JOURNAL OF SPATIAL INFORMATION SCIENCE
Number N (YYYY), pp. xx–yy

doi:10.5311/JOSIS.YYYY.II.NNN

RESEARCH ARTICLE

# Neural Factoid Geospatial Question Answering

Haonan Li[1], Ehsan Hamzei[2], Ivan Majic[2], Hua Hua[3], Jochen Renz[3], Martin Tomko[2], Maria Vasardani[4], Stephan Winter[2], Timothy Baldwin[1]

[1]School of Computing and Information Systems, The University of Melbourne, Australia

[2]Department of Infrastructure Engineering, The University of Melbourne, Australia

[3]Research School of Computer Science, Australian National University, Australia

[4]Department of Geospatial Science, RMIT University, Australia

*Received: AAA; returned: BBB; revised: CCC; accepted: DDD.*

**Abstract:** Existing question answering systems struggle to answer factoid questions when geospatial information is involved. This is because most systems cannot accurately detect the geospatial semantic elements from the natural language questions, or capture the semantic relationships between those elements. In this paper, we propose a geospatial semantic encoding schema and a semantic graph representation which captures the semantic relations and dependencies in geospatial questions. We demonstrate that our proposed graph representation approach aids in the translation from natural language to a formal, executable expression in a query language. To decrease the need for people to provide explanatory information as part of their question and make the translation fully automatic, we treat the semantic encoding of the question as a sequential tagging task, and the graph generation of the query as a semantic dependency parsing task. We apply neural network approaches to automatically encode the geospatial questions into spatial semantic graph representations. Compared with current template-based approaches, our method generalises to a broader range of questions, including those with complex syntax and semantics. Our proposed approach achieves better results on GeoData201 than existing methods.

**Keywords:** question answering, natural language processing, machine learning, geospatial, semantic graph

# 1 Introduction

Geospatial data accounts for a significant fraction of available web data, with broad applications in areas including geospatial search and question answering (QA). This fraction is growing by 20 percent or more each year [40]. Geospatial data can be either structured (e.g., entities and relations in a database, or discrete images of visual fields) [12], or less formal, such as in spatially-grounded natural language. With the rapid development of knowledge bases and corresponding query languages, the retrieval of geospatial information from structured and semi-structured resources is becoming more viable.

However, most people do not have expertise in geographic information retrieval and knowledge bases, meaning that the most naturalistic interface is natural language, in the form of natural language questions. In order to answer these questions, it is often necessary to translate the natural language question into a structured geospatial knowledge representations. For example, if a user were to ask *Which countries does the River Danube flow through?*, a conventional search engine approach would return a list of webpages that may contain the direct answer (in natural language), or more likely partial textual information that the user would have to manually process to derive the answer, taking time and effort. Automatic QA systems aim to provide a way for people to ask questions in natural language and get natural language answers, based either on answer extraction from documents or querying over a knowledge base. However, existing QA systems struggle to answer factoid questions when geospatial information is involved because of the difficulty in accurately detecting geospatial semantic elements and relationships [22, 42, 43, 48]. Systems that can automatically answer geospatial questions are called geographic question answering ("GQA") systems [55].

GQA has been the target of research in geography, geographic information science, geographic information retrieval (GIR), databases, and the semantic web [55]. Based on the type of data that is used to source the answer, QA approaches can be categorized as either information retrieval-based ("IR-based QA" [57]) or knowledge-based ("KB-QA" [66]). Generally, IR-based QA queries the web or other document collection for relevant documents, from which it extracts answers. However, geospatial questions often require interpretation of the topological, directional, and distance relationships between places, where there may not be a direct match with text in a document. As such, IR-based methods are limited in their ability to provide answers to geospatial questions. For KB-QA systems, questions are answered using structured knowledge bases (KBs) like Freebase [7] or DBpedia [1]. Here, the challenge is in translating natural language questions into structured queries [14]. With the increasing availability and richness of geospatial KBs such as Geonames[1] and OpenStreetMap (OSM),[2] more and more geospatial features can be easily queried from these KBs.

Today, most existing geospatial QA systems [24, 55, 68] are based on geospatial resources and open-domain knowledge bases. These are generally encoded as RDF in the form of (subject, predicate, object) triples, and can be queried using query languages such as SPARQL/GeoSPARQL. Although geographic resources are becoming more and more comprehensive, methods for translating natural language questions into formal queries to search such resources are still largely rule-based and brittle. With recent progress in deep learning and natural language processing (NLP), impressive results have been achieved

[1] https://www.geonames.org/

[2] https://www.openstreetmap.org/

for open domain question answering, with the benefit that neural approaches tend to be more robust and better at generalization. In this paper, we bridge the gap between natural language questions and existing translation methods for geospatial QA systems using machine learning methods.

To capture the semantic information in natural language questions, we first propose a geospatial question encoding schema that extends the semantic encoding schema proposed in [25]. In addition to the place name, place type, and spatial relationships used in [25], our new schema encodes additional semantic elements such as numbers, logic relationships, and comparisons. We treat question translation as a sequence labeling task, and translate questions to a sequence of geospatial elements. We additionally propose a novel geospatial semantic graph representation to capture the semantic relations between the elements. Compared with sequence structure representations, the graph structure can capture long-distance relations and non-contiguous relations. For example, given the question *Is there a forest in Lancashire north of Burnley?*, the template-based method of [55] will generate two triples — *(forest, in, Lancashire)* and *(Lancashire, north of, Burnley)* — the latter of which is incorrect (the forest should be both contained in Lancashire and be located north of the township of Burnley; Burnley is in Lancashire so cannot be north of it). In contrast, our graph representation can correctly model long-distance relationships, and predict the second triple as *(forest, north of, Burnley)*.

In addressing the research question "How can we generate spatial queries for factoid geospatial question answering using neural methods?", our contributions are:

- A new geospatial semantic encoding scheme which is amenable to sequence labeling. We propose several neural tagging methods, and show that they beat a rule-based baseline by a large margin.
- A new geospatial semantic graph formalism to model geospatial relations, and an automatic graph generation method using the semantic encodings and semantic dependency parsing. Based on the proposed tagging scheme and semantic graph formalism, we release an annotated dataset based on GeoData201.[3]

We manually evaluate the generated queries and show that our proposed approach achieves better results than existing methods.

## 2 Related work

### 2.1 Geospatial question answering

In GQA, answering geographic questions can be based on diverse information sources such as textual information [18,50], spatial databases [8], and spatially-enabled knowledge bases [19]. Based on the types of geographic questions, existing work on geographic question answering can be classified into four types [48]: (1) factoid GQA [55, 68], which focuses on answering questions based on geographic factoids; (2) geo-analytical QA [60, 64], which focuses on questions with complex spatial analytical intent; (3) scenario-based GQA [10, 11, 32], which associates questions with a scenario described with a map or a paragraph of text; and (4) visual GQA [33, 46], that links questions to an image or video. In this paper, we focus on the first of these: namely factoid GQA, and aim to answer questions through

[3] Data and code are available at https://github.com/haonan-li/neural-factoid-geoqa

a spatially-enabled knowledge base. Throughout this article, the term factoid GQA is used to refer to approaches that produce answers by querying a knowledge base.

Most existing factoid GQA systems [55, 62, 63, 68] are template-based and can only handle a limited array of geographic questions [48]. Commonly, questions are first classified into several categories according to the intent type or syntactic/semantic structure. Then predefined templates are used to match questions, and translate them into executable queries. Due to the small size of existing factoid GQA datasets, there is still a lack of an machine learning-based systems, unlike open-domain QA which is predominantly machine learning-based [4, 9, 61].

To the best of out knowledge, the earliest work on factoid geospatial question answering [62, 63] was based on the GeoQuery dataset [69], which contains 880 natural language queries about U.S. geography, paired with corresponding queries in Prolog. This line of work focused on semantic parsing from natural language to Prolog. Since Prolog is not widely used for database querying and has limited expressivity as a query language, systems based on GeoQuery are heavily limited in their scope. GeoCLEF[4] and GikiCLEF[5] led to a spike in geographic IR research, which gave rise to the first publications on general-purpose geographic question answering systems, based on information extraction from free-text documents. In 2008, the GiKiP pilot defined the task of finding Wikipedia entities that resolve a particular information need with the use of geographic reasoning [58]. The work of Hartrumpf and Leveling [27] in 2009 combined information extraction with GIR methods over a spatial index of documents, and used DBpedia together with Wikipedia to convert RDF to natural language expressions for processing by non-GIR methods.

In 2011, the Open Geospatial Consortium (OGC) proposed the GeoSPARQL[6] standard as an extension of SPARQL with better support for geospatial data. GeoSPARQL defines a vocabulary for representing geospatial data in RDF, as well as an extension to the SPARQL query language for processing geospatial data [5]. However, most people do not have the expertise to write GeoSPARQL queries directly, limiting its uptake. Some recent research [55, 68], therefore, has focused on translating natural language geospatial questions to GeoSPARQL queries. The research reported on the present article also follows the approach of translating natural language geospatial questions to GeoSPARQL queries, through an intermediate semantic representation.

Younis et al. [68] proposed a method for answering geospatial questions over DBpedia, with the ability to handle three types of geographical queries: proximity (e.g., *Find bars within 3km of the Summer Palace*), crossing (e.g., *Find the mouths of the rivers that cross Oxford*), and containment (e.g., *Find universities in Melbourne*). However, this is only a subset of possible geospatial queries, and the method is unable to capture relations about borders or cardinal directions, for example. Also, the method requires manual filtering, and as such is only a partial solution to the problem. In contrast, our approach can capture a broader range of geospatial relationships, and is completely automated.

In 2018, Punjani et al. [55] built a geospatial QA system called GeoQA, which consists of six components: dependency parse tree generator, concept identifier, instance identifier, geospatial relation identifier, SPARQL/GeoSPARQL query generator, and SPARQL/GeoSPARQL query executor. QA is performed by first translating the natural language questions to a set of SPARQL or GeoSPARQL queries, and then ranking and exe-

[4] http://www.clef-initiative.eu/track/GeoCLEF

[5] https://www.linguateca.pt/GikiCLEF/index.php/Main_Page

[6] https://www.opengeospatial.org/standards/geosparql

cuting the top-ranked queries over two endpoints using the SPARQL SERVICE keyword. They also presented a template-based approach to translating natural language questions into GeoSPARQL queries, that can be executed over the combination of GADM,[7] OpenStreetMap, and DBpedia. The use of a geospatial relation identifier helps to detect more geospatial relationships than the method of Younis et al., but the template-based query generator dramatically limits the diversity of questions the method can handle. Based on the authors' own evaluation, only 42.8% of questions match the hand-built templates. Our proposed neural method, on the other hand, can potentially translate any geospatial question into a query.

Hamzei et al. [25] recently proposed a data analysis approach to place-related questions and answers. Although there is no applicable answer generation module, their proposed semantic encoding schema maps tokens in geospatial questions to semantic elements, and is a valuable tool for question and answer analysis. The authors also proposed a rule-based pipelined method to automatic encode sentences, but found it to inevitably propagate errors generated from previous steps. In this paper, we extend their semantic encoding schema in Section 3.2, and show that our neural encoding approach alleviates error-propagation, while improving encoding accuracy.

## 2.2 Geospatial semantics and semantic parsing

Geospatial semantics is a broad area of research that focuses on understanding and formalizing geographic content [30]. Hu [30] identified six major research areas contributing to geospatial semantics through a systematic literature review: (1) *semantic interoperability and ontologies*, focused on identifying geospatial concepts to increase interoperability of geospatial data and services [31, 39]; (2) *digital gazetteers*, that use geospatial ontologies to label geospatial objects and events with spatial and non-spatial properties and relations [21]; (3) *geographic information retrieval*, which focuses on extracting geospatial semantics from Web queries and retrieving content through semantic matching [35, 41]; (4) *geospatial linked data*, which formally captures spatial objects and their relationships to establish spatially-aware knowledge bases [34]; (5) *place semantics*, which is a component of geospatial semantics and aimed at formalizing the notion of place [26, 59]; and (6) *qualitative reasoning over cognitive geographic concepts*, to provide the ability to capture and reason over human qualitative spatial knowledge [17].

In the area of natural language processing, semantic parsing is the task of mapping natural language text to graph-based meaning representations. It is widely used in knowledge-based question answering, as a semantic parser can capture the semantic relations latent in the question [4, 6, 67]. In the past few years, researchers have developed various semantic parsing formalisms and semantic graphbanks. DELPH-IN MRS-derived bi-lexical dependencies (DM), Enju Predicate-Argument Structures (PAS), and Prague Semantic Dependencies (PSD), are three formalisms that were used to annotate graphbanks as part of a 2014 SemEval shared task [52], which use tokens of the sentence as nodes and semantic relations as edges. There are notable differences between the three formalisms, and no clear answer as to which formalism is best for semantic parsing. In terms of dependency labels, PSD is linguistically more fine-grained, PAS has the smallest label inventory and the shallowest semantics, and DM is located somewhere between PSD and PAS in terms of expressivity and tractability, limiting itself to argument structures that are grammatical-

[7] https://www.gadm.org

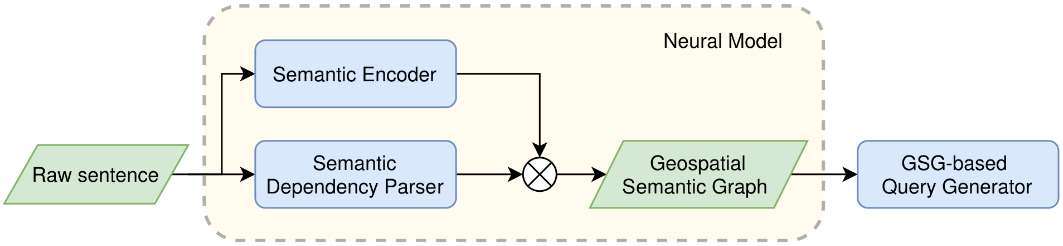

Neural Model
Semantic Encoder
Raw sentence
Semantic Dependency Parser
Geospatial Semantic Graph
GSG-based Query Generator

Figure 1: Neural factoid geospatial question answering model (NeuralGQA) architecture. The semantic encoder and semantic dependency parser are neural network modules, where embeddings are used to represent sentences. The geospatial semantic graph is an intermediate representation of each sentence, and the GSG-based query generator is a rule-based module which takes the geospatial semantic graph as input and outputs executable queries.

ized. Other graphbanks like Abstract Meaning Representations (AMR) [3] and Elementary Dependency Structure (EDS) [20] do not enforce the requirement that the original tokens of the sentence are preserved: in AMR, the alignment between graph nodes and tokens is implicit and noisy [47], while in EDS the alignments are provided, but the complex associations in the graph representation (e.g., the fact that multiple nodes can align with a single token) make them incompatible with dependency parsing methods [44].

To the best of our knowledge, we are the first to apply neural semantic parsing to geographic question answering, or geographical information systems. Note that "semantic parsing" here means semantic dependency parsing, i.e., mapping a sentence into a formal representation of its meaning in the form of a directed graph with arcs between pairs of words, which is different from the term "geoparsing" and "geotagging" in the GIR literature [2,22,49], where the terms represent the task of parsing texts to detect terms which are associated with geographic places.

Another aspect of related work is Spatial Role labeling (SpRL) [37,38,56], which concerns the extraction of the key components of spatial semantics from natural language, such as trajectors, landmarks, and spatial indicators. The main difference between geospatial semantic parsing and the SpRL task is that we analyze all types of semantic dependencies between elements in the sentences, instead of just focusing on the sentence elements that have spatial semantics. This is a direct consequence of our focus on geospatial semantics in the context of question answering, and the need to capture the full syntax and semantics of questions.

# 3 Methodology

In this section, we discuss how to translate natural language geospatial questions to the query language SPARQL/GeoSPARQL. We first introduce a new semantic encoding schema and a graph representation method that captures the semantic and logic relations of geospatial questions in Section 3.1. We then re-define the automatic encoding as a sequence labeling task in Section 3.2. Next, in Section 3.3, we discuss how to automatically generate the graph representations from raw sentences using a semantic encoder and a se-

| Semantic Type | #Sents | Prop | Semantic Type | #Sents | Prop |
|---|---|---|---|---|---|
| PLACE NAME | 200 | 100% | PLACE TYPE | 172 | 86% |
| ACTIVITY | 6 | 3% | SITUATION | 34 | 17% |
| QUESTION WORD | 200 | 100% | SPATIAL RELATIONSHIP | 178 | 89% |
| OTHER OBJECTS | 24 | 12% | COMPARISON | 68 | 34% |
| NUMBER | 26 | 13% | LOGIC RELATIONSHIP | 8 | 4% |

Table 1: Number of questions in GeoData201 which contain each of the semantic elements.

mantic parser. Finally, the transformation from graph representations to SPARQL queries is described in Section 3.4. Figure 1 shows our NeuralGQA model architecture.

## 3.1 Geospatial semantic encoding and graph representation

To semantically encode a geospatial question, we map the sequence of words in the question to a sequence of predefined semantic elements. In most cases, a span of text in the original question can be mapped into a single semantic element. For example, the expression *to the west of* can be encoded into a single SPATIAL RELATIONSHIP, and the text *San Francisco* can be encoded into a single PLACE NAME. There are several existing spatial semantic encoding schemas [16, 25, 37, 56], most of which are designed for general-purpose geographic information systems. Among these, the schema proposed by Hamzei et al. [25] is specifically designed for place-related questions, and a good fit for our work. In this section, we extend the semantic encoding schema of [25], and propose a neural approach to automatically encode questions.

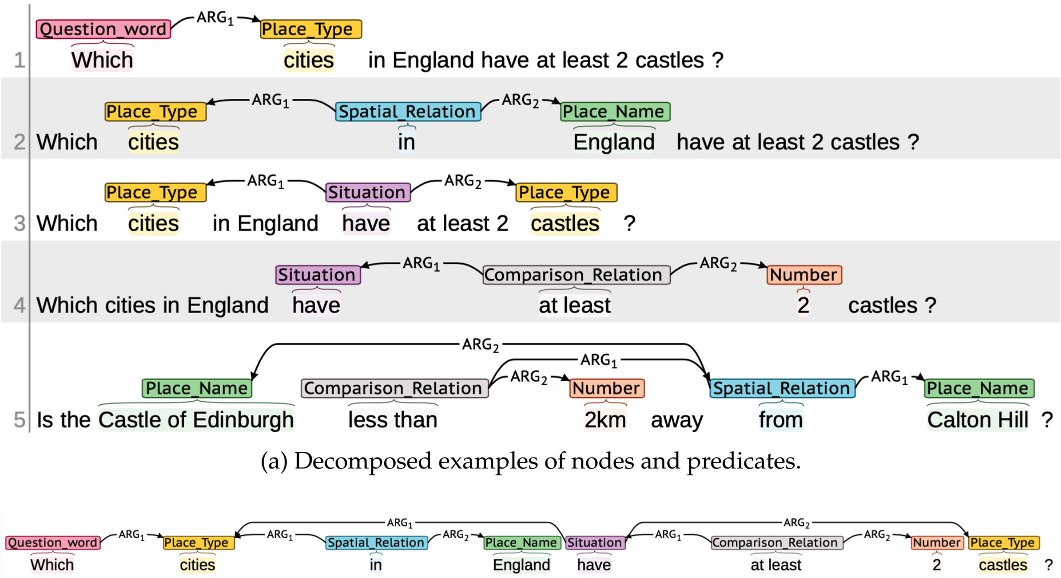
The basic set of semantic elements in our encoding is: (1) PLACE NAMES (e.g., *Melbourne*); (2) PLACE TYPES (e.g., *university*); (3) ACTIVITIES (e.g., *to study*); (4) SITUATIONS (e.g., *to live*); (5) QUALITATIVE SPATIAL RELATIONSHIPS (e.g., *near*); (6) QUALITIES (e.g., *beautiful*); (7) QUESTION WORDS (e.g., *where*); and (8) other OBJECTS. Here, we remove QUALITIES because of disagreement over the definition, and high levels of interpretation between individuals. We extend the semantic encoding to include: (9) COMPARISONS (e.g., *at least*); (10) NUMBERS (WITH UNITS) (e.g., *2km*); and (11) LOGIC RELATIONSHIPS (e.g., *and*). Table 1 details the proportion of questions that include each semantic element in the GeoData201 dataset. As can be seen, a high proportion of questions contain at least one of the new semantic elements of NUMBER, LOGIC RELATIONSHIP, and COMPARISON. Consequently, ignoring these encoding classes in the tagging process will affect the understanding of the questions. For example, in the question *What is the most populated city in the United Kingdom except London?*,[8] if we do not encode the logical relationship word *except* and query for the most populous city in the UK, *London* will be the answer.

The geospatial semantic tagger encodes natural language questions to geospatial semantic elements with a sequence tagger. As is, however, these semantic elements are arranged in the order of the corresponding text in the original question, and do not reflect the logical and semantic dependency structure of the sentence. To tackle this problem, we propose a graph representation named **G**eospatial **S**emantic **G**raph (“GSG”) to comprehensively model the semantic structure of the question sentence. The proposed graph

[8] The wording is slightly awkward, but this is the original form of the question in GeoData201.

representation consists of nodes and edges, where nodes are the semantic elements tagged in Section 3.2 that comprise a piece of text and the corresponding semantic tag, and edges are labelled with semantic relations that hold between the source and target. We first define two types of nodes: predicate nodes and argument nodes. Argument nodes, usually nouns, are arguments that are connected by predicate nodes. Predicate nodes have one or more arguments, which can in turn be argument nodes or other predicate nodes. In most cases, a given node has a unique type, but there are instances where a node is both a predicate and an argument node, as we discuss later. Edges emanate from predicate nodes and point to predicate/argument nodes. The detailed definitions of nodes and edges are listed below, with examples in Figure 2.

- QUESTION WORDS include word sequences like *where*, *what*, *does*, and *is there*, and are always predicate nodes with zero or one argument. For true/false questions — for example when the question starts with *does* — the argument of the question word is usually the rest of the sentence, so we do not explicitly indicate it and treat the number of arguments as zero. For other QUESTION WORDS like *What* or *Which*, the arguments are usually the direct objects of the question words. In Example 1 in Figure 2a, the node *cities* is the argument of *which*.
- PLACE NAMES and PLACE TYPES are always argument nodes. A PLACE TYPE usually serves as a constrain to filter pre-selected candidate places. Although it does not explicitly refer to any place, it implies a set of candidate places of the same type, such that any predicates that connect to a PLACE NAME can also connect to a PLACE TYPE. As shown in Example 2 in Figure 2a, *cities* (PLACE TYPE) and *England* (PLACE NAME) are arguments of the predicate *in*.
- SPATIAL RELATIONSHIPS, as in Example 2 of Figure 2a, are predicate nodes. They have two arguments and together form a triple relation (ARG1, predicate, ARG2). Such predicates denote a spatial relationship between the two arguments, and optionally include additional attributes like the distance between the two arguments. These implicit attributes mean that predicate nodes have the potential to be an argument node of other relations. In Example 5, the predicate node *from* links two place names *Castle of Edinburgh* and *Calton Hill* without any explicit spatial relationship, in addition to implying the distance attribute between these two places. Hence, it can be used as an argument node of other predicates (*less than*) and compared with other numbers.
- ACTIVITIES and SITUATIONS are usually predicate nodes. Similar to SPATIAL RELATIONSHIPS, these words sometimes include attributes, usually specifying the number of instances that satisfy the activity or situation. In this case, they can be a predicate and argument node at the same time. In Example 3 in Figure 2a, *have*, as a predicate node, implies the number of *castles* that a city has. While in Example 4, it plays the role of an argument node.
- COMPARISONS like *at least* and *less than* are always predicate nodes in the graph. They can either connect two nodes of the same type to make a comparison, or connect nodes with different semantic types where one implies the same type of concept as another. As mentioned, a comparison node can connect a number with a predicate node, to imply a distance between two places. In Example 4, *have* implies the number of *castles* in a city, making it an argument that can be compared with other instances of class NUMBER.



(a) Decomposed examples of nodes and predicates.

(b) Whole sentence graph.

Figure 2: Examples geospatial semantic graphs.

- LOGIC RELATIONSHIPS are always predicate nodes with two arguments. Words like *and* or *or* connect two arguments of the same semantic type. Negation words like *not* or *except* connect two arguments with different semantic types. For example, in the sentence *What is the largest city in China after Beijing?*, the node *after* connects a PLACE TYPE with a PLACE NAME.
- NUMBERS and other OBJECTS are always arguments, as in Example 4 in Figure 2a.

Based on the definitions above, we manually annotate the 201 geospatial questions in the GeoData201 dataset with semantic graphs. Intuitively, the semantic relations between two or three nodes are straightforward to identify, as demonstrated in Figure 2a. However, when we take all nodes into consideration (see Figure 2b), the number of possible edges between nodes increases rapidly.

## 3.2 Generating geospatial semantic encodings through sequence labeling

The encoding schema defines the semantic categories to assign sequences of words to. However, manual labeling is time-consuming and labor-intensive, and we desire an automatic approach to tagging semantic elements. Various tagging approaches have been proposed in NLP. The most common approaches are pipelines, as shown in Figure 3 (left). A typical pipeline consists of several steps including stopword removal, part-of-Speech tagging, gazetteer and dictionary lookup, and preposition classification. These pipelined methods are usually paired with hand-engineered rules to assign tags to text. However, these rule-based approaches are not able to handle misspellings, abbreviations, and other

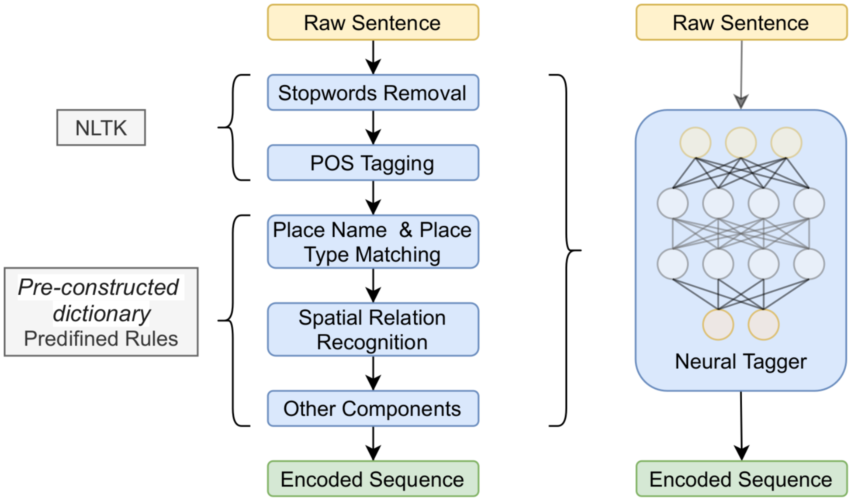


Figure 3: Comparison of tagging methods: (1) to the left is a pipelined, rule-based tagging approach; and (2) to the right is our neural tagging approach. The grey boxes show the tools we used to achieve the corresponding steps.

types of irregular text. Furthermore, the pipelined structure inevitably results in error propagation, such as the rule in [25] that only prepositions anchored to a place name (e.g., *in Shanghai*) should be encoded as a spatial relationship. Using a pipelined encoding method, if a place name is not detected, the spatial relationship will also not be detected because the preposition's anchor will be considered to be missing.

In this paper, we treat semantic encoding as a sequence labeling task. We first project the tokens into a dense vector space in which words with similar meanings have similar representations.[9] For example, in word representation vector spaces, the word *US* is likely to be near to *America* but far from *sport*. After representing words by their embeddings, a sequence labeling model is used to capture the inter-word dependencies and sentence structure. Given an annotated dataset, any sequential tagger can be trained and used end-to-end. As illustrated in Figure 3 (right), compared with the rule-based pipelined encoding method (left), a neural approach is more direct and concise. Figure 4 is an example of the outputs of the rule-based approach of [25] and our neural method (as detailed in Section 4.1). The rule-based tagger based on the original semantic encoding schema fails to detect numbers (e.g., *1km*) and abbreviations (e.g., *LA*) because it lacks gazetteers, and thus fails to capture spatial relationships (e.g., *in a range of*) associated with them. In contrast, our neural tagger with the new tagging schema can tag more diverse semantic elements, even when containing non-canonical text. The end-to-end model method labels all words in a sentence in a single step, effectively eliminating error propagation.

[9] This distributed representation is usually called a "word embedding".

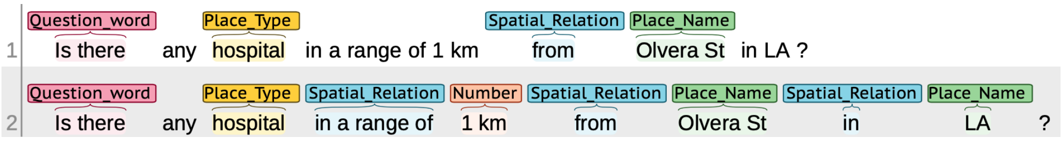


Figure 4: Example geospatial semantic tagging outputs. The first is the result of a rule-based tagger, which can not detect NUMBERS, LOGIC RELATIONSHIP, and COMPARISONS, and the second is the result of our neural tagger.

### 3.3 Enhanced semantic parsing for GSG

To generate the geospatial semantic graph automatically, we first use a semantic dependency parser to map sentences to dependency graphs. Then, we combine the parsing results with the semantic encoding results to construct the final geospatial semantic graph.

In this paper, we use the compositional semantic parser of Lindemann et al. [45] to generate the semantic graph. The idea is to build graphs from smaller graph fragments by representing each graph with its compositional tree structure, and learning to generate the graph using a dependency parser and "supertagger". Here, the supertagger is used to identify elementary graphs for each individual word, and the dependency parser is responsible for identifying the algebraic operations to combine the elementary graphs. As demonstrated in Figure 5, the DM semantic graph of the sentence in Figure 6 (a) is built from the compositional trees and operations that combine these trees. The compositional tree structures of [45] are highly similar to our decomposed graph structure in Figure 2a. Their parser combines the compositional semantic AMR parser [23] with some heuristics that help to compute the latent compositional structures of graphs, and learns latent representations across several semantic datasets based on multi-task learning. Among the five graphbanks mentioned in Section 2.2, AMR and EDS graphbanks map words to concepts during parsing based on a semantic encoding (much like machine translation). In contrast, DM, PAS and PSD are dependency graphs that map words of the sentence onto graph nodes. In addition, DM limits itself to argument structure distinctions that are grammaticalized, making it the best fit for GSG. Hence, we use their basic model to predict the DM graph representations as the intermediate graph representation.

Because the DM graph representation treats each word in the given sentence as a node, it introduces some unwanted nodes that need to be merged in our geospatial graph. An example is shown in Figure 6 a, where in the DM formalism, the tokens *Calton* and *Hill* are nodes which are linked by an edge with label *compound*, indicating that this is a compound word. However, in our geospatial semantic graph representation, multiple words in the sentence may make up one node. To transform DM graph representations to our geospatial semantic graph representations, we combine the generated DM graphs with the semantic encoding results, and apply a series of rules to adjust the edges between nodes.

With the example in Figure 6, the detailed structure transformation rules are:

1. Remove intra-node edges and retain inter-node edges. (Figure 6 b)
2. Remove all tokens not belonging to any nodes. If a non-node token links two nodes, remove the token but add an extra edge between the two nodes; now all edges are inter-node edges. (Figure 6 c)
3. For a predicate node that has two argument nodes, if both nodes are of the same type, or one is PLACE TYPE and the other is PLACE NAME, keep the edges; if the predicate

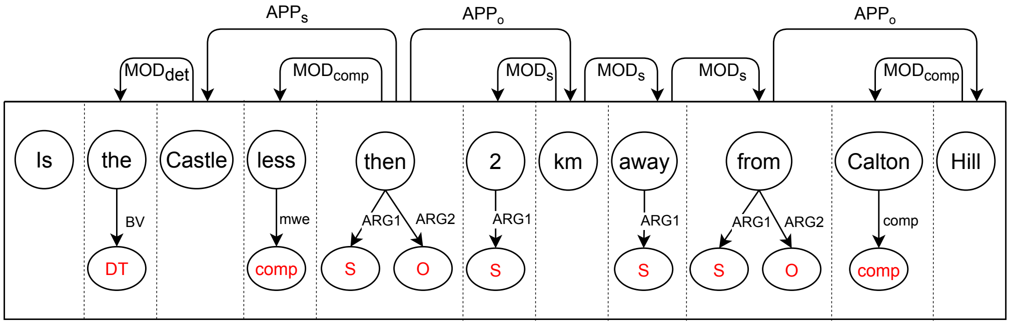


Figure 5: Compositional tree structure (nodes and edges in box), and algebraic operations (on top of the box). There are two operations: the *apply* operation $APP_X$ and *modify* operation $MOD_X$. In the figure, "mwe", "comp", and "det" signify multi-word expression, compound word, and determiner, respectively.

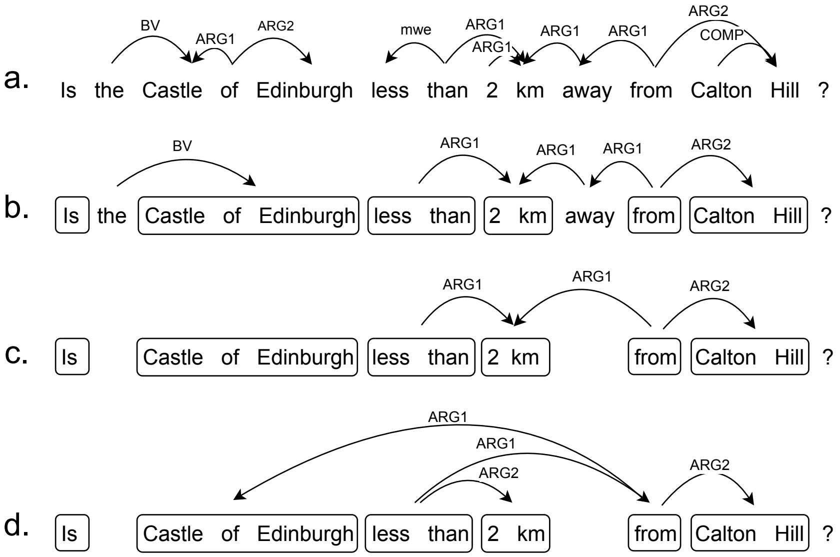

a.
Is the Castle of Edinburgh less than 2 km away from Calton Hill ?
BV
ARG1
ARG2
mwe
ARG1
ARG1
ARG1
ARG1
ARG2
COMP
b.
Is the Castle of Edinburgh less than 2 km away from Calton Hill ?
BV
ARG1
ARG1
ARG1
ARG2
c.
Is Castle of Edinburgh less than 2 km from Calton Hill ?
ARG1
ARG1
ARG2
d.
Is Castle of Edinburgh less than 2 km from Calton Hill ?
ARG1
ARG1
ARG2
ARG2

Figure 6: Example of the generated DM semantic graph and the proposed structure tranformation rules, given the gold semantic encode. Each box indicate a geospatial semantic node, which may be PLACE NAME, COMPARISON, etc. We keep the direction and label of edges from the generated DM graph for convenience. Note that the label direction in the final graph is only depends on the node type (from predicate node to argument node).

node is SPATIAL RELATION and one of the argument nodes is NUMBER, shift it to the nearest unconnected PLACE NAME or PLACE TYPE, and add a new edge to the COMPARISON node if there is no it is only connected with NUMBER. (Figure 6 d)

In this way, we generate the geospatial semantic graph.

## 3.4 From GSG to SPARQL queries

To demonstrate an application of the proposed semantic parsing method, we developed a template-based approach to generate GeoSPARQL queries from GSG representations, focusing on ASK (yes/no) and SELECT (return stored values) queries. GeoSPARQL queries have several constituents including PREFIXES, ASK/SELECT statements, and WHERE clauses. PREFIXES define the namespaces from which to access knowledge bases, ontologies, and implemented functions. The ASK/SELECT statements determine the output of the query, and the WHERE clauses capture the criteria mentioned in the question. These constituents are constructed based on predefined templates, described below.

First, the overall structure of a given query (i.e., ASK vs. SELECT query) is determined from the extracted question words. Next, the WHERE-clause is dynamically generated by concatenating individual concept and relation definition statements. Finally, in case of SELECT queries, the intent of questions is derived using a heuristic approach. Here, we used a predefined set of prefixes to access SPARQL/GeoSPARQL functions and to generate shorter and easier-to-read queries (Query 1).

```
PREFIX geos: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX db: <http://spatial.au/ontology#>
PREFIX spatialf: <http://jena.apache.org/function/spatial#>
PREFIX units: <http://www.opengis.net/def/uom/OGC/1.0/>
```

Query 1: Predefined set of prefixes

The WHERE-clause is part of the general structure for both ASK and SELECT queries. To generate the WHERE-clause, for each extracted geographic concept (i.e., Place names, Place types, and Objects) a unique variable is assigned. Each concept is defined using predefined templates and the corresponding variable name. The concept definition statements define a place based on Place name or Place type using an *is-a* relationship. The extracted Objects such as population and temperature are defined as properties of their corresponding concept variables. The extracted relations among the concepts (i.e., Activities, Situations, Logical, and Spatial relationships) are translated to the formal query language using the participating variables and predefined relation templates. Definition statements for Place names and Place types are presented in Queries 2, and 3, respectively.

```
?<PLACE> db:has_name ?<PLACE>NAME;
geos:hasGeometry ?<PLACE>GEOM.
FILTER(regex(?<PLACE>NAME, "<TOPONYM>", "i" )).
```

Query 2: Place name definition template

```
?<PLACE> db:has_fclass ?<PLACE>TYPE;
geos:hasGeometry ?<PLACE>GEOM;
db:has_name ?<PLACE>NAME.
FILTER(regex(?<PLACE>TYPE, "<TYPE>", "i" )) .
```

Query 3: Place type definition template

The relationships are captured through *has-a* relationships. In the case of spatial relationships, a lookup table is used to map the spatial prepositions to their corresponding spatial relationships implemented in GeoSPARQL. The relationships are constructed by the *predicate* and *argument(s)* which are captured in the semantic parse tree. The template for defining attributes and distance relationships, as examples of the relation templates, are shown in Queries 4 and 5.

```
        ?<PLACE> db:has_<ATTRIBUTE> ?<OBJECT>.
```

Query 4: Attribute relation template

```
        FILTER(geof:distance(<PLACE1>, <PLACE2>, units:<UNIT>) < <DISTANCE>).
```

Query 5: Distance relation template

In the case of SELECT queries, the SELECT-statement could: (1) select some variables, or (2) apply a function (e.g., distance) over the values of selected variables. The selected variables are related to the intent of the questions. Here, we extended the heuristic for extracting intent of geographic questions developed by Xu et al. [64]. The intent is derived using the following rules:

- **Question Word Rule:** Here, we focus on *where*, *what*, and *which* questions. For *where* questions, the intent is the location of the selected variable. For *what* and *which* question, the following rules are used to determine the intent.
- **Less Specificity Rule:** The more specific the concepts are, the less likely they are to be the intent of the question — e.g., an Object or a Place type (if mentioned in the question) is more likely to be the intent of the question in comparison to a Place name. For example, in the question *Which cities in England have at least two castles?*, cities and castles are more likely to be the intent of the question than England. This rule determines the specificity of concepts using their encoding classes, i.e., Objects, Place types, and Place names. Here, the objects (properties of places) are the least specific concept. Next most specific are Place types, and finally Place names.
- **Earlier Position Rule:** This rule is based on the structure of the questions. The subject of the questions is the earliest concept extracted from the questions. For example, in question: *Which cities in England have at least two castles?*, the concept (i.e., cities) is determined as the intent of the question.

Finally, aggregation, sorting, and spatial functions such as distance are considered to be part of the SELECT-statement. The general template for functions is constructed by the function name and a set of variables. The relation between function names and their variables is captured in the parsing results. In the special case of aggregation and sorting, GROUP-BY and HAVING statements, and ORDER-BY and LIMIT-statements are concatenated at the end of the generated query, respectively. Query 6 shows the generated GeoSPARQL query for the question, *Is the Castle of Edinburgh less than 2km away from Calton Hill?*

```
PREFIX geos: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX db: <http://spatial.au/ontology#>
PREFIX spatialf: <http://jena.apache.org/function/spatial#>
PREFIX units: <http://www.opengis.net/def/uom/OGC/1.0/>

ASK {
?castl0 db:has_name ?castl0NAME;
geos:hasGeometry ?castl0GEOM.
FILTER(regex(?castl0NAME, "castle␣of␣edinburgh", "i" )) .

?calto1 db:has_name ?calto1NAME;
geos:hasGeometry ?calto1GEOM.
FILTER(regex(?calto1NAME, "calton␣hill", "i" )) .

FILTER(geof:distance(?calto1GEOM, ?castl0GEOM, units:kilometer) < 2 ).

}
```

Query 6: A generated GeoSPARQL query

# 4 Experiments and results

To test the effectiveness of the proposed semantic encoding schema and graph representation, as well as evaluate the corresponding neural model, we use the annotated GeoData201 dataset and perform 5-fold cross-validation experiments on semantic encoding and graph generation.[10] In this section, we present the technical details, evaluation metrics, and experimental results.

## 4.1 Experimental setup

### 4.1.1 Semantic encoding

Formally, the input for semantic encoding is a question sentence $S = < s_1, s_2, ..., s_n >$ where $s_i$ is the $i$-th token in the sentence, and the output is a list of semantic labels (described in Section 3.2) $< t_1, t_2, ..., t_n >$ corresponding to each token in the input. We baseline against a rule-based encoding method, which we compare with two trained sequence taggers.

We reimplement the rule-based semantic encoder of [25] as a baseline. In detail, we first perform sentence tokenization and POS-tagging and dependency parsing using the NLTK toolkit.[11] [12] then the semantic encoding is done based on the pre-constructed dictionary and predefined rules. A brief overview of the rules is as follows:

- **Noun encoding** Nouns are encoded in the order of place names (toponyms), followed by place types, and objects. The GeoNames gazetteer is used to match toponyms, and remaining nouns are matched against a dictionary of place types, failing which, they are encoded as objects.

[10] Due to the very small data size, we do not perform hyperparameter tuning.

[11] https://www.nltk.org

[12] NLTK is used to perform part-of-speech tagging, using tags such as verb, noun, adjective, determiner, adverb, pronoun, preposition, conjunction, and interjection. Note that these differ from our semantic tags (place name, place type, comparison, etc.), which are the result of the rule-based semantic encoder.

- **Verb encoding:** First, we collect two sets of dynamic verbs (action and stative verbs) from online resources, and generate an ELMo [54] embedding vector for each verb in the verb set. Then we classify the verbs in the natural language questions into activities or situations by calculating the Euclidean distances between the embedding vector of the verb and the vectors in the two sets, and classify based on the $k$-nearest-neighbour algorithm with $k$ equals to 3.
- **Preposition and adjective encoding:** First, we build a dependency parse tree of the question sentences. Then extract the *case* and adjective modifier (*amod*) dependencies for prepositions and adjectives. Finally, we encode prepositions anchored to a place name as spatial relationships, and adjectives modifying a place type or a place name as qualities of places.

For full details, see [25].

For the neural network model, we treat the task as a token-level multi-class classification task for each token in the sentence. We experiment with two model architectures: (1) a recurrent neural network, and (2) a transformer-based neural network. As detailed below:

- The recurrent neural network is implemented by building a dense layer on top of a single long short term memory (LSTM) layer [29]. The LSTM layer can help encode the contextual information, and the dense layer helps to simplify the hidden states and make the classification. We concatenate GloVe [53] and ELMo [54] embeddings as word representations, and train the model for 20 epochs using stochastic gradient descent (SGD) with a learning rate of 0.1 and batch size of 16.
- For the transformer model, we build a dense-layer classifier on top of BERT, a pre-trained language model [13]. We use `BERT-base-uncased` (12 layers and 768 hidden size) and fine-tune the entire model for 10 epochs with a learning rate of $5e^{-5}$ and batch size of 16. All other hyperparameters are set to the default.

### 4.1.2 Semantic graph generation

For geospatial semantic graph generation, we use two baselines. The first is a random baseline, and generated by randomly creating edges between any two nodes. For two ordered nodes $N_1, N_2$, the probability of the edge from $N_1$ to $N_2$ is set to 0.5.[13] A random baseline is commonly used to assess (i.e., calibrate) the complexity of a task and evaluate whether a machine learning model has really learned useful information. We use it here to provide the reader with an estimate of an overall level of challenge. For example, for a binary classification task with an unbalanced dataset, a majority-baseline accuracy of 70% is a proxy for the relative label skew, and it is reasonable to expect results well over 70%.

The second baseline is to create edges by syntactic dependency parsing. We use biaffine classifiers on top of a bidirectional LSTM (BiLSTM) [15] with GLoVe embeddings. We remove intra-node dependencies and keep inter-node dependencies as edges.

As a comparison, our model takes the form of a semantic parser using the pre-trained BiLSTM-based arc-factored semantic dependency parsing model of Kiperwasser and Goldberg [36] with BERT embeddings.[14] The parser predicts the semantic dependency structure of a question, which we then translate into a node-level semantic structure using our

[13] Noting that we do not generate labels, so can only evaluate in terms of unlabelled attachment scores.

[14] BERT decomposes a word into multiple tokens and generates sub-token representations, we then pull out a single vector for each original word.

structure transformation rules described in Section 3.3. Note that the proposed structure transformations are based on the combination of the semantic parsing results and the DM graph representation, while the baseline models are not combined and tested with the DM representation.

#### 4.1.3 Query generation

The query writer takes the semantic graph representation as input and produces the GeoSPARQL query. To make it comparable with previous work [55], we conduct generation and evaluation on half of the dataset, following [55]. As different knowledge bases have different ontologies and resources (e.g., a street name may appear in OpenStreetMap but be missing in DBpedia) we didn't assign the namespaces of the triples to the resulting queries.

### 4.2 Evaluation metrics

The evaluation of semantic encoding is conducted on three levels: token-level, node-level, and sentence-level. For token-level evaluation, we treat each token individually with a corresponding tag and report the accuracy score:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{1}$$

For node-level evaluation, similar to the assessment for NER, each node is a unit, and the prediction is considered to be correct if both the node boundary and tag are correct. We report precision, recall, and F1-score of the prediction results as follows:

$$\text{Precision} = \frac{\text{Number of correct predicted edges}}{\text{Total number of predicted edges}} \tag{2}$$

$$\text{Recall} = \frac{\text{Number of correct predicted edges}}{\text{Total number of true edges}} \tag{3}$$

$$\text{F1-score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

For sentence-level evaluation, a sentence is correct only if all node boundaries and tags are correct. We calculate the sentence-level performance as an accuracy score.

The semantic graph generation results are evaluated at the edge and graph levels. For edge-level evaluation, we report unlabelled precision, recall and F1-score of the generated edges. For graph-level evaluation, a graph is correct only if all predicted edges are correct. We report the whole graph accuracy.

Because we do not have the ground truth answers of the GeoData201 dataset, the evaluation of the generated queries is performed manually. We report the ratio of successfully generated queries and the ratio of correct queries. Three PhD students familiar with SPARQL query language and knowledge base were asked to do the manual evaluation. Given a natural language question, they were asked to judge whether the model generated query is correct. The final decisions were made by majority vote.

| **Method** | Token-level | Node-level | | | Sentence-level |
|---|---|---|---|---|---|
| | Acc | P | R | F | Acc |
| Rule-based | 78.8 | 76.7 | 76.2 | 76.5 | 34.5 |
| LSTM | 92.6 | 83.9 | 85.3 | 84.6 | 48.5 |
| BERT | **96.4** | **93.3** | **95.0** | **94.1** | **78.0** |

Table 2: Semantic encoding results. "Acc", "P", "R", and "F" denote accuracy, precision, recall, and F1-score, respectively.

| **Method** | g.t. | Tag | Edge | | | Graph |
|---|---|---|---|---|---|---|
| | | Acc | P | R | F | Acc |
| Random | yes | 100 | 16.4 | 51.1 | 24.8 | 0.0 |
| SynP | yes | 100 | 26.3 | 39.7 | 31.6 | 0.0 |
| SemP | no | 96.4 | 53.3 | 66.9 | 59.3 | 25.0 |
| | yes | 100 | 56.2 | 69.8 | 62.3 | 29.0 |
| SemP + ST | no | 96.4 | 71.1 | 72.9 | 72.0 | 45.0 |
| | yes | 100 | **75.5** | **75.8** | **75.7** | **50.0** |

Table 3: Graph generation results. "g.t." indicates whether we use gold geospatial semantic tags. "SynP" denotes syntactic parsing, "SemP" denotes semantic parsing, and "ST" denotes structural transformation.

## 4.3 Results

Table 2 shows the semantic encoding results. The neural network approaches beat the rule-based approach across all evaluation metrics, showing that they are both more robust but also more accurate, and able to capture context despite the small data training regimen. The BERT-based model achieves the best performance with absolute improvements over the rule-based baseline of 17.6%, 17.7%, and 43.5% at the token-level, node-level (F1), and sentence-level, respectively. BERT is helpful in many NLP tasks because of the large-scale pre-training and well-designed model structure. For this token-level classification task, the reason it outperforms the LSTM model is likely due to the better contextualized representations.

The graph generation results are presented in Table 3. The two baseline models get poor performance on both edge prediction and graph construction. Compared with syntactic dependency parsing, the random baseline gets a higher recall but lower precision, which is because it predicts a denser graph than the ground truth. In the GeoData201 dataset, one node usually has less than three edges linked to it, but for the random baseline, each node will be expected to have $n/2$ edges (where $n$ is the total number of nodes). The syntactic parser ("SynP") beats the random baseline because of the higher precision. The results of the semantic dependency parser ("SemP") under different settings outperform the two baselines by a substantial margin. When paired with structure transformation rules, both

| | Template coverage | Generate Query | Correct Query | Final* |
|---|---|---|---|---|
| GeoQA | 43.0 | 51.2* | — | 22.0 |
| NeuralGQA | 100.0 | 45.0* | 71.0* | **38.0** |

Table 4: Query generation results. "*" indicates the results are based on the correct predictions of previous step.

edge prediction and graph prediction increase, and we produce a perfectly correct semantic graph (a very stringent evaluation metric!) 50% of the time.

Table 4 shows the evaluation over the generated queries, including results for GeoQA [55] on the same test set (noting that due to the different evaluation metrics and query generation approaches, their results are not directly comparable with ours). In terms of the number of questions that can be handled, their template-based method covers 43% questions in the dataset, while our neural approach can deal with all questions. In other words, our neural method can generate graph representation for any geospatial question, which is a significant advance on this task. Of all the questions that can be handled (43% for GeoQA and 100% for NeuralGQA), the GeoQA method successfully generates answers for 51% of the questions. The final score is the proportion of queries for which an answer was successfully generated. As a comparison, our method correctly generates semantic graphs for 45% of questions, and manual evaluation shows that our graph-to-query transformation achieves 71% accuracy.

# 5 Discussion

## 5.1 Drawback of the pipelined method

The proposed method uses two pipelined steps to build geospatial semantic graphs from raw text inputs, where the first step is semantic encoding, and the second step is graph construction from the encoded sentences. As mentioned above, pipelining leads to error propagation. From Table 2, we saw that the best tagger could achieve only 73% sentence-level accuracy. Table 3 demonstrates that tagging errors result in failures to build semantic graphs. Building a joint model for tagging and parsing [28, 51, 65] is a possible solution to this problem, which we plan to investigate in future work.

## 5.2 Machine learning vs. rule-based approach

Although the experimental results show that the neural method beats the rule-based method, there are some limitations to the neural method. First, the lack of explainability, and perhaps more importantly, lack of fine-grained control in training of the neural network, is a well-known limitation of neural methods. Second, any machine learning approach (whether neural or otherwise) is data-driven and generally requires a large amount of training data. We used a pretrained DM graph generation model to generate intermediate representation in our experiments, rather than an end-to-end method, because of the prohibitively small size of the dataset.

Theoretically, rule-based methods can achieve (near-)perfect results over a closed dataset, but the process of designing hundreds or even thousands of rules is time-consuming, and the resulting system is often brittle and doesn't generalize well to unseen data. Machine learning methods learn patterns from a large number of samples, and there are well-established training methods for reducing overfitting on the training data, and encouraging generalization. In this paper, we demonstrated the utility of a neural method for factoid geospatial question answering. The sequence labeling approach we proposed can be applied to other encoding schemata, where similar results should be achievable with similar amounts of labelled data (160 sentences) and numbers of label types (10).

## 6 Conclusions and future work

In this paper, we addressed the task of translating natural language geospatial questions into executable queries using semantic encoding and parsing. We proposed a new semantic encoding schema and graph representation to model the semantics of geospatial questions, and automated the tasks of semantic encoding tagging and graph representation generation. To evaluate our method, we manually annotated the GeoData201 dataset and trained several neural network models over it, and found our NeuralGQA method can achieve better results than all previous approaches, providing a broad-coverage automatic query generation method.

In the future, we plan to extend the GeoData201 dataset with extra questions and annotations, to better train our machine learning models. With a larger dataset, we will be able to train semantic parsers on it directly, rather than using intermediate representations combined with structure transformations. Further, we will endow the query generator with the ability to automatically assign namespaces using a reinforcement learning approach, to make the system fully automatic.

## References

[1] AUER, S., BIZER, C., KOBILAROV, G., LEHMANN, J., CYGANIAK, R., AND IVES, Z. G. Dbpedia: A nucleus for a web of open data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference* (2007), pp. 722–735. 10.1007/978-3-540-76298-0_52.

[2] AVVENUTI, M., CRESCI, S., NIZZOLI, L., AND TESCONI, M. Gsp (geo-semantic-parsing): geoparsing and geotagging with machine learning on top of linked data. In *European Semantic Web Conference* (2018), Springer, pp. 17–32. 10.1007/978-3-319-93417-4_2.

[3] BANARESCU, L., BONIAL, C., CAI, S., GEORGESCU, M., GRIFFITT, K., HERMJAKOB, U., KNIGHT, K., KOEHN, P., PALMER, M., AND SCHNEIDER, N. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse* (2013), pp. 178–186.

[4] BAO, J., DUAN, N., YAN, Z., ZHOU, M., AND ZHAO, T. Constraint-based question answering with knowledge graph. In *26th International Conference on Computational Linguistics, Proceedings of the Conference* (2016), pp. 2503–2514.

[5] BATTLE, R., AND KOLAS, D. GeoSPARQL: enabling a geospatial semantic web. *Semantic Web Journal 3*, 4 (2011), 355–370.

[6] BERANT, J., CHOU, A., FROSTIG, R., AND LIANG, P. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (2013), pp. 1533–1544.

[7] BOLLACKER, K. D., EVANS, C., PARITOSH, P., STURGE, T., AND TAYLOR, J. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (2008), pp. 1247–1250. 10.1145/1376616.1376746.

[8] CHEN, W., FOSLER-LUSSIER, E., XIAO, N., RAJE, S., RAMNATH, R., AND SUI, D. A synergistic framework for geographic question answering. In *2013 IEEE Seventh International Conference on Semantic Computing* (2013), IEEE, pp. 94–99. 10.1109/ICSC.2013.25.

[9] CHEN, Y., WU, L., AND ZAKI, M. J. Bidirectional attentive memory networks for question answering over knowledge bases. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2019), Association for Computational Linguistics, pp. 2913–2923. 10.18653/v1/n19-1299.

[10] CONTRACTOR, D., GOEL, S., SINGLA, P., ET AL. Joint spatio-textual reasoning for answering tourism questions. *arXiv preprint arXiv:2009.13613* (2020).

[11] CONTRACTOR, D., SHAH, K., PARTAP, A., SINGLA, P., ET AL. Large scale question answering using tourism data. *arXiv preprint arXiv:1909.03527* (2019).

[12] COUCLELIS, H., FRANK, A. U., CAMPARI, I., AND FORMENTINI, U. People manipulate objects (but cultivate fields): Beyond the raster-vector debate in GIS. In *Proceedings of the International Conference on GIS* (1992), vol. 639, pp. 65–77. 10.1007/3-540-55966-3_3.

[13] DEVLIN, J., CHANG, M., LEE, K., AND TOUTANOVA, K. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2019), pp. 4171–4186. 10.18653/v1/n19-1423.

[14] DONG, L., WEI, F., ZHOU, M., AND XU, K. Question answering over freebase with multi-column convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing* (2015), pp. 260–269. 10.3115/v1/p15-1026.

[15] DOZAT, T., AND MANNING, C. D. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations* (2017).

[16] EDWARDES, A. J., AND PURVES, R. S. Eliciting concepts of *place* for text-based image retrieval. In *Proceedings of the 4th ACM Workshop On Geographic Information Retrieval* (2007), pp. 15–18. 10.1145/1316948.1316953.

[17] EGENHOFER, M. J., AND MARK, D. M. Naive geography. In *International Conference on Spatial Information Theory* (1995), Springer, pp. 1–15.

[18] FERRÉS, D., AND RODRÍGUEZ, H. Experiments adapting an open-domain question answering system to the geographical domain using scope-based resources. In *Proceedings of the Workshop on Multilingual Question Answering* (2006), Association for Computational Linguistics, pp. 69–76.

[19] FERRÉS, D., AND RODRÍGUEZ, H. TALP at GikiCLEF 2009. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments* (2010), Springer Berlin Heidelberg, pp. 322–325. 10.1007/978-3-642-15754-7.

[20] FLICKINGER, D., HAJIČ, J., IVANOVA, A., KUHLMANN, M., MIYAO, Y., OEPEN, S., AND ZEMAN, D. Open SDP 1.2, 2017. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[21] GOODCHILD, M. F., AND HILL, L. L. Introduction to digital gazetteer research. *International Journal of Geographical Information Science 22*, 10 (2008), 1039–1044. 10.1080/13658810701850497.

[22] GRITTA, M., PILEHVAR, M. T., LIMSOPATHAM, N., AND COLLIER, N. What's missing in geographical parsing? *Language Resources and Evaluation 52*, 2 (2018), 603–623. 10.1007/s10579-017-9385-8.

[23] GROSCHWITZ, J., LINDEMANN, M., FOWLIE, M., JOHNSON, M., AND KOLLER, A. AMR dependency parsing with a typed semantic algebra. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (2018), pp. 1831–1841. 10.18653/v1/P18-1170.

[24] GRÜTTER, R., PURVES, R. S., AND WOTRUBA, L. Evaluating topological queries in linked data using dbpedia and geonames in switzerland and scotland. *Trans. GIS 21*, 1 (2017), 114–133. 10.1111/tgis.12196.

[25] HAMZEI, E., LI, H., VASARDANI, M., BALDWIN, T., WINTER, S., AND TOMKO, M. Place questions and human-generated answers: A data analysis approach. In *Geospatial Technologies for Local and Regional Development - Proceedings of the 22nd AGILE Conference on Geographic Information Science* (2019), pp. 3–19. 10.1007/978-3-030-14745-7_1.

[26] HAMZEI, E., WINTER, S., AND TOMKO, M. Place facets: a systematic literature review. *Spatial Cognition & Computation 20*, 1 (2020), 33–81. 10.1080/13875868.2019.1688332.

[27] HARTRUMPF, S., AND LEVELING, J. Recursive question decomposition for answering complex geographic questions. In *Workshop of the Cross-Language Evaluation Forum for European Languages* (2009), pp. 310–317. 10.1007/978-3-642-15754-7_36.

[28] HATORI, J., MATSUZAKI, T., MIYAO, Y., AND TSUJII, J. Incremental joint approach to word segmentation, pos tagging, and dependency parsing in chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics* (2012), pp. 1045–1053.

[29] HOCHREITER, S., AND SCHMIDHUBER, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780.

[30] HU, Y. 1.07 - geospatial semantics. In *Comprehensive Geographic Information Systems*. Elsevier, 2018, pp. 80–94. https://doi.org/10.1016/B978-0-12-409548-9.09597-X.

[31] HU, Y., JANOWICZ, K., CARRAL, D., SCHEIDER, S., KUHN, W., BERG-CROSS, G., HITZLER, P., DEAN, M., AND KOLAS, D. A geo-ontology design pattern for semantic trajectories. In *International conference on spatial information theory* (2013), Springer, pp. 438–456.

[32] HUANG, Z., SHEN, Y., LI, X., WEI, Y., CHENG, G., ZHOU, L., DAI, X., AND QU, Y. Geosqa: A benchmark for scenario-based question answering in the geography domain at high school level. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (2019), pp. 5865–5870. 10.18653/v1/D19-1597.

[33] JANOWICZ, K., GAO, S., MCKENZIE, G., HU, Y., AND BHADURI, B. Geoai: Spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond. *Int. J. Geogr. Inf. Sci.* (2020), 625–636. 10.1080/13658816.2019.1684500.

[34] JANOWICZ, K., SCHEIDER, S., PEHLE, T., AND HART, G. Geospatial semantics and linked spatiotemporal data –past, present, and future. *Semantic Web 3*, 4 (2012), 321–332. 10.5555/2590208.259020.

[35] JONES, C. B., PURVES, R., RUAS, A., SANDERSON, M., SESTER, M., VAN KREVELD, M., AND WEIBEL, R. Spatial information retrieval and geographical ontologies an overview of the spirit project. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2002), SIGIR '02, Association for Computing Machinery, p. 387–388. 10.1145/564376.564457.

[36] KIPERWASSER, E., AND GOLDBERG, Y. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Trans. Assoc. Comput. Linguistics 4* (2016), 313–327. 10.1162/tacl_a_00101.

[37] KOLOMIYETS, O., KORDJAMSHIDI, P., MOENS, M. F., AND BETHARD, S. Semeval-2013 task 3: Spatial role labeling. In *Proceedings of the seventh international workshop on semantic evaluation* (2013), pp. 255–262.

[38] KORDJAMSHIDI, P., BETHARD, S., AND MOENS, M.-F. Semeval-2012 task 3: Spatial role labeling. In *Proceedings of the Sixth International Workshop on Semantic Evaluation* (2012), vol. 2, ACL, pp. 365–373.

[39] KUHN, W. Core concepts of spatial information for transdisciplinary research. *International Journal of Geographical Information Science 26*, 12 (2012), 2267–2276. 10.1080/13658816.2012.722637.

[40] LEE, J., AND KANG, M. Geospatial big data: Challenges and opportunities. *Big Data Research 2*, 2 (2015), 74–81. 10.1016/j.bdr.2015.01.003.

[41] LEIDNER, J. L., MARTINS, B., MCDONOUGH, K., AND PURVES, R. S. Text meets space: Geographic content extraction, resolution and information retrieval. In *Advances in Information Retrieval* (2020), Springer International Publishing, pp. 669–673. 10.1007/978-3-030-45442-5_89.

[42] LI, H., VASARDANI, M., TOMKO, M., AND BALDWIN, T. Target word masking for location metonymy resolution. In *Proceedings of the 28th International Conference on Computational Linguistics* (2020), pp. 3696–3707. 10.18653/v1/2020.coling-main.330.

[43] LI, H., WANG, M., BALDWIN, T., TOMKO, M., AND VASARDANI, M. Unimelb at semeval-2019 task 12: Multi-model combination for toponym resolution. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (2019), pp. 1313–1318. 10.18653/v1/s19-2231.

[44] LINDEMANN, M., GROSCHWITZ, J., AND KOLLER, A. Compositional semantic parsing across graphbanks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019), pp. 4576–4585. 10.18653/v1/p19-1450.

[45] LINDEMANN, M., GROSCHWITZ, J., AND KOLLER, A. Compositional semantic parsing across graphbanks. In *Proceedings of the 57th Conference of the Association for Computational Linguistics* (2019), pp. 4576–4585. 10.18653/v1/p19-1450.

[46] LOBRY, S., MARCOS, D., MURRAY, J., AND TUIA, D. Rsvqa: Visual question answering for remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing 58*, 12 (2020), 8555–8566. 10.1109/TGRS.2020.2988782.

[47] LYU, C., AND TITOV, I. AMR parsing as graph prediction with latent alignment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (2018), pp. 397–407. 10.18653/v1/P18-1037.

[48] MAI, G., JANOWICZ, K., ZHU, R., CAI, L., AND LAO, N. Geographic question answering: Challenges, uniqueness, classification, and future directions. *arXiv preprint arXiv:2105.09392* (2021).

[49] MIDDLETON, S. E., MIDDLETON, L., AND MODAFFERI, S. Real-time crisis mapping of natural disasters using social media. *IEEE Intelligent Systems 29*, 2 (2013), 9–17. 10.1109/MIS.2013.126.

[50] MISHRA, A., MISHRA, N., AND AGRAWAL, A. Context-aware restricted geographical domain question answering system. In *2010 International Conference on Computational Intelligence and Communication Networks* (2010), pp. 548–553.

[51] NGUYEN, D. Q., AND VERSPOOR, K. An improved neural network model for joint POS tagging and dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* (2018), pp. 81–91. 10.18653/v1/k18-2008.

[52] OEPEN, S., KUHLMANN, M., MIYAO, Y., ZEMAN, D., CINKOVÁ, S., FLICKINGER, D., HAJIC, J., AND URESOVÁ, Z. Semeval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation* (2015), pp. 915–926. 10.18653/v1/s15-2153.

[53] PENNINGTON, J., SOCHER, R., AND MANNING, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (2014), pp. 1532–1543. 10.3115/v1/d14-1162.

[54] PETERS, M. E., NEUMANN, M., IYYER, M., GARDNER, M., CLARK, C., LEE, K., AND ZETTLEMOYER, L. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2018), pp. 2227–2237. 10.18653/v1/n18-1202.

[55] 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* (2018), p. 7. 10.1145/3281354.3281362.

[56] PUSTEJOVSKY, J., KORDJAMSHIDI, P., MOENS, M.-F., LEVINE, A., DWORMAN, S., AND YOCUM, Z. Semeval-2015 task 8: Spaceeval. In *Proceedings of the 9th International Workshop on Semantic Evaluation* (2015), ACL, pp. 884–894. 10.18653/v1/s15-2149.

[57] SAMARINAS, C., AND TSOUMAKAS, G. WAMBy: An information retrieval approach to web-based question answering. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence* (2018), pp. 40:1–40:8. 10.1145/3200947.3201023.

[58] SANTOS, D., CARDOSO, N., CARVALHO, P., DORNESCU, I., HARTRUMPF, S., LEVELING, J., AND SKALBAN, Y. Getting geographical answers from Wikipedia: the GikiP pilot at CLEF. In *Working Notes for CLEF 2008 Workshop co-located with the 12th European Conference on Digital Libraries* (2008).

[59] SCHEIDER, S., AND JANOWICZ, K. Place reference systems: A constructive activity model of reference to places. *Applied Ontology 9*, 2 (2014), 97–127. 10.5555/2668256.2668257.

[60] SCHEIDER, S., NYAMSUREN, E., KRUIGER, H., AND XU, H. Geo-analytical question-answering with gis. *International Journal of Digital Earth* (2020), 1–14. 10.1080/17538947.2020.1738568.

[61] SOROKIN, D., AND GUREVYCH, I. Modeling semantics with gated graph neural networks for knowledge base question answering. In *Proceedings of the 27th International Conference on Computational Linguistics* (2018), Association for Computational Linguistics, pp. 3306–3317.

[62] TANG, L. R., AND MOONEY, R. J. Automated construction of database interfaces: Intergrating statistical and relational learning for semantic parsing. In *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora* (2000), pp. 133–141. 10.3115/1117794.1117811.

[63] THOMPSON, C. A., CALIFF, M. E., AND MOONEY, R. J. Active learning for natural language parsing and information extraction. In *Proceedings of the Sixteenth International Conference on Machine Learning* (1999), pp. 406–414.

[64] XU, H., HAMZEI, E., NYAMSUREN, E., KRUIGER, H., WINTER, S., TOMKO, M., AND SCHEIDER, S. Extracting interrogative intents and concepts from geo-analytic questions. *AGILE: GIScience Series 1* (2020), 23. 10.5194/agile-giss-1-23-2020.

[65] YANG, L., ZHANG, M., LIU, Y., SUN, M., YU, N., AND FU, G. Joint pos tagging and dependence parsing with transition-based neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 26*, 8 (2017), 1352–1358. 10.1109/TASLP.2017.2788181.

[66] YANG, M., LEE, D., PARK, S., AND RIM, H. Knowledge-based question answering using the semantic embedding space. *Expert Syst. Appl. 42*, 23 (2015), 9086–9104. 10.1016/j.eswa.2015.07.009.

[67] YIH, W., CHANG, M., HE, X., AND GAO, J. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing* (2015), pp. 1321–1331. 10.3115/v1/p15-1128.

[68] YOUNIS, E. M. G., JONES, C. B., TANASESCU, V., AND ABDELMOTY, A. I. Hybrid geo-spatial query methods on the semantic web with a spatially-enhanced index of dbpedia. In *Geographic Information Science - 7th International Conference* (2012), pp. 340–353. 10.1007/978-3-642-33024-7_25.

[69] ZELLE, J. M., AND MOONEY, R. J. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference* (1996), pp. 1050–1055.