionDefiner to address practical regionalization tasks efficiently. In contrast, general conversational bots lack the ability to execute code or access local files, limiting their effectiveness in performing specialized tasks within the regionalization domain.
- **Efficiency:** The response times for RegionDefiner were recorded with a median of 14 seconds, a 75th percentile of 17 seconds, and a 90th percentile of 19 seconds. These times are competitive with those of other agents and are deemed satisfactory given the tasks' complexity.

Discussion

Due to the inherent randomness in language models (LLMs), the responses generated by RegionDefiner may not be the same as those presented in the paper, which is a common characteristic of all generative models and all AI agents based on them. However, the underlying logic and accuracy in addressing regionalization tasks remain consistent. To ensure transparency and reproducibility, we have documented the prompts used in our examples and made them available for independent testing. The responses generated by RegionDefiner should be very similar in content, such as whether it is a regionalization problem and which regionalization algorithm should be used to handle a particular problem, with only slight differences in the details of the wording. Furthermore, according to updates from the OpenAI platform, a ′random_seed′ parameter will be introduced shortly, and this feature will enable the generation of identical responses. Once this feature is available, RegionDefiner's outputs can be reproduced as described in the paper.

Excessive computation time is a common challenge in developing conversational bots for complex tasks in many fields, especially given the nondeterministic polynomial-time (NP) hard nature of the regionalization problem addressed in this paper. It is important to note that the primary purpose of RegionDefiner is to make regionalization methods and tasks accessible to a broader audience. Even if a specific task requires a lot of computational time, RegionDefiner may illustrate the workflow using a smaller, manageable piece of data to help users understand the process. Another issue requiring attention is the expected increase in response time and costs associated with the expanded use of tools and documents when operating GPT-4. To address these concerns, future research may want to consider the adoption of open-source LLMs as alternatives to commercial ones. Furthermore, it is promising to restructure RegionDefiner's features, particularly those related to understanding user intent and selecting appropriate tools, by employing smaller, more efficient natural language processing transformers. This strategy may efficiently manage costs while maintaining or improving the system's effectiveness.

Although RegionDefiner greatly simplifies regionalization tasks, users may still need a fundamental understanding of GIS concepts and spatial data preparation. Thankfully, RegionDefiner can assist users in accomplishing this by offering interactive guidance and explanations. For instance, it provides users with detailed guidance on how to prepare and submit shapefiles by offering step-by-step instructions. It also explains GIS concepts by providing definitions and context for GIS terminology encountered during the analysis. Furthermore, it supports problem framing and interpretation, helping users understand regionalization results and their implications. By catering to users with different levels of competence, RegionDefiner serves as a solution to the challenge of complicated GIS tasks, making advanced regionalization techniques more approachable.

The proposed conversational agent incorporates all regionalization-related techniques from the PySAL **spopt** (spatial optimization) package. Subsequent work might consider the inclusion of other regionalization techniques from various open-source software packages, adhering to the process described in this paper. In addition to regionalization, **spopt** deals with two more categories of optimization problems: facility location and route planning. We refrained from integrating these approaches into the same conversational agent due to their inherent dissimilarity and lack of correlation. As future work, separate conversational bots for different optimization problems are worth developing, further enhancing their utility and applicability in various spatial optimization tasks.

The subscription requirement of GPT-4 has raised concerns about the feasibility of using it to develop conversational agents to improve the accessibility of open-source Python packages. However, it is undeniable that this integration allows users to interact with the virtual agent RegionDefiner, making complex regionalization tasks more approachable and manageable. There are two notable options for addressing these accessibility concerns. First, exploring the feasibility of integrating open-source large language models could allow users to deploy the RegionDefiner agent locally without subscription costs, thereby increasing accessibility for a wider audience. A second option is to register for the 'OpenAI Researcher Access Program' (https://openai.com/form/researcher-access-program/), which could reduce or eliminate the cost barrier for academic and research purposes.

Conclusion

RegionDefiner is a customized conversational agent to assist with regionalization tasks and deliver expert-level instructional information through natural language interactions. Developed based on the OpenAI Assistants API and powered by the GPT-4 model, RegionDefiner identifies specific regionalization inquiries, retrieves up-to-date relevant documents, and executes open-source algorithms through interactive, multi-step conversations. Through detailed conversational experiments, including domain-specific Q&A sessions and comprehensive walkthroughs of addressing regionalization problems, this paper demonstrates RegionDefiner's proficiency in practical problem resolution and deep domain understanding, showcasing its utility in facilitating accessible and user-centric regionalization analyses. RegionDefiner surpasses other platforms, including Microsoft Copilot and ChatGPT versions, in delivering accurate regionalization information and immediately performing tasks with user data. The framework of RegionDefiner is adaptable to many problems in geographical research when equipped with the necessary domain knowledge and tools.

Authors contributions

Dr. Feng: Writing – review & editing, Writing – original draft, Visualization, Validation, Project administration, Methodology, Resources, Formal analysis, Data curation, Conceptualization. **Dr. Cao**: Writing – review & editing, Validation, Methodology, Formal analysis, Conceptualization.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Dr. Xin (Selena) Feng is an Assistant Professor in the Department of Geography and Environmental Sustainability at the University of Oklahoma. She holds a PhD in GIScience from the University of California, Santa Barbara, a B.S. in Cartography and GISystem from Wuhan University, China, an M.S. in Remote Sensing and GISystem from Peking University, China, and an M.A. in Geographical Science and Urban Planning from Arizona State University. Her research focuses on geospatial data science, spatial optimization, location modeling, resource planning and development, transportation, and methods to support interactive planning and decision-making. Twitter: @Feng_GIST.

Dr. Yuanpei Cao is a Staff Machine Learning Engineer at Airbnb, Inc. He earned a PhD in Applied Mathematics and Computational Science from the University of Pennsylvania, a B.S. in

Mathematics from Fudan University, China, and an M.A. in Statistics from the Wharton School of the University of Pennsylvania.

Data and codes availability statement

We use Pysal example datasets. You can find our data from libpysal/examples/mexico at https://pysal.org/notebooks/lib/libpysal/Example_Datasets.html. However, RegionDefiner is capable of operating on any user-specified data on a local device if the guidance on preparing the datasets is followed. The corresponding code of this work is available at the link: https://doi.org/10.5281/zenodo.13901021.

References

Airbnb, 2020. Using chatbots to provide faster COVID-19 community support.

Airbnb, 2021. Task-oriented conversational AI in Airbnb customer support.

Assunção, R.M., *et al.*, 2006. Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees. *International Journal of Geographical Information Science*, 20 (7), 797–811.

Aydin, O., *et al.*, 2021. A quantitative comparison of regionalization methods. *International Journal of Geographical Information Science*, 35 (11), 2287–2315.

Badr, H.S., Zaitchik, B.F., and Dezfuli, A.K., 2015. A tool for hierarchical climate regionalization. *Earth Science Informatics*, 8 (4), 949–958.

Brown, T., *et al.*, 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.

Chen, M., *et al.*, 2021. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374.

Chowdhery, A., *et al.*, 2023. Palm: scaling language modeling with pathways. *Journal of Machine Learning Research*, 24 (240), 1–113.

Dai, H., *et al.*, 2023. AD-AutoGPT: an autonomous GPT for Alzheimer's disease infodemiology. arXiv preprint arXiv:2306.10095.

Daru, B.H., Karunarathne, P., and Schliep, K., 2020. phyloregion: R package for biogeographical regionalization and macroecology. *Methods in Ecology and Evolution*, 11 (11), 1483–1491.

Dong, Q., *et al.*, 2022. A survey for in-context learning. arXiv preprint arXiv:2301.00234.

Duque, J.C., Anselin, L., and Rey, S.J., 2012. The max-p-regions problem. *Journal of Regional Science*, 52 (3), 397–419.

Duque, J.C., Church, R.L., and Middleton, R.S., 2011. The p-Regions Problem. *Geographical Analysis*, 43 (1), 104–126.

Duque, J.C., Ramos, R., and Suriñach, J., 2007. Supervised regionalization methods: a survey. *International Regional Science Review*, 30 (3), 195–220.

Feng, X., Rey, S., and Wei, R., 2022a. The max-p-compact-regions problem. *Transactions in GIS*, 26 (2), 717–734.

Feng, X., *et al.*, 2022b. spopt: a python package for solving spatial optimization problems in PySAL. *Journal of Open Source Software*, 7 (74), 3330.

Feng, Z., *et al.*, 2020. Codebert: a pre-trained model for programming and natural languages. arXiv preprint arXiv:2002.08155.

Fovell, R.G., and Fovell, M.Y.C., 1993. Climate zones of the conterminous United States defined using cluster analysis. *Journal of Climate*, 6 (11), 2103–2135.

Gel, B., *et al.*, 2016. regioneR: an R/Bioconductor package for the association analysis of genomic regions based on permutation tests. *Bioinformatics (Oxford, England)*, 32 (2), 289–291.

Kostbade, J.T., 1968. The regional concept and geographic education. *Journal of Geography*, 67 (1), 6–12.

Lemus-Canovas, M., *et al.*, 2019. synoptReg: an r package for computing a synoptic climate classification and a spatial regionalization of environmental data. *Environmental Modelling & Software*, 118, 114–119.

Li, W., Church, R.L., and Goodchild, M.F., 2014. The p-compact-regions problem. *Geographical Analysis*, 46 (3), 250–273.

Macmillan, W., and Pierce, T., 1994. Optimization modelling in GIS framework: the problem of political redistricting. *In:* S. Fotheringham and P. Rogerson, eds. *Spatial analysis and GIS*. London: Taylor & Francis, 221–246.

Martins-Bedê, F.T., *et al.*, 2009. Risk mapping of schistosomiasis in Minas Gerais, Brazil, using MODIS and socioeconomic spatial data. *IEEE Transactions on Geoscience and Remote Sensing*, 47 (11), 3899–3908.

Microsoft, 2023. Bing AI: the search engine that knows what you need. https://www.bing.com/new/.

Montello, D.R., 2003. Regions in geography: process and content. *In*: M. Duckham, M.F. Goodchild, and M. Worboys, eds. *Foundations of geographic information science*. Boca Raton, USA: CRC Press, 173–189.

Nakano, R., *et al.*, 2021. Webgpt: browser-assisted question-answering with human feedback. arXiv preprint arXiv:2112.09332.

OpenAI, 2022. Optimizing language models for dialog. https://openai.com/blog/chatgpt/.

OpenAI, 2023. Gpt-4 technical report. arXiv, 2023.

Openshaw, S., 1996. Developing GIS-relevant zone-based spatial analysis methods. *In:* P. Longley and M. Batty, eds. *Spatial analysis*: *modelling in a GIS environment*. Cambridge: GeoInformation International, 55–73.

Openshaw, S., and Alvanides, S., 2001. Designing zoning systems for the representation of socio-economic data. *In*: A. Frank, J. Raper, and J. Cheylan, eds. *Time and motion of socio-economic units*. London: Taylor & Francis, 288–307.

Qian, C., *et al.*, 2023. Creator: tool creation for disentangling abstract and concrete reasoning of large language models. *In: Findings of the Association for Computational Linguistics: EMNLP 2023*, December, Singapore, 6922–6939.

Qin, Y., *et al.*, 2023. Tool learning with foundation models. arXiv preprint arXiv:2304.08354.

Radford, A., *et al.*, 2018. Improving language understanding by generative pre-training. OpenAI blog.

Radford, A., *et al.*, 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1 (8), 9.

Rahman, A., 2019. sdpt3r: semidefinite quadratic linear programming in R. *The R Journal*, 10 (2), 371.

Richardson, B.C., 1992. Places and regions. *In:* R.E. Abler, M.G. Marcus, and J.M. Olson, eds. Geography's inner worlds: pervasive themes in contemporary American geography. New Brunswick: Rutgers University Press, 27–49.

Rey, S.J., and Sastré-Gutiérrez, M.L., 2010. Interregional inequality dynamics in Mexico. *Spatial Economic Analysis*, 5 (3), 277–298.

Richards, T.B., 2023. Auto-GPT: an autonomous GPT-4 experiment. Available from: https://github.com/Torantulino/ Auto-GPT [Accessed 13 April 2023].

Shen, Y., *et al.*, 2023. Hugginggpt: solving AI tasks with chatgpt and its friends in huggingface. arXiv preprint arXiv:2303.17580.

Touvron, H., *et al.*, 2023. Llama: open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

Viglione, A., 2009. nsRFA: non-supervised regional frequency analysis. http://www.CRAN. R-project.org/package=nsRFA.

Wei, R., Rey, S., and Knaap, E., 2021. Efficient regionalization for spatially explicit neighborhood delineation. *International Journal of Geographical Information Science*, 35 (1), 135–151.

Wei, R., *et al.*, 2022. Reducing racial segregation of public school districts. *Socio-Economic Planning Sciences*, 84, 101415.

Zeng, Y., *et al.*, 2024. Automated interactive domain-specific conversational agents that understand human dialogs. In *International Symposium on Practical Aspects of Declarative Languages*, January. Cham: Springer Nature Switzerland, 204–222.

Zhao, W.X., *et al.*, 2023. A survey of large language models. arXiv preprint arXiv:2303.18223.