A Spatially-Explicit Reinforcement Learning Model for Geographic Knowledge Graph Summarization

Bo Yan, Krzysztof Janowicz, Gengchen Mai, and Rui Zhu
STKO Lab
Department of Geography
University of California, Santa Barbara

Abstract

Web-scale knowledge graphs such as the global Linked Data cloud consist of billions of individual statements about millions of entities. In recent years, this has fueled the interest in knowledge graph summarization techniques that compute representative subgraphs for a given collection of nodes. In addition, many of the most densely connected entities in knowledge graphs are places and regions, often characterized by thousands of incoming and outgoing relationship to other places, actors, events, and objects. In this paper, we propose a novel summarization method that incorporates spatially-explicit components into a reinforcement learning framework in order to help summarize *geographic* knowledge graphs, a topic that has not been considered in previous work. Our model considers the intrinsic graph structure as well as the extrinsic information to gain a more comprehensive and holistic view of the summarization task. By collecting a standard dataset and evaluating our proposed models, we demonstrate that the spatially-explicit model yields better results than non-spatial models, thereby demonstrating that spatial is indeed special as far as summarization is concerned.

Keywords: Geographic Knowledge Graphs, Graph Summarization, Reinforcement Learning, Spatially-Explicit Models, Spatial Context

1 Introduction

Knowledge graphs and the technologies used to create them are intended to foster the creation, retrieval, and reuse of human and machine readable structured data about real world entities such as places, events, actors, and objects using a graph-based representation (Paulheim, 2017). Recent advances in related technology stacks such as knowledge graph-based question answering systems as well as the adoption by commercial companies have highlighted the success of knowledge graphs in both academia and industry. The formal representation of geographic feature types and their relationships has been a long standing interest of the GIScience community and geographic information has been recognized as a key component of (general purpose) knowledge graphs (Kuhn et al., 2014). In fact, a large number of entities in Wikidata[1] — a sister initiative of Wikipedia to create a repository for structured information — are spatial and dedicated geospatial knowledge graph hubs such as LinkedGeoData[2] contain billions of statements about geographic entities.

In theory, today's abundance of geographic data facilitates new research and more powerful question answering systems. From a more practical perspective, however, sifting through the data deluge becomes increasingly challenging. Ramscar et al. (2014) have shown that too much information may adversely influence our cognitive information-processing capacities and unavoidably result in lags and retrieval errors. As a result, researchers are working on ways to better present data for humans, such as interfaces and visualization tools to make knowledge graphs more user-friendly and more accessible for non-technical audiences. One novel area of study is *knowledge graph summarization*, namely selecting and identifying the property-value pairs that best represent the underlying entity from a large and convoluted graph (Cheng et al., 2011).

The idea of summarizing a knowledge graph in a way such that the subgraph retains the significant substructures and meaning, here prominent nodes and edges, of the original graph is intriguing. However, this task is entangled with a lot of challenges, especially in the geospatial domain. One challenge is related to the inherently complex structure of graph data. Unlike other commonly-used structures such as the 1D sequence of natural languages and the 2D grids of images, graph structures are peculiar in their own ways. For example, on the global level, two graphs can be isomorphic, i.e., have the same structure, while they have distinct representations (e.g., labeling and visual representations). On the local level, substructures such as homophily and structural equivalence (Grover and Leskovec, 2016) coexist in the graph as proxies to encode the underlying patterns. In addition, since most knowledge graphs follow the so-called Open World Assumption (OWA) – which implies that there are possibly missing statements/triples in the knowledge graph without having to assume that those missing statements do not hold true in reality – the original structure of the graph might not represent the complete information. This adds another layer of complexity.

As a result of the versatility and peculiarity of graph data, traditional methods that rely heavily on handcrafted features/rules (such as clustering coefficients and other graph summary statistics) for knowledge graph summarization are not sufficient enough because they do not generalize well. Another challenge is the subjectivity of the summarization criteria. The relative importance of a node (entity) or an edge (relation/property/predicate) in the knowledge graph is not universally defined and different application fields can interpret it differently. For instance, a knowledge graph

[1] https://www.wikidata.org
[2] http://linkedgeodata.org

that primarily models friendship relation among individuals may take advantage of the connectivity information (such as degrees, betweenness, closeness, or eigenvector centrality) to determine important nodes in the summarization process. On the contrary, to summarize the DBpedia[3] knowledge graph — a crowd-sourced community effort to extract structured information from various Wikimedia projects — where there are a large number of distinct relation types and the whole graph is densely connected, latent information embedded in the labels and abstracts of each entity and relation is required to determine the extent to which each component of the graph is related with one another in order to rank the relative importance. Besides the aforementioned challenges, *geographic* knowledge graph summarization has its distinct challenges. Given the inherent richness of geospatial semantics (Yan et al., 2017, 2018), geospatial components such as spatial contexts play a significant role in understanding spatial entities and their dependencies. However, existing (knowledge) graph summarization methods (Liu et al., 2018) are not tailored towards the geospatial domain thus neglecting such special components. For instance, a summary about Santa Barbara, CA is also always a partial summary of California. As humans we give special weight to the places where important historic figures were born even if they spent their entire life somewhere else. Hence, every summary of the city of Ulm, Germany, e.g., the first paragraph of its Wikipedia article, lists Albert Einstein as notable resident despite his family moving to Munich a year after his birth. For Munich in turn, his name is not prominently featured in the city's description. This may be related to the broader phenomenon of duration neglect (Fredrickson and Kahneman, 1993).

In light of this, we propose to adopt a reinforcement learning-based approach to explicitly incorporate spatial contextual information. Our method combines both intrinsic structure and extrinsic information to help summarize *geographic* knowledge graphs as most domain-agnostic work (Cheng et al., 2011; Thalhammer et al., 2012; Thalhammer and Rettinger, 2014; Pirrò, 2015; Bast et al., 2015; Song et al., 2018) fails to consider the inherent richness of geospatial semantics. In fact, we believe that there is no prior work about geographic knowledge graph summarization at all – despite places such as Vienna, Austria being represented by tens of thousands triples in modern knowledge graphs, and, hence, in desperate need for graph summarization. In order to strike the balance between diversity and uniformity in summarizing geographic knowledge graphs, our model utilizes the idea of distance decay and information entropy to determine the relatedness of different spatial/non-spatial entities.

By intrinsic structure, we mean the graph structure where each entity is connected by properties. We embrace the current trend of utilizing vector representations, namely translation-based embedding models (Bordes et al., 2013), to embed the structural information of knowledge graphs. The semantic information – by which we mean latent information encoded in natural language, and, hence, not directly available to structural analysis – of the knowledge graph is captured by the embeddings of entity and relation labels. For extrinsic information, we take advantage of the Wikipedia abstracts of different places (geographic entities) to guide our summarization process since these abstracts are exemplary summaries of each geographic entity produced by human ingenuity, and there is a clear tractable correspondence between Wikipedia articles and knowledge graphs (Auer et al., 2007; Bollacker et al., 2008; Vrandečić and Krötzsch, 2014). By combining reinforcement learning with knowledge graph embeddings, word embeddings, information theory, and spatial contexts, we aim to tackle the challenges mentioned above. Knowledge graph embeddings efficiently encode the hidden structure of the graph. Word embeddings facilitate the transmis-

[3] https://wiki.dbpedia.org/

sion of semantic information in the knowledge graph to the summarization process. Information theory together with the reinforcement learning framework (guided by Wikipedia summaries) is employed to partially mitigate the subjectivity issue that impacts knowledge graph summarization tasks. After all, Wikipedia abstracts provide relatively neutral (Nielsen, 2007; Greenstein and Zhu, 2012), curated, concise, and generic digests that highlight the distinctive and significant aspects of different places. Spatial contexts are used to help recover missing links in the geographic knowledge graph and uncover the hidden geospatial patterns.

The research contributions of this paper are as follows:

- We utilize Wikipedia summaries to guide the geographic knowledge graph summarization process using reinforcement learning. Instead of mostly relying on intrinsic information, such as node groups in grouping and aggregation-based approaches and the number of bits needed to describe the graph in bit compression-based approaches, our approach reaps the complementary strengths of intrinsic information from the graph structure and extrinsic knowledge using Wikipedia summaries by framing the task as a sequential decision making process that can be optimized using reinforcement learning.

- We account for the richness of geospatial semantics in geographic knowledge graphs and incorporate such information in the summarization process in order to better capture the relatedness of geographic entities and provide better results. We do so by following established GIScience methods, namely modeling distance decay, as well as from an information theoretic perspective.

- We create a dataset DBP369[4] that includes 369 place summaries from Wikipedia and a subgraph of DBpedia for geographic knowledge graph summarization tasks and make it openly available. A lack of standard datasets has been one of the obstacles that hinder research development in the area of geographic knowledge graph summarization and to some degree geographic information retrieval in general. By taking the initiative to collect this dataset, we hope it will foster further research in this area.

- We establish different baselines for the geographic knowledge graph summarization task for the DBP369 dataset. Our result shows that by considering spatial contextual components the summarized graph better resembles the Wikipedia summary.

- Finally, to the best of our knowledge this is the first research to consider the problem of geographic knowledge graph summarization. This is remarkable as Web-scale knowledge graphs such as Linked Data store tens of millions of places and often thousands of statements (subject-predicate-object triples) about them.

The remainder of this paper is organized as follows. Section 2 summarizes existing work on knowledge graph summarization, spatially-explicit models, and utilizing reinforcement learning in the context of knowledge graphs. Section 3 describes the basic procedure of our data collection and provides detailed information about the DBP369 dataset. Section 4 explains the proposed method for geographic knowledge graph summarization. Section 5 applies the model to the DBP369 dataset and evaluates the results. Section 6 concludes the research and points to directions for future work.

[4] http://stko.geog.ucsb.edu/gkg/

2 Related Work

Most graph summarization techniques fall into one of the four categories (Liu et al., 2018) namely: grouping or aggregation-based approaches, bit compression-based approaches, simplification or sparsification-based approaches, and influence-based approaches. Knowledge graph summarization usually adopts the simplification or sparsification-based approach for the reason that the prime motivation for summarizing knowledge graphs is to provide a subgraph that highlights the important entities and relations of the original graph. Cheng et al. (2011) and Thalhammer and Rettinger (2014) proposed to utilize the graph structure and performed PageRank to identify relevant entities and summarize the graph. Pirrò (2015) formalized the notion of relatedness in knowledge graphs to better harness the large variety of information. While these papers primarily take advantage of the intrinsic information of knowledge graphs, some work is geared towards extrinsic knowledge. For instance, Bast et al. (2015) utilized textual information from Wikipedia to build logistic regression and generative models to calculate relevance scores for relations in knowledge graph triples. Our work takes the best of both worlds by considering intrinsic knowledge graph structure and extrinsic information simultaneously.

In addition, all the work mentioned above aims at retrieving/ranking entities/relations based on certain criteria such as relevance scores with respect to a user's queries rather than providing a subgraph that captures the essence of the original graph. Our work provides a subgraph that summarizes the relations and connected entities for each geographic entity based on corresponding Wikipedia abstracts. With the recent trend towards learning latent representations of graphs (Hamilton et al., 2017), methods based on matrix factorization strategies (such as Singular Value Decomposition (SVD), CUR (Drineas et al., 2006), and Compact Matrix Decomposition (CMD) (Sun et al., 2007)) have been used in which low-rank approximations of adjacency matrices are viewed as sparse approximation summaries of the original graphs. Our work embraces the idea of adopting a more scalable neural network-based approach, namely the TransE (Bordes et al., 2013) model, to learn low-dimensional latent knowledge graph representations and applying these embeddings within our summarization pipeline.

In order to study the influence of geospatial contexts on identifying different types of places, Yan et al. (2017) proposed a latent representation learning method based on augmented spatial contexts. Similarly, Yan et al. (2018) used spatial sequence patterns of neighborhoods as Bayesian priors and combined them with state-of-the-art convolutional neural network models to help improve image classification for different place types using data collected from Yelp and Google Street View. Mai et al. (2019) incorporated geographic weights into the latent representation learning process in order to provide better knowledge graph embeddings for geographic question answering tasks. Our work, follows the same line of reasoning, namely that *spatially-explicit models* substantially outperform more general models when applied to geographic data. Kejriwal and Szekely (2017) presented a geospatial data source generated using weighted neural embeddings methods on Geonames[5] data. The resulting embeddings encode geographic contextual information.

Researchers working on knowledge graphs have been exploring different ways in which reinforcement learning can be used. For example, Xiong et al. (2017) adopted the REINFORCE (Monte Carlo Policy Gradient) algorithm (Williams, 1992) to make a policy-based agent learn multi-hop relational paths for knowledge graph reasoning tasks by considering accuracy, diversity,

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

and efficiency in their reward function. Das et al. (2017) framed the knowledge graph reasoning task as a finite horizon, deterministic partially observable Markov Decision Process (MDP) and designed a randomized non-stationary history-dependent policy parameterized by a long short-term memory network (LSTM) (Hochreiter and Schmidhuber, 1997). Shen et al. (2018) developed the M-Walk graph-walking agent using recurrent neural network (RNN) to encode the history of the walked path and Monte Carlo Tree Search (MCTS) with a neural policy to generate trajectories yielding more positive rewards to overcome the challenge of sparse rewards under the off-policy Q-learning framework for knowledge graph completion. However, none of these approaches used a geographic dataset. Moreover, our work is based on the novel idea of treating the geographic knowledge graph summarization task as an MDP and the decision at each summarization step is made by the reinforcement learning agent.

3 Dataset

Given the lack of existing work on geographic knowledge summarization and related benchmarks, we collected the dataset DBP369 for our research and hope it can be adopted in similar research studies in the future. Previous research efforts that explored similar datasets focused on city networks (Salvini and Fabrikant, 2016; Zhang and Thill, 2019). We initially picked 500 places from different areas of the world, as shown in Fig. 1. In this work, we would like to explore the possibility of guiding the summarization process of geographic knowledge graphs by means of unstructured human knowledge. There are two parallel parts of our dataset: 1) Wikipedia summaries of each of these places, 2) A geographic knowledge graph containing each of these places and their related entities. These places include well-known metropolitan areas such as New York City

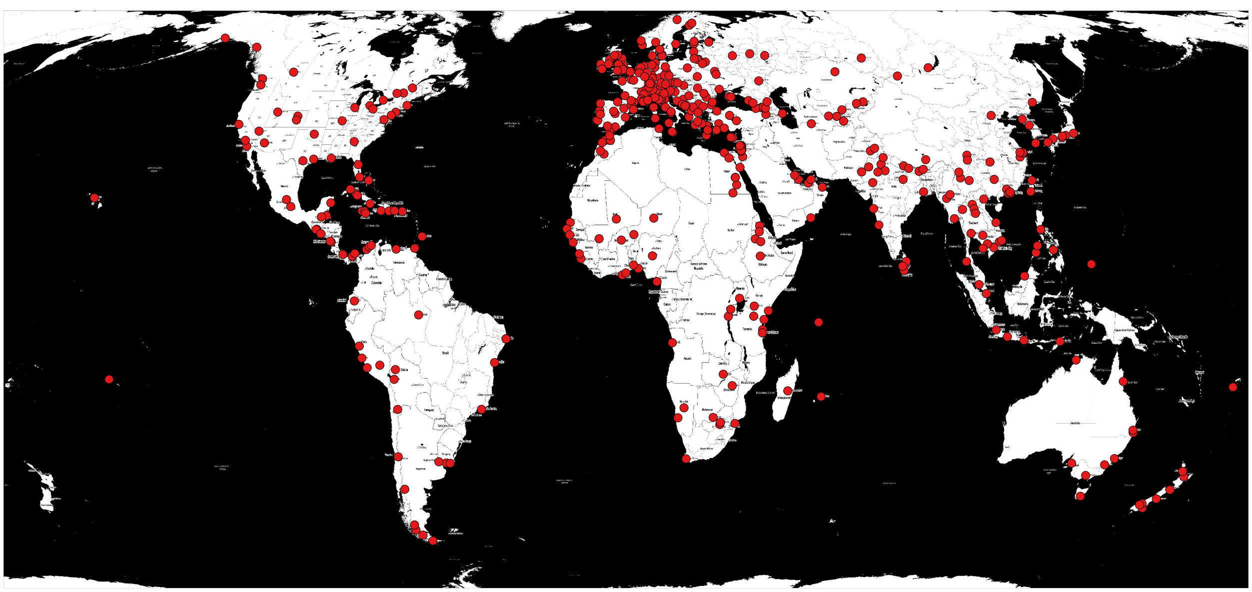

Figure 1: Place distribution map (Eckert IV).

and Los Angeles as well as areas with archaeological and historic significance such as Olympia, Greece. We used the MediaWiki API[6] to find the corresponding Wikipedia pages for these places, from which summary texts were extracted. These summaries provide a human-generated guidance for summarizing geographic knowledge graphs.

For the geographic knowledge graph part, we selected DBpedia as our data source, as it has numerous geographic entities, is being actively maintained and updated, has a clear one-to-one correspondence for each Wikipedia article, and provides a diversified and comprehensive coverage of properties. In order to construct our geographic knowledge graph from DBpedia, we prepared these 500 places from Wikipedia and retrieved all links that appeared in the summaries of these 500 articles. We generated mappings to find the corresponding entities for these places as well as the links. After obtaining these *seed* entities, we generated SPARQL[7] queries to retrieve 1-degree and 2-degree neighbors iteratively in order to form subgraphs surrounding these seed nodes. In DBpedia all statements are organized as *(head, relation, tail)* or *(subject, predicate, object)* triples. Query 1 shows an example query that uses a basic graph pattern to obtain 1-degree (both incoming and outgoing) neighboring nodes of DBpedia entity *dbr:Los_Angeles*.

```
PREFIX  dbr: <http://dbpedia.org/resource/>
SELECT DISTINCT * WHERE {{
dbr:Los_Angeles ?p1 ?o .
FILTER(CONTAINS(str(?p1),'http://dbpedia.org/ontology/') && !isLiteral(?o))}
UNION {
?s ?p2 dbr:Los_Angeles .
FILTER(CONTAINS(str(?p2),'http://dbpedia.org/ontology/') && !isLiteral(?s))}}
```

Listing 1: An example SPARQL query for retrieving the 1-degree neighbors for *dbr:Los_Angeles*, using it as both the head (subject) and the tail (object) entity.

We only considered relations with prefix *http://dbpedia.org/ontology/* since these mapping-based relations have a much higher quality. For the purpose of our modeling strategy, we further removed duplicate triples/statements and filtered out entities that appeared less than 10 times. In the end we obtained a dataset that contains 369 Wikipedia place summaries and a DBpedia subgraph that connects these 369 place entities with various other spatial and non-spatial entities, e.g., historical figures, via different relations, thus forming our geographic knowledge graph.

For the 369 places, the average length for the Wikipedia summary is 299 words and each summary on average contains 28 links. For the geographic knowledge graph, there are all together 419,579 entities, 534 unique relations, and 3,248,715 triples/statements. The data is split into a training set of 334 places and a test set including 35 places. Fig. 2 shows a slice of our dataset. The text in the middle is part of the summary for *Los Angeles (dbr:Los_Angeles)* and the graph surrounding the text illustrates the way in which different entities are connected with each other. We highlight the correspondence between the links in the summary and DBpedia entities.

4 Methods

In this section, we introduce our spatially-explicit reinforcement learning method. Instead of pruning the graph as explored by previous studies (Song et al., 2018), we decide to tackle the problem

[6] https://en.wikipedia.org/w/api.php

[7] https://www.w3.org/TR/rdf-sparql-query/

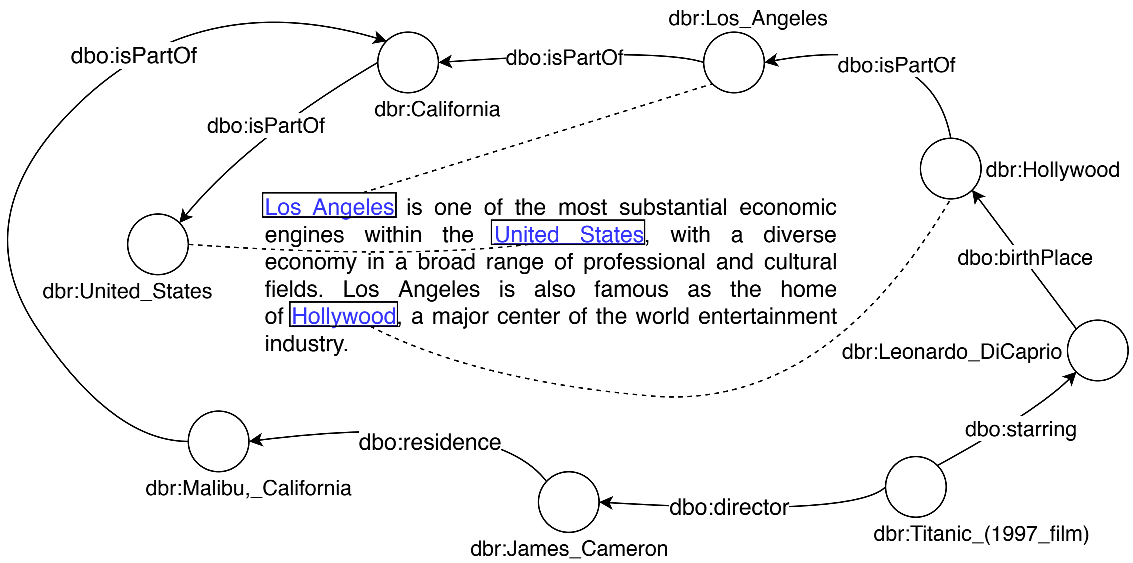

Figure 2: Three links *Los Angeles*, *United States*, and *Hollywood* in this text are mapped to three entities *dbr:Los_Angeles*, *dbr:United_States*, and *dbr:Hollywood* respectively. By retrieving the 1-degree and 2-degree neighbors of these entities, we are able to find their connections as well as information about other related entities.

in a reverse manner. We formulate the task as a sequential decision making problem where we start from the simplest graph, namely a single node (the geographic entity in question), and iteratively propose to make the graph more complex and expressive by sequentially adding new relations (edges) and entities (nodes) through trial and error until the graph representation closely resembles Wikipedia's textual summary. We first introduce the reinforcement learning model by explaining the basic components such as the environment, agent, actions, states, and rewards. Our policy-based agent learns to pick meaningful relations by interacting with the geographic knowledge graph environment. Then we describe the training pipeline where the model is first trained on a supervised policy followed by being retrained using the reward function.

4.1 Reinforcement Learning Framework

The geographic knowledge graph summarization task is formalized as a Markov Decision Process (S, A, P_a, R_a) where two components, namely the environment and the agent, interact with each other, as shown in Fig. 3. $S = \{s_1, s_2, ..., s_n\}$ is a set of states that contains useful information from the history of the MDP. $A = \{a_1, a_2, ..., a_n\}$ is a set of actions that the agent can take for the state provided by the environment. Because of the memorylessness of the MDP, the state transition probability matrix $P_a(s, s') = \Pr(s_{t+1} = s' | s_t = s, a_t = a)$ represents the probability of reaching state s' at time $t + 1$ after the agent takes action a in state s at time t. $R_a(s, s')$ is the immediate reward after taking action a and transitioning from state s to state s'.

To intuitively understand the process, let us suppose the MDP starts with a graph that is composed of the place entity itself and the Wikipedia summary of the place. At each step, the agent analyzes the current state (by considering information about the graph as well as the Wikipedia summary) of the process and decides to add one of the possible relations to the graph to grow it

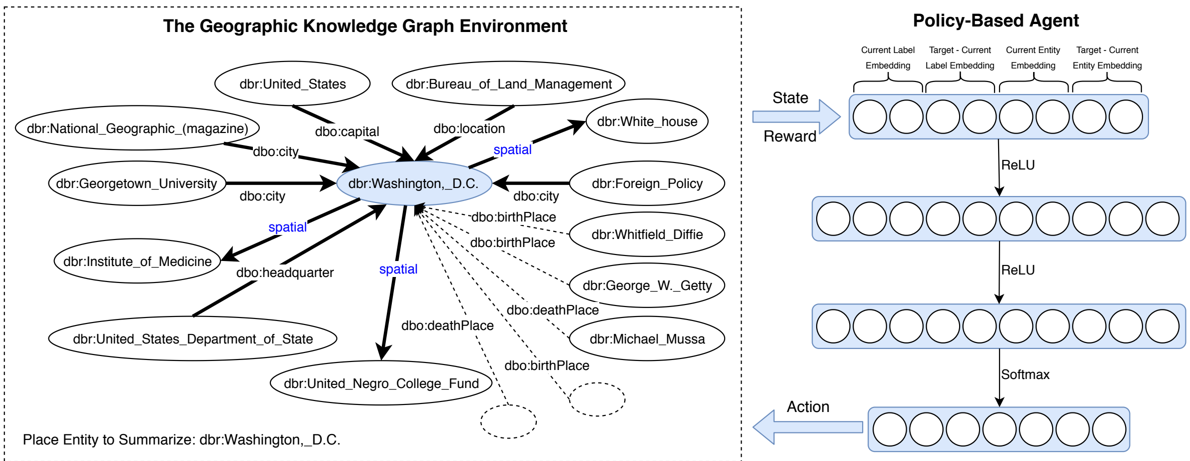

Figure 3: The geographic knowledge graph environment and the policy-based agent interact with each other in the reinforcement learning model. The graph environment on the left shows how the place entity *dbr:Washington,_D.C.* is connected with other spatial/non-spatial entities via various relations. The agent on the right interacts with the environment in the MDP and learns to pick relations to help summarize the graph.

in the hope of more closely resembling the Wikipedia abstract. The agent gets a certain amount of reward depending on the extent to which this step was successful in reaching this goal. When the process terminates, i.e., an episode of MDP has been conducted, the graph is expected to be a good summary of the original geographic knowledge graph for this place. The goal of the agent is to maximize the amount of reward it receives. During this process, the agent is learning to discover the sweet spot on the spectrum between information deficit (a graph with a single node for the place entity itself) and information overload (the whole geographic knowledge graph containing 419,579 nodes) by considering the textual summarization counterpart, namely the Wikipedia abstract. In order to balance the trade-off between exploration and exploitation, the behavior of the agent is defined by the stochastic policy $\pi(a|s) = \Pr(a_t = a|s_t = s)$ which is a probability distribution that determines the likelihood of the agent taking action a in state s at time step t.

In our model, the policy network (shown in Fig. 3) is used to learn an approximation function that captures the dynamics of the interaction and to parameterize the policy $\pi_\theta(a|s)$ of the agent. It is a fully-connected neural network with two hidden layers. Rectified Linear Units (ReLU) are used as activation functions in the hidden layers and the softmax function is used in the output layer to generate probabilities for each possible action. Before diving into the training pipeline, we further explain each concept in the context of our summarization task.

4.2 States

The states capture the information in the MDP. Since our model aims to capture both intrinsic and extrinsic information, we utilize the geographic knowledge graph structure as well as the semantic information from the Wikipedia summaries.

Since there are more than 400,000 entities in our geographic knowledge graph, modeling them as discrete atomic symbols using one-hot vectors in the states is not feasible. In order to provide a condensed representation of the entities, we use the translation-based knowledge graph embedding approach (TransE) (Bordes et al., 2013). The TransE model provides a scalable and generic way to embed nodes and edges in a heterogeneous graph into the same vector space. More concretely, heads, tails and, relations are represented as vectors $\mathbf{v}_{head}$, $\mathbf{v}_{tail}$, and $\mathbf{v}_{relation}$ respectively. The TransE model assumes that $\mathbf{v}_{head} + \mathbf{v}_{relation} = \mathbf{v}_{tail}$ holds for the triple *(head, relation, tail)*. By considering the relations in the graph as translations in the embedding space, the model extracts local and global connectivity patterns between entities. The intrinsic structures of the graph are, thus, embedded in these latent representations of entities and relations. The states in the MDP are supposed to help the agent understand the current environment in order to make decisions about the next step. In this case, the entity embeddings can help capture the progress in the summarization process with respect to the Wikipedia summary. We use the sum of the entity embeddings $\mathbf{z}_t = \sum_{i \in E_t} \mathbf{e}_i$ in the current summarization graph at step t to capture the intrinsic structural information where $\mathbf{e}_i$ is the embedding for entity i in a set of entities E_t.

As these entities also appear as links in Wikipedia summaries, we denote the sum of the embeddings of entities from a target Wikipedia place summary as $\mathbf{z}_{target} = \sum_{i \in E_{target}} \mathbf{e}_i$ where E_{target} is a set of entities that appear in the target Wikipedia place summary. The intrinsic component of the state representation is defined as $\mathbf{s}_t^{intrinsic} = (\mathbf{z}_t, \mathbf{z}_{target} - \mathbf{z}_t)$ where the first component (left) encodes the structure of the summarization graph at step t and the second component (right) encodes the gap between the current graph structure $\mathbf{z}_t$ and the desired structure $\mathbf{z}_{target}$.

For the extrinsic component of the state representation $\mathbf{s}_t^{extrinsic}$, we consider the labels of the entities and relations in the graph as well as the Wikipedia text summary. Neural word embeddings have proven to be an efficient and effective way of encoding meaning of words in our natural languages (Mikolov et al., 2013a,b). We adopt the fastText word embedding model proposed by Bojanowski et al. (2017) as it handles out-of-vocabulary words and considers the morphology of words by viewing each word as a bag of character n-grams.

After obtaining the word embeddings using the fastText model, we use the sum of the entity label and relation label embeddings $\mathbf{h}_t = \sum_{l \in L_t} \mathbf{v}_l$ to help capture the semantic information of the graph at step t. In order to obtain the latent representation of the Wikipedia textual summary, we utilize the Smooth Inverse Frequency (SIF) embedding approach to generate paragraph embeddings $\mathbf{h}_{target}$ using the word embeddings. The theoretical justification of this method is provided by Arora et al. (2017). The idea is to multiply each word vector $\mathbf{v}_w$ by the inverse of its probability of occurrence $p(w)$. Here α is a smoothing constant and is set to 0.001 by default. We then obtain $\mathbf{h}'_{target}$ by summing these normalized and smoothed word vectors and dividing them by the number of words $|W|$:

$$\mathbf{h}'_{target} = \frac{1}{|W|} \sum_{w \in W} \frac{\alpha}{\alpha + p(w)} \mathbf{v}_w \tag{1}$$

As suggested by Arora et al. (2017), we obtain the matrix representation of all 369 Wikipedia summaries and remove the first principal component from this matrix to generate the final embeddings $\mathbf{h}_{target}$ for each Wikipedia place summary because the top singular vector tends to contain syntactic information and removing it cleans up the embeddings' ability to better express semantic information.

Similar to the intrinsic component, the extrinsic component of the state is represented as $\mathbf{s}_t^{extrinsic} = (\mathbf{h}_t, \mathbf{h}_{target} - \mathbf{h}_t)$ and the state representation is a concatenation of these two components:

$$\mathbf{s}_t = (\mathbf{s}_t^{intrinsic}, \mathbf{s}_t^{extrinsic}) = (\mathbf{z}_t, \mathbf{z}_{target} - \mathbf{z}_t, \mathbf{h}_t, \mathbf{h}_{target} - \mathbf{h}_t) \tag{2}$$

After calculating state representations, the cosine distance is calculated between the current graph and the target Wikipedia summary for both entity embeddings and label embeddings, denoted as $dist_{\mathbf{z}_t} = 1 - \cos(\mathbf{z}_t, \mathbf{z}_{target})$ and $dist_{\mathbf{h}_t} = 1 - \cos(\mathbf{h}_t, \mathbf{h}_{target})$ respectively. The termination of the process is decided by:

$$\mathcal{T} = \begin{cases} 1, & \text{if } dist_{\mathbf{z}_t} \leqslant \frac{dist_{\mathbf{z}_1}}{2} \text{ or } dist_{\mathbf{h}_t} \leqslant \frac{dist_{\mathbf{h}_1}}{2} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $dist_{\mathbf{z}_1}$ and $dist_{\mathbf{h}_1}$ denotes the initial cosine distance between the subgraph and the Wikipedia summary for entities and labels respectively. The process terminates if $\mathcal{T} = 1$. This means that if either the cosine distance for entity embeddings or label embeddings is at most half of the initial cosine distance the process will terminate.

4.3 Actions

Given the place entity and Wikipedia summary, the agent aims to choose actions that iteratively leads to a better summary of the geographic knowledge graph for the place in question. Starting from the initial state s_0, the policy network (shown in Fig. 3) outputs the probability of choosing each action a. Since there are 534 unique relations in our geographic knowledge graph, the normal action space is of size 534.

After the agent takes an action and decides to add a relation to the current subgraph, the environment checks possible ways of connecting the entities on the current subgraph with potential new entities via the chosen relation. Let us suppose (by checking the graph) that there are n potential triples to be added to the current subgraph. Each triple contains an entity that is already in the graph, the chosen relation, and a new entity (either a head or a tail entity) to be added. We use the index i to denote the new entity where $1 \leqslant i \leqslant n$ and $triple_i$ to denote the corresponding triple for entity i. Our model picks the triple (and the new entity) among all candidate triples from a distribution where the probability for each triple $p(triple_i)$ is proportional to the information content of the new entity:

$$p(triple_i) = \frac{-\log(p(i))}{\sum_{j=1}^{n} - \log(p(j))} \tag{4}$$

where $p(i)$ is the probability of encountering entity i in the whole geographic knowledge graph and $-\log(p(i))$ is its information content. The rationale behind this approach is that entities that are rich in information content carry latent information that can more efficiently enrich our knowledge about the place we wish to summarize.

In addition to the normal 534 actions, we also propose a novel step by including a dedicated *spatial* action to make the model *spatially-explicit*. This idea stems from the data-driven approach that exploits the hidden patterns of geographic data (Janowicz, 2012) and is inspired by previous work on spatially-explicit models where spatial contextual information facilitates place type embeddings (Yan et al., 2017), image classification for places (Yan et al., 2018), and geographic question answering (Mai et al., 2019). Following a similar school of thought, we aim to utilize spatial context to help improve geographic knowledge graph summarization. Another reason to incorporate this special *spatial* action is that, as mentioned in Section 1, it helps in discovering missing links in the geographic knowledge graph by connecting spatially related entities together.

Simply put, a human (textual) summary of San Diego will include the adjacent border with Mexico. However, such adjacency relation does not exist in DBpedia, and, hence, Tijuana (and Mexico in general) would not be reachable within the graph for the agent.

The *spatial* action itself is modeled as an extra action that the agent can take at any step t. However, if the agent decides to take a *spatial* action, our model only gathers candidates that are geographic entities and are not connected with any entities in the current subgraph directly. We execute a spatial query retrieving all geographic entities within k-meter radius of the place we want to summarize. Our spatially-explicit model selects one geographic entity among these candidate geographic entities from a distribution where the probability for each candidate $p(i)$ is proportional to the inverse of the distance between the candidate and the place q in question:

$$p(i) = \frac{d(i,q)^{-1}}{\sum_{j=1}^{n} d(j,q)^{-1}} \tag{5}$$

where $d(i,q)$ denotes the geodesic distance between candidate i and place q. This inverse distance strategy favors nearby geographic entities over distant ones. While the spatial radius buffer gives a local geographic view around the center place entity, we also propose to incorporate a global view that is modeled by the PageRank score of each entity in the whole geographic knowledge graph (Mai et al., 2018). Intuitively, some places, e.g., landscape features, are characteristic for an entity to be summarized despite their distance due to their overall importance. Mount Fuji is such an example despite its distance of over 100 km from Tokyo. Each entity is assigned a score pr_i after running the PageRank algorithm. This score represents the relative importance of each entity in the graph by examining the incoming and outgoing link connections. By combining the global graph view and the local geographic view, we propose to use a weighted inverse distance in the probability calculation:

$$p(i) = \frac{pr_i d(i,q)^{-1}}{\sum_{j=1}^{n} pr_j d(j,q)^{-1}} \tag{6}$$

After deciding on the relations and entities to add into the subgraph through either spatial or non-spatial actions, new state representations are generated using the methods explained in Section 4.2 and the new state is then presented to the agent to help it decide on the next action.

4.4 Rewards

The reward function plays an important role in guiding the agent to summarize the geographic knowledge graph as the goal of our reinforcement learning model is to find an optimal behavior strategy for the agent to obtain optimal rewards. In our model, there are three components in the reward function, namely similarity score, diversity score, and connection score.

In order to help the agent select the actions (relations) that make the subgraph representation resembles the Wikipedia summary representation from such a large action space, an intuitive way is to incorporate such mechanism in the immediate reward. In addition to cosine distance calculated after the agent takes an action as described in Section 4.2, the cosine similarity is also calculated. The normal similarity score is then defined as the sum of the cosine similarities:

$$r_{similarity}^{normal} = \cos(\mathbf{z}_t, \mathbf{z}_{target}) + \cos(\mathbf{h}_t, \mathbf{h}_{target}) \tag{7}$$

where larger cosine similarity values will result in higher scores for the reward component $r^{normal}_{similarity}$. Moreover, considering the fact that sometimes the TransE model does not handle one-to-many and many-to-many relationships well (Bordes et al., 2013) and summing or averaging the entity embeddings may exacerbate such issues because the connectivity information of individual nodes/entities may be dwarfed by the crude aggregation of other nodes/entities, we propose to substitute the entity similarity score $\cos(\mathbf{z}_t, \mathbf{z}_{target})$ by another measurement to highlight the difference of the intrinsic structure between the subgraph and the Wikipedia summary. Such a measurement is inspired by the Hausdorff distance commonly-used to measure the difference between two geometries. Instead of using a metric such as Euclidean distance as in Hausdorff distance, we use cosine distance because it is insusceptible to the change of magnitude of embedding vectors. This measurement is defined as:

$$sim_{maxmin}(E_t, E_{target}) = 1 - \max_{i \in E_t} \min_{j \in E_{target}} (1 - \cos(\mathbf{e}_i, \mathbf{e}_j)) \tag{8}$$

where E_t is a set of entities on the subgraph at time step t, E_{target} is a set of entities in the Wikipedia summary, and $\mathbf{e}_i$ and $\mathbf{e}_j$ are entity embeddings for entity i and j respectively. The max-min similarity score is then defined as:

$$r^{maxmin}_{similarity} = sim_{maxmin}(E_t, E_{target}) + \cos(\mathbf{h}_t, \mathbf{h}_{target}) \tag{9}$$

While there are 535 possible relations/actions (including the *spatial* action), these relations follow a long-tail distribution, which might lead the agent to pick the most possible relations in order to avoid penalties. In addition, a good graph summary should exhibit a balance between diversity and uniformity. In light of this, we propose to incorporate a diversity score into the reward function:

$$r_{diversity} = \begin{cases} +0.5, & \text{if relation is not already on the subgraph} \\ -0.5, & \text{otherwise} \end{cases} \tag{10}$$

Since it is possible that the model might pick relations and entities that are not directly connected to the place entity in question, we would like to discourage such behavior. For example, to summarize *dbr:Los_Angeles*, the model might add new triples regarding *dbr:California* (because *dbr:California* became part of the subgraph for *dbr:Los_Angeles* at some point) instead of *dbr:Los_Angeles*. This behavior is the result of the data bias in knowledge graphs (Janowicz et al., 2018) as prominent entities are safer for the model to target and would mislead the model to summarize the wrong place. In order to alleviate this potential issue, we propose to include the connection score:

$$r_{connection} = \begin{cases} +0.5, & \text{if entity is directly connected to the place} \\ -0.5, & \text{otherwise} \end{cases} \tag{11}$$

The reward function is then defined as the combination of the three components:

$$R = r_{similarity} + r_{diversity} + r_{connection} \tag{12}$$

It is worth noting that simply reducing relations to be selected from 1-degree queries relative to the entity to be summarized would not be a suitable solution. This would restrict the summary subgraph to a star-shape.

4.5 Training Procedure

As mentioned in Section 4.1, we use a policy-based method to train our spatially-explicit reinforcement learning model. The advantage of policy-based methods over value-based methods such as Q-learning (Watkins and Dayan, 1992) and SARSA (Rummery and Niranjan, 1994) is that they solve an easier problem by optimizing the policy π directly, can provide a stochastic policy, and can be applied to a wider range of problems where the state space is large or even continuous. The objective of the policy-based method is to maximize the total future expected rewards J:

$$J(\theta) = \mathbb{E}_{s \sim \Pr(s), a \sim \pi_\theta(a|s)} R(s, a) \tag{13}$$

Following the REINFORCE (Monte Carlo Policy Gradient) method (Williams, 1992), the policy network is updated using the gradient:

$$\begin{aligned} \nabla_\theta J(\theta) &= \mathbb{E}_{s \sim \Pr(s), a \sim \pi_\theta(a|s)} Q(s, a) \nabla_\theta \log \pi_\theta(a|s) \\ &\approx \frac{1}{N} \sum_{i=0}^{N} \sum_{s, a \in eps_i} Q(s, a) \nabla_\theta \log \pi_\theta(a|s) \end{aligned} \tag{14}$$

where N episodes eps are sampled from the process, $Q(s_t = s, a_t = a) = \mathbb{E}[G_t | s_t = s, a_t = a]$ is the expected return starting from state s after taking action a, and the return $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$ is the total discounted reward from time step t with discount factor $\gamma \in [0, 1]$. A low γ value implies that the agent is myopic in evaluating the situation and values immediate reward over delayed future reward. In addition, similar to the idea of diversity reward in Section 4.4, we include the entropy of the policy as a regularization term in the optimization where we encourage a more diversified set of actions. The entropy is defined as:

$$H(\theta) = -\sum_{a \in A} \pi_\theta(a|s) \log \pi_\theta(a|s) \tag{15}$$

In order to maximize the total future expected rewards J and the entropy H, the loss function is formulated as:

$$\mathcal{L}_{REINFORCE} = -(J + \alpha H) \tag{16}$$

where α is the regularization factor.

Due to the size of the action space, it would be challenging for the policy agent to learn to pick actions purely based on trial and error. In order to solve this problem and inspired by imitation learning (Hussein et al., 2017) and the training pipeline proposed by Silver et al. (2016), we first train our model with supervised learning and then retrain the supervised policy with the proposed reward function to learn summarizing the geographic knowledge graph.

For the supervised learning stage, we use the links in Wikipedia summaries to help gather positive training samples. We query the whole graph to check if the links in the Wikipedia place summary are directly connected to the place entity itself and keep track of these connections. In addition, in order to learn about the *spatial* action as well, we randomly incorporate nearby geographic entities via the special *spatial* relation. This procedure is applied to every place in the training place set in order to get our positive training samples for the supervised learning. A reward of $+1$ is used for each step in these positive training samples. After the supervised training stage, we retrain the model using the reward function described in Section 4.4 to help the agent pick up desired relations to better summarize the graph. Summarizing one place is considered an episode eps. The model starts with a single node (the place entity itself) for the graph and follows the stochastic policy $\pi(a|s)$ to iteratively add relations. We limit the maximum length of the episode with an upper bound max_eps_len to improve the training efficiency.

5 Experiment and Results

In this section, we explain our experiment setup for the model, describe the evaluation metrics used to test the model performance, and present our results and findings.

5.1 Implementation Details

Since we use 50-dimensional vectors for both entity and label embeddings, the resulting state representations are 200-dimensional vectors (see Eq. 2). For *spatial* action, we use a search radius of $k = 100,000$ meters in our geopatial query. The discount factor γ for the cumulative reward we use in the model is 0.99. In the policy network, the first hidden layer has 512 units and the second hidden layer has 1024 units. The Adam Optimizer (Kingma and Ba, 2014) is used to update the parameters in the policy network. The upper bound for the episode length is set to $max\_eps\_len = 20$.

Different alternative settings are proposed for actions and rewards in Section 4.3 and Section 4.4 respectively. The alternatives in the actions component are non-spatial actions vs. spatial actions and unweighted inverse distance (Eq. 5) vs. PageRank-weighted inverse distance (Eq. 6). The alternatives in the reward component are $r^{normal}_{similarity}$ vs. $r^{maxmin}_{similarity}$. In order to better understand the contribution of different component alternatives and to test our assumption that spatially-explicit models are superior in modeling geographic data, we examine our method with different combinations of these alternatives, resulting in 5 models, namely $RL_{nonspatial-normal}$ (model without spatial actions using $r^{normal}_{similarity}$ score), $RL_{spatial-normal}$ (model with spatial actions using $r^{normal}_{similarity}$ score), $RL_{nonspatial-maxmin}$ (model without spatial actions using $r^{maxmin}_{similarity}$ score), $RL_{spatial-maxmin}$ (model with spatial actions using $r^{maxmin}_{similarity}$ score), and $RL_{spatial-maxmin-pr}$ (model with spatial actions and PageRank-weighted inverse distance using $r^{maxmin}_{similarity}$ score).

5.2 Results

To evaluate the models, we consider the intrinsic and extrinsic components separately. For the summarization results, we would like to see the improvements of using our summarization approach compared with the initial information, i.e., we compute the difference between the cosine similarity of the summarized graph and the Wikipedia summary and the cosine similarity of the initial place entity/label and the Wikipedia summary:

$$\text{diff}_{entity} = \cos(\mathbf{z}_T, \mathbf{z}_{target}) - \cos(\mathbf{z}_1, \mathbf{z}_{target}) \tag{17}$$

$$\text{diff}_{label} = \cos(\mathbf{h}_T, \mathbf{h}_{target}) - \cos(\mathbf{h}_1, \mathbf{h}_{target}) \tag{18}$$

where $\text{diff}_{entity} \in [-2, 2]$ and $\text{diff}_{label} \in [-2, 2]$ are the difference of cosine similarities between entity and label embeddings and $\mathbf{z}_T$ and $\mathbf{h}_T$ are the final entity and label embeddings for the summarized graph. Higher diff scores show better summarization results. In addition to this evaluation metrics, we also calculate the Mean Reciprocal Rank (MRR) score for these 5 models. We calculate the cosine similarity scores between the summarized graph of the place with all 35 candidate places in our test set and then rank them. We record the rank position of the corresponding Wikipedia place summary for each place entity, take the reciprocal of the rank, and then calculate the mean of these reciprocal ranks for all 35 places in the test set. Higher MRR scores correspond to better model performance.

Table 2 and Table 3 show the diff_{entity} and diff_{label} scores for all 35 test places. As we can see, on average all 5 models show positive diff_{entity} and diff_{label} scores, implying that these models are effective in creating subgraphs that resemble the Wikipedia summary, thus facilitating the summarization of these places. In general, the scores for the intrinsic component diff_{entity} are lower than the ones for the extrinsic component diff_{label} for the same place and on average. One reason might be that the TransE model takes into account the local and global connectivity information of entities and since the place entity itself is usually closely connected with the Wikipedia links for this place entity the initial single-node graph $\mathbf{z}_0$ tends to be quite similar to $\mathbf{z}_{target}$, making further improvements less prominent. On average, incorporating the *spatial* action or using the $r^{maxmin}_{similarity}$ component in the reward function helps improve the performance and including both further improves the result. The best model is $RL_{spatial-maxmin-pr}$ for both the intrinsic diff_{entity} and extrinsic diff_{label} components. On average it has a 147% and 90% increase compared with the $RL_{nonspatial-normal}$ model for the intrinsic and extrinsic components respectively.

Table 1: MRR result for 5 models.

	Entity	Label
$RL_{nonspatial-normal}$	0.9190	0.6975
$RL_{spatial-normal}$	0.9380	0.7183
$RL_{nonspatial-maxmin}$	0.9428	0.7095
$RL_{spatial-maxmin}$	0.9571	0.7396
$RL_{spatial-maxmin-pr}$	0.9523	0.7742

By examining the results in Table 2 and Table 3 for $RL_{spatial-normal}$ and $RL_{nonspatial-maxmin}$, we can see that adding the spatial action is beneficial for the model to capture more semantic information and using the $r^{maxmin}_{similarity}$ reward component facilitates the model to capture intrinsic structural information as the diff_{label} result is better for $RL_{spatial-normal}$ than for $RL_{nonspatial-maxmin}$ and vice versa in the case of diff_{entity}. The MRR result in Table 1 aligns with our findings.

Fig. 4 and Fig. 5 show the summarization results for *dbr:Washington,_D.C.* and *dbr:Guangzhou* using the $RL_{spatial-maxmin-pr}$ model. The model learns to pick different relations such as *dbo:capital*, *dbo:city*, *dbo:headquarter*, *dbo:location*, *dbo:isPartOf*, and the *spatial* relation. In the case of *dbr:Washington,_D.C.*, the relationship between *dbr:White_House* and *dbr:Washington,_D.C.* is missing in the original geographic knowledge graph. Without the *spatial* relation, such certainly important information would have been lost. Our spatially-explicit model outperforms non-spatial models. In the case of *dbr:Guangzhou*, as we incorporate the connection reward $r_{connection}$ into the model, it refrains from summarizing other entities even though

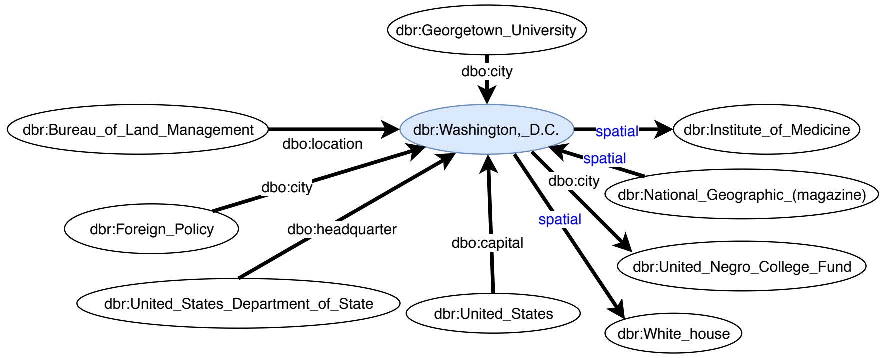

Figure 4: Summarization result for *dbr:Washington,_D.C.*.

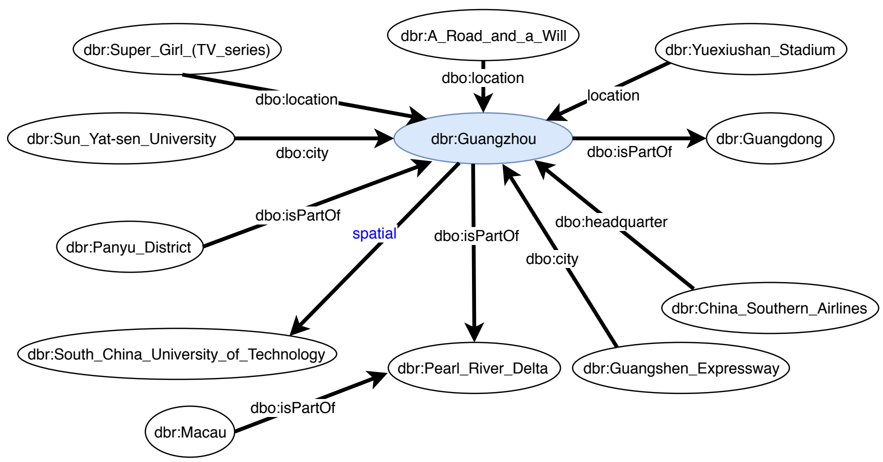

Figure 5: Summarization result for *dbr:Guangzhou*.

dbr:Macau is included in the subgraph at some point.

6 Conclusions and Future Work

In this research, we introduced and motivated the need for geographic knowledge graph summarizations and proposed a spatially-explicit reinforcement learning framework to learn such graph summaries. Due to the lack of benchmark and standard datasets, we collected a dataset that contains Wikipedia place summaries as well as a geographic knowledge graph for 369 places as seed. In order to explore different possibilities of modeling the summarization process, we suggested different alternatives for the actions and rewards formulation in the model. By testing 5 models using different combinations of the alternative components, we conclude that a spatially-explicit model yields superior summarization results compared to non-spatial models, thereby confirming that spatial is indeed special as far as knowledge graph summarization is concerned.

In the future, we would like to test if reducing the variance in the Monte Carlo Policy Gradient method by using an advantage function or the Actor-Critic framework would help improve the performance. Finally, our and other approaches do not consider datatype properties which is an important goal for future research.

References

Arora, S., Liang, Y., Ma, T., 2017. A simple but tough-to-beat baseline for sentence embeddings. In: International Conference on Learning Representations.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z., 2007. Dbpedia: A nucleus for a web of open data. In: The semantic web. Springer, pp. 722–735.

Bast, H., Buchhold, B., Haussmann, E., 2015. Relevance scores for triples from type-like relations. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 243–252.

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., 2017. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5, 135–146.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J., 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. ACM, pp. 1247–1250.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O., 2013. Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems. pp. 2787–2795.

Cheng, G., Tran, T., Qu, Y., 2011. Relin: relatedness and informativeness-based centrality for entity summarization. The Semantic Web–ISWC 2011 , 114–129.

Das, R., Dhuliawala, S., Zaheer, M., Vilnis, L., Durugkar, I., Krishnamurthy, A., Smola, A., McCallum, A., 2017. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. arXiv preprint arXiv:1711.05851 .

Drineas, P., Kannan, R., Mahoney, M. W., 2006. Fast monte carlo algorithms for matrices iii: Computing a compressed approximate matrix decomposition. SIAM Journal on Computing 36 (1), 184–206.

Fredrickson, B. L., Kahneman, D., 1993. Duration neglect in retrospective evaluations of affective episodes. Journal of personality and social psychology 65 (1), 45.

Greenstein, S., Zhu, F., 2012. Is wikipedia biased? American Economic Review 102 (3), 343–48.

Grover, A., Leskovec, J., 2016. node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 855–864.

Hamilton, W. L., Ying, R., Leskovec, J., 2017. Representation learning on graphs: Methods and applications. arXiv preprint arXiv:1709.05584 .

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9 (8), 1735–1780.

Hussein, A., Gaber, M. M., Elyan, E., Jayne, C., 2017. Imitation learning: A survey of learning methods. ACM Computing Surveys (CSUR) 50 (2), 21.

Janowicz, K., 2012. Observation-driven geo-ontology engineering. Transactions in GIS 16 (3), 351–374.

Janowicz, K., Yan, B., Regalia, B., Zhu, R., Mai, G., 2018. Debiasing knowledge graphs: Why female presidents are not like female popes. In: International Semantic Web Conference.

Kejriwal, M., Szekely, P., 2017. Neural embeddings for populated geonames locations. In: International Semantic Web Conference. Springer, pp. 139–146.

Kingma, D. P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

Kuhn, W., Kauppinen, T., Janowicz, K., 2014. Linked data-a paradigm shift for geographic information science. In: International Conference on Geographic Information Science. Springer, pp. 173–186.

Liu, Y., Safavi, T., Dighe, A., Koutra, D., 2018. Graph summarization methods and applications: A survey. ACM Computing Surveys (CSUR) 51 (3), 62.

Mai, G., Janowicz, K., Yan, B., 2018. Support and centrality: Learning weights for knowledge graph embedding models. In: International Conference on Knowledge Engineering and Knowledge Management. Springer, pp. 212–227.

Mai, G., Yan, B., Janowicz, K., Zhu, R., 2019. Relaxing unanswerable geographic questions using a spatially explicit knowledge graph embedding model. In: Proceedings of 22nd AGILE International Conference on Geographic Information Science, in press.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013a. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 .

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J., 2013b. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119.

Nielsen, F. Å., 2007. Scientific citations in wikipedia. arXiv preprint arXiv:0705.2106 .

Paulheim, H., 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic web 8 (3), 489–508.

Pirrò, G., 2015. Explaining and suggesting relatedness in knowledge graphs. In: International Semantic Web Conference. Springer, pp. 622–639.

Ramscar, M., Hendrix, P., Shaoul, C., Milin, P., Baayen, H., 2014. The myth of cognitive decline: Non-linear dynamics of lifelong learning. Topics in cognitive science 6 (1), 5–42.

Rummery, G. A., Niranjan, M., 1994. On-line Q-learning using connectionist systems. Vol. 37. University of Cambridge, Department of Engineering Cambridge, England.

Salvini, M. M., Fabrikant, S. I., 2016. Spatialization of user-generated content to uncover the multirelational world city network. Environment and Planning B: Planning and Design 43 (1), 228–248.

Shen, Y., Chen, J., Huang, P.-S., Guo, Y., Gao, J., 2018. M-walk: Learning to walk over graphs using monte carlo tree search. In: Advances in Neural Information Processing Systems. pp. 6787–6798.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al., 2016. Mastering the game of go with deep neural networks and tree search. nature 529 (7587), 484.

Song, Q., Wu, Y., Lin, P., Dong, L. X., Sun, H., 2018. Mining summaries for knowledge graph search. IEEE Transactions on Knowledge and Data Engineering 30 (10), 1887–1900.

Sun, J., Xie, Y., Zhang, H., Faloutsos, C., 2007. Less is more: Compact matrix decomposition for large sparse graphs. In: Proceedings of the 2007 SIAM International Conference on Data Mining. SIAM, pp. 366–377.

Thalhammer, A., Rettinger, A., 2014. Browsing dbpedia entities with summaries. In: European Semantic Web Conference. Springer, pp. 511–515.

Thalhammer, A., Toma, I., Roa-Valverde, A., Fensel, D., 2012. Leveraging usage data for linked data movie entity summarization. arXiv preprint arXiv:1204.2718 .

Vrandečić, D., Krötzsch, M., 2014. Wikidata: a free collaborative knowledgebase. Communications of the ACM 57 (10), 78–85.

Watkins, C. J., Dayan, P., 1992. Q-learning. Machine learning 8 (3-4), 279–292.

Williams, R. J., 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning 8 (3-4), 229–256.

Xiong, W., Hoang, T., Wang, W. Y., September 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017). ACL, Copenhagen, Denmark.

Yan, B., Janowicz, K., Mai, G., Gao, S., 2017. From itdl to place2vec: Reasoning about place type similarity and relatedness by learning embeddings from augmented spatial contexts. In: Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, p. 35.

Yan, B., Janowicz, K., Mai, G., Zhu, R., 2018. xnet+ sc: Classifying places based on images by incorporating spatial contexts. In: 10th International Conference on Geographic Information Science (GIScience 2018). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Zhang, W., Thill, J.-C., 2019. Mesoscale structures in world city networks. Annals of the American Association of Geographers .

Table 2: Result for diff_{entity} scores for 35 test places for 5 models.

	$RL_{nonspatial-normal}$	$RL_{spatial-normal}$	$RL_{nonspatial-maxmin}$	$RL_{spatial-maxmin}$	$RL_{spatial-maxmin-pr}$
dbr:New_Orleans	0.0287	0.0721	0.0757	0.0866	0.0946
dbr:Boston	0.0008	0.0091	0.0101	0.0114	-0.0012
dbr:Canberra	0.0863	0.1038	0.1020	0.1084	0.1206
dbr:Osaka	0.0759	0.1015	0.0953	0.1135	0.1057
dbr:Lyon	0.0529	0.0676	0.0617	0.0754	0.0764
dbr:Heidelberg	0.0614	0.0858	0.0901	0.1062	0.1086
dbr:Kraków	0.0065	0.0166	0.0409	0.0449	0.0491
dbr:Johannesburg	0.0097	0.0176	0.0131	0.0325	0.0233
dbr:Oxford	0.0231	0.0257	0.0344	0.0602	0.0941
dbr:Milan	0.0043	0.0561	0.0445	0.0746	0.1092
dbr:Montreal	0.0050	0.0192	0.0250	0.0342	0.0451
dbr:Brasília	0.0265	0.0699	0.0651	0.0951	0.1140
dbr:Tel_Aviv	0.0122	0.0238	0.0234	0.0366	0.0391
dbr:Frankfurt	0.0777	0.1073	0.1125	0.1212	0.1235
dbr:Philadelphia	0.0066	0.0168	0.0451	0.0573	0.0682
dbr:Washington,_D.C.	0.0403	0.0545	0.0578	0.0701	0.0949
dbr:Shanghai	0.0323	0.0626	0.0543	0.1002	0.0925
dbr:Saint_Petersburg	0.0279	0.0562	0.0497	0.0699	0.0787
dbr:Seattle	0.0194	0.0310	0.0228	0.0689	0.0406
dbr:San_Diego	0.0350	0.0484	0.0508	0.1267	0.0868
dbr:Seoul	0.0215	0.0291	0.0273	0.0558	0.0599
dbr:Las_Vegas	0.0107	0.0202	0.0332	0.0648	0.0818
dbr:Athens	0.0159	0.0390	0.0499	0.0612	0.0675
dbr:Guangzhou	0.0090	0.0176	0.0378	0.0935	0.0802
dbr:Hangzhou	0.0240	0.0464	0.0500	0.0713	0.0680
dbr:Madrid	0.0380	0.0566	0.0594	0.0670	0.0782
dbr:Edinburgh	0.0335	0.0767	0.0771	0.0931	0.1053
dbr:Barcelona	0.0130	0.0239	0.0383	0.0577	0.0813
dbr:Denver	0.0239	0.0412	0.0498	0.0631	0.0641
dbr:Mexico_City	0.0044	0.0149	0.0177	0.0411	0.0397
dbr:Manila	0.0606	0.0756	0.0859	0.0891	0.0953
dbr:Amsterdam	0.0913	0.1046	0.0966	0.1112	0.1129
dbr:Ho_Chi_Minh_City	0.0495	0.0614	0.0591	0.0848	0.0684
dbr:Kyoto	0.0377	0.0651	0.0561	0.0728	0.0732
dbr:Prague	0.0123	0.0192	0.0183	0.0417	0.0212
Average	0.0307	0.0496	0.0523	0.0732	0.0760

Table 3: Result for diff_{label} scores for 35 test places for 5 models.

	$RL_{nonspatial-normal}$	$RL_{spatial-normal}$	$RL_{nonspatial-maxmin}$	$RL_{spatial-maxmin}$	$RL_{spatial-maxmin-pr}$
dbr:New_Orleans	0.2520	0.3804	0.3725	0.3883	0.3959
dbr:Boston	0.1476	0.3025	0.3027	0.3765	0.4243
dbr:Canberra	0.1033	0.2775	0.2532	0.3138	0.3829
dbr:Osaka	0.0747	0.1172	0.1078	0.1479	0.1971
dbr:Lyon	0.3296	0.4490	0.4396	0.5298	0.5111
dbr:Heidelberg	0.1653	0.2079	0.2140	0.2321	0.2592
dbr:Kraków	0.1041	0.1534	0.1158	0.2258	0.2181
dbr:Johannesburg	0.1593	0.2436	0.2461	0.2931	0.3002
dbr:Oxford	0.1656	0.3358	0.3206	0.3899	0.4136
dbr:Milan	0.2647	0.3249	0.3217	0.3579	0.3823
dbr:Montreal	0.2049	0.2320	0.2753	0.3004	0.3074
dbr:Brasília	0.0676	0.1682	0.0694	0.2071	0.2148
dbr:Tel_Aviv	0.1588	0.2143	0.2069	0.2288	0.2431
dbr:Frankfurt	0.1867	0.3025	0.2900	0.3347	0.3386
dbr:Philadelphia	0.0762	0.1274	0.1214	0.1484	0.1618
dbr:Washington,_D.C.	0.1509	0.3166	0.3290	0.3655	0.3889
dbr:Shanghai	0.1655	0.1840	0.1810	0.2689	0.3629
dbr:Saint_Petersburg	0.1622	0.1981	0.1911	0.2506	0.2381
dbr:Seattle	0.2090	0.2609	0.2634	0.2881	0.2944
dbr:San_Diego	0.2412	0.3575	0.2752	0.3962	0.3986
dbr:Seoul	0.1295	0.1616	0.2061	0.3086	0.2893
dbr:Las_Vegas	0.1652	0.2300	0.2197	0.3613	0.3678
dbr:Athens	0.1770	0.1999	0.2258	0.2466	0.2390
dbr:Guangzhou	0.1122	0.1711	0.1693	0.2193	0.2334
dbr:Hangzhou	0.1045	0.2032	0.1864	0.2151	0.2397
dbr:Madrid	0.1624	0.2214	0.2232	0.2373	0.2364
dbr:Edinburgh	0.1938	0.2737	0.2695	0.3708	0.3944
dbr:Barcelona	0.0697	0.2311	0.2140	0.2528	0.2375
dbr:Denver	0.5028	0.6273	0.6034	0.6688	0.6421
dbr:Mexico_City	0.1383	0.1610	0.1698	0.1869	0.2187
dbr:Manila	0.1013	0.2114	0.1852	0.2595	0.2407
dbr:Amsterdam	0.0745	0.1418	0.1420	0.2233	0.2186
dbr:Ho_Chi_Minh_City	0.2000	0.2974	0.2857	0.3321	0.3603
dbr:Kyoto	0.1350	0.1801	0.1718	0.2304	0.2456
dbr:Prague	0.1537	0.3808	0.3868	0.4317	0.4620
Average	0.1659	0.2527	0.2444	0.3025	0.3159