

Assemble geo-analytical questions through a Blockly-based natural language interface

Haiqi Xu [orcid](#)¹, Enkhbold Nyamsuren [orcid](#)¹, and Simon Scheider [orcid](#)¹

¹Affiliation: Department of Human Geography and Spatial Planning, Utrecht University, Utrecht, The Netherlands

Correspondence: Haiqi Xu (h.xu1@uu.nl)

Abstract. Natural language Interfaces (NLIs) have the ability to make Geographic Information Systems more accessible for interdisciplinary researchers or any inexperienced users. However, the majority of research on NLIs for GIS explored NLIs for visualization or spatial data retrieval. Research on NLIs for geo-analytical questions is still lacking. Google Blockly, an open-source JavaScript library, is frequently used for developing visual programming editors for young students learning programming languages. Students set up program functions by selecting and assembling the programming blocks and the Blockly system automatically translates the block stacks into different programming languages. Similarly, we present a Blockly-based interface that generates a question depending on the blocks the user has assembled. It can be seen that a Blockly-based interface not only naturally represents syntactic structures in geo-analytical questions but also well assists users in familiarizing the blocks and generating clear and complete questions. A comprehensive usability study is still necessary to better evaluate the interface's performance.

Keywords. natural language interface, geo-analytical questions, google blockly, geographic question answering

1 Introduction

Natural Language Interfaces (NLIs) allow end-users to communicate with complex systems and access enormous databases or knowledge structures easily via written or spoken human language (Schofield and Zheng, 2003; Shah et al., 2013; Lin et al., 2018; Liang et al., 2021). In this way, there is no need for non-expert users to understand database structures, learn query languages (e.g., SPARQL, SQL) or domain knowledge. NLI applications such as Google search, Siri and Alexa, are widely used in daily life, which facilitate people in searching information and operating devices. In academia, NLIs to databases (NLIDs) has been extensively studied for the last decades

(Majhadi and Machkour, 2021; Nihalani et al., 2011), and NLIs to knowledge graphs start become popular with the advent of Semantic Web (Kaufmann and Bernstein, 2007). Besides, many researchers also focus on domain-tailored NLIs that can handle specialized terms and perform analytics (Yates et al., 2003; Setlur et al., 2016; Yuan et al., 2019).

Developing such transparent and intuitive interfaces is one of the long-term goals for Geographic Information System (GIS) (Wang et al., 2008). Previous research on NLI for GIS has been centered either on retrieving spatial databases (spatial DBs) or visualization. Most of visualization NLIs support interactive dialogues such as “show me a map of Florida” and “add major cities in Florida” (Wang et al., 2008). For spatial databases NLIs, users can pose questions such as “How many people live in Mississippi?” and “Is New York or London bigger?” (Li et al., 2019). Yet, no NLI seems to be designed for *geo-analytical questions* such as “Where are the clusters of elderly people in Amsterdam?” or “From where are the wind farms visible in Amsterdam?”. These questions always require professional GIS analysis and input resources to generate the right output. For example, GIS experts will need a digital surface model and input representing “wind farms” and “Amsterdam”, as well as a tool implementing visibility analysis to determining visible area. Geographic analysis is one of the core applications of GIS, however, to the best of our knowledge, little attention has been devoted to geo-analytical question NLIs. Even the research on NLIs for visualization and spatial DBs, tends to focus mainly on semantic extraction, interpretation and data retrieval or response control, with little discussion of the requirements for the user interface itself. This paper will only discuss the natural language interface for forming geo-analytical questions. Question interpretation and resources matching will not be included here.

Designing NLIs for geo-analytical questions faces the following challenges. First of all, geo-analytical questions are more complex than spatial databases questions and visualization dialogues in terms of syntactic structures and anal-

ysis intention (Xu et al., 2020). Visualization dialogues explicitly stated actions such as “show” and “add” and simple geographic phenomena such as “city” and “Florida”. Spatial databases questions follow homogeneous structures and query types that are highly dependent on spatial databases (Xu et al., 2020). However, geo-analytical questions are very diverse in terms of analysis types (e.g., distribution analysis, visibility analysis, network analysis, and raster analysis), as well as explicit and potential geo-analytical concepts (e.g, “cluster”, “elderly people”, “wind farms”, “visible surface”). Therefore, NLIs for geo-analytical questions should support natural expressions of various GIS analysis types and concepts.

Secondly, inexperienced users may not be able to generate a complete and clear geo-analytical question, thus impairing information retrieval (Wang, 2000). For example, “Amsterdam” in the previous example is important for determining the spatial extent of “elderly people”. However, user sometimes may forget to specify a spatial extent for a geo-analytical question. Moreover, there might be ambiguous expressions about size, distance, etc. in questions (Wang, 2000). For example, “What is the number of flower stores within the service area of warehouses in Paris?”. Service area¹ can be generated by determining a travel cost in network analysis, while the travel cost is missing in the question. Therefore, NLIs for geo-analytical questions should help user avoid incomplete and ambiguous expressions, and ideally guide user in formulating eligible questions.

Thirdly, non-expert users do not know the scope of questions that can be answered by a question answering system, because they are not familiar with spatial databases or GIS analysis components in back-end. In this regard, it would be convenient if NLIs for geo-analytical questions have clarified the question types that can be supported.

To address these challenges, we propose an novel idea that using the *Google Blockly library*² to developing a natural language interface for formulating geo-analytical questions. To the best of our knowledge, this paper is the first to explore the user interface for geo-analytical questions and discuss the possibilities and benefits of using the Google Blockly in natural language interface. This paper is organized as followed: Google Blockly and a Blockly-based NLI are explained with a demo and examples in Section 2, followed by the discussions of how Blockly can help address the challenges of geo-analytical question NLIs in Section 3. Conclusion and future work is summarized in Section 4.

2 A Blockly-based natural language interface

In this section, we first introduce the Blockly library and its common applications and then explain how to use it in designing a natural language interface for geo-analytical questions.

2.1 The Blockly library and its applications

Blockly is an open source library that allow users to customize interlocking blocks of coding concepts and automatically generate programming language of your choice. It is designed to be flexible, extensible and easily to be added to your websites or mobile applications (Pasternak et al., 2017). Blockly has considerably contributed to the educational use of visual programming for young students (Seraj et al., 2019). As shown in Figure 1, a Blockly application usually consists of three parts. Namely, a toolbox that provide all the available blocks; a workspace that used for combining blocks together; and a code interpreter output the codes corresponding to the combined blocks in the programming language you chosen. A young students can drag programming blocks from the toolbox and drop them to the workspace. Some predefined rules such as type checks³ and connection directions⁴ will make sure the blocks are connected logically. For example, the purple block “set Count to” is from the “Variable” category, and can be connected with the blue “number” block from the “Math” category. Together, they represent declaring a variable “Count” and set its value as “1”. You can also connect “set Count to” with the “Hello world” block, which means the variable “Count” now have a string value. Besides, it also supports if-else and while-do statement, which can be found in “Logic” and “Loops” category.

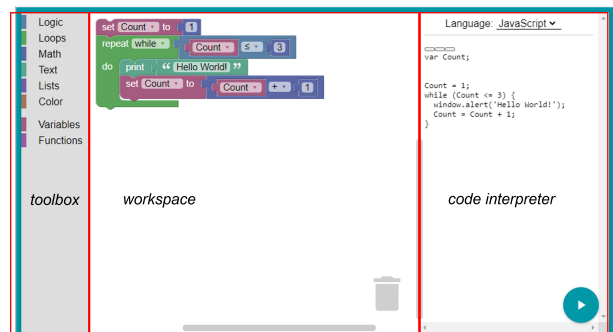


Figure 1. a Blockly-based programming editor that generates JavaScript code by connecting blocks in the workspace⁵.

It can be seen that comparing with learning data structures and grammar of programming language, it is much easier and fun for elementary students to use such visual

¹<https://pro.arcgis.com/en/pro-app/2.7/help/analysis/networks/service-area-analysis-layer.htm>

²<https://developers.google.com/blockly>

³<https://developers.google.com/blockly/guides/create-custom-blocks/type-checks?hl=en>

⁴Blocks can be nested within each other, or connected from top to bottom or left to right, depending on their shape and types.

⁵<https://developers.google.com/blockly>

programming editors and create scripts with no syntactic errors. Therefore, as long as we can generate blocks representing syntactic structures from geo-analytical questions, it is possible for non-expert users to combine a question with good quality via a Blockly-based interface.

2.2 Design a Blockly-based interface for geo-analytical questions

Instead of having coding concepts in toolbox, we investigated a geo-analytical question corpus GeoAnQu (Xu et al., 2020) and customized four types of blocks as basic components in forming a geo-analytical questions. The blocks can be defined using the provided JSON formats and JavaScript functions⁶. As shown in Figure 2, the green blocks representing *question words* (e.g., “where”, “what”, “which”) with its auxiliary words (e.g., “is”, “are”, “do not”). The blue blocks are designed for *noun phrases* that can represent geographic concepts with three syntactic structures, such as, “wind farms”, “cluster of fire calls” and “the shortest distance from car accidents to the nearest hospital”. Purple blocks represent *relationships* and *qualities* that restrict geographic concepts in geo-analytical questions. We provide topological relationships such as “intersect with” and “within 200 meters of”, comparison relationships such as “larger than”, as well as general qualities such as “visible”, “for sale”. Yellow blocks are designed for *spatial extent* that accepts any place names in the text box, and an optional *temporal extent* that accepts month and year. The workspace in Figure 2 shows the example question can be generated by combining the four types of blocks, and the natural language question is created by via Blockly APIs and displayed in the *question generator* simultaneously.



Figure 2. Generating the question “From where are the wind farms visible in Amsterdam?” by assembling the customized blocks.

3 Benefits of the Blockly-based natural language interface

We can argue that the Blockly-based NLI is very helpful in addressing the aforementioned challenges. First of all,

⁶<https://developers.google.com/blockly/guides/create-custom-blocks/define-blocks?hl=en>

the Blockly-based NLI is capable of designing blocks with different syntactic structures and geo-analytical concepts. Blockly allows users define fixed and flexible text for each block. For example, a question work block “From where are” consist of fixed text “From where” and a drop-down box listing all the available options of auxiliary verbs. Users can choose “is” or “was” based on the context. Another example is shown in Figure 3, the noun phrase block is designed for “concept + preposition + concept” structure. User can enter “cluster” or “Euclidean distance” or other concepts in the input boxes and choose a suitable adjective (e.g., “average”) and preposition (e.g., “of”, “to”) from the drop-down list.

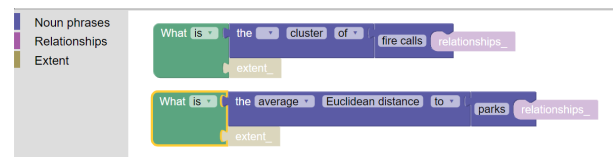


Figure 3. Users can change geo-analytical concepts in the noun phrase block for different questions.

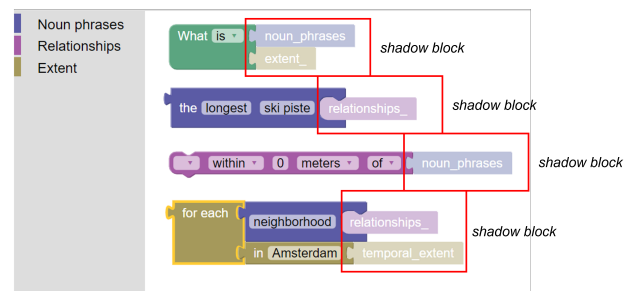


Figure 4. The shadows blocks can assist inexperienced users in correctly combining blocks.

Secondly, the Blockly-based NLI can assist users in creating a clear and complete question. To ensure this, Blockly provides *shadow block* that can be used as hints of the subsequent blocks. For example, the question word block “What is” in Figure 4 is connected with two shadow blocks by default. The first one suggests a position of a noun phrase block and second one suggests a position for extent. Users are not allowed to replace the “noun-phrases” shadow block by an “extent” block because the predefined types of extent and noun phrases blocks are not matched. We also gave a default concept or value for input boxes so that users may get impressions of the accepted input. In addition, we made categories or blocks invisible and disable when they are not suitable for the current block combination in the workspace. In Figure 5, “Question word” category and “Noun phrase” category are removed because they have been already added in the workspace and are no longer relevant. All the support blocks in the “Extent” category are also disable considering one of them is already used and adding another one is pointless. If users delete the support block from the workspace, all the support blocks will be enabled automatically. With these reminders and



Figure 5. Remove categories and disable blocks when they are inapplicable.

constraints, users are well informed of right blocks and avoid wrong connections to a large extent.

Thirdly, considering all the available blocks are displayed in the toolbox, it should be clear that what syntactic strictures and question types can be interpreted by the back-end server. This will help prevent inexperienced users from being disappointed by getting no result from a correct question. Except for addressing the challenges of geo-analytical question NLI, a blockly-based interface could be used for template-based question answering systems. Although deep learning methods are widely used in question interpretation, query translation and information matching in question answering related research (Ishwari et al., 2019), template-based methods are still very important since they generate more transparent results (Hamzei et al., 2022; Zhekova and Totkov, 2021; Athreya et al., 2021; Punjani et al., 2018). Consider natural language questions are highly flexible and can hardly be interpreted by template-based models without reformulating them, a free-form NLI such as the Google search interface is not applicable. A Blockly-based interface can directly take advantage of the defined templates and thus allow users generate questions with interpretable structures.

4 Conclusion and future work

In this paper, we first discussed the challenges for developing a robust natural language interface for geo-analytical questions. Then, we introduced a novel idea of developing a NLI with Google Blockly and clarified the usages and benefits of such a interface. Blockly seems provide a promising solution for addressing the difficulties, however, the effectiveness and usability of the interface still need to be evaluated via scientific and systematic methods. In future, the geo-analytical blocks and their connection

rules will continue to be tested, modified and extended, until they fulfill the formulation of the training questions from the GeoAnQu corpus. Next, an independent test corpus will be collected and used for the interface evaluation. A usability study including standard questionnaires will be designed and conducted to draw a solid conclusion for the performance of the geo-analytical question NLI.

References

- Athreya, R. G., Bansal, S. K., Ngomo, A.-C. N., and Usbeck, R.: Template-based question answering using recursive neural networks, in: 2021 IEEE 15th International Conference on Semantic Computing (ICSC), pp. 195–198, IEEE, 2021.
- Hamzei, E., Winter, S., and Tomko, M.: Templates of generic geographic information for answering where-questions, *International Journal of Geographical Information Science*, 36, 188–214, 2022.
- Ishwari, K., Aneez, A., Sudheesan, S., Karunaratne, H., Nugalayadde, A., and Mallawarachchi, Y.: Advances in natural language question answering: A review, arXiv preprint arXiv:1904.05276, 2019.
- Kaufmann, E. and Bernstein, A.: How useful are natural language interfaces to the semantic web for casual end-users?, in: *The Semantic Web*, pp. 281–294, Springer, 2007.
- Li, J., Wang, W., Ku, W.-S., Tian, Y., and Wang, H.: Spatialnli: A spatial domain natural language interface to databases using spatial comprehension, in: *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 339–348, 2019.
- Liang, S., Stockinger, K., de Farias, T. M., Anisimova, M., and Gil, M.: Querying knowledge graphs in natural language, *Journal of big data*, 8, 1–23, 2021.
- Lin, X. V., Wang, C., Zettlemoyer, L., and Ernst, M. D.: Nl2bash: A corpus and semantic parser for natural language interface to

- the linux operating system, arXiv preprint arXiv:1802.08979, 2018.
- Majhadi, K. and Machkour, M.: The history and recent advances of Natural Language Interfaces for Databases Querying, in: E3S Web of Conferences, vol. 229, p. 01039, EDP Sciences, 2021.
- Nihalani, N., Silakari, S., and Motwani, M.: Natural language interface for database: a brief review, *International Journal of Computer Science Issues (IJCSI)*, 8, 600, 2011.
- Pasternak, E., Fenichel, R., and Marshall, A. N.: Tips for creating a block language with blockly, in: 2017 IEEE blocks and beyond workshop (B&B), pp. 21–24, IEEE, 2017.
- Punjani, D., Singh, K., Both, A., Koubarakis, M., Angelidis, I., Bereta, K., Beris, T., Bilidas, D., Ioannidis, T., Karalis, N., et al.: Template-based question answering over linked geospatial data, in: *Proceedings of the 12th Workshop on Geographic Information Retrieval*, pp. 1–10, 2018.
- Schofield, E. and Zheng, Z.: A speech interface for open-domain question-answering, in: *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pp. 177–180, 2003.
- Seraj, M., Katterfeldt, E.-S., Bub, K., Autexier, S., and Drechsler, R.: Scratch and Google Blockly: How Girls' Programming Skills and Attitudes are Influenced, in: *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, pp. 1–10, 2019.
- Setlur, V., Battersby, S. E., Tory, M., Gossweiler, R., and Chang, A. X.: Eviza: A Natural Language Interface for Visual Analysis, in: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology, UIST '16*, p. 365–377, Association for Computing Machinery, New York, NY, USA, <https://doi.org/10.1145/2984511.2984588>, 2016.
- Shah, A., Pareek, J., Patel, H., and Panchal, N.: NLKBIDB-Natural language and keyword based interface to database, in: *2013 International conference on advances in computing, communications and informatics (ICACCI)*, pp. 1569–1576, IEEE, 2013.
- Wang, F.: A fuzzy grammar and possibility theory-based natural language user interface for spatial queries, *Fuzzy sets and systems*, 113, 147–159, 2000.
- Wang, H., Cai, G., and MacEachren, A. M.: GeoDialogue: a software agent enabling collaborative dialogues between a user and a conversational GIS, in: *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, vol. 2, pp. 357–360, IEEE, 2008.
- Xu, H., Hamzei, E., Nyamsuren, E., Krueger, H., Winter, S., Tomko, M., and Scheider, S.: Extracting interrogative intents and concepts from geo-analytic questions, *AGILE: GI-Science Series*, 1, 23, <https://doi.org/10.5194/agile-giss-1-23-2020>, 2020.
- Yates, A., Etzioni, O., and Weld, D.: A Reliable Natural Language Interface to Household Appliances, in: *Proceedings of the 8th International Conference on Intelligent User Interfaces, IUI '03*, p. 189–196, Association for Computing Machinery, New York, NY, USA, <https://doi.org/10.1145/604045.604075>, 2003.
- Yuan, C., Ryan, P. B., Ta, C., Guo, Y., Li, Z., Hardin, J., Makadia, R., Jin, P., Shang, N., Kang, T., and Weng, C.: Criteria2Query: a natural language interface to clinical databases for cohort definition, *Journal of the American Medical Informatics Association*, 26, 294–305, <https://doi.org/10.1093/jamia/ocy178>, 2019.
- Zhekova, M. and Totkov, G.: QUESTION PATTERNS FOR NATURAL LANGUAGE TRANSLATION IN SQL QUERIES, *International Journal on Information Technologies & Security*, 13, 2021.