Author/s:
Hamzei, E;Tomko, M;Winter, S

Title:
Translating Place-Related Questions to GeoSPARQL Queries

Date:
2022-04-25

Citation:
Hamzei, E., Tomko, M. & Winter, S. (2022). Translating Place-Related Questions to GeoSPARQL Queries. Www 2022 Proceedings of the ACM Web Conference 2022, pp.902-911. ASSOC COMPUTING MACHINERY. https://doi.org/10.1145/3485447.3511933.

Persistent Link:
https://hdl.handle.net/11343/311356

Ehsan Hamzei, Martin Tomko, and Stephan Winter. Translating Place-Related Questions to GeoSPARQL Queries. In Proceedings of the ACM Web Conference 2022 (WWW '22). ACM, 902–911.
This is a post-review, pre-copyedit version of the article.
The final version is available online at: https://doi.org/10.1145/3485447.3511933.

TRANSLATING PLACE-RELATED QUESTION TO GEOSPARQL QUERIES

Ehsan Hamzei, Martin Tomko, Stephan Winter
The University of Melbourne
Parkville
{ehsan.hamzei, tomkom, winter}@unimelb.edu.au

ABSTRACT

Many place-related questions can only be answered by complex spatial reasoning, a task poorly supported by factoid question retrieval. Such reasoning using combinations of spatial and non-spatial criteria pertinent to place-related questions is increasingly possible on linked data knowledge bases. Yet, to enable question answering based on linked knowledge bases, natural language questions must first be re-formulated as formal queries. Here, we first present an enhanced version of YAGO2geo, the geospatially-enabled variant of the YAGO2 knowledge base, by linking and adding more than one million places from OpenStreetMap data to YAGO2. We then propose a novel approach to translate the place-related questions into logical representations, theoretically grounded in the *core concepts of spatial information*. Next, we use a dynamic template-based approach to generate fully executable GeoSPARQL queries from the logical representations. We test our approach using the Geospatial Gold Standard dataset and report substantial improvements over existing methods.

***K*eywords** geographic question answering, place-based search, query generation, geospatial knowledge bases

1 Introduction

People frequently search on the Web seeking answers to pragmatic information needs, such as evaluating the number of facilities (i.e., pharmacies) in a suburb where they may want to move to. Such place-related questions often require the use of structured Web content (linked data). Consider the following question, taken from the Geospatial Gold Standard dataset [Punjani et al., 2018]:

Question: *How many pharmacies are in 200 meter radius of High Street in Oxford?*

Structured data stored in databases and knowledge bases are more suitable than unstructured text retrieval to answer such questions. Using structured data, Geographic Question Answering (GeoQA) systems can perform spatial (i.e., 200 meter radius) and non-spatial operations (i.e., count *how many*), and retrieve information based on specific criteria (e.g., *High Street in Oxford*).

The complementary role of structured data for Question Answering (QA) systems has been emphatically noted [e.g., Dimitrakis et al., 2019, Chen, 2014], yet answering natural language questions using structured geospatial data remains challenging. Similar to open-domain QA systems, GeoQA systems require the ability to generate structured queries from natural language questions. However, vagueness in place types both in terms of their boundaries (e.g., *downtown*), meaning (e.g., *bakeries* vs. *cafes*), and relations (e.g., *near*) makes this task significantly more difficult [Montello et al., 2003, Hollenstein and Purves, 2010].

Translating natural language questions to formal queries often involves two major steps: (1) parsing the questions into an intermediate structure, and (2) generating queries using the intermediate structure [Punjani et al., 2018]. In the first step, concepts such as place names are extracted from the natural language questions, and their relations are expressed in graph or tree data structures. In the second step, the extracted concepts are used to define variables, and their relations are translated into the targeted structured language(s) based on their predefined syntax.

A parsing method in a domain-specific QA should be linked to the domain concepts to ease the query generation step. In this paper, the object-based conceptualization of place [Purves et al., 2019, Kuhn et al., 2021] is used as the grounding to design a parsing method for place-related questions. We extend previous studies on analyzing place-related questions [Hamzei et al., 2020a, Xu et al., 2020] to capture the extracted concepts and relations. We use the state-of-the-art language models to identify concepts and their relations. We capture the results of the parsing step in a logical representation that is both machine- and human-readable. Next, we devise a dynamic approach to translate the parsed questions to GeoSPARQL queries. The dynamic approach uses templates for partial GeoSPARQL *constructs*, e.g., defining a concept, rather than a template for the *whole query*. We show how our approach leads to significant improvements in translating questions to GeoSPARQL queries in comparison to the previous works.

The novelty of the proposed method is twofold: (1) our method is grounded in geographic domain knowledge. Thus, instead of trying to fit a model on a dataset, we use our conceptualization to determine how concepts relate; (2) our method is reusable to translate the questions to other structured languages(s) with minimal efforts using logical representation that formally captures the gist of the place-related questions. In short, the paper:

- Improves existing methods in translating natural language question to GeoSPARQL queries over the Geospatial Gold Standard dataset [Punjani et al., 2018].
- Proposes an intermediate logical representation using the available domain knowledge that can be used to translate question to structured queries.
- Enriches an available knowledge base (YAGO2geo [Punjani et al., 2018]) by linking more than a million places from 500 place types using OpenStreetMap data.
- Uses the state-of-the-art language model (i.e., BERT embedding) to perform ontology mapping and evaluates their performance in mapping place types and properties.

2 Related Works

GeoQA is a sub-domain of Question Answering (QA) that focuses on generating answers to geographic questions [Ferrés Domènech, 2017, Mai et al., 2021]. Diverse information sources such as textual information [Ferrés and Rodríguez, 2006, Mishra et al., 2010], geodatabases [Chen et al., 2013], and spatially-enabled knowledge bases [Ferrés and Rodríguez, 2010] have been investigated to enable GeoQA. Answers to geographic questions can then be presented in natural language, structured into tables and graphs, or visualised as maps [Scheider et al., 2020, Chen, 2014, Ferrés Domènech, 2017].

Early research on the translation of geographic questions to structured queries links back to Zelle and Mooney [1996] and Tang and Mooney [2001], who designed a method to capture geographic questions in logical form. This logical form captured factual statements extracted from the questions in terms of objects and predicates. This early work had limited generality, restricting the diversity of geographic questions that could be captured to only those relating to a narrow set of types of geographic places (e.g., *cities*, *countries*, *states* and *rivers*). The resulting formalism only offered a simplistic set of twenty properties and predefined relations (predicates), including the ability to define objects, get their properties such as population and area, and for defining spatial and logical relations such as *capital-of* and *equal* [Zelle and Mooney, 1996, Tang and Mooney, 2001].

Later, Chen [2014] devised a method to translate geographic questions into *spatial SQL* queries. This method was only developed for four types of questions: (1) identifying coordinates of a place, (2) finding the distance between places, (3) finding the closest place to another place and (4) finding places in a predefined neighbourhood of another place. The method was based on annotated questions and predefined spatial SQL templates.

Recently, Punjani et al. [2018] introduced a Gold Standard dataset for translating geographic questions to GeoSPARQL queries. They developed a template-based approach to generate GeoSPARQL queries from geographic questions. Their approach includes two steps of natural language processing (information extraction and entity resolution), and a query generator to parameterize GeoSPARQL templates. The initial work of Punjani et al. [2018] was further refined by using a more comprehensive list of templates for geographic queries and diverse set of natural language processing toolkits [Punjani et al., 2021]. While small, the hand-curated set of questions proposed by Punjani et al. [2018] remains the only evaluation Gold Standard dataset for GeoQA.

Li et al. [2021] designed a parsing method using deep neural networks (DNN) to combine the preprocessing, information extraction, and relation identification steps into a DNN model. They produced GeoSPARQL queries out of the parsing results using a dynamic approach for query generation, yet the queries were not directly executable, as they lacked ontology mapping and concept identification. While recent studies in translating open-domain questions to structured queries [e.g., Xu et al., 2019, Lyu et al., 2020, Cheng et al., 2020] show that DNN models perform much better in comparison to traditional rule-based methods, the small Gold Standard dataset seems to be insufficient for training a DNN model. Consequently, the method proposed by Punjani et al. [2021] performed better over the same dataset.

While several studies explored how to model geographic questions, a proper intermediate representation of the questions that utilizes the available domain knowledge is still missing. It is specially important for GeoQA because the available datasets are small and without a theoretical grounding the coverage of proposed methods cannot be evaluated. Here, we utilize available domain knowledge to propose an intermediate representation that captures semantics of questions without considering the technical features of a destination query language. We then use this intermediate representation to translate questions to GeoSPARQL queries.

3 Preliminaries

Core concepts of spatial information, proposed by Kuhn [2012], include the sole base concept of *location*, and a set of content and quality concepts. The content concepts are *object*, *field* and *event*. A spatial object has an identity and is bounded in space (e.g., *Mount Everest*), while a spatial field represents a geographic phenomenon that encompasses the whole space but its magnitude may differ from one location to another (e.g., *terrain height*). Events are bounded not only in space but in time, and they may cause changes in spatial objects and fields (e.g., a *hurricane*). Finally, the quality concepts determine the granularity and value of spatial information.

Using the core concepts of spatial information, geographic 'places' are conceptualized as *spatial objects with socially-constructed identities* [Purves et al., 2019]. While geographic places include diverse types with varied and possibly heterogeneous characteristics [Hamzei et al., 2020b], this conceptualization captures geographic places at a high level of abstraction, without any bias or favor to specific place types. Purves et al. [2019] conceptualize places as spatial objects that:

- may have associated *properties*, and *relations*, incl. spatial;
- have a *location* and are bounded, yet their boundary may be fuzzy (e.g., *downtown*);
- may have 'parts' or 'aggregates';
- can participate in *events*, and their properties or relation may be subjected to changes;
- can be carved out of fields (e.g., *climate zones*).

Thus, to study place-related questions based on the object-based conceptualization, we must consider 'places', their 'location', other 'properties', 'relations', and also 'events'.

Hamzei et al. [2020a] proposed an encoding schema to analyze the content of place-related questions, which was later extended by Xu et al. [2020] to increase its capabilities for analyzing a wider range of geographic questions. This schema captures the syntactic structure of the questions by labelling the tokens and phrases in natural language questions using a predefined set of encoding classes. The essential elements of this encoding schema are:

- **place name** is a direct reference to a geographic place (e.g., *New York*, *Big Apple*).
- **place type** is a generic reference to a category of a taxonomy that captures places with similar functional, spatial and physical properties (e.g., *mountain*).
- **properties** describe diverse characteristics of places, and a place may be described by a set of criteria imposed on these properties (e.g., *population*, or *area*).
- **activities** are afforded by places, and places may be queried for their affordances (e.g., [a place] *to buy hardware*).
- **situations** are another way to describe places by reference to what is available or can be experienced there (e.g., [a place] *to see birds*), instead of what one can do there.
- **qualities** can be a quality of an activity, a situation or a property of place that narrows down the search domain for identifying relevant places (e.g., the *most populated city*, the *old building*, or the *best cafe*).
- **spatial relations** describe how places are located in a relative space, and includes a diverse set of topological (e.g., *inside*), directional (e.g., *north of*) and metric (e.g., *in 200 meter*) relations [Ligozat, 2013].

4 Questions to Queries

4.1 Dataset

We use the Geospatial Gold Standard dataset [Punjani et al., 2018] to test the proposed translation method. This dataset contains 200 place-related questions collected for translating questions into GeoSPARQL queries. This is a handcrafted dataset generated by students of an Artificial Intelligence course. The participants were instructed to formulate questions about geographic places that can be answered using information available in knowledge bases or through spatial analysis supported in GeoSPARQL. The questions are about geographic places in the United Kingdom and The Republic of Ireland.

Punjani et al. [2018] also provide a knowledge base that contains both thematic and spatial information about places in the UK and Ireland. They link detailed spatial information extracted from the OpenStreetMap (only natural features) and GADM (administrative levels) datasets to the YAGO knowledge base [Hoffart et al., 2013] which contains accurate thematic information extracted from DBPedia, GeoNames and WordNet [Biega et al., 2013].

We extended this dataset by adding more than one million places in the UK and Ireland that belong to roughly 500 place types to enrich the YAGO2geo that originally contains spatial information for natural and administrative places. We collected data using the OSM Overpass Turbo API[1], and used the strategy of Punjani et al. [2018] to link these spatial data to the YAGO knowledge base. The enriched dataset now contains accurate spatial information about point of interests (e.g., *tourist attractions*), amenities (e.g., *restaurants*), historic places (e.g., *historic monuments*), buildings (e.g., *schools*) and shops (e.g., *boutiques*)[2]. Thematic information about places also extended by addresses, contact numbers and websites, wherever they are provided[3].

4.2 Method

Figure 1 shows the workflow of the proposed method to translate place-related questions to structured queries[4]. The workflow starts with extracting encodings to identify and annotate place-related semantics. Next, the relations among the encodings are identified through grammatical parsing. Then, the encodings and their relationships are expressed in logical statements. Finally, the logical statements are translated into GeoSPARQL queries.

4.3 Encoding Extraction

The encoding classes proposed by [Hamzei et al., 2020a] are extended to capture events and event types, as well as dates and numbers. Logical operations such as *and*, *or*, *negation* and *comparison* are then added to the encoding schema. Table 1 shows the schema of encoding classes and their codes. Identifying the extended encodings enable us to support the identified concepts in the object-based conceptualization of place (i.e., location, place and event), and also support complex questions that contain multiple criteria.

To extract the encodings, the pre-trained models for fine-grained named entity recognition (NER) [Lample et al., 2016a] and part-of-speech tagging [Joshi et al., 2018] are used. The relation between the encoding classes and part-of-speech is then used to extract information from the questions as follows:

- **Noun encoding:** Noun phrases can be place names, place types, event names, event types or properties. Place names and event names are detected using a fine-grained NER [Lample et al., 2016b]. Generic nouns are identified using part-of-speech tagging. These noun phrases include place types and event types. Here, a look-up approach is used to test whether a generic noun refers to a place type or an event type. The lists of place types and event types are extracted from the OpenStreetMap tags and the YAGO ontology, respectively. The remaining unlabelled noun phrases are labelled as properties.
- **Verb encoding:** As proposed and tested by Hamzei et al. [2020a], sets of active and stative verbs can be used to differentiate situations from activities. Here, BERT representations of the words are used to derive cosine similarity of any identified verb to predefined sets of active and stative verbs [Hamzei et al., 2020a], and classify them as situations or activities based on maximum similarity.
- **Preposition encoding:** Preposition phrases can be related to spatial relations, temporal relations, and comparisons. If a preposition refers to place(s) (either by place names or *generic* place types), the preposition is

[1] `http://overpass-turbo.eu/`

[2] OSM data keys: `https://wiki.openstreetmap.org/wiki/Category:Key_descriptions`

[3] Available at `https://github.com/hamzeiehsan/Questions-To-GeoSPARQL`

[4] Demo is available at: `https://tomko.org/demo/`

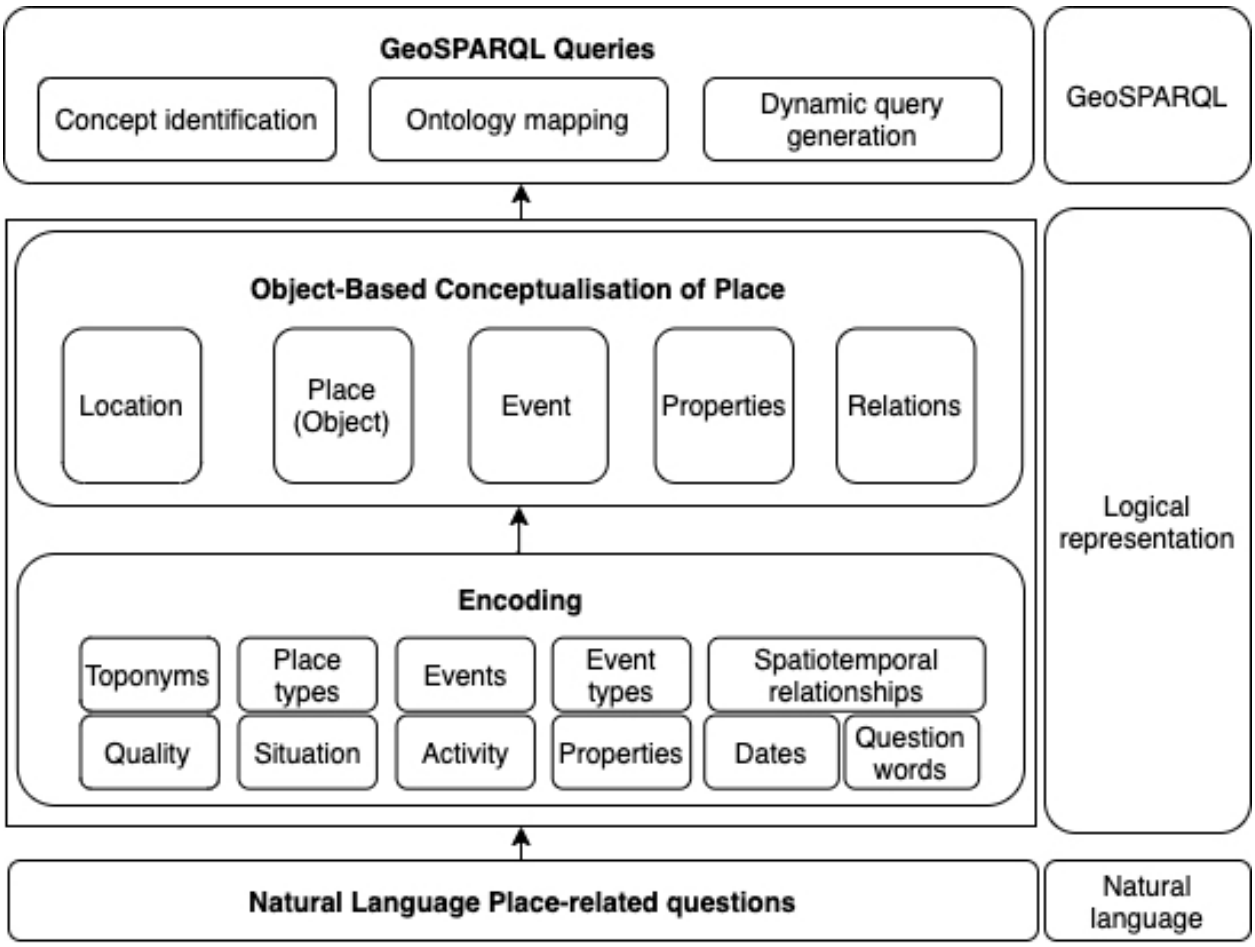

Figure 1: Method workflow to translate place-related question to GeoSPARQL queries.

Table 1: Encoding classes

Encoding class	Code	Encoding class	Code
where	1	*place name*	P
what	2	*place type*	p
which	3	*event*	E
when	4	*event type*	e
how	5	*properties*	o
how+adj	6	*activity*	a
why	7	*situation*	s
is/are	8	*spatial relation*	R
date	d	*temporal relation*	r
place quality	Q	*properties/events quality*	q
comparison	<, >, =	*and*	&
or	|	*negation*	!

captured as a candidate for spatial relations. If the preposition phrase refers to a date then the preposition is captured as a temporal relation. Otherwise, if the preposition phrase includes *comparative adjectives*, it is captured as a comparison (e.g., *greater than*). Here, constituency parsing [Joshi et al., 2018] is used to find what phrase a preposition refers to.

- **Adjective encoding:** Superlative (e.g., *smallest*) and descriptive (e.g., *small*) adjective phrases that are referring to place names and place types are captured as place qualities, and otherwise they are captured as property qualities. Comparative adjectives are a constituent of preposition phrases (e.g., *smaller than*) which are encoded as comparisons due to their different role in the questions.
- **Conjunction encoding:** Conjunctions are identified through part-of-speech tagging, and their encodings are identified through a look-up approach. The conjunctions are encoded either as *or*, *and*, or *negation*.

4.4 Constituency and Dependency Parsing

Constituency parsing and dependency parsing are two approaches to study the structure of natural language sentences. Constituency parsing [Joshi et al., 2018] captures how tokens can be combined to construct phrases and how phrases form more complex phrases or sentences (Figure 2). Dependency parsing [Dozat and Manning, 2017] identifies relations between tokens and captures their long distance relations (Figure 3). In our method, if a relation between identified concepts (e.g., places and events) can be extracted from phrases and their constituents, constituency parsing is used; otherwise we use dependency parsing to make sure the long distance relations are captured.

The parsing workflow starts with constituency parsing and identification of phrase-level information such as *conjunctions phrases*, *quality phrases* and *location phrases*. The following steps are performed in analysing constituency parsing results:

1. **Preprocessing:** In this step, the constituency parse tree is trimmed and encoded. During trimming, extracted compound phrases (e.g., High Street) are captured as leaf nodes in the tree representation. Hence, the constituency tree expands to a meaningful phrase level that is not necessarily consisting of individual tokens. Then, the extracted encodings are labelled in the tree representation.
2. **Conjunction phrases:** First, leaf nodes labelled with conjunction encodings (i.e., `&`, `|`, and `!`) are selected from the constituency tree. For *and*/*or* conjunctions, if the parent node includes constituents of the same encoding class (e.g., multiple place names), the parent node is labelled as a *conjunction phrase* – e.g., towns or cities in *what are the towns or cities in UK?*. For negations, if the parent nodes contain place names or event names the parent nodes are labelled as *negation phrase* – e.g., 'except London' in *What is the largest city in UK except London?*
3. **Quality phrases:** Leaf nodes encoded as qualities (i.e., `q` and `Q`) are retrieved from the constituency tree. If their parent nodes include nodes with relevant encoding classes (i.e., `p`, `P`, `e`, `E` and `o`), the parent nodes are captured as quality phrases.
4. **Location phrases:** Location phrases are constructed by spatiotemporal relations and their corresponding anchor places/dates – e.g., *in 200 meter radius of High Street*. Here, location is considered in space and time. Location phrases are captured by finding an ancestor node of spatiotemporal relations which includes the anchor places and dates. Figure 4 shows two labelled locations, *in Oxford* and *in 200 meter radius of High Street in Oxford*.
5. **Measure Phrases:** Phrases that are only constructed with cardinal numbers and properties/types are detected as measures phrases. In the introductory example, *200 meter* is a phrase with numeric (200) and property (meter) constituents.
6. **Comparison phrases:** Comparison is a binary relation with a source and a target. While the target and comparison tokens (e.g., *less than*) often form a phrase, the source can have a long distance relation with comparison tokens, depending on the structure of the questions. Hence, the relation between comparison tokens and the target is captured through constituency parsing, and later dependency parsing is used to detect the source of comparison. If the parent node of an identified comparison encoding includes a valid target phrase (i.e., places, events, properties, dates or measure phrases), the parent node is labelled as a comparison phrase (e.g., *more than ten districts*).

The results of detecting phrase-level information is shown in the labelled constituency tree in Figure 4.

Next, we extract the *intent* of the place-related questions using the following heuristics:

1. **Question word rule:** The question word gives an initial signal about the intent of the questions – e.g., *where*, *what* and *is/are*. For yes/no questions (*is/are*) not only intent is clear but the answer domain – a Boolean value

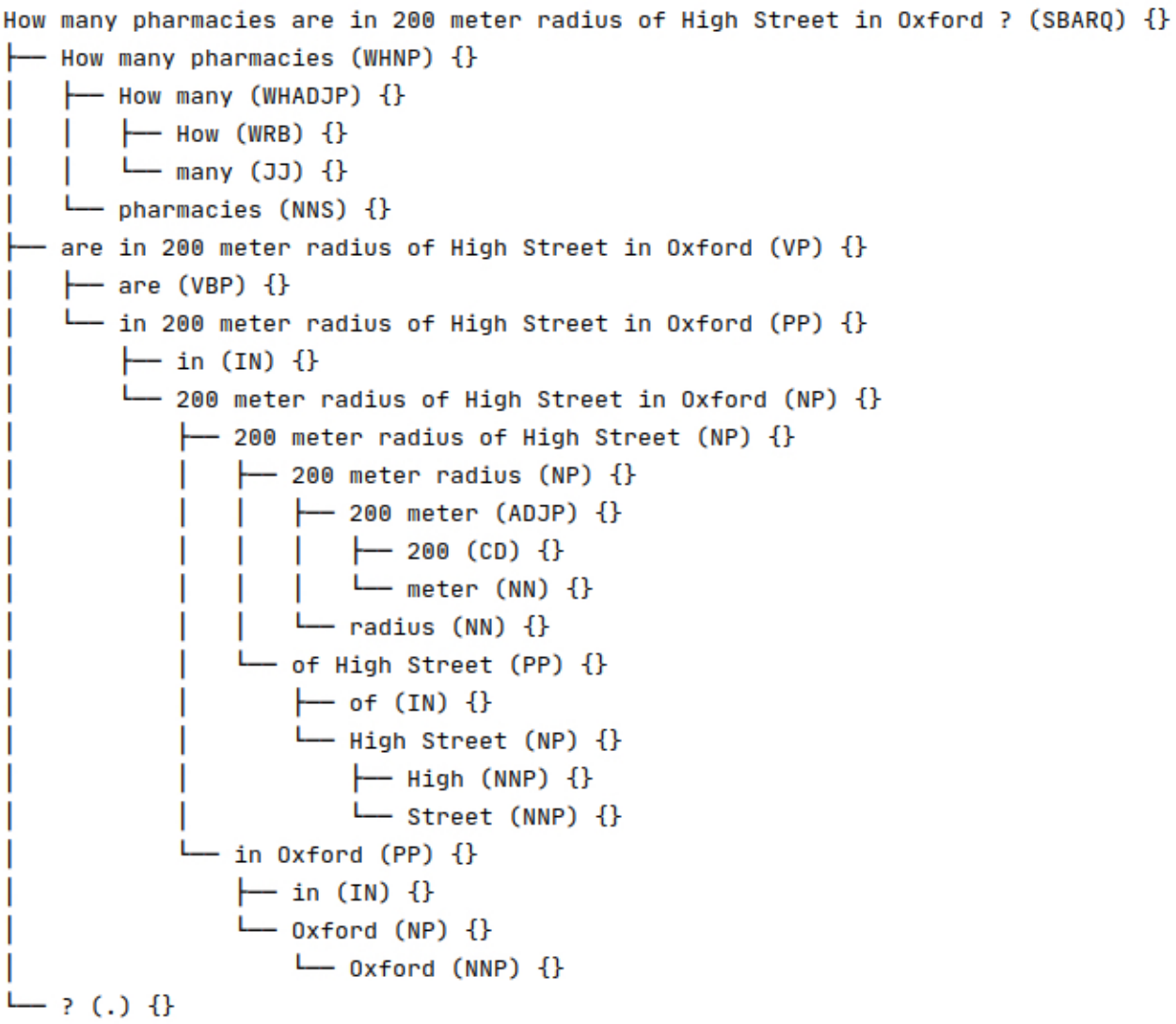

Figure 2: Constituency parsing

– is evident as well. For other question words, we need to identify the concept that is the intent of the question. For *where* questions, the intent is a location of a place or an event. *How+adjective* question words determine specific operations such as counting (e.g., how many) and distance (e.g., far) that must be applied to the intent concepts.

2. **Specificity rule:** The more specific the concepts are, the less likely they are to be the intent of the question — e.g., a *property* such as population or *place/event type* (e.g., *cafe*) is more likely to be the intent of a question in comparison to a *place/event 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., *properties*, *place/event types* and *place/event names*. The properties are the least specific concept. The next more specific concept is *place/event type*, and finally the most specific concept is *place/event name*.
3. **Phrase rule:** This rule determines whether a candidate concept is valid to be considered as the intent concept. If the candidates belong to place types or place names (e.g., multiple place types are identified), location phrases are used to reduce ambiguity. If a place type or place name is the intent, it must not belong to a *location phrase*. For example, in *Where in the UK is Wolverhampton?*, the UK is part of a location phrase (*in the UK*) and must be removed from the list of intent candidates. Such location phrases are spatial criteria, and must be

```
are (root) {['AUX']} []
├── How (advmod) {['ADV']} []
├── pharmacies (nsubj) {['NOUN']} []
│   └── many (dep) {['ADJ']} []
├── in (prep) {['ADP']} []
│   └── radius (dep) {['NOUN']} []
│       ├── meter (nn) {['NOUN']} []
│       │   └── 200 (dep) {['NUM']} []
│       └── of (prep) {['ADP']} []
│           └── Street (pobj) {['PROPN']} []
│               └── High (amod) {['PROPN']} []
├── in (prep) {['ADP']} []
│   └── Oxford (pobj) {['PROPN']} []
└── ? (dep) {['PUNCT']} []
```

Figure 3: Dependency parsing

avoided in the intent recognition process. In the same manner, properties which belong to activity/situation phrases, complex spatial relations and comparison phrases are removed from the candidate list.

4. **Phrase position rule:** Finally, if the intent is still ambiguous, and multiple valid concepts are identified, a position rule is used, favouring concepts appearing earlier in the question phrase. The subject of the questions is then the earliest concept extracted from the questions. For example, in the question: *Which river crosses the most cities in England?*, the earliest concept (*river*) is determined as the intent.

In the introductory example, the intent is therefore identified as *How many pharmacies* using these rules.

Next, dependency parsing is used to detect the relation between (1) places/events with location phrases, (2) situation/activities with properties, (3) places with situations/activities, and (4) comparison phrases and their source.

1. **Preprocessing:** The dependency parse tree is trimmed and enriched by the extracted encoding and the identified phrases from constituency parsing. In the trimming task, each of the extracted compound phrases (e.g., *High Street*) or phrase-level information from constituency parsing is captured as a single node in the dependency tree representation. Figure 5 shows the dependency tree of the introductory example after the preprocessing step.
2. **Places/events and location phrases:** Using the terminology introduced by Vasardani et al. [2013], a spatial description includes three elements, the *locatum*, the *spatial relation*, and the *relatum*. A location phrase is a combination of the spatial relation and the relatum that describe the location of a place or an event in space and time (e.g., *in 200 meter radius of High Street*). If the location phrase is a dependent (*prep* relation[5]) of a place or an event, their relation is captured as a locatum and location phrase relation (e.g., *pharmacies* (locatum), *in 200 meter radius of High Street* (location phrase)).
3. **Situations/activities with properties:** Situations and activities often include references to non-place objects that are captured as properties. In these cases, the verbs do not completely describe the situation and activities – e.g., to buy [coffee]. If a property is a dependent (an object phrase of the verb, *dobj* relation) of a situation/activity verb, then the relation is identified between the verb and property phrase.
4. **Places with activities/situations:** If a situation/activity phrase is a dependant (subject phrase of the verb, *nsubject* relation) of an identified place, their dependency is captured as a place relation with situations/activities. The same grammatical rule is applied for events and situation verbs (e.g., [...] *hurricanes occurred* [...]).

[5]more information can be found at: `https://universaldependencies.org/u/dep/`

```
How many pharmacies are in 200 meter radius of High Street in Oxford ? (SBARQ) {}
├── How many pharmacies (WHNP) {}
│   ├── How many (WH) {6}
│   └── pharmacies (NNS) {p}
├── are in 200 meter radius of High Street in Oxford (VP) {}
│   ├── are (VBP) {}
│   └── in 200 meter radius of High Street in Oxford (PP) {LOCATION}
│       ├── in 200 meter radius of (IN) {R}
│       │   ├── in (IN) {}
│       │   ├── 200 meter radius (NP) {}
│       │   │   ├── 200 meter (ADJP) {MEASURE}
│       │   │   │   ├── 200 (CD) {n}
│       │   │   │   └── meter (NN) {o}
│       │   │   └── radius (NN) {o}
│       │   └── of (IN) {R}
│       ├── High Street (NP) {P}
│       └── in Oxford (PP) {LOCATION}
│           ├── in (IN) {R}
│           └── Oxford (NP) {P}
└── ? (.) {}
```

Figure 4: Labelled constituency parse tree

```
are (root) {['AUX']} []
├── How many (advmod) {['ADV']} [6]
├── pharmacies (nsubj) {['NOUN']} [p]
├── in 200 meter radius of (prep) {['ADP']} [R]
│   └── High Street (pobj) {['NOUN']} [P]
├── in (prep) {['ADP']} [R]
│   └── Oxford (pobj) {['NOUN']} [P]
└── ? (dep) {['PUNCT']} []
```

Figure 5: Dependency parse tree after preprocessing

5. **Source of comparison phrases:** If a comparison phrase is a direct dependent of a valid encoding class (i.e., `p`, `P`, `e`, `E` and `o`), their relation is captured. In *does England have more counties than Ireland*, England has a valid encoding class (i.e., `P`) and the comparison phrase (i.e., *more counties than Ireland*) is its dependent in the parse tree.

4.5 Generating Logical Statements

Logical statements are constructed using *terms* and *functions*. In place-related QA, the terms are places, events or their properties. A term can be either a constant (e.g., *High Street*) or a variable (e.g., x that represents pharmacies). Place and event names are the constants and the types and properties are the variables terms. Referring to the introductory example, *High Street* and *Oxford* are the constants, and for the *pharmacies*, variable $x0$ is assigned to represent the generic reference. Functions are symbols that either declare terms or describe their relations – e.g., declaration:

$Place(HighStreet)$ and relation: $IN(HighStreet, Oxford)$. The logical statements are either query statements that return values of variable terms or Boolean True/False statements.

We declare terms using two special functions, $PLACE()$ and $EVENT()$, that describe place and event concepts. Generic references are declared using the assigned variables and the extracted type – e.g., $PHARMACY(x0)$. A generic declaration is either a place or an event using a predefined rule, e.g., $\forall x\ PHARMACY(x) \rightarrow PLACE(x)$.

Spatiotemporal relations, situation/activity relations, comparisons, and qualities are defined as functions. Qualities describe a single term, and are represented by a function with an argument. Spatial relations are either binary or ternary functions. The name of the functions are the spatial prepositions, with the identified *locatum* and *relatum* as the two arguments. In case of complex spatial relations, the additional information is presented as the third argument. In the example, the complex relation is represented as $IN\_RADIUS\_OF(x0,\ High\ Street,\ 200\ meter)$. Comparisons are also defined as binary relations with the comparison source as the first argument and target as the second argument.

The extracted conjunction relations are treated differently when generating logical statements. If a constituent of a conjunction participates in relations to other terms, then automatically a similar relation is applied to the other constituent of the conjunction. For example in *Are there any rivers that cross both England and Wales?*, the spatial relations between the rivers and *England* is replicated for *Wales* because of their conjunction relations – i.e., $CROSS(x0, England) \wedge CROSS(x0, Wales)$.

The logical representation is formulated using the identified intent and concatenating declarations and functions. If the question is a yes/no question, then the statement is simply derived by appending term declarations and function definitions. Otherwise, the logical statement is a query statement that returns the intended terms (e.g., $x0$) or functions (e.g., $COUNT(x0)$). The logical statement for the introductory question is presented as:

$$\begin{aligned} &COUNT(x0) : PLACE(High\ Street) \wedge PLACE(Oxford) \\ &\wedge PHARMACY(x0) \wedge IN\_RADIUS\_OF(x0, High\ Street, 200\,meter) \\ &\wedge IN(High\ Street, Oxford) \end{aligned} \tag{1}$$

4.6 Generating GeoSPARQL Queries

To generate GeoSPARQL queries, two necessary steps are needed: 1) concept identification and ontology mapping, to match extracted information to available information in the knowledge base; and 2) the dynamic generation of GeoSPARQL queries.

Concept Identification and Ontology Mapping: Concept identification is completed prior to query generation. We use Apache Solr to index the names and identifiers of places and events in the knowledge base and perform string matching using Solr search. Solr improves the performance of concept identification through its powerful string indexing. While concept identification could be undertaken on the query itself, string matching can take long on large knowledge bases.

We consider a one-to-many mapping to match extracted place/event types and properties to the knowledge base ontology. Our workflow performs ontology mapping in three sequential steps: 1) an *exact matching* from extracted information to the knowledge base ontology, followed by 2) a *label matching* using cosine similarity between the contextual BERT representations [Kenton and Toutanova, 2019] of the extracted information from the questions and the labels in our ontology; and finally 3) a *glossary matching* using cosine similarity between the BERT representations of the definitions for the extracted information from WordNet and Wikipedia snippet search with glossary in our ontology. Both label and glossary matching are based on thresholds that are tuned using randomly selected and manually mapped types and properties (10% of the available set).

Query Generation: 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. We use a set of predefined prefixes that includes GeoSPARQL functions, YAGO and YAGO2geo ontologies and resources. The ASK/SELECT statements determine the output of the query, and the WHERE clauses capture the criteria mentioned in the question.

We propose three sequential steps to translate logical statements to GeoSPARQL queries: 1) the overall structure of a query (i.e., ASK vs. SELECT query) is determined from the extracted intent. In case of SELECT queries, the intent of questions determines what variables are queried for their values; 2) the WHERE-clause is dynamically generated by concatenating individual concept and relation definition statements; and finally 3) sorting and aggregation (ORDER-BY and GROUP-BY clauses) are generated for queries that require these. Details of the predefined templates are presented in Appendix A.

The WHERE-clause is part of the general structure for both ASK and SELECT queries. To generate the WHERE-clause, a unique variable is assigned to each extracted geographic concept. Each of these concepts is defined using the

Table 2: Extraction performance

Encoding class	Average precision	Average recall	Average f-score	Count
Place name	98.9	99.5	99.2	263
Place type	100.0	99.4	99.7	178
Event name	–	–	–	0
Event type	–	–	–	0
Properties	95.9	97.9	96.9	55
Number	100.0	100.0	100.0	26
Spatiotemporal relation	100.0	96.8	98.4	191
Situation	94.2	98.0	96.1	51
Activity	50.0	100.0	66.7	1
Place quality	100.0	91.7	95.7	25
Object quality	85.7	100.0	92.3	6
Comparison	100.0	100.0	100.0	24
Conjunction	100.0	100.0	100.0	6

predefined templates and a corresponding variable name. The concept definition statements define a place/event based on the *name* or *type*. The extracted *properties* are defined based on their corresponding place/event variables. The extracted relations among the concepts (e.g., *spatiotemporal relations*) are translated to GeoSPARQL query using the participating variables, and the predefined relation templates.

Finally, when aggregation (e.g., counting) or sorting (e.g., superlative qualities) is needed their corresponding templates are used. 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. Sorting is needed when superlative qualities (e.g., longest river) are found, and aggregation is necessary when comparison-based situations (e.g., cities that have more than 10 suburbs) are extracted. The generated GeoSPARQL query for the introductory example is shown in Query 7 (see Appendix A).

5 Results and Discussion

5.1 Extraction Results and Logical Statements

Table 2 shows the results of evaluating encoding extraction using Geospatial Gold Standard dataset [Punjani et al., 2018]. The macro averaging strategy is used to derive the precision, recall and f-score of encoding extraction for each encoding class. The count shows the frequency of each class in the question dataset based on manual annotation.

As shown in Table 2, the pre-trained models perform well in place name identification, and the part-of-speech rules are successful in identifying numbers, comparisons and conjunctions. The dependency parsing and constituency parsing are highly successful in identifying more complex encoding classes such as spatiotemporal relations and qualities.

Table 2 shows that event-based place-related questions are completely missing in the dataset. Moreover, activities are also rarely observed in the dataset, which is another limitation of the Geospatial Gold Standard dataset [Punjani et al., 2018]. Identifying such limitations shows that by using the object-based conceptualization our method is less prone to the biases in the dataset.

Table 3 shows the average precision, recall and f-score of logical term declarations and function definitions. The results show that declarations, spatiotemporal relations and comparisons can be formulated in logical statements with high precision and recall. Thus, the combination of grammatical rules from dependency parsing and constituency parsing are shown to be successful.

Table 3 shows that qualities are defined with high precision (100%), yet the recall is lower (82.8%). The reason is that using constituency parsing alone may not be sufficient for all cases. For example in *Which site of Manchester is the most popular?*, the long distance relation between the quality (*most popular*) and the place type it describes (*site*) cannot be captured through constituency parsing. Similarly for situations, we observe higher precision and lower recall.

Table 3: Evaluation of the generated logical statements

Term/Function/Statement	Average precision	Average recall	Average f-score
Declaration	98.2	99.7	98.9
Spatiotemporal relations	97.6	91.9	94.7
Quality	100.0	82.8	90.6
Comparison	91.7	91.7	91.7
Conjunction	80.0	100.0	88.9
Situation	100.0	64.7	78.6
Overall[6]	85.0	–	–

Table 4: Query generation results

Step	Average precision	Average recall	Average f-score
Concept identification	97.5	96.3	96.9
Ontology mapping	83.7	97.2	89.9
Query	79.5	–	–

The reason is that in some cases dependency parsing fails to link situation verbs to properties. Hence, when the link is not captured, the situation is missing and the recall is lower.

Conjunctions are reflected in the logical statements with high recall (i.e., 100%) and lower precision (i.e., 80%). The conjunctions are reflected in logical statements by applying the same functions for both sides of the conjunction which are bounded by either *and* or *or* logical operators. Hence, errors in defining other functions (e.g., spatial relations) can be propagated through conjunctions. Consequently, errors in these functions impact on the precision of the conjunction statements as well.

5.2 Query Generation Results

Table 4 shows the results of analyzing concept identification, ontology mapping and query generation. The concept identification is highly accurate, yet in this evaluation we did not consider toponym disambiguation. Hence, the minor issues in concept identification is due to incorrect or missing data in our knowledge base (e.g., an incorrect alternative name for a place). For questions that contain spatial relations between two place names, toponym disambiguation is automatically resolved when the query is executed. However, if we only have one place name in the question, every place with the same name may be considered as a correct match – simply, because of lack of additional context.

The results show that the ontology mapping method using BERT embedding performs well in matching extracted information to the predefined ontology. However, the evaluation of ontology mapping is sometimes subjective, specially when multiple knowledge sources are integrated. For example, whether 'clinic' can be a valid match for 'hospital' is subjective. Here, if the services associated to the place types are similar, the match is considered correct. Yet, further larger-scale evaluation studies are needed to measure human agreement on this task.

We identify the following reasons for the precision drop when generating queries from logical statements (drop from 85% to 79.5%)[7]:

- Flexibility of natural language: For example, *Scottish counties* can be captured in logical form, yet the adjective implies a spatial relation which must be stated in the corresponding GeoSPARQL query – i.e., *counties of Scotland*. In such cases, our rule-based approach fails to properly generate the query.
- Missing concepts and types: In some cases the place names and types that are mentioned in the questions are not found in the knowledge base, and thus query generation fails – e.g., *underground lines* cannot be matched to a place type in our knowledge base.

[7]Check Appendix B for the evaluation of generating GeoSPARQL constituents.

Table 5: Query generation results. "*" indicates that the results are only for correct queries, not the whole dataset.

	Template coverage	Correct query	Answers
GeoQA [Punjani et al., 2018]	43.0	22.0 (51.2% of 43% generated query)	37.4*
NeuralGQA [Li et al., 2021]	100.0	38.0 (71% of 45% generated query)	–
GeoQA [Punjani et al., 2021]	76.5	65.5%	64.6*
Our approach	**100.0**	**79.5**	**78.6***

5.3 Comparing with Previous Works

Table 5 shows the results of analysing generated GeoSPARQL queries in comparison to previous work. To the best our knowledge, three papers have proposed methods for translating questions to GeoSPARQL and used the same dataset. Table 5 shows a considerable improvement on the benchmark dataset in comparison to previous works [Punjani et al., 2018, 2021, Li et al., 2021].

In terms of the number of questions that can be handled, the methods presented by Punjani et al. [2018] and Punjani et al. [2021] cover 43% and 76.5% of the questions, respectively. Using dynamic query generation, our approach and [Li et al., 2021] were able to generate queries for all of the questions in the dataset. The precision of the answers is only reported by Punjani et al. [2021, 2018], and the comparison shows an improvement of 14% of precision in retrieving the answers, attributable to our method.

The comparison shows that integrating dependency parsing and constituency parsing is useful to identify the relations. The benchmark methods [Punjani et al., 2018, 2021] only use dependency parsing to extract information from the questions. Hence, using phrase level information from constituency parsing can improve the quality of information extraction.

Using the object-based conceptualization [Purves et al., 2019], our method covers more diverse types of place-related questions and consequently is able to handle all questions in the dataset. Moreover, the proposed logical representation has two main advantages which ease such translations. First, the representation is machine digestible and can also be empowered with logical reasoning. Second, it can capture both intent and criteria of the natural language questions which is also required in translating natural language questions to other structured query language(s).

6 Conclusions

In this paper, we present a method to translate place-related questions to queries using the object-based conceptualization of place [Purves et al., 2019]. Using the domain knowledge, the proposed method is less biased to the dataset and covers more diverse types of place-related questions, and we identified missing types of questions in the Geospatial Gold Standard dataset [Punjani et al., 2018] (i.e., questions about events and activities).

In our method, we use and test state-of-the-art pre-trained models, and the results shows that the quality of information extraction and relation identification is improved in comparison to the benchmark methods. Moreover, the grammatical rules derived from the dependency and constituency parsing lead to more accurate results in digesting place-related questions. However, the available dataset is relatively small and the method should be tested whenever larger datasets are available for GeoQA. Enriching the current benchmark dataset to cover questions about activities and events is also a necessary step for future work in translating place-related questions to queries. Using local context (e.g., user location) to present relevant and personalized answers to geographic questions remains as a future work of this study.

Acknowledgement

The support by the Australian Research Council grant DP210101156 is acknowledged.

References

J. Biega, E. Kuzey, and F. M. Suchanek. Inside yago2s: A transparent information extraction architecture. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13 Companion, page 325–328, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450320382. doi: 10.1145/2487788.2487935. URL `https://doi.org/10.1145/2487788.2487935`.

W. Chen. Parameterized spatial sql translation for geographic question answering. In *2014 IEEE International Conference on Semantic Computing*, pages 23–27, 2014.

W. Chen, E. Fosler-Lussier, N. Xiao, S. Raje, R. Ramnath, and D. Sui. A synergistic framework for geographic question answering. In *Proceedings of IEEE 7th International Conference on Semantic Computing*, pages 94–99, 2013.

L. Cheng, Z. Chen, and J. Ren. Enhancing question answering over knowledge base using dynamical relation reasoning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020. doi: 10.1109/IJCNN48605.2020.9207428.

E. Dimitrakis, K. Sgontzos, and Y. Tzitzikas. A survey on question answering systems over linked data and documents. *Journal of Intelligent Information Systems*, pages 1–27, 2019.

T. Dozat and C. D. Manning. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=Hk95PK9le`.

D. Ferrés and H. Rodríguez. 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*, MLQA '06, pages 69–76, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. ISBN 2-9524532-4-1.

D. Ferrés and H. Rodríguez. TALP at GikiCLEF 2009. In C. Peters, G. M. Di Nunzio, M. Kurimo, T. Mandl, D. Mostefa, A. Peñas, and G. Roda, editors, *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, pages 322–325, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15754-7.

D. Ferrés Domènech. *Knowledge-based and data-driven approaches for geographical information access*. PhD thesis, 2017.

E. Hamzei, H. Li, M. Vasardani, T. Baldwin, S. Winter, and M. Tomko. Place questions and human-generated answers: A data analysis approach. In P. Kyriakidis, D. Hadjimitsis, D. Skarlatos, and A. Mansourian, editors, *Geospatial Technologies for Local and Regional Development*, pages 3–19, Cham, 2020a. Springer International Publishing. ISBN 978-3-030-14745-7. doi: https://doi.org/10.1007/978-3-030-14745-7_1.

E. Hamzei, S. Winter, and M. Tomko. Place facets: a systematic literature review. *Spatial Cognition & Computation*, 20(1):33–81, 2020b. doi: 10.1080/13875868.2019.1688332.

J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2012.06.001. URL `https://www.sciencedirect.com/science/article/pii/S0004370212000719`. Artificial Intelligence, Wikipedia and Semi-Structured Resources.

L. Hollenstein and R. Purves. Exploring place through user-generated content: Using Flickr tags to describe city cores. *Journal of Spatial Information Science*, 1(1):21–48, 2010. doi: 10.5311/JOSIS.2010.1.3.

V. Joshi, M. E. Peters, and M. Hopkins. Extending a parser to distant domains using a few dozen partially annotated examples. In *ACL*, 2018.

J. D. M.-W. C. Kenton and L. K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

W. Kuhn. Core concepts of spatial information for transdisciplinary research. *International Journal of Geographical Information Science*, 26(12):2267–2276, 2012.

W. Kuhn, E. Hamzei, M. Tomko, S. Winter, and H. Li. The semantics of place-related questions. *Journal of Spatial Information Science*, (23):157–168, 2021. doi: 10.5311/JOSIS.2021.23.161.

G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. *ArXiv*, abs/1603.01360, 2016a.

G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, 2016b.

H. Li, E. Hamzei, I. Majic, H. Hua, J. Renz, M. Tomko, M. Vasardani, S. Winter, and T. Baldwin. Neural factoid geospatial question answering. *Journal of Spatial Information Science*, (23):65–90, 2021. doi: 10.5311/JOSIS.2021.23.159.

G. Ligozat. *Qualitative Spatial and Temporal Reasoning*. John Wiley & Sons, Inc., Hoboken, NJ, 2013. ISBN 9781118601457. doi: 10.1002/9781118601457.

Q. Lyu, K. Chakrabarti, S. Hathi, S. Kundu, J. Zhang, and Z. Chen. Hybrid ranking network for text-to-sql. *arXiv preprint arXiv:2008.04759*, 2020.

G. Mai, K. Janowicz, R. Zhu, L. Cai, and N. Lao. Geographic question answering: Challenges, uniqueness, classification, and future directions. *AGILE: GIScience Series*, 2:8, 2021. doi: 10.5194/agile-giss-2-8-2021. URL `https://agile-giss.copernicus.org/articles/2/8/2021/`.

A. Mishra, N. Mishra, and A. Agrawal. Context-aware restricted geographical domain question answering system. In *2010 International Conference on Computational Intelligence and Communication Networks*, pages 548–553, 2010.

D. R. Montello, M. F. Goodchild, J. Gottsegen, and P. Fohl. Where's downtown?: Behavioral methods for determining referents of vague spatial queries. *Spatial Cognition & Computation*, 3(2-3):185–204, 2003. ISSN 1387-5868. doi: 10.1080/13875868.2003.9683761.

D. Punjani, K. Singh, A. Both, M. Koubarakis, I. Angelidis, K. Bereta, T. Beris, D. Bilidas, T. Ioannidis, N. Karalis, et al. Template-based question answering over linked geospatial data. In *Proceedings of the 12th Workshop on Geographic Information Retrieval*, page 7, New York, NY, USA, 2018. ACM.

D. Punjani, M. Iliakis, T. Stefou, K. Singh, A. Both, M. Koubarakis, I. Angelidis, K. Bereta, T. Beris, D. Bilidas, T. Ioannidis, N. Karalis, C. Lange, D.-A. Pantazi, C. Papaloukas, and G. Stamoulis. Template-based question answering over linked geospatial data, 2021.

R. S. Purves, S. Winter, and W. Kuhn. Places in information science. *Journal of the Association for Information Science and Technology*, 70(11):1173–1182, 2019. doi: 10.1002/asi.24194.

S. Scheider, E. Nyamsuren, H. Kruiger, and H. Xu. Geo-analytical question-answering with gis. *International Journal of Digital Earth*, 0(0):1–14, 2020. doi: 10.1080/17538947.2020.1738568.

L. R. Tang and R. J. Mooney. Using multiple clause constructors in inductive logic programming for semantic parsing. In L. De Raedt and P. Flach, editors, *Machine Learning: ECML 2001*, pages 466–477, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. ISBN 978-3-540-44795-5.

M. Vasardani, S. Timpf, S. Winter, and M. Tomko. From descriptions to depictions: A conceptual framework. In T. Tenbrink, J. Stell, A. Galton, and Z. Wood, editors, *Spatial Information Theory*, pages 299–319. Springer International Publishing, 2013. ISBN 978-3-319-01790-7.

B. Xu, R. Cai, Z. Zhang, X. Yang, Z. Hao, Z. Li, and Z. Liang. NADAQ: Natural language database querying based on deep learning. *IEEE Access*, 7:35012–35017, 2019.

H. Xu, E. Hamzei, E. Nyamsuren, H. Kruiger, S. Winter, M. Tomko, and S. Scheider. Extracting interrogative intents and concepts from geo-analytic questions. *AGILE: GIScience Series*, 1:1 – 23, 2020. doi: 10.5194/agile-giss-1-23-2020.

J. M. Zelle and R. J. Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, page 1050–1055. AAAI Press, 1996. ISBN 026251091X.

A Predefined templates

Definition statements for places using their *name* and *type* are presented in Queries 1 and 2, respectively. Here, *PI* refers to variable name for place identifiers, and *URIS* refers to the results of concept identification in Query 1 and ontology mapping in Query 2. In place declarations, the geometry is always expressed using the Well-Known Text (WKT) encoding (*?<PI>GEOM*).

```
VALUES ?<PI>  {<URIS>}.
?<PI> geosparql:hasGeometry ?<PI>G .
<PI>G geosparql:asWKT ?<PI>GEOM .
```

Query 1: Place definition template using name

```
?<PI> rdf:type ?<PI>TYPE;
    geosparql:hasGeometry ?<PI>G .
?<PI>G geosparql:asWKT ?<PI>GEOM .
VALUES ?<PI>TYPE {<URIS>} .
```

Query 2: Place definition template using type

Properties are captured through *has-a* relations by binding identified properties to the results of ontology mapping (see Query 3). In the case of spatial relations, a lookup table is used to map spatial prepositions to their corresponding spatial functions implemented in GeoSPARQL. The relations are constructed by the *predicate* and *argument(s)* which are captured in the corresponding function definition. Similarly, the situation/activities and comparisons are defined using

their corresponding templates from their definitions. The template for distance relations, as examples of the relation templates, is shown in Query 4.

```
VALUES ?<ATTRIBUTE> {<URIS>}.
?<PI> ?<ATTRIBUTE> ?<PROPERTY>.
```

Query 3: Attribute relation template

```
FILTER(geof:distance(?<PI1>GEOM, ?<PI2>GEOM, <UNIT>) < <DISTANCE>).
```

Query 4: Distance relation template

The template for aggregation and sorting are shown in Queries 5 and 6, respectively.

```
GROUP BY <VARIABLE>
HAVING (COUNT(*) > <LIMIT>)
```

Query 5: Aggregation template

```
ORDER BY <ASC/DESC> (<VARIABLES>) LIMIT <LIMIT>.
```

Query 6: Sorting template

```
PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX units: <http://www.opengis.net/def/uom/OGC/1.0/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX yago:<http://yago-knowledge.org/resource/>
PREFIX geont: <http://kr.di.uoa.gr/yago2geo/ontology/>
PREFIX yago2geo:<http://kr.di.uoa.gr/yago2geo/resource/>
SELECT DISTINCT (COUNT(distinct ?x0) as ?countx0)
WHERE {
VALUES ?c0 {yago2geo:OSM_HighStreet246
            yago2geo:OSM_HighStreet301}.
?c0 geosparql:hasGeometry ?c0G .
?c0G geosparql:asWKT ?c0GEOM .
VALUES ?c1  {yago:Oxford
            yago2geo:OSM_Oxford813}.
?c1 geosparql:hasGeometry ?c1G .
?c1G geosparql:asWKT ?c1GEOM .
?x0 rdf:type ?x0TYPE;
    geosparql:hasGeometry ?x0G .
?x0G geosparql:asWKT ?x0GEOM .
VALUES ?x0TYPE {yago:wordnet_drugstore_103249342
                geont:OSM_amenity_veterinary
                geont:OSM_amenity_pharmacy} .
FILTER(geof:distance(?x0GEOM, ?c0GEOM, units:meter) < 200).
FILTER (geof:sfContains(?c1GEOM, ?x0GEOM)). }
```

Query 7: GeoSPARQL query of the introductory example

B Evaluation of GeoSPARQL components

The performance for generating each part of the GeoSPARQL queries (e.g., WHERE clause) is presented in Table 6. All GeoSPARQL queries include intent (SELECT/ASK statement) and criteria (WHERE clause). Hence, the recall and f-score is meaningless in evaluating these parts (i.e., 100% recall). On the other hand, sorting (29 questions) and aggregation (6 questions) are applicable for specific questions, where recall and f-score are reported (Table 6).

Table 6 shows that the intent heuristic performs well and therefore the errors in formulating the criteria (WHERE clause) are the main source of incorrect GeoSPARQL queries. Criteria may include multiple components, and even a minor mistake in formulating a criterion in a WHERE clause produces an incorrect query.

Table 6: GeoSPARQL queries evaluation

GeoSPARQL query	Average precision	Average recall	Average f-score
Intent (SELECT/ASK)	97.5	–	–
Criteria (WHERE)	84.3	–	–
Aggregation (Group By)	100.0	83.3	91.5
Sorting (Order By)	92.9	89.7	91.2