Journal of the Korea Industrial Information Systems Research Vol.27 No.3, Jun. 2022 :1-10
http://dx.doi.org/10.9723/jksiis.2022.27.3.001 ISSN:1229-3741

웹 문서상의 공간 텍스트 위치 맵핑과 질의 기법+

(Techniques for Location Mapping and Querying of Geo-Texts in Web Documents)

하 태 석[1], 남 광 우[2]*
(Tae Seok Ha and Kwang Woo Nam)

요 약 웹 기술의 발전과 함께 대량의 웹 문서들이 생산되고 있다. 이 웹 문서에는 다양한 공간적 텍스트들을 포함하고 있으며, 이 텍스트들을 공간정보로 변환함으로서 공간질의로 텍스트 문서를 검색할 수 있는 기반이 된다. 이러한 공간 텍스트들에는 행정지명이나 관심 지역(POI)이름 뿐만이 아니라 우편번호나 지역 전화번호 등까지 폭넓은 영역으로 구성되어 있다. 이 논문은 웹 문서내 내에 존재하는 공간 텍스트 정보를 기반으로 위치를 맵핑 할 수 있는 알고리즘들을 제시하고 있다. 이 알고리즘들을 통해 웹 문서들을 일반 웹 단어 기반 문서 검색 뿐만 아니라, 지도상에서 공간 영역과 텍스트의 복합형태로 해당 지역을 설명하는 문서들을 검색할 수 있게 된다. 마지막으로 이 논문에서는 제안된 알고리즘들을 이용하여 웹 공간 텍스트 질의 시스템을 구현함으로써 유용함을 보였다.

핵심주제어 : 지오웹, 공간 텍스트, 위치 맵핑, 웹 문서 공간 질의

Abstract With the development of web technology, large amounts of web documents are being produced. This web document contains various spatial texts, and by converting these texts into spatial information, it is the basis for searching for text documents with spatial query. These spatial texts consist of a wide range of areas, including postal codes and local phone numbers, as well as administrative place names and POI names. This paper presents algorithms that can map locations based on spatial text information existing within web documents. Through these algorithms, web documents can be searched for documents describing the region on a map rather than a general web search. In this paper, we demonstrated the presented algorithms are useful by implementing a web geo-text query system.

Keywords : geoweb, geo-text, location mapping, web document spatial queries

* Corresponding Author : kwnam@kunsan.ac.kr
+ 이 연구는 2020년도 정부(교육부)의 재원으로 한국연구재단의 기초연구사업(No.2020R1F1A1048432)과 2021년 한국국토정보공사 공간정보연구원의 산학협력 R&D 지원사업 자유과제 지원에 의하여 수행된 연구임

Manuscript received January 25, 2022 / revised April 08, 2022 / accepted April 20, 2022
1) (주)쿠첸, 제1저자
2) 군산대학교 컴퓨터정보통신공학부, 교신저자

1. 서 론

인터넷상의 웹 문서들은 함축적이거나 명시적으로 다양한 공간정보를 포함하고 있다(Lee, 2018). 예를 들면 뉴스나 블로그의 내용 중에 '군산시'나 '다보탑' 등의 단어가 존재한다고 가정하자. 이 단어들은 우리나라 사람들 대부분이 아는 명사들에 속하며 지도상의 특정 영역이나 특정 지점을 명확하게 특정할 수 있다. 즉, '군산시'나 '다보탑'이라는 텍스트를 이용하여 공간정보로 변환할 수 있다는 것을 의미한다. 이러한 공간정보로 유추할 수 있는 텍스트 정보를 공간 텍스트(geo-text)라고 하며, 공간 텍스트와 공간정보의 변환성을 이용하여 지도상에서 그 지역에 대한 문서를 검색하고자 할 때 공간 지역질의 결과로 검색되도록 할 수 있다(Borges, 2006, Ha, 2010, Borges et al., 2007, Cui et al., 2019). 앞의 예와 같이 공간 텍스트들을 포함하고 있어 공간정보로 변환이 가능한 웹 문서들을 지오웹(geoweb) 문서라고 한다.

지오웹 문서들은 텍스트 공간정보를 실제 공간정보로 변환할 수 있도록 함으로서 공간정보시스템을 웹의 영역까지 확장할 수 있게 한다(Laender et al. 2002, Lee, 2020, McCurley, 2001). 즉, 기존의 GIS와 결합하여 자신과 가까운 문서의 정보를 검색 또는 관심주제의 문서 내 위치 등을 확인하는데 사용할 수 있다(Cong et al., 2009, Ma et al., 2020). 텍스트와 지리적 의미는 웹 페이지에 내에서 어디든지 발생할 수 있기 때문에 공간정보의 인식은 복잡한 문제를 가지고 있다(Moon et. al, 2019, Rahimi, 2015, Yang et al., 2009).

간단하면서도 공간정보를 직접적으로 유추할 수 있는 공간 텍스트 정보들은 주소, 전화번호, 우편번호 등이 있다. 또한, 유명한 산의 이름이나 주요 명소와 같은 관심지역(POI: point-of-interests)들은 그 자체만으로도 공간좌표로 변환이 가능하다. 예를 들면 지오웹 문서내에 '설악산'이나 '경복궁'등의 단어가 존재한다면 그 단어들은 용이하게 공간좌표로 변환될 수 있다. 그러나 공간정보 텍스트가 지명이라면, 한 지명이 여러 지역에서 동시에 사용될 수 있기 때문에 더 복잡해진다.

이 논문에서는 공간 텍스트 온톨로지 모델을 제안하여 이를 기반으로 공간정보로 맵핑할 수 있는 프레임워크와 세부적인 알고리즘들을 제시하고 있다. 또한, 이러한 웹 지역정보에 포함된 공간텍스트들 활용하여 공간 웹 온톨로지 기반의 지오웹 문서들이 데이터베이스에 저장되어 있다고 할 때 이 문서들에 대한 웹 기반의 공간 텍스트 질의를 할 수 있는 시스템 제안한다.

2. 공간 텍스트 온톨로지 모델

웹 페이지에는 텍스트로 표현된 다양한 공간정보를 함께 포함하고 있으며, 이를 이용하여 웹 페이지가 어느 공간적 위치를 대상으로 작성된 것인지 유추할 수 있다. 웹 텍스트에 표현된 공간정보는 크게 두 가지로 구분될 수 있다(Martinez-Rodirguez et al., 2020). 첫 번째는 고유명사 형태의 텍스트로 도시나 지역명, 주소, 또는 특정 상점의 이름과 같은 관심 지점(POI)들이 해당되며, 두 번째는 전화번호와 같은 숫자형태의 공간정보의 텍스트들이다. POI 이름과 주소들은 지금까지 다양한 문헌들에서 다루어졌으나 숫자 형태의 공간정보 표현은 거의 언급되지 않았다. 예를들면, 미국의 경우 도시 이름 대신 우편주소를 함축적인 공간적인 표현으로 빈번하게 사용한다. 또 다른 예는 전화번호에 있는 지역번호 등도 그러한 예라고 할 수 있다. 우편 번호나 지역번호는 당연히 강력한 지역의 징표이다. 우편번호 형태의 텍스트 공간정보는 국가마다 다른 표현 방법을 갖지만 웹 페이지에 직접 적시되어 있으며 계층적 의미까지 포함하고 있으므로 매우 세부적인 위치를 대표한다고 볼 수 있다. 02, 031과 같은 일반 유선전화 번호는 함축적인 지역 정보를 가지고 있다. 유선 지역번호는 우편번호 보다는 넓지만 광역 지역에 따라 별도로 관리하기 때문에 고정된 코드 식별자로써 다른 텍스트 공간정보와 결합하여 지역적 범위를 한정할 수 있는 강력한 수단이 될 수 있다. 이 번호는 계좌번호, 도로번호 숫자들과 혼돈하지 않도록 유선 전화번호의 패턴을 인식하고 처리를 해야 한다.

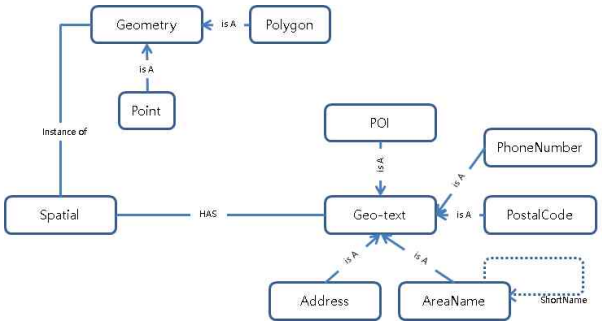

Fig. 1 Ontology model for geo-texts

웹 페이지에서 데이터를 추출하기 위한 여러 가지 방법들이 있다(Embley et al., 1999, Laender et al. 2002). 이 논문에서는 온톨로지를 기반으로 웹 공간 텍스트 인식을 위한 모델을 제안하고, 이를 기반으로 웹 텍스트로부터 공간정보를 추출한다.

Fig. 1은 이 논문에서 제안하고 있는 웹 텍스트의 공간정보 온톨로지 모델을 보이고 있다. Fig 1에서 공간 클래스는 지리 좌표 공간상에서 공간 영역을 표현하기 위한 방법으로 Geometry 클래스로 표현되며, 이 모델에서는 POI 표현을 중심으로 표현하기 위해 Point와 Polygon 클래스 형태를 지원하고 있다. 공간 텍스트 클래스는 POI 클래스, 주소 클래스, 지명 클래스, 전화번호 클래스, 우편번호 클래스 등으로 구성된다.

공간 텍스트 클래스에서 웹 텍스트상에 나타는 기본 주소 클래스는 고전적인 지번 주소들과 도로명 주소 형태들로 구성될 수 있다. 지명 클래스는 고전적인 시/군/구와 읍/면/동 체계 등에서 존재하는 지명들을 키워드 형태로 지원하는 클래스이다. '군산시', '군산', '수송동' 과 같은 텍스트가 웹 페이지내에 포함되었다고 할 때, 이 텍스트로부터 대략적인 웹 페이지 내용이 지칭하는 공간정보를 파악할 수 있다. POI 클래스는 식당이나 주요 기관의 명칭을 위한 클래스로 큰 지역적 구분을 표현하는 지명 클래스와는 구분되는 특성이 있다. 우편번호 클래스와 전화번호 클래스도 공간정보를 유추하기 위해 사용될 수 있다. 우편번호 클래스는 앞 세자리를 이용하여 시/군/구를 구분할 수 있으며, 전화번호의 지역번호는 광역시/도단위에서 공간정보를 구분할 수 있는 표식이 될 수 있다. 다음 장에서 제안된 온톨로지 모델을 지원하는 위치 맵핑 및 질의 시스템에 대하여 보다 구체적으로 설명한다.

3. 위치 맵핑 및 질의 시스템과 알고리즘

3.1 시스템 구조

웹 문서의 위치 맵핑 및 질의 시스템은 웹 문서의 텍스트들을 분석하여 위치를 맵핑하여 텍스트 정보와 함께 데이터베이스에 저장하는 부분과 사용자들이 웹을 이용하여 관심있는 위치에 대한 웹 문서들을 질의하면 데이터베이스에서 해당 웹 문서를 검색해서 반환하는 부분으로 구성된다. Fig. 2는 이 논문이 제안하는 위치 맵핑 및 질의 시스템의 구조를 보이고 있다.

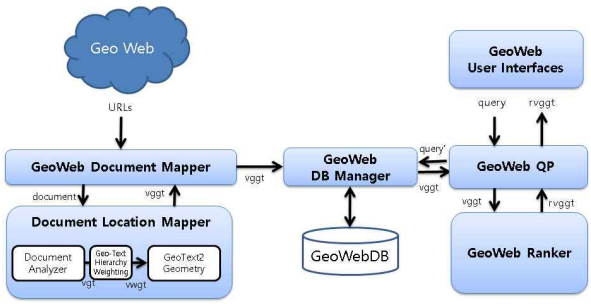

Fig. 2 System architecture

이 논문의 위치 맵핑 및 질의 시스템은 크게 지오웹 문서 맵퍼(GeoWeb Document Mapper)와 지오웹 DB 매니저(GeoWeb DB Manager), 그리고 지오웹 질의 처리기(GeoWeb QP)로 구성된다. 각 구성요소들은 다음과 같은 역할을 수행한다.

· 지오웹 문서 맵퍼(GeoWeb Document Mapper) : 웹 문서를 웹으로부터 다운로드하여, 해당 문서의 텍스트를 분석하여 위치를 맵핑하고 태깅한 뒤 GeoWeb DB Manager를 이용하여 DB에 저장한다.
· 지오웹 DB 매니저(GeoWeb DB Manager) :

공간 좌표화된 공간텍스트를 포함하는 문서들의 집합을 관리 저장하고 Geo Web QP으로부터 입력되는 검색 질의를 수행한다.

· 지오웹 질의 처리기(GeoWeb QP) : Geo Web User Interfaces부터 공간 영역 질의를 받아 GeoWeb Ranker를 통해 해당 공간영역에 근접한 순으로 랭킹화된 문서들을 반환한다.

지오웹 문서 맵퍼는 이 논문에서 제안하고 있는 웹 문서 위치 맵핑을 위한 핵심적인 구성요소이다. Fig. 3은 지오웹 문서 맵퍼의 알고리즘을 보이고 있다. Table 1은 이해를 돕기위해 Fig. 3과 이후의 알고리즘에서 사용되는 약자들에 대해 먼저 설명하고 있다.

Table 1 Notations

기호	설 명
vt	A vector of texts in a parsed document
vgt	A vector of geo-texts in a document
vwgt	A vector of weighted geo-texts by geo-hierachy weighting
vggt	A vector of geo-coordinated geo-texts
vvggt	A vector of documents which were transformed into *vggt*
rvvggt	A vector of ranked vggt by given query area and distances from stored documents
q	A geo-text query with a minimum bounding rectangle area and a keyword

GeoWebDocumentMapper는 Fig. 3의 알고리즘에서 보이는 것과 같이 지오웹 문서들의 URL들을 입력으로 받아 위치 맵핑을 수행한 후 GeoWebDBManager를 이용하여 DB에 저장한다.

Algorithm : GeoWebDocumentMapper

Input : Vector of URLs *urls*
Output : Vector of OID of Geo Text in DB *VOID*
foreach *url* ∈ *urls* **do**
  doc ← GetDocument(*url*);
  vggt ← DocumentLocationMapper(*doc*);
  oid ← GeoWebDBManager.insert(*vggt*);
  VOID ← *VOID* ∪ *oid*
return *VOID*

Fig. 3 GeoWebDocumentMapper algorithm

위의 알고리즘에서 GetDocument는 지오웹 문서의 URL들을 받아 HTML 태그와 필요 없는 구획 문자들을 제거하고 위치 맵핑에 필수적인 요소들을 파싱하여 변수 *doc*에 반환한다. 그 다음 DocumentLocationMapper 함수를 통해 지오웹 문서 *doc*의 공간 텍스트를 기반으로 위치맵핑을 수행하고 공간좌표 형태로 변환된 *vggt* (vector of geometrized geo-text)를 반환받는다. 반환된 *vggt*는 GeoWebDBManager의 insert 함수를 이용하여 웹 질의를 위해 사용될 수 있게하기 위해 DB에 저장된다. insert 함수는 저장을 요청한 함수에서 *vggt*가 잘 저장되었는지 확인하거나 나중에 직접 문서를 요청할 수 있게 하기 위해 해당 *vggt*의 데이터베이스내 객체 식별자들(*oid* : object identifier)를 반환한다. 마지막으로 GeoWebDocumentMapper는 다수의 url들을 입력으로 받아 위치맵핑과 데이터베이스 저장을 하므로 해당 결과를 한번에 반환하기 위하여 반환된 oid를 리스트 자료구조인 *VOID*(vector of oids)에 저장한다. 다음 절에서는 위치 맵핑을 위한 DocumentLocationMapper 알고리즘에 대해서 다음 절에서 좀 더 자세하게 설명한다.

3.2 문서분석기 알고리즘

지오웹 문서의 텍스트들을 이용하여 문서의 공간적 위치를 추정하는 DocumentLocationMapper 알고리즘은 아래의 Fig. 4에서 보이는 것과 같이 문서분석기(DocumentAnalyzer), 공간계층가중치 분석기(GeoTextHierarchyWeighting), 공간텍스트좌표변환기(GeoText2Geometry)로 구성되어 있다.

Algorithm : DocumentLocationMapper

Input : A document *doc*, A query *q*
Output : Vector of Geo Texts *VOID*
vgt ← ∅
vgt ← DocumentAnalyzer(*doc*)
vwgt ← GeoTextHierarchyWeighting(vgt);
vggt ← GeoText2Geometry(*vwgt*);
return *vggt*

Fig. 4 DocumentLocationMapper algorithm

DocumentLocationMapper 알고리즘에서 가장 먼저 수행하는 것은 문서분석기이다. 문서 분석기는 지오웹 문서를 입력으로 받아 파싱한 후 전화번호나 우편번호 등의 숫자 텍스트들과 지명 등을 추출하여 *vgt*로 반환한다. 두 번째 단계는 공간계층가중치분석기로로서 문서 분석기의 공간 텍스트 힌트들을 이용해 공간 텍스트들의 가중치 분석을 수행하여 *vwgt*를 생성한다. 마지막으로 공간텍스트좌표변환기를 통해 공간좌표로 변환하여 이 결과를 *vggt*로 반환하게 된다.

Algorithm : DocumentAnalyzer
Input : A document *doc*
Output : Vector of Geo Texts *VOID*
vgt ← ⦰
vt ← kkmParser(*doc*);
vgt.add(PhoneNumberExtractor(*vt*));
vgt.add(PostalCodeExtractor(*vt*));
vgt.add(LocalNameCheck(*vt*));
return *vgt*

Fig. 5 DocumentAnalyzer Algorithm

문서분석기 알고리즘은 Fig. 5에서 보이는 것과 같이 문서 *doc*을 입력으로 kkmParser를 이용하여 구문 파싱을 수행한다. kkmParser는 서울대학교에서 개발하여 2010년에 공개한 관계형 데이터베이스를 활용한 세종 말뭉치 활용도구로서 공개 소프트웨어로 배포되고 있는 소프트웨어이다(Lee et al. 2010). 이 논문의 시스템은 문서를 파싱하는 기본 도구로 꼬꼬마 분석기를 사용하고 있다. 이렇게 파싱된 문서는 텍스트들의 모음으로서 *vt*로 반환된다.

문서분석기는 파싱된 결과인 vt들 내에서 전화번호와 우편번호 패턴을 갖는 텍스트를 검출하며, 이 중에서 각 지역 번호나 우편번호를 분리하여 횟수 정보와 함께 저장하여 공간적 정보를 추정할 수 있는 힌트로 사용한다. 마지막으로 지명 검증기(LocalNameCheck)를 통해 텍스트내에 존재하는 명사들 중에서 지명을 추출하고 횟수를 계산한다. 문서 내에 존재하는 전화번호, 우편편호들의 지명과 명사의 위치와 횟수는 가장 관련 깊은 지역을 선택하는데 중요한 요소 중에 하나이다.

Algorithm : PostalCodeExtractor
Input : A vector of texts *vt*
Output : Vector of Geo Texts *VOID*
vgt ← ⦰
zippattern ← Pattern('s/\ +([0-9]{5})$/\1\n/p');
M = zippattern.matcher(*vt*);
foreach *m* ∈ *M* **do**
  if zipDB.find(*m*) then
    vgt.add(GeoText(*m*));
return *vgt*

Fig. 6 PostalCodeExtractor Algorithm

우편번호와 지역번호를 포함한 전화번호는 일반 명사 형태의 지명보다 공간정보를 보다 분명히 구분할 수 있는 표식으로 사용될 수 있다. 예를 들면 지오웹 문서내에 '광주시'라는 지명이 존재한다고 했을 때 경기도 광주시와 전남 광주시 중에서 어떠한 지명을 의미하는지 분명히 하기 매우 어렵다. 이 때 동일 문서에서 '062'나 '031'과 같은 지역번호로 시작하는 전화번호가 함께 발견된다면 '광주시'의 지역적 위치를 광주광역시(062)이나 경기도(031) 중 하나의 가중치를 명확하게 높일 수 있는 근거가 된다.

우편번호와 전화번호 추출 알고리즘은 유사하다. Fig. 6에서는 대표적으로 우편번호 검출기 알고리즘으로 보이고 있다. PostalCodeExtractor 알고리즘에서 *zippattern*은 현재 사용되고 있는 다섯 자리의 숫자로 구성된 우편번호 패턴을 추출하는 것을 지원하도록 설정된다. 이 *zippattern*을 이용하여 문서로 부터 우편번호가 될 수 있는 후보들을 모두 추출하여 *M*에 반환한다. *M*은 다섯자리 숫자로 된 후보이며, 우편번호로 확정하기 위해서는 zipDB에 실제 존재하는지 확인하여야 한다. 이 확인 과정을 거쳐 우편번호 데이터베이스에 존재한다면 우편번호로 가정하고 공간 텍스트 태깅을 하여 *vgt*에 추가한다. PostalCodeExtractor 알고리즘은 최종적으로 공간 텍스트들의 벡터인 *vgt*를 반환하게 되며, 전화번호 추출 알고리즘도 유사하게 구현될 수 있다.

LocalNameCheck 함수는 문서 분석기의 꼬꼬마 형태소 분석기를 통해 출력된 명사들 중에 지

명을 검출하고 지명의 level를 처리하는 구성 요소이다. 지명 데이터 구축을 위해 우편번호 데이터베이스와 관심 지역 (POI; Point of Interest)데이터베이스가 사용되었다. 제안된 시스템은 일반적으로 많이 사용되는 POI 데이터베이스외에 우편번호 데이터베이스가 함께 사용되었다. 우편번호 데이터베이스는 주소를 구성하는 지명들이 행정단위의 구성체계인 시도, 구군, 읍면동으로 구분되어 구축되어 있으므로 단순한 지명용 명사뿐만 아니라, 지명들간의 계층적 공간 지명 체계에 대한 정보까지 함께 검증할 수 있는 장점이 있다.

지명검증기의 핵심적인 기능은 입력받은 명사들을 우편번호 데이터베이스와 POI 데이터베이스에 있는 명사들 중에서 존재하는 지 검증하는 것이다. 즉 입력된 문자열과 대량의 명사 문자열들의 집합들을 비교하는데 적지않은 시간 비용을 필요로 하므로 이에 대한 최적화된 알고리즘이 필요하다.

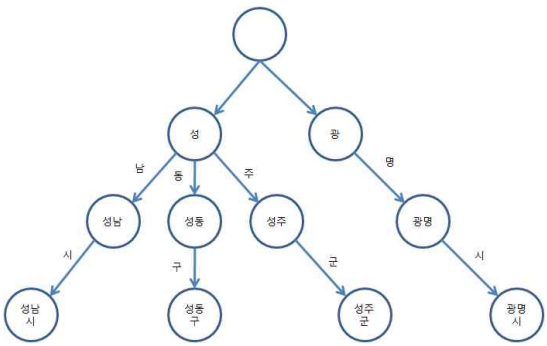

Fig. 7 Trie structure for local name check

제안된 시스템에서는 빠른 검색을 위해 Trie 메모리 구조를 사용한다. Trie 메모리 구조는 보통 문자열을 처리하기 위해 사용되며, 속도가 빠르고 공간을 적게 차지하기 때문에 검색엔진에서는 사전과 역화일의 인덱스 정보를 저장하는데 주로 사용한다. 제안 시스템에서 사용하고 있는 Trie의 구조의 예는 Fig. 7과 같다. 예를 들면, “성남시, 성동구, 성주군, 광명시”를 저장하는 트라이 문자열에서 각 음절을 하나의 depth(자식노드)로 볼 수 있으며 같은 depth의 음절은 바로 앞의 음절이 같을 경우 형제노드로 볼 수 있다. 제시된 명사들은 최상위 루트 노드부터 한글자씩 비교해서 하단으로 진행하며 문자열을 비교하게 된다. Trie 메모리 구조를 구성하기위한 조건으로 도시 이하의 행정단위 명칭이 웹 텍스트 내에서 정확하게 표기 한다는 것을 가정하고 있다. 즉 도시의 정식 명칭과 약어 명칭 모두를 가지는 구조를 가진다. 예로 전라북도와 전북, 전주시 완산구를 전주시, 전주, 완산구와 같이 정식 명칭과 약어 명칭 모두를 Trie 메모리 구조로 유지한다. 가장 마지막 노드에는 해당 지명의 행정체계상 레벨 정보를 함께 갖고 있으며, 발견된 명사에 해당 공간 및 레벨정보를 태깅하여 반환한다.

3.3 계층형 공간 텍스트 변환 알고리즘

지오웹 문서에서 다수의 지명들로 구성된 공간 텍스트들이 발견되었다고 할 때, 이 문서가 어느 공간적 위치에 대한 문서인지를 추정하는 가장 단순한 방법은 문서에 출현된 지명과 지명의 횟수 그리고 문서의 길이를 사용하는 방법이다. 즉, 기존에 사용되는 가장 단순한 방법은 중요한 지명일 경우 문서 내에서 여러 번 출현 한다고 가정하는 것이다. 이러한 방법들은 서로 다른 지역에 존재하는 동일한 지명을 구분하거나, 두 지역에 대해서 함께 설명하고 있는 경우와 같은 복합적인 경우에 적용하기 매우 어려운 단점이 있다.

이 논문에서는 한 문서 내에 다수의 지명이 발견되었다고 할 때 공간적 정확도를 높이기 위해 지명들간의 상위, 하위 행정 단위 지명의 계층형 관계를 파악하는 알고리즘을 제안한다. 다음의 Fig. 8은 공간 텍스트의 계층적 구조의 예를 보이고 있다.

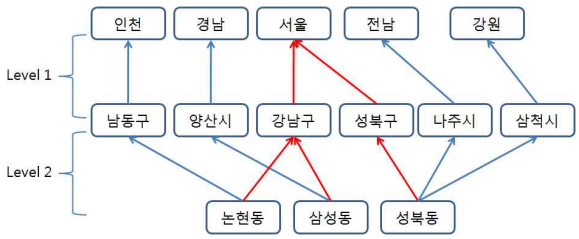

Fig. 8 Geo-text hierarchy structure

이 논문의 공간 텍스트의 계층적 구조는 우편번호 데이터베이스를 기본으로 사용하고 있다. 이 논문에서는 우편번호 지명을 3단계 계층으로 분류하여 구축되었다. 특별시, 광역시, 도를 level 0으로 구분하고 시, 군, 구을 level 1로 구분하고 읍, 면, 동, 리을 level 2로 구분한다. 이렇게 계층적 지명 체계가 구축되어 있을 때 이 지명 계층 정보와 문서내에 포함된 공간 텍스트들을 비교함으로서 공간적 포함관계와 인접 관계를 이용하여 보다 정확한 공간적 정보를 추정할 수 있다.

먼저 지오웹 문서에서 '성북동'이라는 공간 텍스트가 발견되었다고 가정하자. 공간 위치 맵핑 시스템이 '성북동'에 관한 문서의 공간적 위치를 추정하는 것은 매우 복잡한 문제이다. '성북동'이라는 지명이 Fig. 8에서 보이는 것과 같이 서울 성북구와 전남 나주시, 강원 삼척시 등 세 곳에 존재하는 동명이기 때문이다. 그러므로 위치 맵핑 시스템이 공간적 위치를 보다 정확하게 추정하기 위해서는 보다 많은 정보를 필요로 하게된다. 동일한 문서내에서 '성북동'이라는 지명과 함께 '성북구' 또는 '서울'이라는 단어가 출현했다고 가정하자. 그렇다면 가장 앞의 '성북동'이 실제로는 서울의 성북동임을 보다 확실히 확증할 수 있게 된다.

또 다른 예는 동일한 레벨의 지명이 중복해서 발견되는 경우이다. 지오웹 문서 내에 '논현동'과 '삼성동'이라는 공간 텍스트가 발견되었다고 가정하자. '논현동'이라는 지명은 Fig. 8에서 보이는 것과 같이 인천 남동구와 서울 강남구 두곳에 존재하는 지명이며, '삼성동'은 경남 양산시와 서울 강남구 두 곳에 존재하는 지명이다. 계층적 지명 구조가 없다면 이 문서의 공간적 위치를 좁히기 매우 어렵다고 할 수 있다. 그러나 이 논문에서 제안하고 있는 계층적 지명 구조에서 '논현동'과 '삼성동'은 동일한 '강남구'에 존재하는 지명임을 확인할 수 있으며 별도의 추가적인 정보 없이도 경남이나 인천이 아닌 서울 강남구에 있는 두 동에 대한 문서임을 보다 높은 확률로 확증할 수 있게 된다.

```
Algorithm : GeoTextHierarchyWeighting
Input : Vector of Geo Texts vgt
Output : Vector of Weighted Geo Texts vwgt
tree ← Load('Geo Texts Hierachy DB')
vwgt ← ∅
weightedTree ← ∅
foreach gt ∈ vgt do
    candidates ← tree.find( gt )
    foreach c ∈ candidates do
        weightedTree.addAndScoreUp( c )
vwgt ← addressTagging( vgt, weightedTree )
return vwgt
```

Fig. 9 GeoTextHierarchyWeighting algorithm

앞의 Fig. 9은 공간계층가중치분석기의 알고리즘을 보이고 있다. GeoTextHierarchyWeighting 알고리즘은 공간 텍스트들의 벡터인 *vgt*를 입력으로 받아, 계층적 공간 텍스트 트리 구조에 존재하는 명사들을 발견하여 공간적 가중치를 계산하고 이를 이용하여 지오웹 문서의 공간대상 후보들인 *vwgt*를 반환하는 알고리즘이다. 알고리즘의 첫 번째 부분에서는 계층적 공간 텍스트 데이터베이스를 로딩하여 *tree*에 반환하며, 반복문을 수행하며 입력된 *vgt*내의 각 명사 들에 대하여 *tree*내에 존재하는 *candiates*들을 찾게된다. 즉 '성북동'이라는 명사를 tree에서 찾는다면 세 개의 *candiates*들이 반환될 것이다. 이 *candiates*들은 *weightedTree*에 추가되며 가중치가 계산된다. 마지막으로 모든 공간 텍스트들을 이용하여 *weightedTree*의 구축이 완료되면, 이를 기반으로 *addressTagging* 함수를 이용하여 최종적으로 후보 공간 영역을 확정하여 *vwgt*로 반환하게 된다.

3.4 공간 텍스트의 공간좌표 변환 알고리즘

이 절에서는 가중치를 갖는 공간 텍스트들의 모음인 *vwgt*를 평가하여 공간정보로 변환하는 알고리즘을 설명한다. 앞의 알고리즘들은 공간 텍스트의 공간적인 후보 지역들의 후보들을 만들고 가중치를 추론해내는 과정의 알고리즘들이었다. 이렇게 만들어진 가중치를 이용하여 지오웹 문서를 실질적인 공간정보로 변환하기 위해서

는 마지막 단계로 공간좌표로 변환해야한다. Fig. 10은 *vwgt*를 실제 공간정보로 변환하는 GeoText2Geometry 알고리즘을 보이고 있다.

Algorithm : GeoText2Geometry

Input : Vector of Weighted Geo Texts *vwgt*
Output : Vector of Geometrized GeoText *vggt*
vggt ← ∅
foreach *wgt* ∈ *vwgt* **do**
  ggt ← *geometrize*(*wgt*)
  if *ggt* is exist **then**
    vggt ← *vggt* ∪ *ggt*
return *simplifying(vggt)*

Fig. 10 GeoText2Geometry algorithm

GeoText2Geometry 알고리즘은 *vwgt*를 입력으로 받아 공간좌표화 된 텍스트들의 벡터인 *vggt*를 반환하는 알고리즘이다. 알고리즘에서 보이는 것과 같이 *vwgt*내의 *wgt*를 하나씩 받아서 처리한다. 이때 POI들은 POI DB를 이용하여 공간좌표로 변환되며, 행정명 등은 행정공간정보를 이용하여 polygon 타입의 공간정보로 변환된다. 주소 정보 등이 존재할 때에는 DAUM/NAVER 지오코딩 API를 이용하여 공간좌표로 변환된다. 다음 절에서 질의와 구현 시스템에 대하여 기술한다.

4. 공간텍스트 질의 시스템 구현

이 장에서는 제안된 공간 웹 온톨로지 기반의 텍스 공간정보 위치 맵핑시스템 구현에 대해 기술한다. 제안 시스템의 지오웹 문서 분석 등의 알고리즘들과 웹 서비스 부분은 Java를 이용하여 J2EE Servlet의 형태로 구현되었으며, Tomcat을 이용하여 구동된다. 데이터의 저장은 대중적으로 많이 사용되는 공개 DBMS인 PostgreSQL상에서 공간 질의를 지원하는 PostGIS가 함께 사용되었다. 지오코딩과 같은 일부 기능은 NAVER와 카카오 Open API를 이용하여 시스템을 구현하였다.

지오웹 사용자 인터페이스를 통해 사용자로부터 공간정보 범위 *mbr*과 공간 텍스트 *q*가 입력되었을 때 지오웹 질의 처리기는 Fig. 11과 같은 알고리즘으로 질의를 처리한다.

Algorithm : GeoWebQuery

Input : Query MBR *mbr*, GeoText Query *q*
Output : Ranked Vector of Geometrized GeoTexts *rvggt*
vvggt ← GeoWebDBManager.select(*mbr, q*)
rvvggt ← GeoWebRanker(*mbr, q, vvggt*)
return *rvvggt*

Fig. 11 GeoWebQuery algorithm

GeoWebQuery 알고리즘은 입력된 조건 *mbr*과 *q*를 이용하여 GeoWebDBManager의 select 함수를 호출한다. select 함수는 데이터베이스에 저장된 공간 텍스트 문서들을 기반으로 공간 문서 질의를 수행하여 제시된 조건 *mbr*과 *q*를 만족하는 공간 텍스트 문서들인 *vvggt*를 반환한다. GeoWebRanker 함수는 반환된 *vvggt*들에 대하여 질의 조건에 가장 적합한 순으로 정렬하여 *rvvggt*를 생성하여 반환한다. 제안된 공간 텍스트 질의 시스템은 GeoWebQuery를 수행하여 반환된 *rvvggt*를 지도와 함께 웹상에 표출하는 시스템으로 구현되었다. 다음에서는 웹 기반 공간 텍스트 질의 시스템의 실행 결과들을 보인다.

그림 Fig. 12는 실험용 데이터를 이용하여 구축된 DB를 대상으로 경기 전지역을 질의 사각형으로 하고 '맛집'이라는 키워드를 질의로 하여 변환된 결과의 화면이다. 이 실험 구현에 사용된 데이터들은 네이버 블로그의 RSS(Rich Site Summary) 인터페이스에 의해 수집된 정보들을 이용하여 약 10,000개의 데이터로 구축되었다. 이 데이터를 대상으로 입력된 지역과 '맛집' 질의를 대상으로 Fig. 12에 보이는 것과 같은 10개의 문서가 반환되었다. 해당 검색어를 만족하는 문서는 300여개 정도 였으며 가시화를 위해 상위 가중치값을 갖는 10개만 반환되도록 구현되었다. 또한 공간텍스트 질의 시스템에 대한 반복적인 질의 시간 테스트에서 질의로부터 결과를 반환하기까지 걸린 시간은 대략 2초내에 완료되었다.

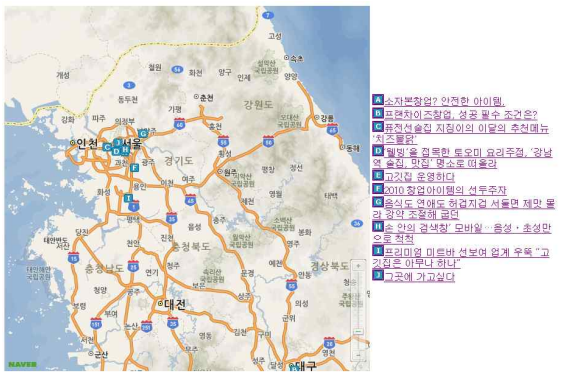

Fig. 12 Result of a GeoWebQuery for Kyongi-do

Fig. 13은 질의 영역의 공간적 지역을 좀 더 좁혀 서울시의 중심영역을 대상으로 하여 동일한 '맛집' 질의의 결과이다.

Fig. 13 Result of a GeoWebQuery for Seoul

위의 그림 Fig. 12와 Fig. 13외에 다양한 공간영역과 공간질의 키워드에 대하여 지오웹 문서의 결과들이 비교적 정확하게 반환되는 것을 시범 서비스를 통해 확인할 수 있었다. 제안된 시스템은 지도의 중심 지역을 대상으로 영역이 이동하거나 줌인 줌아웃을 수행할 경우 질의를 반복적으로 수행하는 편의 기능도 지원하고 있다.

5. 결 론

본 연구에서는 텍스트 공간정보 위치 맵핑 시스템을 구축하여 효율적인 텍스트 공간정보 위치 맵핑 기법을 제안하였다. 제안 알고리즘들은 웹 텍스트 내에 존재하는 지명들의 상관관계와 관련성으로 문서와 가장 관련 있는 지역을 추출하여 공간정보로 변환하고 데이터베이스에 저장하고, 이 정보를 기반으로 다양한 공간 영역 및 키워드 질의 서비스를 구축할 수 있었다. 향후 연구로는 다양한 복합 공간정보 구조를 지원하도록 확장하고, 보다 정확한 형태소 파싱을 수행할 수 있도록 하는 연구가 요구된다.

References

Borges, K.A.V. (2006). Use of an ontology of urban places for recognition and extraction of geospatial evidences on the web(in Portuguese). PhD Thesis, Federal University of Minas Gerais: Belo Horizonte(MG), Brazil.

Borges, K. A. V., Laender, A. H. F. , Medeiros, C. Bauzer , Davis, C. A. (2007). Discovering geographic locations in web pages using urban addresses. GIR. 31-36.

Borges, K. A. V., Davis, C. A., Laender, A. H. F., and Medeiros, C. B. (2011). Ontology-driven discovery of geospatial evidence in web pages. GeoInformatica, 15(4) 609 - 631.

Cong, G., Jensen, C. S., Wu, D. (2009). Efficient retrieval of the top-k most relevant spatial web objects. PVLDB, 2(1), 337-348.

Chen, Y., Suel, T., Markowetz, A. (2006). Efficient query processing in geographic web search engines. SIGMOD, 277-288.

Cui, N., Li, J., Yang, X., Wang, B., Reynolds, M., and Xiang, Y. (2019). When geo-text meets security: Privacy-preserving boolean spatial keyword queries. Proceedings - International Conference on Data Engineering 2019-April 1046 - 1057.

Dongjoo, L., Yeon, J., Hwang, I., and Lee, S.

(2010). KKMA : A tool for utilizing Sejong corpus based on relational database. Journal of KIISE: Computing Practices and Letters 16(11) 1046 - 1050.

Embley, D. W., Campbell, D. M., Jiang, Y. S., Liddle, S. W., Lonsdale, D. W., Ng, Y. K., and Smith, R. D. (1999). Conceptual-model based data extraction from multiple-record Web pages. Data and Knowledge Engineering 31(3) 227 - 251.

Ha, T. S. (2010). Location mapping techniques for textual spatial information based on spatial web ontology(Masters dissertation). Kunsan National University, Gunsan, Korea.

Hu, Y. (2018). Geo-text data and data-driven geospatial semantics. Geography Compass 12(11).

Laender, A. H. F., Ribeiro-Neto, B. A., Da Silva, A. S., and Teixeira, J. S. (2002). A brief survey of web data extraction tools. SIGMOD Record, 31(2), 84 - 93.

Lee, J. H. (2018). Building an SNS crawling system using Python. Journal of the Korea Industrial Information Systems, 23(5), 61-76.

Lee, T. (2020). A study on analysis of topic modeling using customer reviews based on sharing economy: focusing on sharing parking. Journal of the Korea Industrial Information Systems, 25(3), 39-51.

Ma, C., Zhao, Y., AL-Dohuki, S., Yang, J., Ye, X., Kamw, F., and Amiruzzaman, M. (2020). GTMapLens: Interactive lens for geo-text data browsing on map. Computer Graphics Forum, 39(3), 469 - 481.

McCurley, K. S. (2001). Geospatial mapping and navigation on the web. In Proc. of the Tenth Int'l World Wide Web Conference. 221-229.

Moon, C. B., Lee, J. Y., and Kim, B. M. (2019). Multimedia contents recommendation method using mood vector in social networks. Journal of the Korea Industrial Information System, 23(5), 11-24.

Rahimi, A., Cohn, T., and Baldwin, T. (2015). Twitter user geolocation using a unified text and network prediction model. ACL-IJCNLP, 630 - 636.

Yang, D. H, and Kim, Y. S. (2009). Evoluation of IS: geospatial web & u-GIS, Journal of Internet Computing and Services, 9(1), 44-55.

하 태 석 (Tae Seok Ha)

- 군산대학교 컴퓨터정보공학과 학사
- 군산대학교 대학원 컴퓨터정보공학과 석사
- 네이버시스템(주)
- (현재) (주)쿠첸
- 관심분야 : 데이터베이스, 공간정보시스템, 모바일 서비스

남 광 우 (Kwang Woo Nam)

- 충북대학교 컴퓨터과학과 학사
- 충북대학교 대학원 전자계산학과 석사
- 충북대학교 대학원 전자계산학과 박사
- 한국전자통신연구원 선임연구원
- (현재)군산대학교 컴퓨터정보통신공학부 교수
- 관심분야 : 데이터베이스, 인공지능, 공간정보시스템