공학석사 학위논문

지리공간질의응답 범위 확장을 위한 지리공간지식그래프 구축방안: 대한민국 지역을 대상으로

Constructing the Geographic Knowledge Graph for a comprehensive Geographic Question Answering: In Korea Region

2023년 2월

서울대학교 대학원

건설환경공학부

김 동 현

지리공간질의응답 범위 확장을 위한 지리공간지식그래프 구축방안: 대한민국 지역을 대상으로

지도교수 유 기 윤

이 논문을 공학석사 학위논문으로 제출함
2022년 10월

서울대학교 대학원
건설환경공학부
김 동 현

김동현의 석사 학위논문을 인준함
2023년 1월

위 원 장 _____ (인)

부위원장 _____ (인)

위 원 _____ (인)

국문초록

질의응답(Question Answering, QA) 시스템은 자연어 형태로 들어온 질의에 대한 답을 찾는 정보탐색기술로 활발하게 연구되며 좋은 성능을 보이고 있는 자연어처리 분야이다. 하지만, 전체 질의 중 많은 비중을 차지하는 지리공간과 관련된 질의에 대해 구글을 비롯한 검색엔진과 기존의 질의응답 시스템은 적절한 답을 반환하는데 어려움을 겪고 있다. 기존의 질의응답 시스템이 지리공간 질의에 답할 때 가지고 있는 한계를 극복하고자 지리공간 지식그래프(geographic knowledge graph, 이하 GeoKG)를 기반으로 한 지리공간 질의응답시스템(GeoQA system)이 연구되었으나, 여전히 사실기반질의(factoid question)와 지리공간분석질의(geo-analytic question)에 대해 적절히 답을 반환하지 못하고 있는 실정이다. 지식베이스(knowledge base)가 보유하고 있는 관심지점(point of interest, POI)과 공간 객체의 종류가 부족하기에 factoid question에 답하는데 어려움을 겪고 있으며, 보유하고 있는 공간 객체의 정확도가 떨어지고 공간 연산을 수행하기에 어려움이 있어 geo-analytic question에 답하는데 어려움을 겪고 있다.

본 연구는 기존의 GeoKG가 어려움을 겪는 factoid 및 geo-analytic 유형의 문제를 해결하기 위한 GeoKG 구축방안을 제시하였다. 더 많은 factoid question에 응답하기 위해 기존의 GeoKG와 공공데이터를 융합하였고, geo-analytic question은 GeoKG에 사전연산된 공간 관계를 추가하여 factoid question과 같은 방식으로 답할 수 있도록 설계하였다. 또한, GeoQuestions201의 질의를 분석해 geo-analytic question의 성능평가를 위한 질의 시나리오를 제작하였다. 이후 대한민국 전역을 대상으로 실험을 수

행하였다.

기존의 GeoKG 중 지리객체를 많이 보유하고 있는 WorldKG를 속성그래프 형태로 변환한 후 공공데이터를 기반으로 주요 POI 정보 및 행정구역과 폴리곤을 추출한 정보를 추가하였다. 또한, GeoQuestions201 및 MS Marco 데이터셋을 분석해 높은 빈도로 출현하는 공간관계를 정보 추출 및 공간연산을 수행해 관계로 적재하였다. 본 연구에서 구축한 GeoKG를 기존의 GeoKG인 WorldKG와 비교한 결과 factoid question과 geo-analytic question 모두 본 연구에서 구축한 GeoKG가 WorldKG보다 많은 질의에 답할 수 있음을 확인하였다. 또한, 표준 질의 언어가 없는 속성 그래프의 단점을 보완하기 위해 여러 속성그래프 데이터베이스에서 범용성을 가진 GraphQL을 통한 질의도 수행하였다.

본 연구는 기존의 GeoKG에 공공데이터와 사전연산된 공간 관계를 적재해 구축한 GeoKG로 대응할 수 있는 factoid 및 geo-analytic 유형 질의의 범위를 넓힌 구축방안을 제시한 것에 의의가 있다.

주요어 : 지리공간 질의응답, 지리공간지식그래프, 사실기반 질의, 지리공간분석 질의, 사전연산, 공공데이터

학 번 : 2021-29171

목 차

표 목 차

그 림 목 차

용어 정의

■ 지식베이스(knowledge base, KB)

: 지식베이스란 데이터베이스의 일종으로, 사용할 분야의 전문 지식과 규칙 등을 저장하는 것이다. 최근에는 웹 상의 자원을 기반으로 대규모의 데이터를 구축하고 있으며, 많은 연구에서 그래프 형태로 데이터를 구축한 지식그래프와 혼용하고 있다.

■ 지식그래프(knowledge graph, KG)

: 지식그래프란 객체, 관계, 의미 기술(semantic description)로 사실을 표현하는 구조화된 형태이다. 객체는 현실 세계의 물건이나 추상적 개념을 나타내고, 관계는 객체 사이의 관계를 나타낸다.

■ 공간 관계

: 공간 관계란 두 공간 객체 사이의 관계를 의미한다. 개방형 공간정보 컨소시엄(Open Geospatial Consortium, 이하 OGC)은 중첩, 포함, 동일, 교차 등 8가지의 공간 관계를 정의하였다. 본 연구에서는 복잡한 절차를 필요로 하지 않는 포함관계, 인접관계, 교차관계만을 범위로 연구하였다.

■ SPARQL Protocol and RDF Query Langauge(SPARQL)

: 자원 기술 프레임워크(RDF) 형식의 데이터를 검색 및 조작할 수 있는 질의언어로, 월드와이드 웹 컨소시엄(W3C)에 의해 표준화되었다.

■ GeoSPARQL

: 시맨틱 웹에서 RDF 형식의 지리공간 데이터를 표현하고, 지리공간 데이터를 처리할 수 있는 OGC의 표준 언어이다. 위상 관계, 교차, 버퍼(buffer), 중첩 등의 공간 연산을 수행할 수 있다.

■ 미세 폴리곤(sliver polygon)

: 데이터를 구축할 때 오류로 인해 아주 작게 생기는 폴리곤으로, 폴리곤 연산을 할 때 실제와 다른 결과를 반환하는 요인 중 하나이다.

■ KBQA (지식베이스 기반 질의응답)
: 사용자의 질의를 지식베이스에 맞는 질의로 변환하여 지식베이스에서 답을 찾는 질의응답 방식이다.

■ IR-based QA (정보검색 기반 질의응답)
: 사용자의 질의에 대한 답을 웹상의 텍스트 정보나 문서 데이터베이스에서 찾아 답변하는 질의응답 방식이다.

1. 서론

1.1 연구 배경 및 목적

자연어 처리(natural language processing)의 주요한 주제 중 하나인 질의응답(question answering, 이하 QA) 시스템은 많은 관심과 연구를 기반으로 큰 발전을 이루어냈고, Google의 Assistant, Apple의 Siri, Amazon의 Alexa 등의 지능형 가상 비서(intelligent virtual assistant) 모델들로 상용화가 이루어져 사람들의 일상에 큰 영향을 미치고 있다. 자연어로 들어온 질문에 대한 적절한 답을 도출하는 시스템을 QA 시스템이라고 하며 정보기술, 인공지능, 자연어 처리, 데이터베이스 관리, 인지과학을 포괄하는 분야이다(Gupta et al., 2012).

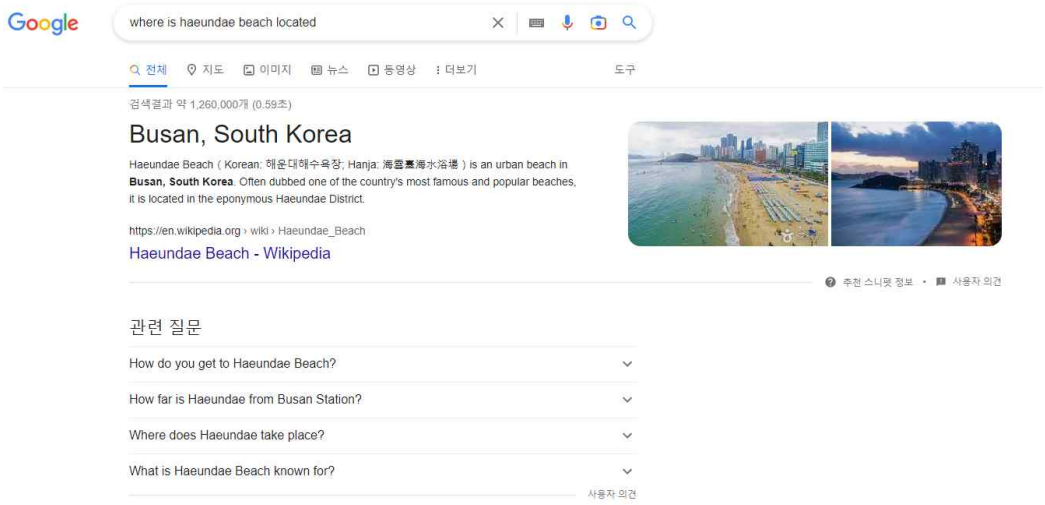
지리공간 질의응답(geograhpic question answering, 이하 GeoQA)은 지리객체를 질의응답 대상으로 하는 질의응답 분야 중 하나이다. Bing과 Cortana에 검색된 질의를 기반으로 제작된 질의응답 데이터셋인 MS Marco(Nguyen et al., 2016)에 포함된 1,010,916개의 질문 중 'Location'으로 분류된 질문유형은 6.17%이고, 'where'를 포함한 질문유형은 4.4%로 많은 비중을 차지함에도 다른 질의응답 분야와 비교하면 활발한 연구가 수행되지 않고 있다.

이는 GeoQA 분야가 가진 어려움이 있기 때문이다. Mai et al.(2021)은 일반적인 QA 시스템은 공간 객체의 폴리곤 등의 기하학적 정보가 부족하고, 인접 및 방위 등의 공간연산은 많은 연산비용이 소모되어 GeoQA를 해결하는 데 한계가 있다고 정의하였다. 또한, 모든 관계를 저장하는 데 한계가 있어 문서나 웹의 정보에서 답을 추출하는 정보검색기반(information retrieval based, 이하 IR-based) 질의응답은 지리공간질의에 응답하기 힘들고, '가깝다'와 같은 모호한 표현에 대한 대처에 어려움이 있다고 서술하였다.

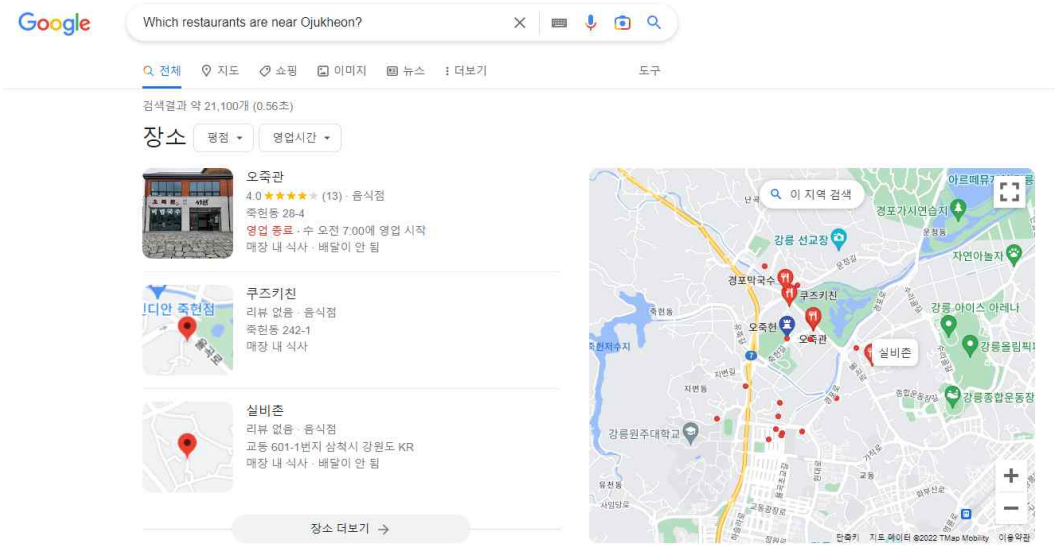
[그림 1-1]과 [그림 1-2]는 대표적인 QA 시스템인 구글에

GeoQuestions201 질의 데이터셋에 포함되어있는 질의를 한국의 지명으로 변환해 검색을 해봤을 때의 결과이다. [그림 1-1]과 같이 POI가 어느 지역에 위치하는지에 대한 질의, 또는 특정 POI에 가까이 있는 공간 객체를 찾는 질의에 대한 답을 잘 반환하는 것을 확인할 수 있다. 하지만 [그림 1-2]의 예시와 같이 행정구역 간의 인접, 강과 행정구역의 중첩 등 공간 연산이 필요한 질의에 대해 문서 검색의 방식으로는 답을 적절히 반환하지 못함을 알 수 있다.

지리공간 질의(이하 Geo question)는 질의유형에 따라 사실기반질의(이하 factoid question)와 지리공간분석질의(이하 geo-analytic question)로 분류된다(Li et al., 2021). Factoid question은 일반적으로 지식베이스(Knowledge Base, 이하 KB)에 저장된 정보를 탐색하여 답을 도출할 수 있는 것으로 [그림 1-1 (a)]의 질의에 해당하며, geo-analytic question은 그 이외의 분석이 필요한 질문으로 [그림 1-1 (b)]와 [그림 1-2]의 질의에 해당한다.

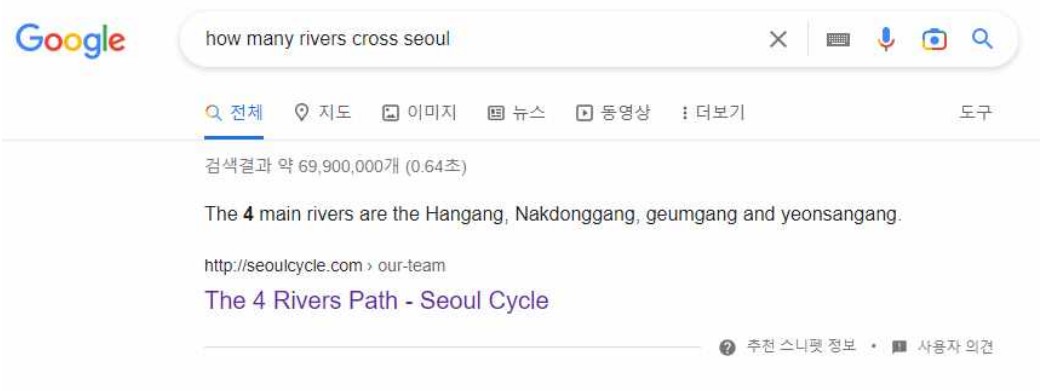


(a) POI의 위치를 묻는 질의
(해운대 해수욕장이 위치한 도시는?)



(b) POI 인근의 공간 객체를 묻는 질의
(오죽헌 근처의 식당은?)

[그림 1-1] 검색엔진이 잘 응답하는 공간 질의



(a) 행정구역과 강의 중첩 관계를 묻는 질의
(서울을 지나는 강은 얼마나 있는가?)

(b) 행정구역 사이의 인접 관계를 묻는 질의
(관악구와 강남구는 인접한가?)

[그림 1-2] 검색엔진이 잘 응답하지 못하는 지리공간질의

QA 접근방식은 크게 웹 또는 문서에서 답을 찾는 방식인 IR-based QA와 KB 안의 정보로 답을 찾는 방식인 knowledge based question answering(이하 KBQA)으로 나뉜다(Park et al., 2015). GeoQA가 지닌 고유한 어려움 때문에 IR-based QA 방식으로는 한계가 있었고, 현재 대부분의 GeoQA는 KB를 기반으로 하는 KBQA의 방식을 채택하였다(Dsouza et al., 2021; Hamzei et al., 2022).

이러한 배경에서 지리공간지식그래프(Geographic Knowledge Graph, 이하 GeoKG)는 GeoQA를 수행하기 위해 구축되었지만 저장하고 있는 공간 객체의 수가 한정적이고, 공간 관계의 종류가 매우 적다는 한계점이 있다. 또한, 자원 기술 프레임워크(Resource Definitoin Framework, 이하 RDF) 데이터 포맷에서 교차(intersect), 통합(union) 등의 공간 연산을 지원하는 GeoSPARQL 연산을 수행하였을 때 많은 시간이 소요되거나 데이터의 오류로 인해 실제와 다른 결과가 나오는 사례도 있었다(Regalia et al., 2019). 공간 객체의 범위가 한정되어있어 음식점, 관광명소 등의 주요한 POI의 정보가 부족하여 factoid question에 답을 반환할 때에도 한계가 있었다.

따라서 본 연구는 대한민국 지역을 대상으로 factoid, geo-analytic 질의에 대응할 수 있는 범위를 넓힌 GeoKG를 구축하는 것을 목적으로 한다.

1.2 연구 동향

1.2.1 지리공간질의응답(GeoQA)

GeoQA는 QA분야 중 지리공간질의를 다루는 분야이다. Mai et al.(2021)은 geo-question이란 자연어 형태의 질의 안에 '서울', '관악구' 등의 공간 객체, '빌딩', '행정구역' 등의 지리개념, '남쪽', '인접'과 같은 객체 간의 관계가 포함된 것으로 정의하였다.

GeoQA는 질의유형에 따라 factoid QA와 geo-analytic QA, scenario QA, visual QA로 분류된다. 이 중에서도 factoid QA와 geo-analytic QA에 관한 연구가 주로 이루어져 왔으며, 각 대답유형의 정의는 아래와 같다.

- Factoid question

Mai et al.(2021)은 factoid question이 factoid geographic knowledge에 기반하여 답할 수 있는 공간정보 질의라고 정의하였고, Li et al.(2021)은 지리적인 사실에 대한 질문에 대답하는 것이라고 정의하였다.

두 연구에서 제시한 factoid question에 대한 정의를 바탕으로 본 연구에서는 지식베이스 또는 지식그래프에 저장되어있는 지리적 사실을 기반으로 답을 반환할 수 있는 질의를 factoid question으로 정의하였다. 예를 들어 '대한민국의 수도는?' 또는 '부산의 인구는?'과 같은 질의가 있다.

- Geo-analytic question

Li et al.(2021)은 geo-analytic question을 복잡한 공간 분석을 요구하는 질의로 정의하였고, Mai et al.(2021)은 복잡한 지오프로세싱(geo-processing) 수행이 필요한 질의로 정의하였다. Xu et al.(2020)은 공간 패턴이나 명확하지 않은 관계를 찾기 위해 등장하였으며 두 점 사

이의 거리를 구하는 단순한 연산부터 길찾기, 패턴 찾기 등 복잡한 연산까지 포함하는 개념으로 정의하였다. Scheider et al.(2021)은 간접 질의응답(indirect QA)의 하나로서, 데이터 변환과정을 거친 후 분석을 수행해 답하는 질의라고 정의하였다.

위의 연구들에서 제시한 geo-analytic question에 대한 정의를 종합하여 본 연구에서는 geo-analytic question을 공간분석과정이 필요한 모든 공간질의로 정의하였다. 예를 들어, '서울대학교에서 가까운 맛집은?', '마포구와 종로구 사이의 행정구는?'과 같은 질의가 geo-analytic question에 포함된다.

Mai et al.(2021)은 IR-based QA 방식은 포인트, 라인, 폴리곤 객체를 거리 연산, 중첩 연산하는 과정에서 문서 정보를 추출하여 답을 하는 것에 한계가 있다고 주장한 바 있다.

복잡한 KBQA를 해결하기 위한 방식은 다시 semantic parsing(이하 SP) 기반 KBQA 방식과 information retrieval(IR) 기반 KBQA 방식으로 분류된다(Lan et al., 2021). SP 기반 KBQA 방식은 자연어 형태의 질의를 KB에 질의할 수 있는 논리적 형태(logic form)로 변환하여 KB 안에서 정보를 찾는 방식이다. 반면에 IR KBQA 기반 방식은 질문과 관련된 그래프를 추출한 후 이를 기반으로 질문에 대한 답을 추론하는 방식이다(Lan et al., 2021). 이때 위에서 웹 또는 문서에서 질의에 대한 답을 검색하는 IR-based QA와 복잡한 KBQA를 해결하기 위한 방식인 IR 기반 KBQA 방식은 서로 다른 개념이다. IR 기반 KBQA 방식은 추론에 의존하는 방식이기에 학습에 어려움이 있어 Binchuan et al.(2019), Hamzei et al.(2022) 등의 연구에서 SP 기반 KBQA 방식을 사용하였다.

KBQA를 수행할 때 주로 KG를 사용하지만, 지리공간질의응답 분야에서 기존의 DBpedia, YAGO와 같이 일반적인 내용을 다루는 다목적 지식그래프(general KG)는 지리객체가 부족하다는 한계를 지닌다(Tempelmeir et al., 2021). 따라서 Karalis et al.(2019), Dsouza et al.(2021) 등은 GeoQA를 수행하기 위한 GeoKG를 구축하는 연구를 수행

하였다.

1.2.2 지리공간지식그래프(GeoKG)

지식그래프(Knowledge Graph, 이하 KG)는 구글이 검색에 시맨틱 정보를 사용하기 위해 제시한 기술로 현재는 DBpedia(Auer et al., 2007), YAGO(Suchanek et al., 2007), Wikidata(Vrandečić and Krötzsch, 2014) 등의 시맨틱 웹 지식베이스를 지칭한다(Zou, 2020). 정보를 시맨틱 그래프 형태로 표현하여 관련된 정보에 접근하기 쉽고 저장된 지식을 기반으로 추론이 가능한 장점이 있으며 방대한 데이터를 보유하고 있다. 특히, 지능형 검색, 개인 맞춤 추천, 소셜 네트워크 분석, 질의응답 등의 분야에서 좋은 성능을 보여주어 활발한 연구가 진행되어왔다(Lu et al. 2019). 다목적 지식그래프는 기하학적 정보를 가지고 있지만 다루는 공간 객체의 종류가 한정적이고 기하학적 정보의 수가 적으며 폴리곤의 지오매트리 정보가 부정확해 GeoKG로 사용하기에 한계가 있다.

GeoQA 분야에서 KBQA 방식으로 응답하기 위해 구축된 GeoKG는 지식그래프에 지리적 지식을 확장한 것으로 WorldKG, YAGO2geo, LinkedGeoData, GeoKG, GEKG 등이 있다.

GeoKG는 구축방법에 따라 두 가지로 나눌 수 있다. LinkedGeoData, WorldKG와 같이 온톨로지를 구축한 후 OSM 등의 지도에서 추출해 새로운 GeoKG를 만드는 방식과 YAGO2geo와 같이 기존의 GeoKG에 GAG, GADM 등의 외부데이터를 추가하는 방식으로 나뉜다. [표 1-1]에 GeoKG의 구축방법과 데이터 소스를 분류하였다.

[표 1-1] 기존 GeoKG의 구축방법 및 데이터 소스

GeoKG	구축방법	데이터 소스
WorldKG (Dsouza et al., 2021)	온톨로지 기반 오픈스트리트맵 정보 추출	오픈스트리트맵, Wikidata
YAGO2geo (Karalis et al., 2019)	YAGO + 행정구역 데이터	YAGO, GADM, GAG, Ordanance Survey
LinkedGeoData (Auer et al., 2009)	OSM 정보 추출	오픈스트리트맵
Binchuan, 2019	Virgual Geographic Environments에서 공간 관계 추출	Virgual Geographic Environments

1.2.2.1 Factoid question 관련 GeoKG

지리공간적 객체에 중점을 두어 factoid question을 다루는 GeoKG는 다음과 같다.

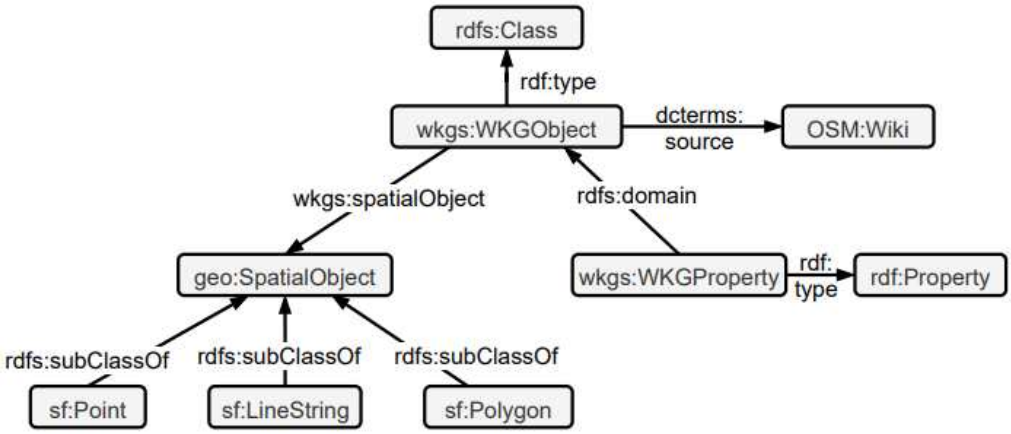
대한민국 국토교통부, 한국관광공사, 외교부 등에서 RDF 데이터 형식의 LOD[1](Linked Open Data)를 제공하고 있다. 국토교통부가 제공하는 LOD는 약 64억개의 RDF 트리플과 6억 개의 지리적 객체를 보유하고 있으나 건물, 주택, 토지, 시설물 정보만을 포인트 형식으로 제공하여 많은 종류의 POI 데이터를 다루지 못한다.

LinkedGeoData(Auer et al., 2009)는 가장 큰 크라우드 소싱 기반 지도인 오픈스트리트맵(OpenStreetMap)의 정보를 RDF 데이터 형식으로 변환하고 저장하였다.

WorldKG(Dsouza et al., 2021)는 공간 객체가 부족한 다목적 KG와 지리적 객체의 종류가 부족한 LinkedGeoData와 YAGO2geo와 같은

1) http://www.nsdi.go.kr/lxportal/?menuno=4038

GeoKG의 단점을 해결하기 위해 오픈스트리트맵을 온톨로지에 맞추어 KG로 구축하였다. 오픈스트리트맵을 기반으로 구축하여 GeoKG 중 가장 많은 종류의 공간 객체를 다룬다는 장점이 있다. 온톨로지는 포인트, 라인, 폴리곤 데이터를 다루도록 제작되었지만, 실제 기하학적 정보는 포인트 데이터만 적재되었다. [그림 1-3]은 WorldKG의 온톨로지이며 [표 1-2]는 WorldKG에서 제공하는 SPARQL Endpoint[2]에 포인트, 라인스트링, 폴리곤 정보의 존재 여부를 확인하는 질의 및 결과이다. 이외에도 행정구역 사이의 포함관계, 인접 관계 등의 공간 관계가 없어 geo-analytic question을 수행할 수 없다는 한계가 있다.



[그림 1-3] WorldKG의 온톨로지(출처: Dsouza et al.(2021))

2) https://www.worldkg.org/

[표 1-2] WorldKG의 기하학적 정보 존재여부 확인

질의	응답

포인트 정보 존재여부 확인
 	
ASK {
?wkg rdf:type sf:Point
}
	true
	true
	Query
	ASK {
	?wkg rdf:type sf:Point
	}

라인 정보 존재여부 확인
 	
ASK {
?wkg rdf:type sf:LineString
}
	false
	false
	Query
	ASK {
	?wkg rdf:type sf:LineString
	}

폴리곤 정보 존재여부 확인
 	
ASK {
?wkg rdf:type sf:Polygon
}
	false
	false
	Query
	ASK {
	?wkg rdf:type sf:Polygon
	}

1.2.2.2 Geo-analytic question 관련 GeoKG

Mai et al.(2021)은 geo-analytic question과 관련하여 다음과 같은 어려움을 정의하였다. 첫째, QA 시스템들은 일반적으로 포인트, 라인, 폴리곤 등의 공간 객체의 기하학적 정보가 부족하다. 둘째, 폴리곤의 경우 인접, 위상관계와 같은 공간 연산을 수행할 때 큰 비용이 소모된다. 마지막으로, 많은 공간 관계들이 모호해서 KG에 연산을 저장하기 어렵고, 학

습에도 어려움이 있다. 예를 들어, ‘서울역에서 가까운 대학교’라는 질의에서 ‘가까운’의 기준이 명확하지 않아 답변에 어려움이 있다. 또한, ‘파주시 북쪽에 있는 시는?’과 같은 질의에 대한 답을 위해 모든 객체와 연산을 수행하기에 한계가 있다. 관계를 저장한다고 해도, 모든 객체별로 관계를 저장하려면 데이터의 크기가 기하급수적으로 커진다는 단점이 있다.

YAGO2geo(Karalis et al., 2019)는 다목적 지식그래프 중 하나인 YAGO에 포함된 지리공간정보의 종류 및 개수가 부족하여 GAG, Ordanance Survey에서 그리스 및 영국지역의 행정구역 정보를 추가하고, GADM과 오픈스트리트맵의 정보로 라인 및 폴리곤 데이터를 추가하였다. 또한, [그림 1-4]의 ‘Antiparos’가 ‘South Aegean’에 속해있다는 관계를 나타내는 ‘within(행정구역 사이 포함)’ 외에도 ‘touches(행정구역 사이 인접’) 관계를 추가하였다. 하지만, YAGO2geo는 그리스, 영국 및 아일랜드 지역에 대해서만 데이터셋을 제공하여 대한민국 지역을 대상으로 질의응답을 수행할 수 없다는 한계점이 존재한다. 또한, Brigham et al.(2011)은 각 나라의 정부에서 제공하는 행정구역 경계 데이터셋을 기준으로 GADM 데이터셋을 다른 경계 데이터셋인 Global Administrative Unit Layers(GAUL)과 UNSALB와 비교했을 때 정확도가 떨어지는 것을 확인하였다.

<http://yago-knowledge.org/resource/geoentity_Dimos_Antiparos_8133698>
<http://www.opengis.net/ont/geosparql#sfWithin>
<http://yago-knowledge.org/resource/South_Aegean>

(a) RDF 트리플의 그래프 예시(Antiparos가 South Aegean에 속해있다)

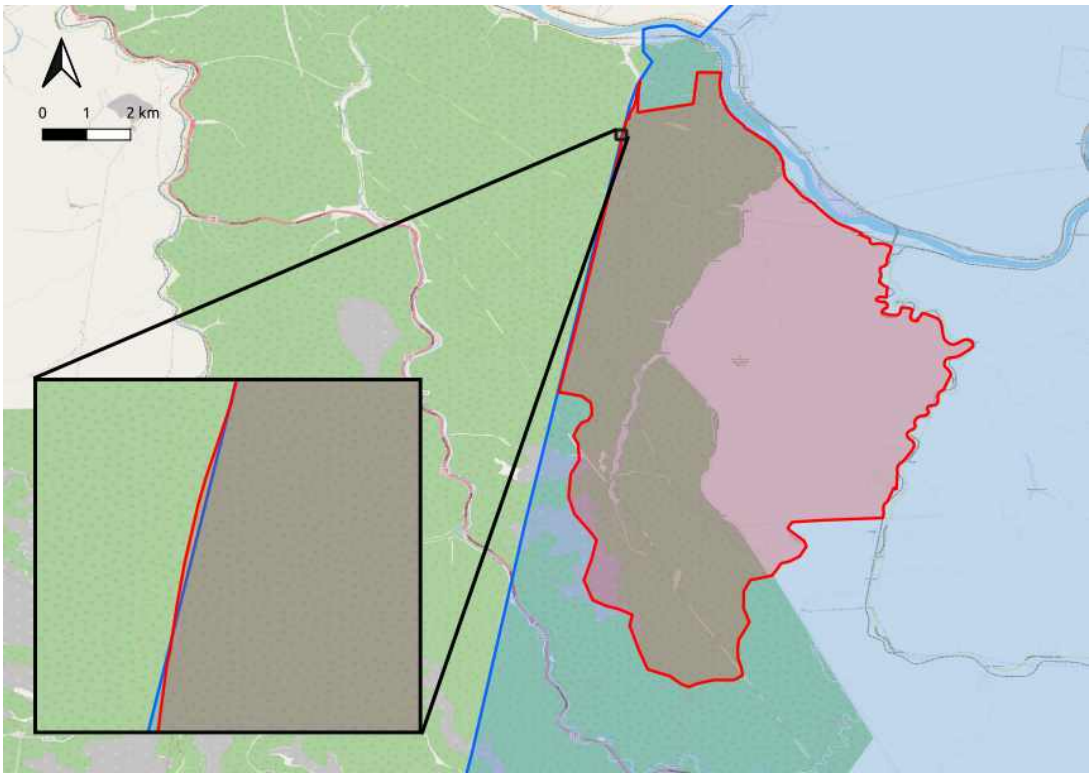
(b) South Aegean 지역 및 Antiparos의 위치

[그림 1-4] YAGO2geo 예시

현재 YAGO, DBPedia, YAGO2geo는 여러 공간 관계 중 행정구역 간의 방위관계와 포함관계만을 저장하고 있고, WorldKG 및 LinkedGeoData는 공간 관계를 전혀 포함하지 않고 있다. 또한, 폴리곤 데이터를 포함하지 않거나, 부정확한 폴리곤 데이터를 활용하고 있어 지식그래프를 기반으로 공간 관계를 도출하거나 연산하기에 어려움이 있다.

Regalia et al.(2019)은 [그림 1-5]의 예시와 같이 미세 폴리곤(sliver polygon) 등의 데이터 문제로 인해 GeoSPARQL을 사용해 공간 연산을 수행할 때 실제와 다른 결과를 반환하거나 큰 규모의 데이터셋에서 연산 시간이 크게 소요되는 문제를 해결하기 위해 라인 및 폴리곤 사이의 관계를 사전연산해 GeoKG에 저장하였다. 이때, 도시, 군(county), 공원만

을 멀티폴리곤 도형으로 선정하였고, 도로와 하천을 라인 객체로 선정하여 연구를 수행하였다. 그 결과, 인접한 관계에 대한 질의에 대해서 사전연산을 수행한 후의 결과와 GeoSPARQL을 수행한 결과를 비교하였을 때 약 20배 빠른 성능을 보여주며 공간 관계에 대한 사전연산의 유효성을 입증하였다. [그림 1-5]의 빨간색 폴리곤은 Powellton 지역이고, 파란색 폴리곤은 Fayette County 지역으로 본래 Powellton 지역이 Fayette County에 포함되어있지만, 포함관계를 연산하는 GeoSPARQL의 'geof:sfWithin' 연산을 수행하였을 때 미세폴리곤으로 인해 '포함하지 않음'의 결과를 반출하였다(Mai et al., 2021).



[그림 1-5] 미세 폴리곤으로 인한 오류의 예시 (출처: Mai et al., 2021)

Binchuan et al.(2019)은 VGE에서 공간정보를 추출해 지식그래프로 구축할 때 'near'와 'far'의 모호함을 해결하기 위해 객체의 단위별로 거리 집합을 만들어 모호한 거리관계를 추출하였다. 예를 들어, "한국과 가

까운 국가"라는 질문에서 '가깝다'의 거리 기준과 "서울과 가까운 도시"라는 질문에서 '가깝다'의 거리 기준이 서로 다르기 때문에 공간 객체의 종류별로 다른 결과를 반환하는 방식을 제안하였다.

하지만, Regalia et al.(2019)과 Binchuan et al.(2019)은 실제 질의에서 나타나는 공간 관계를 고려하지 않고 공간 관계를 구축하였다는 한계가 있다.

본 연구에서는 geo-analytic question에 답하기 위해 정확하고 빠른 연산 결과를 반환할 수 있는 장점을 가진 사전연산을 수행하여 답을 반환하고자 한다. 주요한 공간 관계를 사전연산해 GeoKG에 저장하고, 관련된 질의가 들어왔을 때 추가적인 분석과정을 거치지 않고 정보를 탐색하는 방식으로 응답할 수 있는 GeoKG를 구축하는 방안을 제안한다.

1.2.3 RDF와 속성그래프

Wikidata, Freebase, YAGO를 비롯한 다목적 지식그래프와 WorldKG, YAGO2geo와 같은 GeoKG는 대부분 RDF 형식으로 구축되어있다.

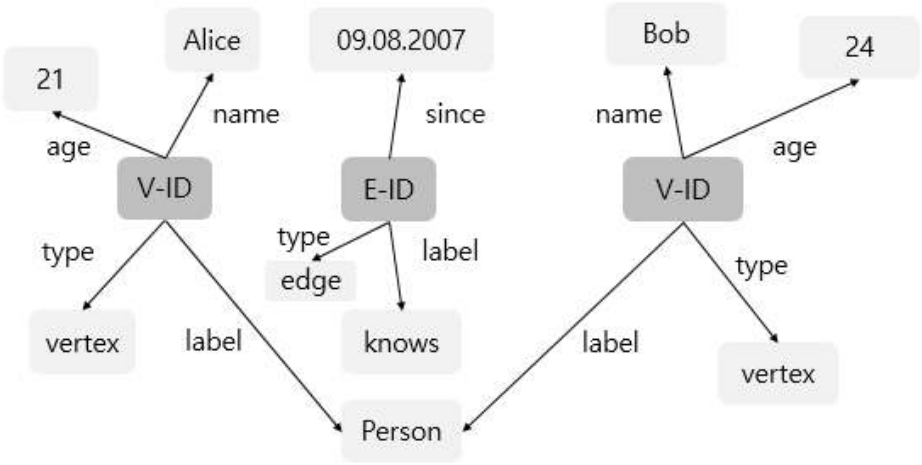
RDF는 웹상에서 데이터의 상호교환을 위해 만들어진 World Wide Web Consortium(W3C) 표준 모델이다. 트리플로 구성된 방향그래프(directed graph)이며, 트리플은 Subject, Predicate, Object로 구성된다. Subject와 Object는 노드를, Predicate은 Subject에서 Object 사이 관계 정보를 나타낸다(Wylot et al., 2018).

RDF는 일반적으로 불필요한 데이터가 많고, RDF 데이터를 검색하기 위한 SPARQL 언어는 그래프를 탐색하거나 그래프 분석 알고리즘을 적용하기에 적절하지 않다(Matsumoto et al., 2018). 또한, RDF 데이터 포맷에는 관계에 속성을 저장할 수 없어 실제 세계의 데이터를 그래프 형태로 표현하는데 한계를 가지고 있다. Haihong et al.(2020)은 만일 두 객체 사이에 '결혼'과 같은 관계가 중복하여 생성되었을 때, 이를 속성그래프로 저장한다면 '결혼'이라는 서로 다른 시간대의 속성을 가진 관계로 표현할 수 있지만, RDF 데이터 형식으로 표현할 때에는 시간대를 속성으로 표현할 수 없어 서로 다른 관계로 저장하는데 어려움이 있다고 서술하였다. 많은 연구들이 RDF 데이터 포맷 형식의 데이터를 속성그래프로 변환하는 연구를 수행하였다(Tomasuzuk, 2016; Haihong et al., 2020).

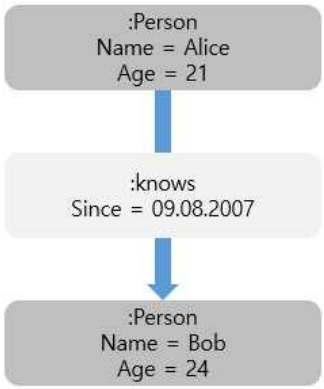
속성그래프는 노드(node)와 엣지(edge)를 가지며, 노드와 엣지를 라벨(label)을 통해 서로 다른 부분집합으로 정의하고, 속성(property)으로 실제 세계의 특성을 키-값 형태로 가지고 있는 방향그래프이다. 엄격한 스키마를 요구하지 않아 데이터 추가가 용이하고, 관계에 속성을 저장하기 쉬워 데이터 표현에 장점이 있다. [그림 1-6]은 같은 내용을 RDF 트리플과 속성그래프로 다르게 표현한 예시이다.

속성그래프는 다양한 소스로부터 얻은 공간 객체를 DB에 저장하고,

객체 간의 관계를 표현하기 적합하여 본 연구에서는 대표적인 속성그래프 데이터베이스인 Neo4j를 사용하여 GeoKG를 구축 및 저장하였으며, Neosemantics[3] 라이브러리를 사용하여 RDF 형식으로 구축되어있는 GeoKG를 속성그래프 형식으로 변환하였다.



(a) RDF 트리플의 그래프 예시



(b) 속성그래프의 예시

[그림 1-6] RDF 데이터와 속성그래프 예시 (출처: Besta et al., 2019)

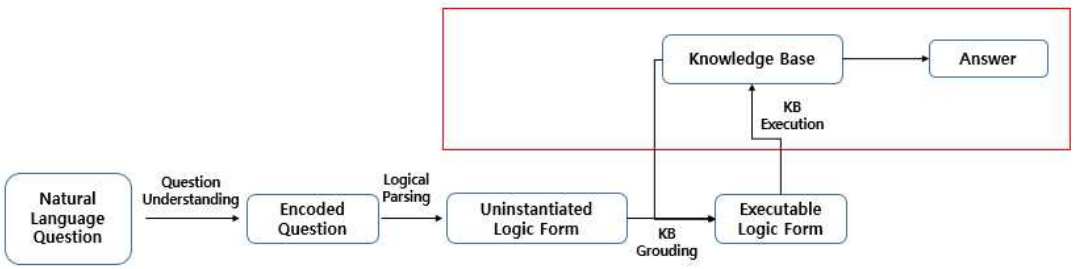
3) https://github.com/neo4j-labs/neosemantics

1.3 연구 범위 및 방법

본 연구에서는 factoid question과 geo-analytic question에 응답할 수 있는 범위를 확장한 GeoKG 구축방안을 제시한다. Factoid question을 해결하기 위해 GeoKG 중 하나인 WorldKG에 행정구역 및 주요 POI 정보를 적재하였다. 이후 공간 관계를 GeoKG에 저장하여 geo-analytic question을 factoid question과 같이 GeoKG 안의 정보를 탐색하는 방식으로 응답하는 방법을 제안하였다.

대표적인 속성그래프 데이터베이스인 Neo4j를 사용해 RDF 데이터 형식인 WorldKG를 속성그래프 데이터 형식으로 변환하였으며 변환된 WorldKG의 저장 및 쿼리 또한 Neo4j 상에서 수행하였다.

[그림 1-7]은 자연어 질의가 들어왔을 때 답을 도출하는 SP 기반 KBQA의 과정이다.(Lan et al., 2021) 본 연구에서는 전체 과정 중에서 KB를 구축하는 과정과 실행 가능한 logical form으로 KB에 쿼리하여 답을 반환하는 과정을 다룬다.



[그림 1-7] Semantic Parsing 기반 KBQA의 흐름도

본 연구의 GeoKG 구축범위는 대한민국 전역을 대상으로 한다. 지리공간질의 데이터셋인 GeoQuestions201의 질의를 분석해 총 7개의 질의 시나리오를 선정하였고, 새로운 GeoKG를 구축하기 위해 가장 많은

POI 종류를 보유하고 있는 WorldKG와 최신 데이터를 제공하는 공공데이터를 융합하였다. 행정구역은 시도, 시군구, 읍면동 단위까지 구축하였다. 읍면동은 쉐이프파일(shapefile)을 사용할 수 있는 법정동을 기준으로 구축하였다.

Geofabrik[4]에서 제공하는 Protocolbuffer Binary Format(이하 PBF) 형식의 대한민국 데이터 파일과 Dsouza et al.(2021)[5]에서 제공하는 라이브러리로 대한민국 지역에 해당하는 WorldKG를 추출하였다. 통계분류포털[6]에서 내려받은 행정구역표를 사용해 행정구역의 정보 및 포함관계를 설정하였다.

행정구역의 폴리곤 데이터는 국가공간정보포털[7]에서 제공하는 쉐이프파일을 QGIS를 활용해 GeoJSON 파일로 변환하였으며, 폴리곤의 집합 형태인 멀티폴리곤 형식의 기하학적 정보는 데이터베이스의 범용성을 위해 Python 기반 Shapely 라이브러리[8]를 통해 서로 다른 폴리곤으로 분리하였다.

POI 대분류 및 대표 POI는 오성호(2006)와 Ordanance Survey(2021)에서 정의한 POI의 분류표를 참고하여 선정하였으며, 대분류별로 속해있는 소분류 1개씩을 선정하였다.

본 논문에서는 geo-analytic 유형의 질의에 답할 수 있는 범위를 확장하기 위해 질의 데이터셋을 분석하여 Hamzei et al.(2019)에서 정의된 2개의 공간 관계와 GeoQuestions201에서 추출한 3개의 공간관계를 선정하였다. 선정한 관계는 POI와 행정구역 사이의 포함관계(In), 행정구역 사이의 포함관계(Belong to), POI 사이의 인근 관계(Near), 행정구역 사이의 인접 관계(Border), 하천과 행정구역의 중첩 관계(Cross)이다.

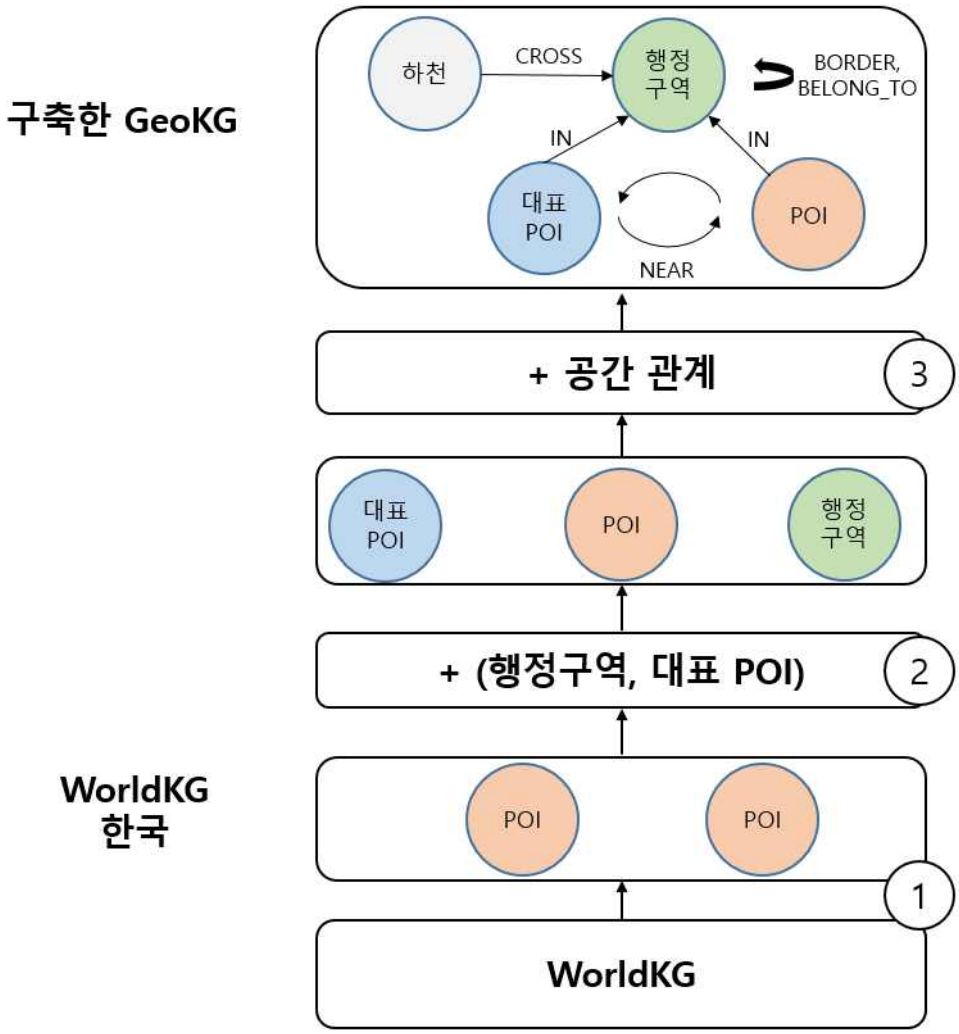
GeoKG를 구축한 후에는 질의 시나리오를 기반으로 기존의 GeoKG들과 성능을 비교하였으며, 속성그래프 데이터베이스의 질의언어에 의존

4) https://www.geofabrik.de/
5) https://github.com/alishiba14/WorldKG-Knowledge-Graph
6) https://kssc.kostat.go.kr/
7) http://data.nsdi.go.kr/dataset
8) https://shapely.readthedocs.io/en/stable/

적이지 않은 GraphQL을 사용한 실험을 수행하였다.

2. 연구 방법

이 장에서는 다양한 종류의 공간 객체를 다루는 GeoKG를 구축하고, 주요한 질의에 대한 사전연산을 수행해 관계를 적재하는 구축방안을 제시하였다. 본 연구에서 제안하는 GeoKG 구축의 프레임워크는 [그림 2-1]과 같다.



[그림 2-1] 본 연구의 GeoKG 구축 프레임워크

[그림 2-1]과 같이 GeoKG 구축의 프레임워크는 (1) 대한민국 지역

에 해당하는 WorldKG를 추출한다. (2) 행정구역 및 주요 POI 정보들을 구축한다. (3) 데이터 추출, 사전연산으로 공간 연산이 필요한 공간 관계들을 GeoKG에 적재한다.

1.2.1절에서 정의한 바와 같이 factoid question과 geo-analytic question은 지식베이스에 관련 정보의 포함 유무에 따라 나뉜다. 만일 한 GeoKG에서 geo-analytic question으로 분류되는 질의여도, 다른 GeoKG에서 관련된 정보를 저장하고 있다면 factoid question으로 분류될 수 있다.

본 연구에서는 GeoKG에 사전연산한 공간 관계를 추가하여, 기존에 geo-analytic question으로 분류되는 질의에 대해 factoid question에 답하는 방식과 같이 데이터베이스 안의 정보를 찾는 방식으로 응답한다.

2.1 질의 데이터셋 분석 및 질의 시나리오 선정

본 연구에서는 구축한 GeoKG의 성능평가 및 기존 GeoKG와의 geo-analyltic question에 대한 비교를 위해 GeoQuestions201을 기반으로 질의 시나리오를 선정하였다.

GeoQuestions201은 Punjani et al.(2018)이 제안한 GeoQA 분야 성능검증 데이터셋으로 총 200개의 질문으로 구성되어있다. GeoQuestions201은 자연어를 GeoSPARQL로 변환하는 연구(Hamzei et al., 2022)와 자연어 질문에서 시맨틱한 관계를 탐지하는 연구(Li et al., 2021) 등에서 사용되고 있는 GeoQA 데이터셋이다. 질의 데이터셋에 속한 질의들은 영국(잉글랜드, 아일랜드, 스코틀랜드) 지역의 지명과 행정구역 체계를 기반으로 제작되어, 한국을 대상으로 한 GeoKG의 성능을 검증할 때 적합하지 않아 지명 및 장소명을 한국의 실정에 맞추어 변환하였다. 변환하는 과정에서 질문의 의도를 변형하지 않기 위해 POI의 종류는 유지하고 이름만 변환하였으며, 행정구역의 단위도 [표 2-1]의 기준에 맞추어 수동으로 같은 단위에 있는 대한민국의 행정구역으로 변경하였다.

[표 2-1] 질의 데이터셋의 행정구역 단위와 상응하는 한국 행정구역 분류

질의 데이터셋	한국 행정구역 분류
Province	도
City	시
County	군
Town	구
Village	읍, 면, 동, 리

예를 들어 ‘영국 박물관(the British Museum)’은 ‘국립박물관(the National Museum of Korea’으로 변경하였고, ‘체셔 카운티(county Cheshire)’는 ‘무안군(county Muan)’으로 변환하였다. 변환한 질의의 예

시는 [표 2-2]와 같다.

[표 2-2] GeoQuestions201 질의 데이터셋의 변환 전후 질의 예시

변환 전 GeoQuestions201	한국 지명 변환 후 질의 예시
런던에서 가장 가까운 공항은? (Which is the closest airport to **London**?)	**서울**에서 가장 가까운 공항은? Which is the closest airport to **Seoul**?
스코트랜드의 카운티 중 **잉글랜드**와 접하는 지역은? (Which counties of **Scotland** border **England**?)	**충청북도**의 군 중 **강원도**와 접하는 지역은? (Which counties of **Chungcheong-bukdo** border **Gangwondo**?)
템즈강이 지나는 도시들은 얼마나 되는가? (How many cities does **Thames river** crosses?)	**낙동강**이 지나는 도시들은 얼마나 되는가? (How many cities does **Nakdonggang river** crosses?)
체셔 카운티에 숲이 있는가? (Is there a forest in county **Cheshire**?)	**무안군**에 숲이 있는가? (Is there a forest in county **Muan**?)
영국박물관에서 가장 가까운 **런던의** 지하철역은? (What **London** underground stations are closest to the **British Museum**?)	**국립박물관**에서 가장 가까운 **서울의** 지하철역은? (What **Seoul** Subway stations are closest to the **National Museum of Korea**)

GeoQuestions201에 속하는 질의에 대한 답을 도출하기 위해 필요한 기하학적 정보 간의 관계를 분류하였으며 분류한 결과의 예시는 [표 2-3]과 같다.

[표 2-3] GeoQuestions201의 질의 관계 분류 예시

질의	분류
서울의 지하철 노선 개수는? (How many underground lines does Seoul have?)	라인 <-> 폴리곤
N타워 근처 호텔은? (Which hotels are near N Tower?)	포인트 <-> 포인트
경기도에서 가장 큰 시군구는? (Which is the biggest county in the Gyeonggi-do?)	폴리곤 <-> 폴리곤
국립박물관에서 가장 가까운 서울의 지하철역은? (What Seoul Subway stations are closest to the National Museum of Korea?)	포인트 <-> 포인트, 포인트 <-> 폴리곤
서울에서 가장 높은 빌딩은? (Which is the highest building in Seoul?)	포인트 <-> 폴리곤, 속성정보

각 기하학적 정보 사이의 관계별 개수 및 예시는 [표 2-4]에 정리하였다. [표 2-3]의 “국립박물관에서 가장 가까운 서울의 지하철역은?”이라는 질의는 ‘국립박물관’이라는 포인트와 ‘지하철역’이라는 포인트의 관계와 ‘서울’이라는 폴리곤과 ‘지하철역’이라는 폴리곤의 관계가 모두 해당한다. 이 경우, 두 관계 모두 개수에 포함하여 [표 2-4]에 정리한 기하학적 정보 사이의 관계가 전체 질의 개수인 200개보다 많은 215개로 분류되었다.

Regalia et al.(2019)은 시군구, 공원을 멀티폴리곤 객체로, 도로와 시내(stream)을 폴리라인 객체로 설정하였다. 본 연구에서는 Regalia et al.(2019)과 같이 도로, 철도, 다리는 라인객체로, 행정구역, 호수, 산, 산림은 폴리곤 객체로 설정하였다. 하천의 경우 질의에서 하천의 길이 또는 면적을 물어보는 질의가 있어 폴리곤 객체로 설정하였다.

[표 2-4] 기하학적 정보 간의 관계별 개수 및 예시

기하학적 정보 사이의 관계	개수	예시
폴리곤 <-> 폴리곤	77	잉글랜드가 아일랜드보다 더 많은 카운티를 가졌는가? (Does England have more counties than Ireland?)
포인트 <-> 폴리곤	59	런던에서 가장 가까운 공항은?) (Which is the closest airport to London?)
속성정보	34	런던에서 가장 높은 건물은? (Which is the highest building in London?)
포인트 <-> 포인트	34	빅벤에서 가장 가까운 호텔은? (Which hotels are near Big Ben?)
라인 <-> 폴리곤	6	템즈강을 지나는 다리는? (Which bridges cross River Thames?)
포인트 <-> 라인	5	워털루 다리에서 1km 범위 안에 있는 주차장은? (Is there a car park at most 1km from Waterloo Bridge?)
계	215	

총 200개의 질의에 포함된 215개의 기하학적 정보 사이의 관계 중 라인 데이터와 관련된 관계는 11개로, 폴리곤과 관련된 관계가 142개, 포인트와 관련된 관계가 98개인 것에 비해 현저하게 적은 것을 확인할 수 있다.

GeoKG 구축 후 geo-analytic question에 대한 성능을 평가하기 위해 GeoQuestions201을 기반으로 질의 시나리오를 선정하였다. GeoQA 데이터셋으로는 GeoQuestions201 외에도 GeoAnQu 데이터셋이 있지만 GeoAnQu는 공간적인 관계보다는 인구밀도, 생활패턴, 접근성 등의 통계 및 분석적인 요소나 범죄, 화재와 같은 사건과 관련된 요소가 64.1%를

차지하며 공간적인 관계를 GeoKG에 저장하는 본 연구의 목적과 다르다고 판단하여 본 연구에서는 GeoQuestions201만을 사용해 질의 시나리오를 구성하였다.

질의 시나리오는 속성정보를 제외한 모든 기하학적 정보별로 1개씩 선정하였다. 폴리곤의 경우 인접, 포함 등의 관계가 다양하여 폴리곤 사이 관계는 가장 많이 나오는 행정구역 사이 인접(13개), 행정구역 사이 포함(34개), 행정구역과 자연물 사이 중첩(22개)으로 나누어 시나리오로 선정하였다.

선정한 질의 시나리오와 시나리오별 기하학적 정보 관계 및 질의는 [표 2-5]와 같다. 질의는 하나의 기하학적 정보 관계를 포함하도록 설정하였으며, 질의의 지명 및 행정구역명은 무작위하게 선정하였다.

[표 2-5] 선정한 질의 시나리오별 기하학적 정보 관계 및 질의 내용

시나리오	기하학적 정보 관계	질의
시나리오 1	포인트 <-> 포인트	동대문역사문화공원역 근처 쇼핑몰은?
시나리오 2	포인트 <-> 라인	성수대교에서 10분 거리 안에 있는 경찰서는?
시나리오 3	포인트 <-> 폴리곤	대학동에 속한 대학교는?
시나리오 4	라인 <-> 폴리곤	서울에 속한 지하철 노선은?
시나리오 5	폴리곤 <-> 폴리곤 (행정구역의 인접관계)	영월군과 인접한 시군구는?
시나리오 6	폴리곤 <-> 폴리곤 (행정구역 사이의 포함관계)	옥천군이 속한 도는?
시나리오 7	폴리곤 <-> 폴리곤 (행정구역과 자연물 중첩관계)	한강이 흐르는 법정동은?

2.2 GeoKG 설계

다양한 POI를 다루며 효율적인 방법으로 GeoKG를 구축하기 위해 본 연구에서는 기존의 GeoKG와 공공데이터를 융합하는 방식으로 GeoKG를 구축하였다. WorldKG는 오픈스트리트맵을 기반으로 제작하여 다목적 지식그래프나 공공데이터 등에서 다루지 않는 POI를 다루고 있기에 다루는 질의의 범위를 고려했을 때 구축의 용이성 측면에서 적합하다고 판단하여 기반이 되는 GeoKG로 WorldKG를 선정하였다. WorldKG에 공공데이터를 사용하여 행정구역 및 주요 POI 등의 정보와 질의 데이터셋을 분석해 선정한 공간 관계를 추가하였다.

2.1절의 [표 2-3]에서 분석한 바와 같이 질의 데이터셋에 포함된 라인 데이터와 관련된 질의가 포인트, 폴리곤 데이터에 비해 매우 적어 본 연구에서는 GeoKG를 구축할 때 라인 데이터를 추가하지 않았다. 또한, 산, 하천, 호수 등의 여러 폴리곤 데이터 중에서 시범적으로 행정구역과 하천만을 구축하였다.

2.2.1 행정구역 정보 구축

GeoQuestions201에 속한 대부분의 질의가 시도, 시군구 단위의 질문이고, '읍면동' 단위에 해당하는 'Village'는 전체 질의 중 3개의 질의만이 포함되어있지만, 풍부한 쿼리를 지원하기 위해 행정구역의 데이터는 시도, 시군구, 읍면동 단위까지 구축하였다. 읍면동의 정보는 국가공간정보포털[9]에서 법정동을 기반으로 쉐이프파일을 제공하여 통계분류포털[10]의 법정동 기준 행정구역 분류표를 기반으로 행정구역 노드를 생성하고 관계를 연결하였다. WorldKG에 행정구역 노드가 존재하지만, 같은 행정구역명을 가진 행정구역에 대한 구분이 되어있지 않고, 일부 행정구역 노드가 구축되어있지 않았다. 따라서 행정구역 정보 구축 및 관계 연결 과

9) http://data.nsdi.go.kr/dataset/
10) https://kssc.kostat.go.kr/

정에서 같은 행정구역이 두 번 구축되거나, 관계가 없는 노드끼리 관계가 생성되는 문제를 방지하고자 기존의 노드를 삭제한 후 새로운 정보를 기준으로 새롭게 정보를 구축하였다.

행정구역 정보를 구축하는 과정에서 다음과 같은 문제가 발생하였다.

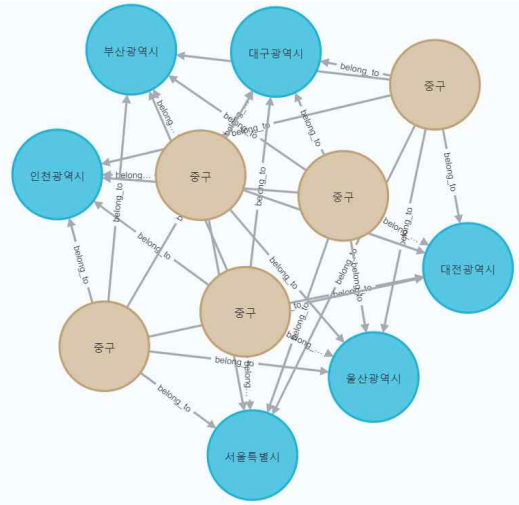
2.2.1.1 중복 행정구역 문제

‘중구’라는 지명을 가진 행정구역은 서울특별시, 인천광역시, 부산광역시, 대구광역시, 대전광역시, 울산광역시 총 6곳에 존재한다. 이외에도 동구, 북구, 서구 남구, 강서구 등 많은 행정구역의 이름이 중복된다. 행정구역 정보를 먼저 구축한 후 포함관계를 연결할 경우, 같은 행정구역명을 가진 행정구역 사이에 [그림 2-4]의 (a), (c)와 같이 관련된 모든 행정구역에 관계가 생성되는 문제가 발생하였다.

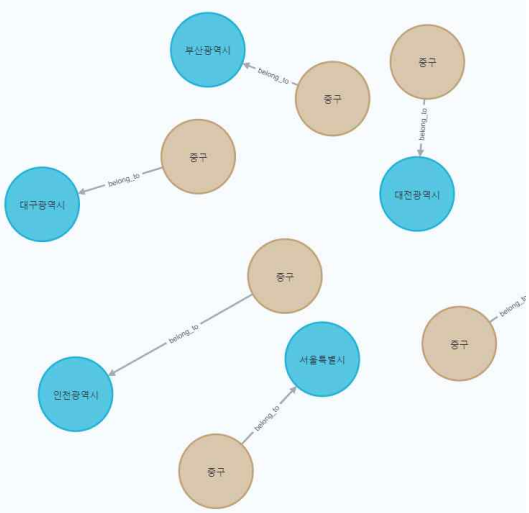
[그림 2-2]의 (a)는 동일한 행정구역명을 가진 서로 다른 행정구역 정보를 구축할 때 생기는 문제이다. ‘중구’라는 행정구역에 대한 정보를 구축할 때 서울특별시, 인천광역시 등 각 상위행정구역별로 한 개의 관계가 구축되어야하지만, 생성된 모든 ‘중구’ 노드가 관련된 모든 상위 행정구역과 연결되어있음을 볼 수 있다.

[그림 2-2]의 (c)는 상위 행정구역명이 같을 때 생기는 문제이다. 예를 들어, ‘서울특별시 중구 명동’과 ‘부산광역시 중구 남포동’의 정보를 구축할 때 ‘명동’은 ‘서울특별시’에 속한 ‘중구’와 연결되어야 하고, ‘남포동’은 ‘부산광역시’에 속한 ‘중구’와 연결되어야 하지만 모든 ‘중구’와 연결되어있음을 확인할 수 있다.

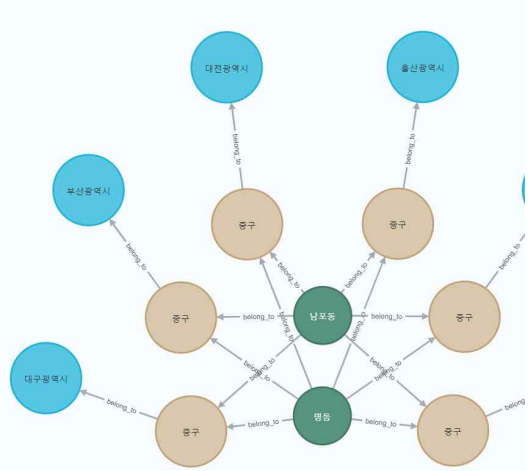
본 연구에서는 행정구역 노드를 구축할 때 2.2.3절에 서술할 ‘Belong_to’관계를 연결하여 잘못된 관계의 생성을 방지하였다. ‘서울특별시 중구’에 대한 정보를 구축할 때 ‘중구’ 노드를 구축하며 ‘서울특별시’와 ‘Belong_to’ 관계를 연결하였고, ‘서울특별시 중구 명동’의 노드를 생성할 때는 ‘서울특별시’ 노드와 ‘Belong_to’ 관계로 연결된 ‘중구’ 노드를 찾고, 해당 ‘중구’ 노드와 ‘명동’ 노드를 ‘Belong_to’ 관계를 연결하였다.



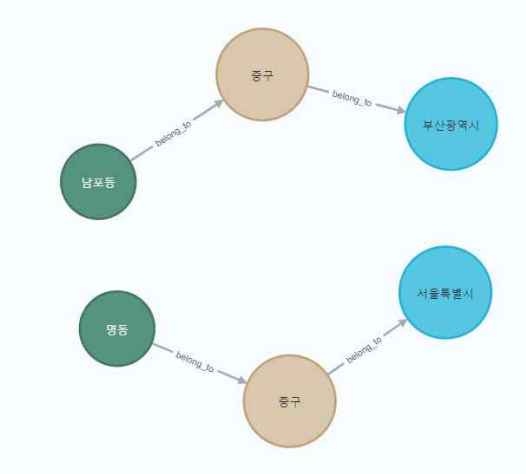
(a) 같은 행정구역명을 가진 노드를 구축할 때의 문제



(b) 같은 행정구역명을 가진 노드를 올바르게 구축한 결과



(c) 상위 행정구역명이 같을 때 생기는 문제



(d) 상위 행정구역명이 같은 노드를 올바르게 구축한 결과

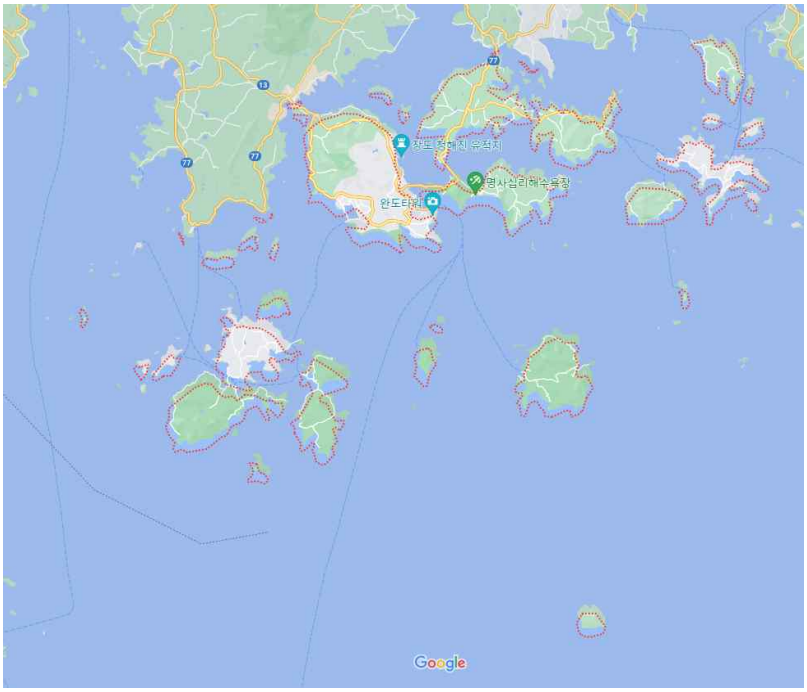
[그림 2-2] 중복행정구역명으로 인한 구축 시 문제점

2.2.1.2 폴리곤 정보 적재

Mai et al.(2021)은 기하학적 정보 데이터의 부재로 인해 지리공간질의에 답하는데 한계가 있다고 하였다. 따라서 본 연구에서는 GeoQA 분야의 한계를 극복하고자 폴리곤 사이의 최단거리를 구하는 질의 또는 폴리곤 면적을 구하는 질의 등 사전연산된 관계 이외의 질의에도 풍부하게

답할 수 있도록 행정구역에 폴리곤 정보를 추가하였다. 행정구역의 쉐이프파일을 QGIS를 활용해 GeoJSON 파일로 변환 후 포인트 집합의 형태로 폴리곤 정보를 저장하였다. GeoJSON 파일은 포인트, 라인, 폴리곤 등의 요소를 기하학적 정보(위경도 좌표)의 테이블(table)로 표현하는 파일 형식이다.

폴리곤 데이터는 읍면동(법정동 기준) 단위만 구축하였다. 국가공간정보포털의 경우, 시군구와 읍면동 쉐이프파일의 데이터 제공기관은 국토교통부이고 시도 쉐이프파일의 데이터 제공기관은 국토지리정보원으로 데이터 제공기관이 다르다. 상이한 데이터 제공기관에서 제공하는 데이터를 사용할 때 행정구역의 경계가 다른 문제를 방지하기 위해 국토교통부에서 제공하는 읍면동 파일만을 사용하여 폴리곤 데이터를 구축하였다. 만일 포인트가 속한 시도, 시군구 단위의 폴리곤을 찾는 연산(point in polygon)을 수행할 때 시도 또는 시군구 단위의 폴리곤에 직접 연산하는 방법 대신 읍면동 단위에서 point in polygon 연산을 수행하고, 해당 폴리곤이 속한 시도 또는 시군구를 'Belong_to' 관계를 통해 찾는 방식이다.



(a) 완도군의 행정구역 지도

{"type":"Feature","geometry":{"type":"MultiPolygon","coordinates":[[[[126.92181359437245,33.98876244691312]

.

.

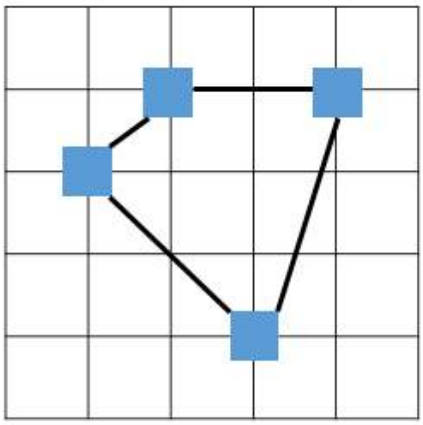
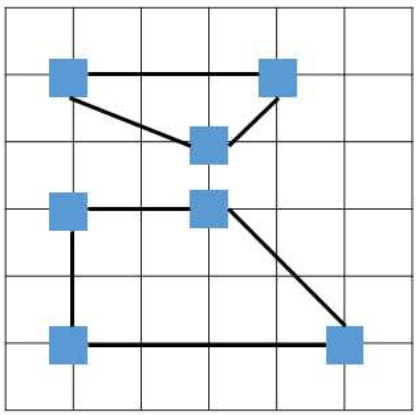
.

[127.04773613842956,34.46062373552936]]]],"properties":{"SIG_CD":"46890","SIG_ENG_NM":"Wando-gun","SIG_KOR_NM":"완도군"}}

(b) 완도군의 GeoJSON

[그림 2-3] 완도군의 행정구역 지도 및 GeoJSON 파일

폴리곤 정보를 적재하는 과정에서 [그림 2-3 (a)]의 완도군과 같이 섬이 많거나 경계가 복잡한 행정구역은 [그림 2-3 (b)]와 같이 멀티폴리곤 형식으로 구축되어있음을 확인하였다. [그림 2-4]는 GeoJSON 형식에서 폴리곤 및 멀티폴리곤의 표현 방식과 대응하는 도형의 예시이다. GeoJSON 형식에서 멀티폴리곤은 개별 폴리곤의 집합임을 확인할 수 있다.

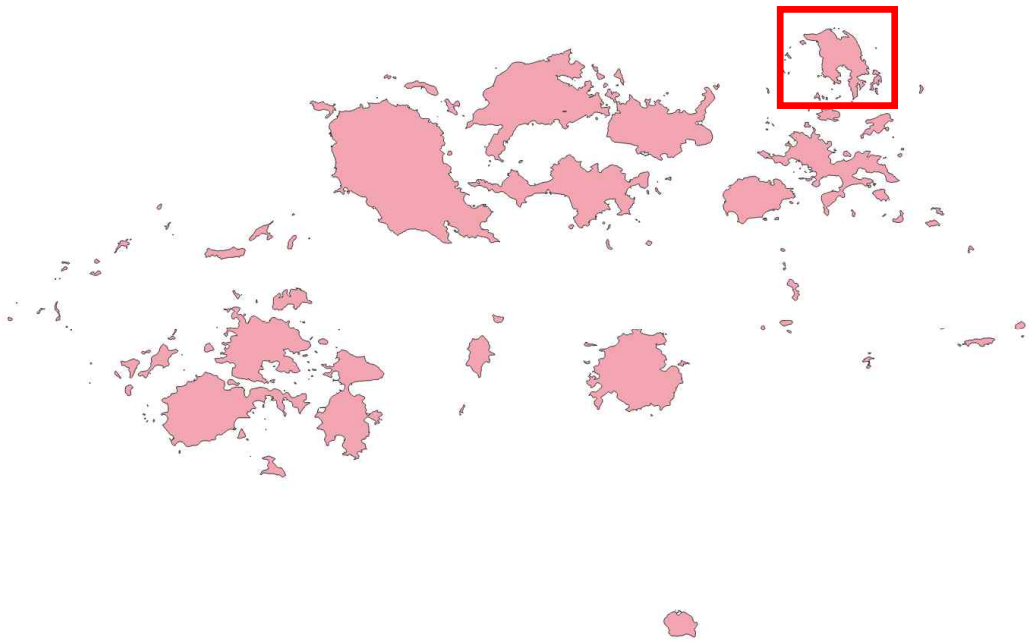
형식	GeoJSON 예시	GeoJSON이 표현하는 도형
폴리곤 (Polygon)	{ “type”: “Polygon”, “coordinates”: [[[3, 1], [4, 4], [2, 4], [1, 3], [3, 1]] }	
멀티폴리곤 (MultiPolygon)	{ “type”: “MultiPolygon”, “coordinates”: [[[[3, 4], [4, 5], [1, 5], [3, 4]]], [[[5, 1], [3, 3], [1, 3], [1, 1], [5, 1]]]] }	

[그림 2-4] 폴리곤 및 멀티폴리곤의 GeoJSON 예시

대부분의 그래프 데이터베이스는 멀티폴리곤 형식의 지원에 대한 기술적인 문서를 지원하지 않았다. 따라서 데이터베이스와 관계없이 멀티폴리곤에 대해 ‘Point in polygon(within)’ 연산이나 폴리곤 면적 계산 등의 폴리곤 연산을 지원하기 위해 멀티폴리곤을 폴리곤으로 분리하여 적재하였다.

평면상의 기하 정보를 다루는 ‘Shapely’ 파이썬 라이브러리를 사용해 [그림 2-5 (a)]와 같은 멀티폴리곤을 [그림 2-5 (b)]와 같이 서로 다

른 폴리곤으로 분리한 새로운 GeoJSON 파일을 생성한 후 Neo4j에 적재하였다. 기존 GeoJSON 파일에 행정구역마다 속성정보로써 법정동코드와 행정구역의 한국어 지명 및 영어 지명이 포함되어있다. 멀티폴리곤을 각각의 폴리곤으로 분리한 후 데이터를 적재할 때 행정구역을 구분하기 위해 본래의 멀티폴리곤에 포함되어있던 속성정보를 각 폴리곤의 속성정보로 추가하였다.



(a) 분리 전 멀티폴리곤(완도군)

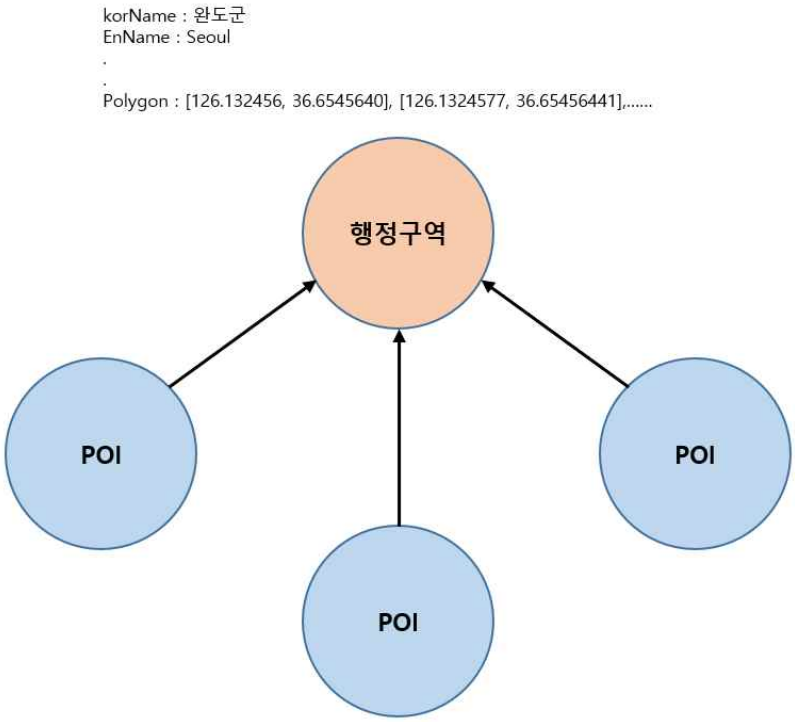
(b) 멀티폴리곤 분리 후 일부 폴리곤

[그림 2-5] 멀티폴리곤 분할 예시

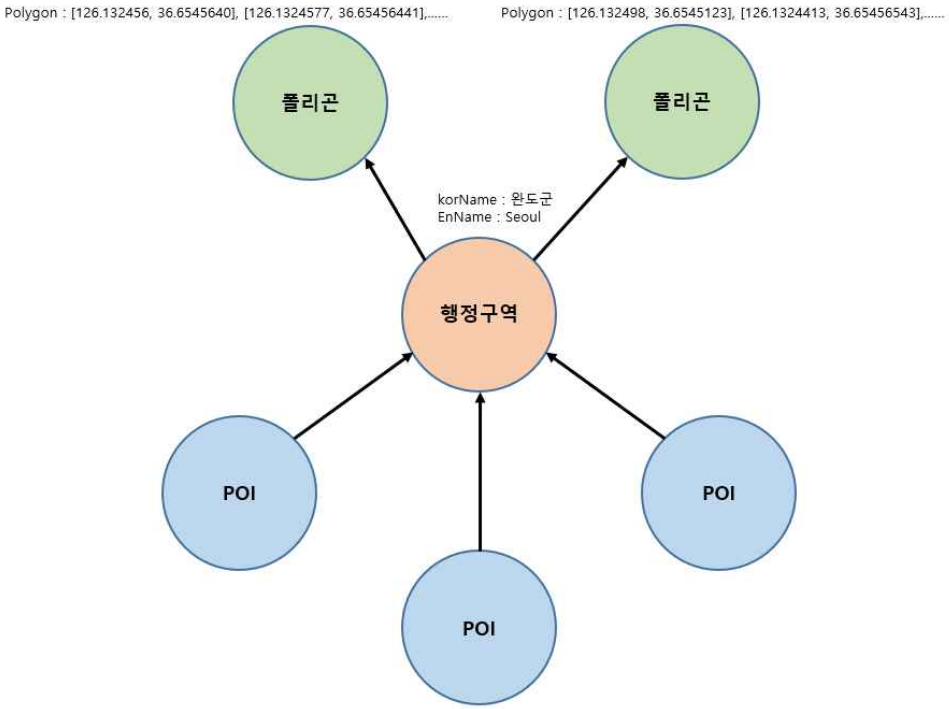
[그림 2-6]은 멀티폴리곤 분리 전후의 적재 방식의 예시이다. [그림

2-6 (a)]와 같이 멀티폴리곤 형식의 데이터를 행정구역 노드의 속성으로 추가하는 대신 [그림 2-6 (b)]의 예시처럼 멀티폴리곤을 개별 폴리곤으로 분리한 후 각 폴리곤마다 독립적인 폴리곤 노드를 생성하고, 해당 폴리곤 노드에 폴리곤 데이터를 속성으로 적재한 후 행정구역 노드와 연결하는 방식으로 멀티폴리곤 데이터를 행정구역 노드에 정보를 추가하였다.

GeoJSON 파일의 속성정보의 행정구역명은 '도화동', '오포읍'과 같이 상위 행정구역명이 병기되어있지 않아 같은 행정구역명을 가진 다른 노드에 정보가 적재될 수 있다. 따라서, 본 연구에서는 Neo4j의 행정구역 노드에 폴리곤 데이터를 구축할 때 행정구역명 대신 고유한 번호를 가진 법정동 코드를 사용하여 행정구역 노드와 매치(match)하였다.



(a) 분리 전 행정구역의 폴리곤 정보 적재 예시



(b) 분리 후 행정구역의 폴리곤 정보 적재 예시

[그림 2-6] 멀티폴리곤 적재 방식 예시

2.2.2 POI 정보 구축

합리적인 노드 및 관계 구축을 위하여 음식점과 같이 변화가 잦거나 대학교, 관공서와 같이 중요하다고 판단되는 대표 POI를 선정하고, 지방행정 인허가데이터개방[11]과 국가공간정보포털[12]의 데이터를 기반으로 POI 데이터를 취득하였다. POI 선정 기준은 국내 실정을 반영한 오성호(2006)와 매년 POI 분류표를 발행하는 Ordanance Survey(2021)의 분류를 참고하였다.

Ordanance Survey(2021)는 9개의 대분류, 52개의 중분류, 600개 이상의 소분류로 POI를 세분화하였다. 대분류 및 대표 중분류는 [표 2-6]과 같다.

[표 2-6] Ordanance Survey(2021)의 POI 대분류 및 중분류

대분류	중분류
숙박시설 및 음식점 (Accommodation, eating and drinking)	Accommodation, Eating and drinking
상업 서비스 (Commercial services)	Construction services, Consultancies, Employment and career agencies, Engineering services, Contract services, IT, advertising, marketing and media services, Legal and financial, Personal, consumer and other services, Property and development services, Recycling services, Repair and servicing, Research and design, Transport, storage and delivery, Hire services

11) https://www.localdata.go.kr/
12) http://www.nsdi.go.kr/lxportal/

관광지 (Attractions)	Botanical and zoological, Historical and Cultural, Recreational, Landscape features, Tourism, Bodies of water
스포츠 및 오락 (Sport and entertainment)	Sport and entertaiment support services, Gambling, Outdoor pursuits, Sports complex, Venues, stage and screen
교육 및 건강 (Education and health)	Animal welfare, Education support services, Health Practitioners and establishments, Health support services, Primary, secondary and tertiary education, Recreational and vocational education
공공 시설 (Public infrastructure)	Central and local government, Infrastructure and facilities, Organisations
제조 및 생산 (Manufacturing and production)	Consumer products, Extractive industries, Farming, Foodstuffs, Industrial features, Industrial products
소매 (Retail)	Clothing and accessories, Food, drink and multi item retail, Household, office, leisure and garden, Monitoring
운송 (Transport)	Air, Road and rail, Water, Public transport, stations and infrastructure, Bus transport

오성호(2006)는 대상의 성격에 따라 총 8개의 대분류, 43개의 중분류, 448개의 소분류, 1,373개의 세분류로 POI를 분류하였으며 이는 [표 2-7]과 같다.

[표 2-7] 오성호(2006)의 POI 분류체계

대분류	중분류	비고
교통편의	교통시설, 자동차시설	공항, 기차역, 지하철역, 버스터미널 등
생활편의	음식점, 카페, 술집, 쇼핑, 생활서비스, 가정의례, 종교, 문화생활시설, 서비스센터, 법률서비스, 기타서비스	음식점, 쇼핑, 예식장, 장례식장, 극장 등
여행/레저	숙박, 관광명소, 데이트, 레저/스포츠	호텔, 동물원, 스키장, 야구장, 산 등
의료편의	의료시설, 기타의료시설	종합병원, 일반의원, 한의원, 보건소 등
금융편의	주요 금융, 제2금융	주요 은행, 보험사, 증권사, 새마을금고 등
공공편의	공공기관, 기타공공기관, 교육기관, 언론기관, 외국기관, 기타	경찰서, 법원, 시/군/구청, 각종 정부기관 등
기업/단체	기업, 공업단지, 각종단체, 연구기관, 사회시설, 연수원, 기타	상장기업, 각종 공사, 그룹, 단체 등
기타	주요건물, 농/어업, 설비, 창고, 차량기지단지, 지명, 지형지물, 기타	주요 건물, 아파트, 지명, 지형지물 등

본 연구에서는 [표 2-6]과 [표 2-7]을 참고하여 총 10개의 대분류 및 대표 POI를 선정하였고, 이를 [표 2-8]에 정리하였다. Ordanance Survey(2021)의 분류체계를 기반으로 오성호(2006)의 분류체계를 일부 반영하여 대분류를 선정하였다. 예를 들어, Ordanace Survey(2021)의 '숙박 시설 및 음식점'을 '숙박 시설'과 '음식점'으로 나누었고, '음식점'은 오성호(2006)의 '생활편의' 대분류에 속하여 대분류로 따로 분류하지 않았다. 오성호(2006)의 '기업/단체' 분류는 '기타'의 분류와 의미가 일부 중복되고, Ordanace Survey(2021)의 '상업 서비스'는 '운송'과 '제조 및 생산'

분류와 일부 중복되어 대분류에 포함하지 않았다. 이외에 ‘교통’, ‘여행’, ‘레저’, ‘공공편의’, ‘교육’은 [표 2-6]과 [표 2-7]에 모두 포함되어 있어 대분류로 선정하였다. ‘주거 시설’은 오성호(2006)의 ‘기타’에 속해있는 ‘아파트’와 Ordanance Survey(2021)의 ‘숙박시설’을 합하여 분류하였다.

컴퓨팅 자원의 한계로 본 연구에서는 모든 POI에 대한 정보를 구축하는 방법 대신 대표 POI를 선정해 시범적으로 구축하여 실험을 진행하였다.

[표 2-8] 선정한 대분류 및 대표 POI

대분류	대표 POI
숙박시설	호텔
생활 편의 시설	일반음식점
교통 편의 시설	지하철역
의료 시설	병원
레저 시설	종합체육시설
관공서	경찰서
주거 시설	아파트
교육 시설	대학교
유통 편의 시설	대형쇼핑몰
여행 시설	놀이공원

2.2.3 관계 생성

GeoKG에 적재할 관계는 GeoQuestions201(Punjani et al., 2018)와 Hamzei et al.(2019)을 참고하였다.

GeoQuestions201을 분석할 때 관계를 세분화해서 정의하였는데 그 내용 및 빈도수는 [표 2-9]와 같다. 명확한 분류를 위해 “경기도에서 가장 큰 군은?(Which is the biggest county in the Gyeonggi-do?)”과 같이 질의에 ‘in’이라는 단어가 포함되어있어도 폴리곤과 폴리곤 사이의 포함 관계이기에 ‘BELONG_TO’로 분류하였다. 또한, 2.2에서 분석한 바와 같이 필요한 모든 공간 관계를 정의하였다. 예를 들어, “강원도에 인접한 충청북도의 읍면동은?”과 같은 질의는 ‘충청북도’라는 행정구역과 하위 행정구역인 ‘읍면동’과의 관계인 ‘BELONG_TO’와 ‘충청북도’와 ‘강원도’의 ‘BORDER’의 관계를 포함한다.

[표 2-9] GeoQuestions201 상의 공간 관계 분류 및 출현횟수

공간 관계	상세	기하학적 정보 관계	출현횟수
In	행정구역과 POI 포함관계	포인트 <-> 폴리곤	103
Near	POI와 POI 간의 인접(거리) 관계	포인트 <-> 포인트	35
Belong to	행정구역 간의 포함관계	폴리곤 <-> 폴리곤	34
Largest/Highest/Longest, Area, Length, Population	속성정보	속성정보	34
Cross	강과 행정구역 간의 인접(포함)관계	폴리곤 <-> 폴리곤	22
Border	행정구역 간의 인접관계	폴리곤 <-> 폴리곤	13
Azimuth	행정구역 간의 방위관계	폴리곤 <-> 폴리곤	12
Radius	행정구역, POI, 강 등으로부터 반경	포인트<->포인트, 포인트<->폴리곤, 라인<->폴리곤	12
Closest	포인트/폴리곤에서 가장 가까운 포인트 관계	포인트<->포인트, 포인트<->폴리곤	4
Distance	객체 사이의 거리관계	포인트<->포인트, 포인트<->폴리곤	3

Hamzei et al.(2019)은 MS Marco와 GeoQuestions201의 질의 데이터셋에 등장하는 공간관계를 분석하였고 [표 2-10]의 결과를 도출했다.

[표 2-10] MS Marco 및 GeoQuestions201의 공간관계 및 출현횟수(Hamzei et al., 2019)

공간관계	
질의에 포함된 관계	**응답에 포함된 관계**
In (3916)	In (10851)
Near (153)	On (379)
At (142)	At (362)
On (109)	Near (275)
Between(38)	Between (251)

[표 2-10]의 분석에서는 단순히 단어를 추출한 방식으로 분류하여 'In'의 경우 본 연구에서 [표 2-9]에 분류한 공간 관계 중 POI와 행정구역 사이의 포함관계를 나타내는 'In'과 행정구역 사이의 포함관계를 나타내는 'Belong_to'를 모두 포함한다. 'At', 'On'은 'In'과 유사한 의미로 사용하는 관계라고 판단하였다.

따라서 본 연구에서는 출현 빈도를 기반으로 [표 2-9]에서 Belong to, Cross, Border의 관계를, [표 2-10]에서 In, Near의 관계를 GeoKG에 저장할 공간 관계로 선정하였다. 'Between'의 경우 'In', 'On' 등의 관계와 비교하였을 때 빈도가 적어 선정하지 않았다.

각 공간 관계별 구축방법은 다음과 같다.

- Belong to

행정구역 간의 포함관계를 나타내는 관계이다. '서울'에 속한 '중구'가 '부산'에 관계가 연결되지 않도록 행정구역 노드를 구축하는 과정에서 관계를 생성하였다.

대분류	시도	중분류	시군구	소분류	읍면동	영문 표기	한자 표기	비고 (7자리)
11	**서울특별시**					Seoul	서울特別市	11
11	**서울특별시**	**11010**	**종로구**			Jongno-gu	鍾路區	11010
11	서울특별시	11010	종로구	11010720	청운효자동	Cheongunhyoja-dong	淸雲孝子洞	1101072
11	서울특별시	11010	종로구	11010530	사직동	Sajik-dong	社稷洞	1101053
11	서울특별시	11010	종로구	11010540	삼청동	Samcheong-dong	三淸洞	1101054
11	서울특별시	11010	종로구	11010550	부암동	Buam-dong	付岩洞	1101055
11	서울특별시	11010	종로구	11010560	평창동	Pyeongchang-dong	平倉洞	1101056
11	서울특별시	11010	종로구	11010570	무악동	Muak-dong	毋岳洞	1101057
11	서울특별시	11010	종로구	11010580	교남동	Gyonam-dong	橋南洞	1101058
11	서울특별시	11010	종로구	11010600	가회동	Gahoe-dong	嘉會洞	1101060
11	서울특별시	11010	종로구	11010610	종로1.2.3.4가동	Jongno 1.2.3.4(ilisamsa)-ga-dong	鍾路1.2.3.4街洞	1101061
11	서울특별시	11010	종로구	11010630	종로5.6가동	Jongno 5.6(oryuk)-ga-dong	鍾路5.6街洞	1101063
11	서울특별시	11010	종로구	11010640	이화동	Ihwa-dong	梨花洞	1101064
11	서울특별시	11010	종로구	11010730	혜화동	Hyehwa-dong	惠化洞	1101073
11	서울특별시	11010	종로구	11010670	창신1동	Changsin 1(il)-dong	昌信1洞	1101067
11	서울특별시	11010	종로구	11010680	창신2동	Changsin 2(i)-dong	昌信2洞	1101068
11	서울특별시	11010	종로구	11010690	창신3동	Changsin 3(sam)-dong	昌信3洞	1101069
11	서울특별시	11010	종로구	11010700	숭인1동	Sungin 1(il)-dong	崇仁1洞	1101070
11	서울특별시	11010	종로구	11010710	숭인2동	Sungin 2(i)-dong	崇仁2洞	1101071

[그림 2-7] 서울시의 행정구역 분류표 예시

예를 들어, [그림 2-7]의 행정구역분류표에는 '서울특별시, 종로구, 사직동'의 형태로 데이터가 구축되어있을 때 행정구역 노드를 생성하며 '서울특별시<-[r:BELONG_TO]-종로구', '종로구<-[r:BELONG_TO]-사직동'과 같이 상위 행정구역 노드를 연결하였다. 행정구역 노드를 연결하는 과정에서 발생하는 중복 행정구역에 대한 문제를 겪지 않고 올바른 포함관계를 생성하였다.

- In

POI와 행정구역의 포함관계를 나타내는 관계로, Neo4j spatial[13] 플러그인의 'withinPolygon' 연산을 수행해 POI가 속한 읍면동 단위의 행정구역을 연결하였다. 예를 들어, "사하구에 속해있는 병원 개수는?"과 같은 질의를 응답하려면 '사하구'에 속한 법정동을 찾고, 각 법정동별로 속해있는 병원의 개수를 합해 대답을 반환한다.

- Near

Binchuan et al.(2019)은 공간 객체별로 '가깝다', '멀다'의 기준을 공간 객체별로 임의로 설정하였으며, Punjani et al.(2018) 또한 POI에 따라 '가깝다'의 범위를 임의로 설정하였다.

13) https://github.com/neo4j-contrib/spatial-algorithms

본 연구에서는 관계 데이터가 기하급수적으로 증가하는 것을 방지하기 위해 대표 POI에 대해서만 'Near' 거리 기준을 설정하고 관계를 연결하였다. Punjani et al.(2018)은 'Near'의 범위를 [표 2-11]과 같이 선정하였다.

[표 2-11] Punjani et al.(2018)의 'Near'의 거리 기준

POI 명	식당	도시	호텔	랜드마크	공원
기준	500m	5km	1km	1km	500m

음주 장소와 범죄율 사이의 관계를 분석한 Elizabeth(2011)는 대중교통시설까지 걸어갈 수 있는 최대 거리를 402m(0.25 마일)로 제시하였다. 이는 마일 단위가 아닌 미터 단위로 변환하였을 때, Punjani et al.(2018)이 설정한 식당에서 'Near'의 기준인 500m와 근사한 것을 확인할 수 있다. 따라서 본 연구에서는 Punjani et al.(2018)의 기준을 참고하여 'Near'의 범위를 [표 2-11]에 정리하였다. "호텔 근처 병원", "주차장 가까이에 있는 편의점"과 같은 거리 기준이 모호한 질의에 답을 하기 위한 공간 관계이다. [표 2-11]의 '식당'과 '공원'의 기준을 참고하여 본 연구에서는 일반음식점, 아파트, 대학교의 'Near' 기준을 500m로 설정하였다. 또한 [표 2-11]의 '호텔'과 '랜드마크'의 기준을 참고하여 호텔, 지하철역, 경찰서의 'Near' 기준을 1km로 설정하였고, 그 이외의 규모가 큰 병원, 종합체육시설, 놀이공원, 대형쇼핑몰의 기준은 2km로 설정하였다.

Neo4j의 두 점 사이의 거리를 구하는 'distance' 연산을 사용하여 기준 거리 안의 모든 POI 데이터와의 관계를 연결하였다.

[표 2-12] POI 별 'Near' 기준

	호텔	일반 음식점	지하철역	병원	종합 체육 시설	경찰서	아파트	대학교	대형 쇼핑몰	놀이 공원
기준	1km	500m	1km	2km	2km	1km	500m	500m	2km	2km

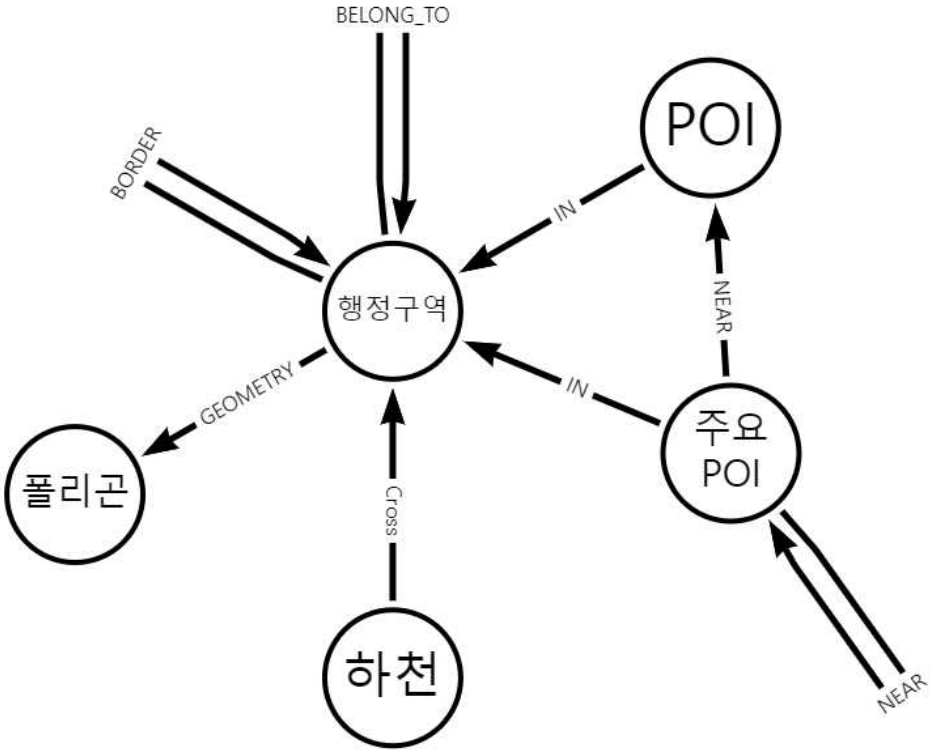
- Border

행정구역간의 인접한 관계를 나타내기 위한 관계이다. GeoSPARQL을 사용해 인접 정보를 추출할 때, Regalia et al.(2019)이 서술한 바와 같이 미세폴리곤과 같은 데이터의 오류로 인해 답을 반환하지 못하거나 실제와 다른 답을 반환할 수 있다. 따라서 본 연구에서는 시군구 및 읍면동 사이의 인접관계를 저장할 때 내려받은 시군구 및 법정동의 쉐이프 파일을 ArcGIS의 'Repair geometry' 기능을 사용해 공간연산 전에 폴리곤을 수정한 뒤, 'Intersection' 공간연산을 수행해 인접한 폴리곤 데이터를 취득한 후 관계를 생성하였다.

- Cross

자연물과 행정구역 간의 인접 또는 교차 관계를 나타내기 위한 관계이다. 국가공간정보포털의 하천망도 데이터(한강홍수통제소 제공) 중 국가하천에 해당하는 강들을 대상으로 관계를 생성하였으며, 국가하천과 시군구 사이의 관계만 구축하였다. 위의 'Border'와 같은 방식으로 ArcGIS의 'Repair geometry'와 'Intersection' 연산을 수행해 강과 맞닿아 있거나 강이 흐르는 행정구역의 데이터를 취득해 관계를 생성하였다.

노드 및 관계들을 최종적으로 적재하였을 때, 본 논문에서 구축한 GeoKG의 스키마는 [그림 2-8]과 같다.

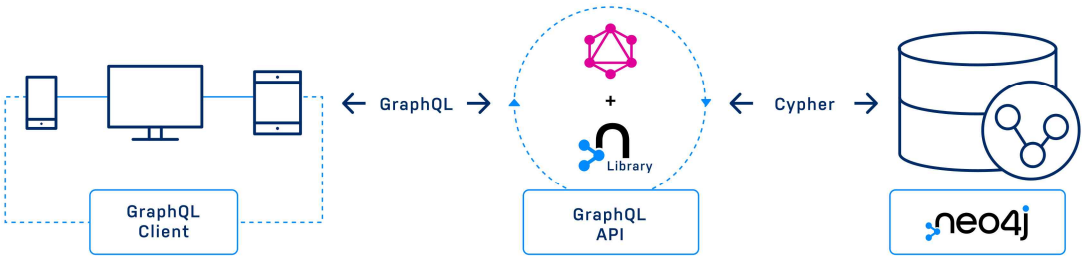


[그림 2-8] 본 연구에서 구축한 GeoKG의 스키마

2.3 GraphQL

속성그래프는 RDF 데이터 형식과 비교했을 때 데이터를 표현하기 용이하고, 스키마가 비교적 자유롭다는 장점이 있다. 하지만 RDF는 데이터를 검색하고 데이터를 수정할 수 있는 표준화된 질의언어인 SPARQL이 있는데 반해 속성그래프는 데이터베이스마다 서로 다른 질의언어를 채택하고 있어 각 데이터베이스에 맞는 질의로 변환해야 하는 단점이 있다.

GraphQL은 각 속성그래프 데이터베이스의 질의언어에 의존하지 않고 일정하게 쿼리를 수행할 수 있어 속성그래프 데이터베이스의 범용성 문제를 해결할 수 있다. 따라서 본 연구에서는 다양한 속성그래프 데이터베이스에서 작동할 수 있도록 설계된 GraphQL을 사용하여 구축한 GeoKG를 질의하였다.



[그림 2-9] Neo4j와 GraphQL API 사용 흐름도

[그림 2-9]는 사용자가 GraphQL을 사용해 Neo4j에 저장되어있는 데이터베이스에 접근하는 예시이다.

구축한 GeoKG를 GraphQL에서 동작하기 위해서는 GeoKG의 스키마 파일이 필요하다. Introspector[14]는 Neo4j의 스키마를 추출하는 라이브러리이다. 본 연구에서는 Introspector 라이브러리를 이용해 구축한 GeoKG의 스키마 파일을 제작하였고, 스키마 파일의 예시는 [그림 2-10]과 같다.

14) https://neo4j.com/docs/graphql-manual/current/introspector/

```
type Gu {
        area: String!
        borderGus: [Gu!]! @relationship(type: "Border", direction: OUT)
        dong_num: String!
        belongToCities: [City!]! @relationship(type: "belong_to", direction:OUT)
        gusBorder: [Gu!]! @relationship(type: "Border", direction: IN)
        name: String!
        name_En: String!
        point: Point!
        geometryPolygons: [Polygon!]! @relationship(type: "Geometry", direction:OUT)
        population: String!
}
```

[그림 2-10] Introspector를 통해 추출한 GeoKG 스키마의 예시

[그림 2-10]은 스키마 중 행정구에 대한 임의의 예시이다. 노드의 속성인 면적, 지명, 영문지명, 포인트, 인구 정보 등이 스키마에 반영되었다. 면적과 이름 등의 정보는 문자열 형식으로, 위치 정보는 포인트 형식으로 적재되어있고, 스키마에 반영되었음을 확인할 수 있다.

공간 관계인 'Border'와 'belong_to', 폴리곤 정보를 나타내는 관계인 'Geometry'도 스키마에서 확인할 수 있다. 'belongToCities'는 'City' 노드와 연결되어있으며 관계의 종류는 'belong_to', 관계(direction)의 방향은 행정구에서 도시 노드로 향함을 보여준다.

GraphQL 인터페이스에서는 스키마 파일을 기반으로 데이터베이스에 접근해 질의할 수 있다. [그림 2-11]은 관악구가 속한 시의 이름을 묻는 Neo4j의 Cypher 질의문과 GraphQL 질의문의 예시이다.

```
match (n:Gu {name:"관악구"})-[r:belong_to]→(m:City)
return m.name
```

(a) Neo4j Cypher를 이용한 쿼리

```
query {
 gus(where: {name:"관악구"}){
    belongToCities{
      name
    }
  }
}
```

(b) GraphQL을 이용한 쿼리

[그림 2-11] Neo4j Cypher와 GraphQL의 쿼리문

3. GeoKG 구축 및 결과

3.1 실험 수행

본 연구에서는 2장에서 제안한 방법론을 토대로 대한민국 전역을 대상으로 새로운 GeoKG를 구축하였다. 데이터베이스로는 Neo4j 4.0.7 Enterprise 버전을 사용했으며, GeoJSON 변환 및 공간연산은 QGIS 및 ArcGIS로 수행하였다. 실험 환경은 아래와 같다.

OS: Microsoft Windows 10 Education

CPU: Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz, 4 코어, 4 논리프로세서

RAM: 8.00GB

3.1.1 GeoKG 추출 및 전처리

WorldKG github에서 제공하는 모듈과 Geofabrik[15]에서 제공하는 대한민국 지역의 osm.pbf 형식의 파일을 사용해 대한민국 지역에 해당하는 WorldKG 데이터를 추출하였다. Neo4j에 RDF 데이터 형식의 WorldKG를 적재하기 위해 Neo4j의 'neosemantics' 플러그인[16]으로 데이터를 불러왔다. 데이터의 포인트 정보는 WGS84 좌표계를 사용했다.

WorldKG의 대학교를 지칭하는 노드 라벨이 'AmenityCollege' 및 'AmenityUniversity'와 같이, 한 공간 객체가 서로 다른 용어로 분류되어 있는 사례가 일부 존재하였다. 대표 POI 정보를 적재하는 과정에서 GeoKG 구축 후 같은 객체를 지칭하는 노드가 서로 다른 라벨로 존재하지 않도록 대표 POI와 같은 의미를 가진 노드 라벨을 통합하였다.

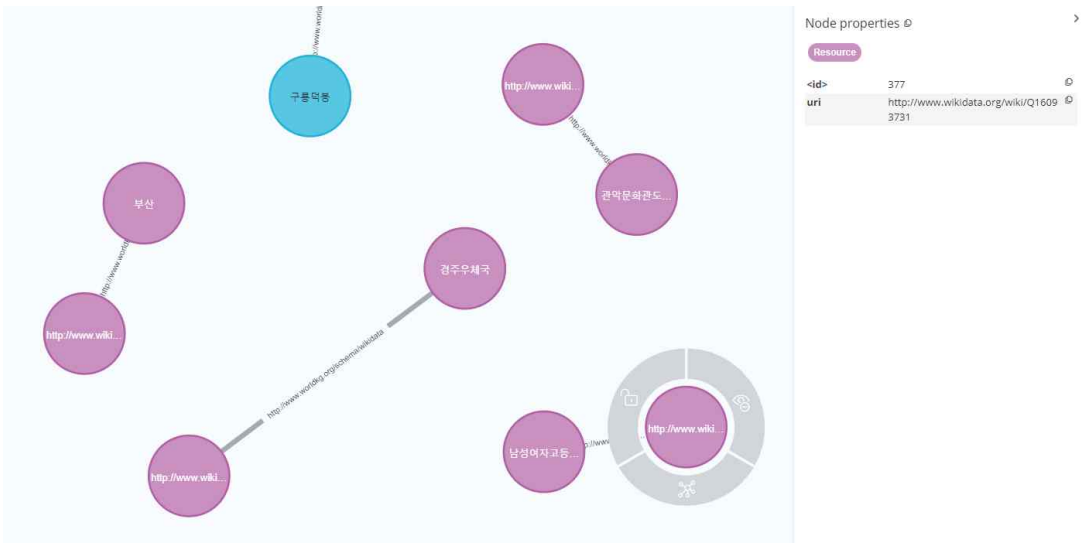
---

15) https://www.geofabrik.de/

16) https://github.com/neo4j-labs/neosemantics

[그림 3-1]에서 특정 공간 객체의 노드와 대응되는 오픈스트리트맵 및 Wikidata의 URI 정보가 담긴 노드를 확인할 수 있다. 노드가 과도하게 증가하는 것을 방지하기 위해 오픈스트리트맵과 Wikidata의 URI 노드를 삭제하고, 해당 정보를 연결된 공간 객체 노드의 속성으로 추가하였다.

또한, 'type' 관계는 노드의 종류를 정의하는데, WorldKG를 속성그래프 데이터 형식으로 변환하는 과정에서 type 정보들이 노드의 라벨로 정의되어 불필요한 관계이기에 모든 type 관계를 삭제하였다.



[그림 3-1] WorldKG 내의 Wikidata 노드 및 관계 예시

3.1.2 행정구역 정보 구축

대한민국 행정구역 정보는 통계분류포털[17]의 한국행정구역분류를 참고하였다. 구축 과정에서 WorldKG에 이미 구축되어있는 정보와 새로운 정보 사이의 혼용을 방지하기 위해 기존 WorldKG의 행정구역 노드를 삭제한 후 행정구역 정보를 새롭게 구축하였다.

행정구역 분류표를 제공하는 시기와 행정구역 쉐이프파일을 제공하

17) https://kssc.kostat.go.kr

는 시기가 서로 달라 변경내역을 반영하지 못하여 행정구역 노드를 구축하는 과정에서 일부 노드가 생성되지 않는 문제가 발생하였다. 본 논문에서는 최신의 쉐이프파일 업데이트 일자를 기준으로 삼고, 해당 일자의 행정구역 분류표를 사용해 행정구역 정보를 구축하였다. 행정구역 정보는 2022년 7월 1일에 제공된 데이터를 기준으로 구축하였다.

행정구역 노드 5,354개와 폴리곤 노드 30,174개를 합쳐 총 35,528개의 노드를 추가하였다.

3.1.3 POI 정보 구축

지방행정 인허가 데이터[18]는 지속해서 최신의 정보를 업데이트를 하기에 최신성을 가지고 있어 숙박시설, 음식점, 병원, 종합체육시설, 대형쇼핑몰, 놀이공원은 지방행정 인허가 데이터개방의 자료를 사용하였다. 지방행정 인허가 데이터에서 취득할 수 없는 지하철역, 경찰서, 아파트, 대학교의 정보는 공공데이터포털[19]에서 내려받아 정보를 적재하였다. POI의 속성으로 좌표, 전화번호 등의 정보를 추가하였다. 총 527,127개의 노드를 구축하였다.

3.1.4 관계 구축

- In

Neo4j spatial 플러그인의 ‘withInPolygon’ 함수를 사용해 POI의 경위도 좌표와 행정구역의 폴리곤 데이터 사이의 포함관계를 생성하였다. 총 848,810개의 관계를 생성하였다.

- Belong_to

통계분류포털의 한국행정구역분류를 참고해 관계를 생성하였다. 총

18) https://www.localdata.go.kr
19) https://www.data.go.kr

5,337개의 관계를 생성하였다.

- Near

2.2.3에서 정의한 대표 POI별 'Near' 범위를 기준으로 Neo4j에 내장된 거리함수를 이용해 적재되어있는 관계를 생성하고자 하였다. 하지만, 대표 POI의 개수가 많아 'Near'의 관계가 과도하게 많아지는 문제가 발생하였다.

따라서, 대표 POI별로 연결할 POI의 종류를 2개로 제한하여 관계를 생성하였으며 연결할 POI는 무작위하게 선정하였다. 각 POI별로 연결할 POI의 종류 및 기준은 [표 3-1]에 다시 정의하였다. 총 2,723,786개의 관계를 생성하였다.

[표 3-1] POI 별 'Near' 기준 및 연결 POI

	숙박시설	음식점	지하철역	병원	종합체육시설	경찰서	아파트	대학교	대형쇼핑몰	놀이공원
연결 POI	음식점, 놀이공원	지하철역, 아파트	경찰서, 아파트	아파트, 경찰서	대형쇼핑몰, 숙박시설	병원, 대학교	지하철역, 종합체육시설	음식점, 병원	지하철역,	지하철역, 숙박시설
기준	1km	500m	1km	2km	2km	1km	500m	500m	2km	2km

- Border

ArcGIS의 Repair Geometry 함수로 미세 폴리곤 등의 오류를 제거하고, 교차(intersection) 연산으로 취득한 인접 관계를 Neo4j에 적재하였다. 총 14,641개의 관계를 생성하였다.

- Cross

'Border'와 마찬가지로 ArcGIS의 Repair Geometry 함수 및 교차 연산으로 강에 인접한 행정구역의 데이터를 취득해 Neo4j에 적재하였다.

총 1,877개의 관계를 생성하였다.

이외에도 행정구역 노드와 폴리곤 노드를 연결하는 'Geometry' 관계는 30,174개 생성되었다.

각 공간 관계별 생성된 관계의 개수는 [표 3-2]에 정리하였다.

[표 3-2] 공간 관계별 생성된 관계의 개수

관계 종류	개수
In	848,810
Belong_to	5,337
Near	2,723,786
Border	14,641
Cross	1,877
Geometry	30,174
계	3,624,625

3.2 구축 결과

3.2.1 DB 구축 결과

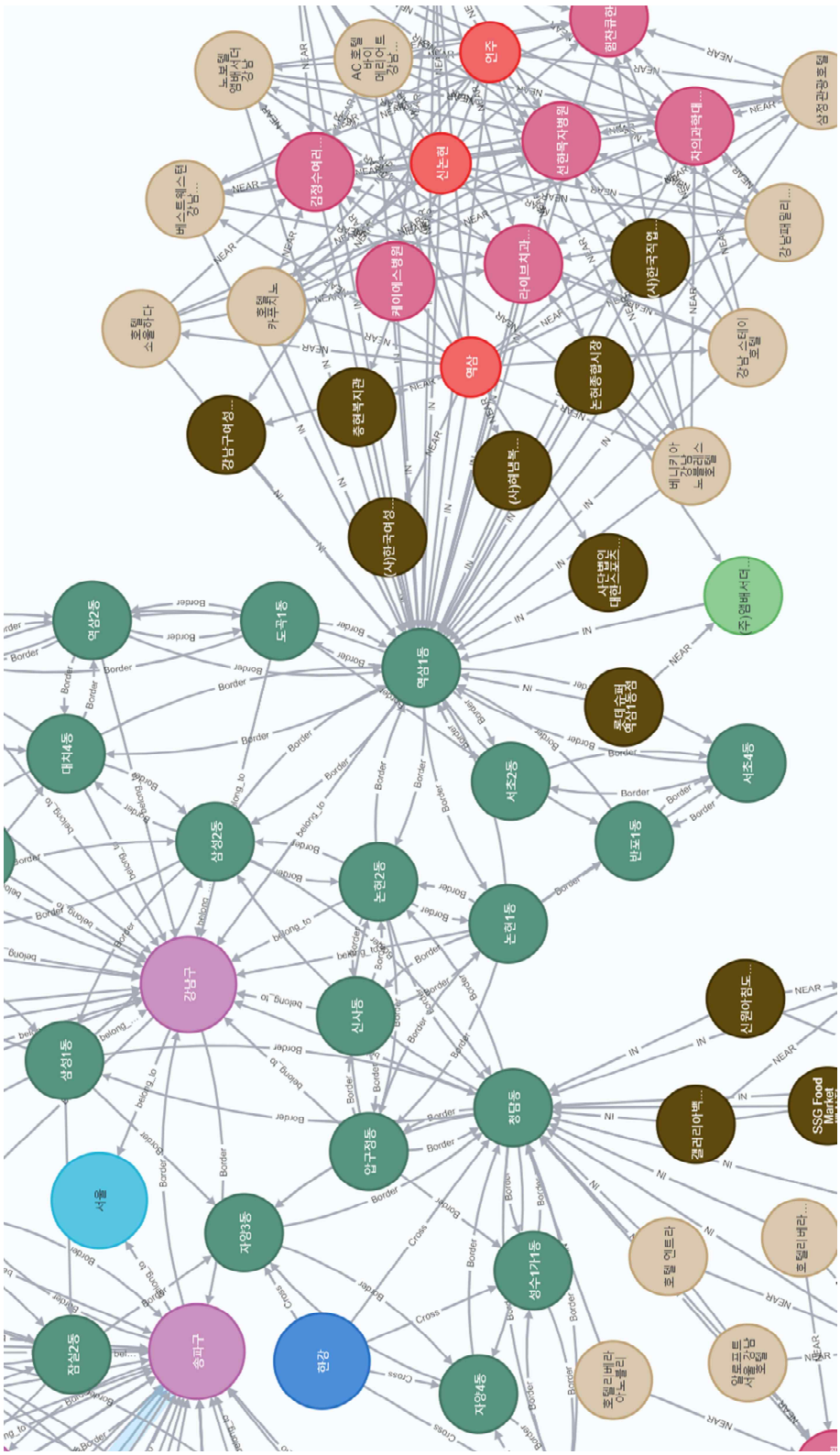
[표 3-3] 구축 전후 GeoKG(WorldKG) 노드 및 관계 비교

	WorldKG	본 연구의 GeoKG
노드 개수	321,683	884,338
라벨(노드 종류) 개수	604	615
관계 개수	0	3,624,625
관계 종류 개수	0	6

[표 3-3]에 WorldKG에 정보를 추가하기 전 노드 및 관계의 개수와 본 연구에서 구축한 GeoKG의 노드 및 관계의 개수를 비교하였다. 주요 POI와 행정구역 노드가 추가됨에 따라 노드의 개수가 562,655개 증가하였다.

관계는 3,624,625개 증가하였다. 관계를 추가하기 전 WorldKG에는 오픈스트리트맵 또는 Wikidata의 링크, 노드의 종류 이외에는 시맨틱한 관계가 존재하지 않았다. 전처리 과정에서 오픈스트리트맵 및 Wikidata의 정보를 속성으로 추가하며 모든 관계를 삭제해 GeoKG 구축 이전의 WorldKG 관계의 개수는 0개이다. 공간 관계 이외에도 폴리곤 정보와 행정구역을 잇는 관계를 추가하여 관계의 종류는 총 6개이다.

[그림 3-2]는 구축한 GeoKG의 예시이다.



[그림 3-2] 본 연구에서 구축한 GeoKG의 예시

3.2.2 GraphQL 질의 결과

본 절에서는 GraphQL을 사용해 2.2절에서 선정한 질의 시나리오를 수행한 결과를 제시하였으며 결과는 [표 3-4]와 같다.

[표 3-4] GraphQL의 시나리오 수행 결과

(a) 시나리오 1

(포인트 <-> 포인트 관계, 동대문역사문화공원역 근처 쇼핑몰은?)

(b) 시나리오 3

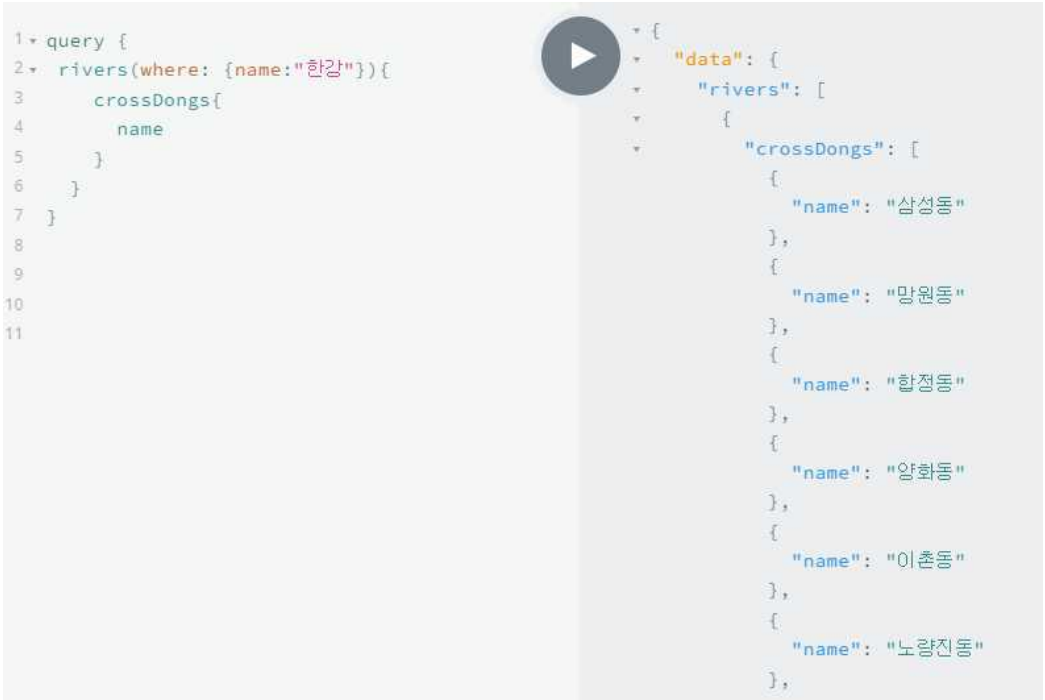
(포인트 <-> 폴리곤 관계, 신촌동에 속한 대학교는?)

(c) 시나리오 5

(행정구역 간의 인접 관계, 영월군과 인접한 시군구는?)

(d) 시나리오 6

(행정구역 사이의 포함관계, 옥천군이 속한 도는?)



(e) 시나리오 7

(행정구역과 자연물의 중첩 관계, 한강이 흐르는 법정동은?)

본 연구에서는 WorldKG에 POI 및 행정구역 데이터를 추가하고, 공간 연산이 필요한 관계를 추가하였다. POI 데이터를 추가하여 시나리오 1에 해당하는 포인트와 포인트 사이의 관계를 성공적으로 수행할 수 있었다. 또한, 행정구역 사이 포함관계를 추가하고 행정구역의 폴리곤 정보를 적재하여 시나리오 3에 해당하는 질의를 수행할 수 있었다.

행정구역 정보를 구축하는 과정에서 행정구역간의 인접관계를 적재하였고, GeoKG상에 적재된 관계를 기반으로 시나리오 5에 해당하는 폴리곤과 폴리곤 사이의 인접관계를 묻는 질의에 응답하였다. ArcGIS를 사용해 전처리 및 교차(intersection)연산을 수행했고, 시나리오 6, 7과 같이 답을 적절히 반환함을 확인할 수 있다.

하지만, 본 연구에서 적재하지 않은 라인 데이터와 관련된 시나리오 2, 4에 대한 답을 반환하지 못하였다.

3.2.3 구축 결과 비교

본 절에서는 기존 GeoKG와 구축한 GeoKG의 factoid question과 geo-analytic question의 대응 여부를 기준으로 비교하였다. WorldKG는 YAGO, YAGO2geo, LinkedGeoData 등의 기존 KG 및 GeoKG가 보유한 지리적 객체의 개수 및 종류가 부족한 단점을 해결하기 위해 구축된 것이기에 factoid question을 비교할 때 WorldKG가 가장 적합하였다. Geo-analytic question을 비교할 때 YAGO2geo는 대한민국 지역을 대상으로 데이터셋을 제공하지 않고, 국토교통부의 LOD는 매우 제한된 종류의 객체를 대상으로 포인트 데이터만을 제공하여 두 GeoKG 모두 부적합하다고 판단하였다. 따라서, factoid question과 geo-analytic question 모두 WorldKG와 비교를 수행하였다.

3.2.3.1 Factoid question 비교

Factoid question은 노드의 개수와 종류로 비교하였다. [표 3-3]에서 분석하였듯, 본 연구에서 구축한 GeoKG는 기존의 WorldKG에 10 종류의 대표 POI를 추가하였으며 그 개수는 527,127개에 달한다. 또한, 시도, 시군구, 읍면동 단위로 행정구역 정보를 구축하였으며, 총 5,354개의 행정구역 정보를 구축하였다. 행정구역의 폴리곤 정보는 총 30,174개 추가하였다.

기존 WorldKG의 노드는 고유한 OSM URI 정보가 존재하였지만, 시맨틱한 정보를 가지고 있지 않아 '서울특별시 중구'에 해당하는 노드를 탐색할 수 없었다. 본 연구에서는 공간 관계 'Belong_to'를 연결하여 같은 행정구역명을 가진 노드를 구분할 수 있도록 구축하였다.

본 연구에서 구축한 GeoKG는 노드의 개수 및 종류 뿐 아니라 공간 관계를 사용하여 기존의 GeoKG와 비교하였을 때 factoid question에 대응할 수 있는 범위를 확장하였음을 확인할 수 있다.

3.2.3.2 Geo-analytic question 비교

본 연구는 geo-analytic question에 대한 비교를 위해 2.2절에서 시나리오를 선정하였다. WorldKG와 본 논문에서 구축한 GeoKG를 비교하였고 [표 3-5]에 정리하였다.

[표 3-5] 시나리오별 GeoKG의 질의 수행가능여부

	본 연구에서 구축한 GeoKG	WorldKG
시나리오 1 (포인트 <-> 포인트)	O	O
시나리오 2 (포인트 <-> 라인)	X	X
시나리오 3 (포인트 <-> 폴리곤)	O	X
시나리오 4 (라인 <-> 폴리곤)	X	X
시나리오 5 (폴리곤 <-> 폴리곤, 행정구역 사이의 인접관계)	O	X
시나리오 6 (폴리곤 <-> 폴리곤, 행정구역 사이의 포함관계)	O	X
시나리오 7 (폴리곤<-> 폴리곤, 행정구역과 자연물 중첩관계)	O	X

본 논문에서 제안한 GeoKG는 오픈스트리트맵을 기반으로 구축한 WorldKG와 공공데이터를 사용하여 시나리오 1을 수행할 수 있었다. 또한, 공간연산이 필요한 관계를 정보 추출 또는 사전연산으로 GeoKG에 저장하였기에 시나리오 3, 5, 6, 7을 수행할 수 있었다. 다만 본 논문에서는 라인 데이터를 다루지 않아 시나리오 2, 4에 대한 답을 반환하지 못하였다.

WorldKG의 경우 많은 공간 객체를 다루고 있어 시나리오 1을 해결할 수 있었지만, 포인트 정보만을 적재하고 있어 나머지 시나리오에는 답하지 못하였다. WorldKG와 비교하였을 때 본 연구에서 구축한 GeoKG의 공간 연산과 관련된 관계가 약 3,600,000개로, geo-analytic question에 답할 수 있는 범위가 확장된 것을 확인할 수 있다.

[표 3-6]는 WorldKG Endpoint에 SPARQL 형식의 질의문을 기반으로 시나리오 1을 수행한 결과이다. 기존 시나리오 1의 질의는 “동대문역사문화공원역 근처 쇼핑몰은?”이었지만, 기존 WorldKG는 ‘동대문역사문화공원역’에 대한 정보를 포함하지 않아 ‘마포역’으로 교체하였다. 또한, 지하철역에 대한 정보가 있는 것이 아닌 지하철역의 출구로 되어있어 ‘마포역 1번 출구’로 바꾸어 질의하였다.

[표 3-6]의 질의문과 GraphQL을 사용한 [표 3-4 (a)]의 질의문을 비교해 보았을 때 SPARQL을 사용한 [표 3-6]의 질의문의 길이가 더욱 짧은 것을 확인할 수 있다.

[표 3-6] WorldKG의 시나리오 1 수행 결과

	질의문	수행 결과

시나리오 1
(포인트 <->
포인트)
	SELECT ?closeObject ?mall
	(bif:st_distance(?cWKT, ?fWKT, uom:metre)
	AS ?distance)
	WHERE {
	?poi wkgs:nameEn "Mapo Station gate 1".
	?poi wkgs:spatialObject [
	geo:asWKT ?cWKT
	] .
	?closeObject rdf:type wkgs:Mall.
	?closeObject rdfs:label ?mall.
	?closeObject wkgs:spatialObject ?fGeom.
	?fGeom geo:asWKT ?fWKT .
	}
	ORDER BY ASC(
	bif:st_distance(?cWKT, ?fWKT, uom:metre))
	LIMIT 10
		“EXIT”
		“롯데아울렛 서울역점”
		“회현지하상가 9번 출입구”
		“회현지하상가 12번 출입구”
		“회현지하상가 11번 출입구”
		“회현지하상가 2번 출입구”
		“회현지하상가 1번 출입구”
		“소공 지하쇼핑센터”
		“명동 지하상가”
		“1898광장”

4. 결론

구글과 같은 검색엔진과 일반적인 QA 시스템은 공간질의가 가진 특성 때문에 geo-question, 특히 geo-analytical question을 잘 수행하지 못하였다. 기존의 QA 시스템이 지리공간질의를 다룰 때 생기는 한계를 극복하고자 GeoKG를 기반으로 한 연구가 진행되었으나, 여전히 라인스트링, 폴리곤과 같은 공간 객체의 부족과 데이터의 정확성 문제로 인해 geo-analytical question을 수행하는데 문제가 있었다. 또한 GeoKG에 포함된 POI 데이터가 다양하지 않아 지리공간질의의 다른 유형인 factoid question에 대한 문제도 발생하였다. 따라서 본 연구에서는 대한민국 지역을 대상으로 기존의 QA 시스템 및 GeoKG의 한계를 극복하고자 한국 실정에 맞는 새로운 GeoKG를 구축하는 방식을 제안하였다.

기존 GeoKG 관련 선행연구는 오픈스트리트맵, Wikidata로부터 정보를 추출하거나, 다목적 지식그래프에 행정구역 폴리곤 정보를 추가하거나, 방향에 대한 관계를 사전연산하여 적재하는 방식으로 GeoKG를 구축하였다. 하지만 선행 연구의 방법론은 다음과 같은 한계점이 존재하였다. 첫째, 전처리한 데이터를 사용하지 않아 공간연산을 수행할 때 실제와 다른 답변이 도출되었다. 둘째, 실제 질의에 대한 고려 없이 '방향'이나 '포함'과 같은 제한적인 공간 관계만을 저장하였다. 이러한 한계점을 극복하고자 본 연구에서는 질의 데이터셋을 분석해 선정한 공간 관계 5가지를 정보 추출 또는 사전연산을 통해 GeoKG에 저장하였다.

본 연구는 네 단계에 걸쳐 진행되었다. 첫 번째 단계는 GeoQA 데이터셋을 분석해 질의시나리오를 선정하고, 추가할 공간관계 및 POI 종류를 선정하는 단계이다.

두 번째 단계는 기존 GeoKG가 factoid question을 수행할 때 발생한 문제점을 해결하기 위한 GeoKG를 구축하는 단계이다. 구축의 용이성을 위해 기 구축되어있는 GeoKG와 공공데이터를 융합하는 방식을 제안하였다. WorldKG는 오픈스트리트맵의 정보를 추출해 구축하여 다른 GeoKG나 공공데이터에서 취득할 수 없는 POI를 보유하는 장점이 있어

구축할 때 기본이 되는 GeoKG로 선정하였다. 최신의 행정구역 분류표를 참고한 행정구역 정보와 POI 분류표를 기준으로 선정한 대표 POI 정보를 공공데이터에서 내려받아 GeoKG에 적재하였다. 또한, 풍부한 공간연산을 지원하기 위해 쉐이프파일을 GeoJSON 파일로 변환하여 행정구역의 폴리곤 정보를 적재하였다. 이때, 데이터베이스에 상관없이 멀티폴리곤 연산을 수행할 수 있도록 서로 다른 폴리곤으로 분리하여 적재하였다.

세 번째 단계는 공간 관계를 적재하는 단계이다. 기존에 geo-analytic 유형으로 분류되던 질의를 factoid question과 같은 방식으로 답을 반환하기 위해 정보추출 또는 사전연산을 수행해 취득한 공간관계를 GeoKG에 저장하였다. 저장 공간과 질의속도를 고려하였을 때 모든 공간관계를 연산하고 저장하기에 한계가 있으므로 GeoQA 데이터셋을 분석해 'Near', 'Border', 'Belong_to', 'In', 'Cross' 총 5개의 관계를 선정하였다. 사전연산이 불필요한 'Belong_to'는 행정구역 분류표의 정보를 기반으로 관계를 연결하였고, 'Near'와 'In'은 Neo4j의 spatial 플러그인을 사용하였으며, 폴리곤 간의 연산인 'Border'와 'Cross'는 연산 오류를 방지하기 위해 ArcGIS의 'Repair geometry' 기능을 사용하여 전처리 과정을 거친 후 연산을 수행하였다. 이후 연산된 정보를 GeoKG에 적재하였다.

마지막 단계는 속성그래프 데이터베이스에 구애받지 않고 사용할 수 있는 GraphQL을 활용해 데이터베이스에 쿼리하고, 시나리오별로 GeoKG의 비교를 수행하였다.

본 연구의 기여점은 다음과 같다. 행정구역 및 POI 데이터를 추가하여 factoid question에 답할 수 있는 범위를 확장하였다. 또한, geo-analytic question으로 분류되는 질의를 GeoKG에 사전 연산하여 저장된 공간 관계를 통해 factoid question과 같이 응답할 수 있는 방식을 제안하였다. 기존의 GeoKG나 GeoSPARQL을 사용했을 때 답하지 못하는 질의에 답변하여 GeoKG가 다룰 수 있는 질의의 범위를 넓혔다는 의의가 있다. 기존에 사전연산을 통해 공간관계를 GeoKG에 저장한 연구

가 수행되었지만, 실제 질의에서 요구하는 공간 연산의 종류에 대한 고려 없이 공간 관계를 구축하였다는 한계가 존재한다. 본 연구에서는 geo-analytic question에 답하기 위한 공간 관계를 구축할 때 GeoQuestions201과 MS Marco에 등장하는 공간 관계를 분석하고, 높은 빈도로 등장하는 공간관계를 구축하여 실제 질의를 반영하였다.

본 연구는 다음과 같이 활용 가능할 것으로 기대된다. 첫째, 저장된 공간 관계를 기반으로 '마포구와 종로구 사이의 행정구는?'과 같은 모호하고 복잡한 geo-question에 답하는 것으로 확장할 수 있을 것이다. 둘째, 여러 분석과정이 필요한 geo-question에 답하는 파이프라인에 활용될 수 있을 것이다. 예를 들어, 나라 간의 거리를 묻는 질의에서 단순히 임의의 점 사이의 거리를 반환하는 것이 아니라 인접 관계까지 고려하여 인접한다면 '0' 또는 '인접'을 반환하는 결과를 도출할 수 있다. 마지막으로, 행정구역과 POI의 정보는 시간이 지남에 따라 바뀌기에 정보에 시간 속성을 포함한다면 시계열적인 분석이 가능할 것이다.

본 연구에서 구축한 GeoKG는 많은 질의 시나리오를 해결하며 기존의 GeoKG와 비교하였을 때 좋은 성능을 보였지만, 저장한 공간 관계 이외의 복잡하고 다중 공간연산이 필요한 geo-analytic question에 대해서는 답을 반환할 수 없다는 한계점을 가지고 있어 복잡한 공간 연산에 대한 연구가 필요하다. 또한, 데이터의 저장 공간과 질의 성능을 고려하여 어느 정도 수준까지 관계를 사전연산해 저장할지에 대한 논의가 필요하다. 마지막으로, 행정구역의 분류와 그 범위가 시간에 따라 변하고 식당과 같은 변화가 빠른 POI 정보에 대한 정확한 답을 반환하려면 지속적인 데이터 갱신에 대한 논의도 필요하다.

향후 연구는 질의 데이터셋을 분석하여 더욱 다양한 종류의 공간 관계를 구축하고, 세세한 POI의 데이터를 구축하는 것이다. 본 연구에서는 컴퓨팅 자원의 한계로 일부의 POI와 한정된 공간 관계를 구축하였지만, 많은 데이터를 포함한다면 더 넓은 범위의 질의를 다룰 수 있을 것이다. 또한, 'Near'의 범위를 여러 공간 분석을 기반으로 POI에 따라 합리적인 기준을 마련한다면 사용자의 요구에 부응하는 GeoQA 시스템을 구축할

수 있을 것이다.

참 고 문 헌

오성호(2006), 인프라 21 세미나, 우리나라 POI 구축현황 및 향후 추진방향, 국토연구원, pp.152-157.

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, Berlin, Heidelberg, pp. 722-735.

Auer, S., Lehmann, J., & Hellmann, S. (2009, October). Linkedgeodata: Adding a spatial dimension to the web of data. In International Semantic Web Conference. Springer, Berlin, Heidelberg, pp. 731-746.

Besta, M., Peter, E., Gerstenberger, R., Fischer, M., Podstawski, M., Barthels, C., ... & Hoefler, T. (2019). Demystifying graph databases: Analysis and taxonomy of data organization, system designs, and graph queries. arXiv preprint arXiv:1910.09017.

Brigham, C., Gilbert, S., & Xu, Q. (2011). Open geospatial data: An assessment of global boundary datasets. World Bank Institute.

Dsouza, A., Tempelmeier, N., Yu, R., Gottschalk, S., & Demidova, E. (2021, October). Worldkg: A world-scale geographic knowledge graph. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 4475-4484.

Franciscus, N., Ren, X., & Stantic, B. (2017, April). Answering temporal analytic queries over big data based on precomputing architecture. In Asian Conference on Intelligent Information and Database Systems, Springer, Cham, pp. 281-290.

Groff, E. (2011). Exploring 'near': Characterizing the spatial extent of drinking place influence on crime. Australian & New Zealand Journal of

Criminology, 44(2), pp.156-179.

Gupta, P., & Gupta, V. (2012). A survey of text question answering techniques. International Journal of Computer Applications, 53(4).

Hamzei, E., Li, H., Vasardani, M., Baldwin, T., Winter, S., & Tomko, M. (2019, June). Place questions and human-generated answers: A data analysis approach. In International Conference on Geographic Information Science. Springer, Cham, pp. 3-19

Hamzei, E., Tomko, M., & Winter, S. (2022, April). Translating Place-Related Questions to GeoSPARQL Queries. In Proceedings of the ACM Web Conference 2022, pp. 902-911.

Hartig, O., & Pérez, J. (2018, April). Semantics and complexity of GraphQL. In Proceedings of the 2018 World Wide Web Conference, pp. 1155-1164.

Jiang, B., Tan, L., Ren, Y., & Li, F. (2019). Intelligent interaction with virtual geographical environments based on geographic knowledge graph. ISPRS International Journal of Geo-Information, 8(10), p.428.

Karalis, N., Mandilaras, G., & Koubarakis, M. (2019, October). Extending the YAGO2 knowledge graph with precise geospatial knowledge. In International Semantic Web Conference (pp. 181-197). Springer, Cham.

Lan, Y., He, G., Jiang, J., Jiang, J., Zhao, W. X., & Wen, J. R. (2021). A survey on complex knowledge base question answering: Methods, challenges and solutions. arXiv preprint arXiv:2105.11644.

Li, H., Hamzei, E., Majic, I., Hua, H., Renz, J., Tomko, M., ... & Baldwin, T. (2021). Neural factoid geospatial question answering. Journal of Spatial Information Science, (23), pp.65-90.

Lu, R., Cai, Z., & Zhao, S. (2019, July). A Survey of Knowledge Reasoning based on KG. In IOP Conference Series: Materials Science and Engineering, Vol. 569, No. 5, p. 052058.

Mai, G., Janowicz, K., Zhu, R., Cai, L., & Lao, N. (2021). Geographic question answering: challenges, uniqueness, classification, and future directions. AGILE: GIScience Series, 2, pp.1-21.

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., & Deng, L. (2016, January). MS MARCO: A human generated machine reading comprehension dataset. In CoCo@ NIPs.

Ordanance Survey, 2021, Point of Interest Classification Scheme, Research report, Ordanance Survey, United Kingdom, pp. 1-16.

Park, S., Kwon, S., Kim, B., Han, S., Shim, H., & Lee, G. G. (2015, June). Question answering system using multiple information source and open type answer merge. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pp. 111-115.

Punjani, D., Singh, K., Both, A., Koubarakis, M., Angelidis, I., Bereta, K., ... & Stamoulis, G. (2018, November). Template-based question answering over linked geospatial data. In Proceedings of the 12th Workshop on Geographic Information Retrieval, pp. 1-10.

Reddy, G. S., Srinivasu, R., Rao, M. P. C., & Rikkula, S. R. (2010). Data Warehousing, Data Mining, OLAP and OLTP Technologies are essential elements to support decision-making process in industries. International Journal on Computer Science and Engineering, 2(9), pp. 2865-2873.

Regalia, B., Janowicz, K., & McKenzie, G. (2019). Computing and querying strict, approximate, and metrically refined topological relations in linked geographic data. Transactions in GIS, 23(3), pp.601-619.

Scheider, S., Nyamsuren, E., Kruiger, H., & Xu, H. (2021). Geo-analytical question-answering with GIS. International Journal of Digital Earth, 14(1), pp.1-14.

Suchanek, F. M., Kasneci, G., & Weikum, G. (2007, May). Yago: a core of semantic knowledge. In Proceedings of the 16th international conference on World Wide Web, pp. 697-706.

Taelman, R., Vander Sande, M., & Verborgh, R. (2018). GraphQL-LD: linked data querying with GraphQL. In ISWC2018, the 17th International Semantic Web Conference, pp. 1-4.

Tomaszuk, D. (2016, November). RDF data in property graph model. In Research Conference on Metadata and Semantics Research. Springer, Cham, pp. 104-115.

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

Wang, S., Zhang, X., Ye, P., Du, M., Lu, Y., & Xue, H. (2019). Geographic knowledge graph (GeoKG): a formalized geographic knowledge representation. ISPRS International Journal of Geo-Information, 8(4), p.184.

Wylot, M., Hauswirth, M., Cudré-Mauroux, P., & Sakr, S. (2018). RDF data storage and query processing schemes: A survey. ACM Computing Surveys (CSUR), 51(4), pp.1-36.

Xu, H., Hamzei, E., Nyamsuren, E., Kruiger, H., Winter, S., Tomko, M., & Scheider, S. (2020). Extracting interrogative intents and concepts from geo-analytic questions. AGILE: GIScience Series, 1, pp.1-21.

Zheng, K., Xie, M. H., Zhang, J. B., Xie, J., & Xia, S. H. (2022). A knowledge representation model based on the geographic spatiotemporal

process. International Journal of Geographical Information Science, 36(4), pp.674-691.

Zou, X. (2020, March). A survey on application of knowledge graph. In Journal of Physics: Conference Series, Vol. 1487, No. 1, p. 012016.

Abstract

Constructing the Geographic Knowledge Graph for a Comprehensive Geographic Question Answering: In Korea Region

Kim, Donghyun

Department of civil and Enviromental Engineering

The Graduate School

Seoul National University

Question Answering (QA) system is an information search technology that finds answers to questions that come in the form of natural language, and is actively researched and performing well in natural language processing. However, search engines, including Google, and existing Q&A systems, have difficulty returning appropriate answers to questions related to geospatial space, which account for a large portion of the total query. In order to overcome the limitations of existing question and answer systems when answering geospatial questions, a geospatial knowledge graph (GeoKG) has been studied, but it still does not properly answer

factoid questions and geo-analytic questions. Due to the lack of point of interest (POI) and types of spatial objects held by the knowledge base, it is difficult to answer the factoid question, and it is difficult to answer the geo-analytic question due to the poor accuracy of spatial objects held and difficulty in performing spatial operations.

This study proposed a GeoKG construction plan to solve the problem of factoid and geo-analytic types in which existing GeoKGs suffer. In order to respond to more factoid questions, we fused public data with existing GeoKGs, and geo-analytic questions were designed by adding pre-computed spatial relations to GeoKG. After that, experiments were conducted across Korea region.

After converting WorldKG, which has a large number of geographical objects among existing GeoKGs, into an property graph form, major POI information and information extracted from administrative districts and polygons were added based on public data. In addition, GeoQuestions201 and MS Marco datasets were analyzed to load spatial relations that appear at high frequency as relationships by performing information extraction and spatial computation. The query of GeoQuestions201 was analyzed to create a query scenario for performance evaluation of geo-analytic question. As a result of comparing the GeoKG constructed in this study with WorldKG, it was confirmed that the GeoKG constructed in this study could answer more factoid and geo-analytic questions than WorldKG. In addition, to compensate for the shortcomings of property graphs without standard query language, queries through GraphQL with versatility were also performed in several property graph databases.

This study is meaningful in that it proposed a construction plan that expanded the scope of factoid and geo-analytic questions that can be responded to GeoKG, which was constructed by loading public

data and pre-computed spatial relationships in the existing GeoKG.

keywords : GeoQA, GeoKG, Factoid question, Geo-analytic question, Pre-computation, Public data

Student Number : 2021-29171