

시맨틱 기술 개발을 지원하기 위한 Open API

박정원, 남세진, 최기선

KAIST 시맨틱 웹 첨단 연구센터

jwpark@world.kaist.ac.kr, jordse@gmail.com, kschoi@cs.kaist.ac.kr

Opne API for semantic technology

Jeong-Won Park, Se-jin Nam, Key-Sun Choi
Semantic Web Research Center, KAIST

요 약

현재 Open API는 프로그램 개발 및 정보의 제공 등 다양한 방향에서 많이 활용 되고 있다. Open API는 개발된 기술을 웹으로 공개하여 네트워크가 연결된 곳이라면 어디서든지 그 기술을 활용할 수 있다. 잘 개발된 Open API는 매쉬업을 통해 또 다른 기술과 합쳐 새로운 서비스를 개발하는 등 그 활용빈도가 높다. 이미 개발된 기술을 활용할 수 있어서 개발 속도가 단축되고 새로운 서비스를 개발하면서 기존의 부족한 부분을 합쳐 더욱 창의적인 정보 제공이 가능하다. 이러한 Open API의 특성을 이용하여 시맨틱 기술 개발을 지원할 수 있으며, 개발 시간 단축 및 관계 설정, 온톨로지 제공 등을 네트워크를 통해 사용자들에게 제공하고 활용할 수 있는 방법을 제시한다.

시맨틱 기술은 사람이 글을 읽고 의미를 이해하는 것처럼 컴퓨터가 이해할 수 있는 언어와 규칙들을 정해 놓아 컴퓨터 사이의 의사소통을 가능하게 하는 지능형 기술이다. 이런 시맨틱 기술 개발을 지원하기 위하여 Open API를 통해 온톨로지의 개념 및 속성, 관계 설정, 코퍼스 등을 제공하기 하기 위해 CoreOnto API라는 것을 구축했다.

CoreOnto API는 네트워크에서 사용할 수 있는 Open API로 세가지의 큰 형태의 Open API를 제공하고 있다. 첫 번째, 온톨로지에 대한 클래스, 속성, 관계를 제공하는 APIs, 두 번째, 트리플(두개의 용어가 어떤 관계인지를 표현한 형태로 구성(용어1, 관계, 용어2)) 생성 및 구축된 트리플을 검색할 수 APIs, 세 번째, 온톨로지를 생성하기 위해 필요한 자원(온톨로지 명, 개체명, 관계, 코퍼스 등)을 제공하는 APIs로 구성했다. 이상과 같이 시맨틱 기술 개발을 지원하기 위한 큰 형태의 세 가지 APIs를 구축하는 방법과 APIs의 활용, 그리고 서비스를 테스트한 결과를 제시한다.

주제어: Ontology, Semantic, Open api

1. 서 론

인터넷의 보급이 이루어지면서 인터넷을 통해 많은 일을 할 수 있다. 사용자들은 인터넷을 이용하여 다양한 정보를 얻고 또한 새로운 정보를 만들어 공개한다 그리고 인터넷을 활용하여 소스 관리, Database 관리, 문서 관리 등의 다양한 정보를 공유하고 소유할 수 있게 되었고, 인터넷이 연결된 곳이라면 장소와 시간에 구애 받지 않고 공유할 수 있게 되었다.

기존의 시스템 개발 및 연구의 경우 개발자 각자가 개발한 모듈을 이용하여 새로운 시스템을 개발하고 확장하였으며 각각의 개발된 모듈을 통합하여 시스템을 개발하는 방식을 많이 사용했다. 이처럼 개발에 필요한 모듈 분석, 작업 분담, 개발, 통합의 절차를 밟는 것이 일반적인 프로젝트 진행 절차 중 하나의 방법이었다 이 경우, 각 개발된 모듈은 별도의 통합 절차가 필요했으며 통합의 어려움과 최신 데이터의 유지 등이 어려웠다

이 문제점을 해결하고 모듈의 중복 개발 방지를 위해 Open API 형식으로 필요 모듈을 개발하였으며, 시맨틱 기술 개발의 시간 단축을 하고, 시맨틱 기술 개발에 활용할 수 있도록 지원했다. 그래서 개발된 Open API를 CoreOnto API¹⁾라고 하였다.

본 논문은 다음과 같이 구성된다. 2장에서는 현재 사

용되고 있는 Open API 동향에 대해서 언급하고, 3장에서는 CoreOnto API의 설명과 구축에 대해 언급하며, 4장에서는 시맨틱 기술을 지원을 위한 네트워크 시스템 테스트 결과에 대해 제시한다 그리고 마지막 5장에서는 결론과 향후 발전 방향에 대해서 제시한다

2. Open API 동향

Open API는 많은 웹 서비스 업체에서 제공하는 공개 API 서비스로 대표적인 Open API는 Google²⁾과 Yahoo³⁾의 Map API, Amazon⁴⁾의 Web Service API, Swoogle⁵⁾의 Semantic web search API 등이 있다. 그중 시맨틱 기술을 지원을 위한 온톨로지를 검색할 수 있는 Swoogle이 있다. 이 Open API는 공개 네트워크인 인터넷의 장점을 최대한 활용하여 하나의 자리를 만들어 가고 있다

1) <http://coreontoapi.kaist.ac.kr/>

2) <http://code.google.com/intl/ko/apis/maps/>

3) <http://developer.yahoo.com/maps/>

4) <http://aws.amazon.com/>

5) <http://swoogle.umbc.edu/>

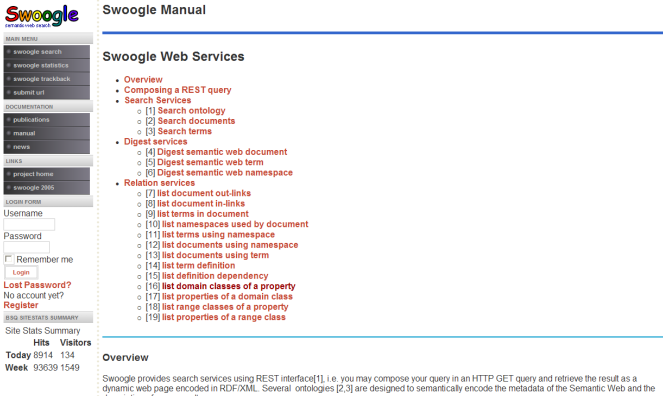


그림 1. Swoogle의 온톨로지 검색 Open API

개발자는 자신이 필요한 부분의 모듈을 이미 개발된 모듈의 Open API를 자신의 시스템에 붙이는 것으로 해당 부분의 모듈 개발이 완료 된다. 그러면 새로운 모듈을 개발하는 시간을 단축하고, 개발에 필요한 비용을 줄일 수 있다.

특히 Swoogle의 경우 온톨로지를 검색할 수 있는 Open API를 제공함으로써, 온톨로지 확장이나 기존 구축된 온톨로지를 비교 분석할 수 있어 시맨틱 기술 개발을 지원하는 데 도움을 주고 있다.

이처럼 OpenAPI를 선택하여 모듈에 붙이는 것만으로도 개발이 완료 될 수 있는 최대의 장점을 가지고 있다

3. CoreOnto API 시스템 구현

3.1 시스템 입력

CoreOnto API를 구현하기 위해 첫번째 고려 대상은 CoreOnto API의 접근 방법이다. CoreOnto API는 인터넷을 통해 접근하기 때문에 인터넷의 특성을 파악해야 한다. 그러므로 사용자의 입력과 결과를 사용자에게 주고받기 위해 간단하면서도 표준화된 방식이 필요하다.

그래서, 입력 방식으로는 쉽게 접근하기 위해 HTTP 프로토콜을 활용한 REST 방식을 채택했다. REST 방식은 HTTP의 주소에 입력변수 값을 추가하는 방식으로 데이터를 전송하게 되어, 사용자는 쉽고 간편한 방식으로 입력값을 전송할 수 있다.

시스템 입력 방식의 예는 다음과 같다

```
http://coreontoapi.kaist.ac.kr:8080/DBpedia/userapi?searchAPI=articleLabel&searchType=exact&searchString=Asus&offset=0&limit=10
```

위의 입력 변수의 설명은 아래와 같다

변수 이름	변수 설명
http://coreontoapi.kaist.ac.kr:8080/DBpedia/userapi	서비스 URI(호출 주소)
searchAPI(String)	API 종류 선택
searchString(String)	검색 단어
searchType(String)	exact(전체매칭) or partial(부분매칭)
offset(Int)	검색의 시작 위치
limit(Int)	출력 개수

표 1. 입력 변수 및 값

3.2 시스템 출력

CoreOnto API를 구현하기 위해 두 번째 고려 대상은 CoreOnto API의 출력 방법이다. 시스템의 출력 방식으로 XML, RDF/OWL과 JSON방식을 고려했다. 이 중 XML과 RDF/OWL의 경우 출력 결과가 크고 사용자들의 별도의 파싱 프로그램을 사용하여야 하며 후처리가 필요했다. 그래서, OWL, XML 또는 RDF 문법이 필요한 경우 XML, RDF/OWL를 사용했고, 그 외의 경우 사용이 용이하고 호환성이 높은 JSON을 선택하였다.

JSON은 JavaScript Object Notation의 약자로서 인터넷에서 자료를 주고받을 때 간편하게 사용할 수 있으며 JSON은 자바스크립트의 구문 형식을 따르지만 프로그래밍 언어나 플랫폼에 독립적이므로 C, C++, 자바, 자바스크립트, 펄, 파이썬 등 다양한 언어에서 활용이 가능하다는 장점이 있어 XML 보다 다양한 사용자들이 원하는 자료를 간단하게 제공받을 수 있다.

```
{"name2": 50, "name3": "값3", "name1": true}
```

표 2. JSON 출력 방식의 예

3.3 시스템 구조

세 번째로 전체 시스템 내부 구조이다. 이상과 같이 입력과 출력이 결정이 되었으므로 CoreOnto API의 시스템 구성을 하였으며, 총 4개의 Layer로 구성했다.

Layer 4, 앞에서 정의한 정의한 입출력을 담당한다.

Layer 3, 웹 서비스를 구성하여 입력을 전달하고 처리된 결과를 제공하게 될 웹서비스를 구성하고 로그, 예외 처리 등의 이벤트를 관리를 위해 AXIS를 추가했다.

Layer 2, 요청된 CoreOnto API를 처리를 위한 모듈 구성으로 Ontology Access, RDF Access, Triplet Extractor의 모듈을 개발했고, 요청된 API에 대한 처리를 할 수 있도록 했다.

Layer 1, 온톨로지를 저장할 수 있는 저장소와 Indexing Server로 활용할 Lucene을 구성했다.[그림. 2]

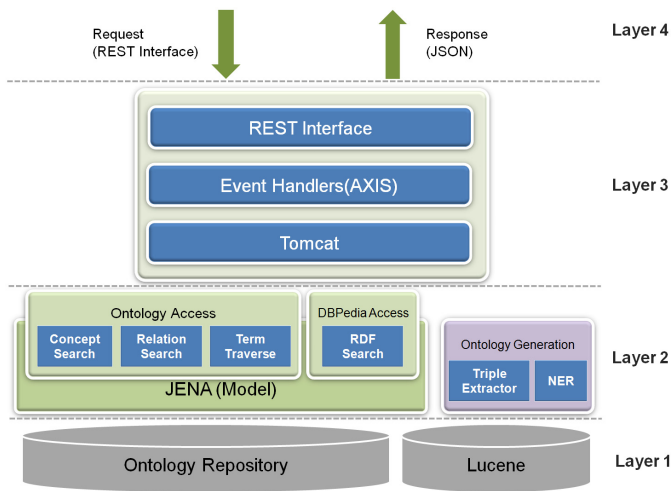


그림 2. Open API 시스템 구조

3.4 CoreOnto API 구성

시맨틱 기술을 지원하기 위해 필요한 CoreOnto API를 다음과 같이 구성했다. 온톨로지를 제공할 Ontology Access CoreOnto APIs, 온톨로지 확장을 지원하기 위한 RDF Access CoreOnto APIs, 관계 추출을 지원할 Ontology Generation CoreOnto APIs로 구성했다.

첫 번째로, Ontology Access Open APIs를 이용하여 온톨로지의 클래스 검색 및 프로퍼티 검색의 결과를 이용하여 온톨로지 연결을 지원하는 등 온톨로지에 대한 지원을 하는 CoreOnto API로 구성되어 있다.

Ontology Access Open APIs 리스트에 대한 상세한 정보는 아래의 [표 3]과 같다.

API 정의	API 종류 이름	입력 값	출력 Type
온톨로지 클래스 검색	findClasses	검색 단어 출력 개수 출력 위치	TEXT/JSON
온톨로지 상위 클래스 검색	getSuperClasses	Class의 URI 주소	TEXT/JSON
온톨로지 하위 클래스	getSubClasses	Class의 URI 주소	TEXT/JSON
온톨로지 상위 프로퍼티	getSuperProperties	Property의 URI 주소	TEXT/JSON
온톨로지 하위 프로퍼티	getSubProeprties	Property의 URI 주소	TEXT/JSON
온톨로지 클래스 정보	getClassInfo	Class의 URI 주소	TEXT/JSON
온톨로지 프로퍼티 정보	getPropertyInfo	Property의 URI 주소	TEXT/JSON

표 3. Ontology Acces APIs

두 번째로, RDF Access Open APIs로 온톨로지를 확장하기 위한 자원을 활용할 수 있는 API로 구성되어 있으며,

현재는 DBPedia의 자원을 가져와서 활용할 수 있도록 지원하고 있다.

RDF Access Open APIs 리스트에 대한 상세한 정보는 아래의 [표 4]와 같다.

API 정의	API 종류 이름	입력 값	출력 Type
위키피디아 Article	articleLabel	검색 단어	TEXT/JSON
위키피디아 Article를 가리키는 Redirect Article	redirect	검색 단어	TEXT/JSON
Category의 Upper Category	Categories	Category의 URI	TEXT/JSON
Category의 Sub Category	subCategories	Category의 URI	TEXT/JSON
Article이 가지는 Category	articleCategories	Article의 URI	TEXT/JSON
Category가 포함하고 있는 Article	categoryArticles	Category의 URI	TEXT/JSON
위키피디아 Article의 내용 (500자 내)	shortAbstract	Article의 URI	TEXT/JSON
위키피디아 Article의 내용 (3000자 내)	extendedAbstract	Article의 URI	TEXT/JSON
위키피디아의 구조화된(표) 데이터 모음	Infobox	Article의 URI	TEXT/JSON
Infobox에서 사용하는 Property 정보	Properties	Infobox의 Property URI	TEXT/JSON
WordNet	WordNet	Article의 URI	TEXT/JSON

표 4. RDF Access APIs

세 번째로, 문장에서 관계 추출과 관계 설정을 지원하고 및 NER(Named Entity Recognizer)를 제공하는 CoreOnto API로 구성하여 시맨틱 기술 개발을 지원하고 있다

Ontology Generation Open APIs 리스트에 대한 상세한 정보는 아래의 [표 5]와 같다.

API 정의	API 종류 이름	입력 값	출력 Type
문장에서 트리플 자동 추출	getTriple	문장	RDF/OWL
트리플 검색	getTripletList	검색 단어 출력 개수 출력 위치	TEXT/JSON
NER	getNameEntity	문장	TEXT/XML

표 5. Ontology Generation APIs

4. CoreOnto API 배포 테스트

CoreOnto API는 공개된 네트워크를 통해 접근하여 사용하는 모듈이므로 네트워크의 부하 및 서버의 접근에 대한 관리 능력 및 처리 능력을 고려해야 한다 이것은 접속자가 한꺼번에 요청을 하거나 대량의 작업을 처리하기 위해 지속적으로 작업 요청을 할 때 발생할 수 있는 문제점을 사전에 파악하기 위함이다. 네트워크에 대한 처리 능력은 한계가 있으며, 요청에 따라 다른 작업의 경로를 확인할 필요가 있기 때문이다. 그러므로 CoreOnto API에 접속 테스트를 했다.

다음은 CoreOnto API에 대해 접근에 대한 부하 테스트의 결과이다.

System1	시스템 구성	사양
	Cpu	Intel Core2Duo 1.8Ghz
	Memory	1GB
	OS	Windows XP
	TEST Web Server	TOMCAT 5.5
System2	시스템 구성	사양
	Cpu	Intel Pentium 3.4Ghz
	Memory	1GB
	OS	Windows XP
	TEST Web Server	OPENSTA 1.4

표 5. 테스트 시스템 구성

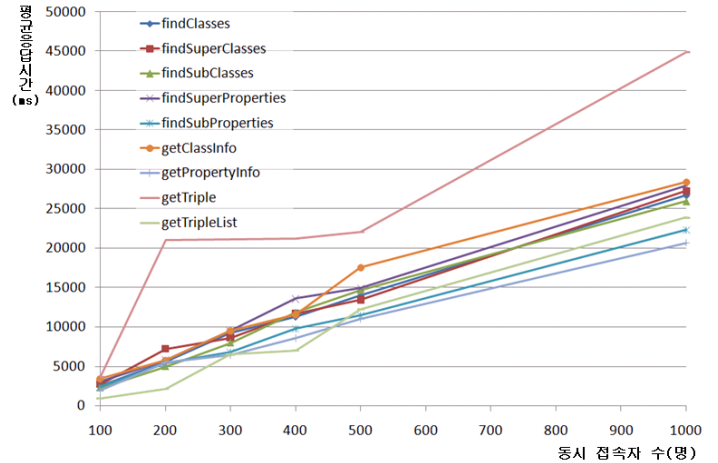


그림 3. CoreOnto API 서버 부하 테스트 결과

위 테스트는 동시에 사용자가 갑자기 접속 하였을 때 접속 결과를 보여 주고 있다. 100명 이하의 인원이 동시에 접속하였을 경우 0.5초 내의 응답 시간을 나타내고 있으나 1000명의 사용자가 동시에 접속하였을 경우에는 약 2초가량의 응답 시간이 필요한 것으로 나타내고 있다

그리고 CoreOnto API중 getTriple의 경우 NER 모듈을 이용한 것으로 문장에서 triple을 추출하는 API이다. 이 API는 사용자의 입력 문장을 별도로 분석하기 위한 분석 시간과 관계를 설정해 주는 시간이 필요해 다른 API 보다 속도가 늦은 것으로 확인했다.

5. 결론 및 향후 계획

Open API를 이용할 경우 기존 수평적으로 관리되던 자원과 온톨로지를 하나로 관리하고 제공함으로써 관리의 용이성과 기존 모듈의 활용으로 시맨틱 기술 연구에 도움을 줄 수 있었다. 시맨틱 기술을 개발하면서 필요한 모듈을 CoreOnto API로 제공함으로써, 연구의 속도를 높여주고 흩어져 있는 모듈을 통합하여 사용자는 원하는 API를 선택하여 사용할 수 있었다. 시맨틱 기술 개발에 꼭 필요한 온톨로지의 확장과 검색에도 도움이 되어 최신의 온톨로지를 제공받음으로서 온톨로지의 중복을 방지하고 최신의 온톨로지를 제공받아 연구할 수 있었다.

이상과 같이 본 논문에서는 시맨틱 기술을 지원하기 위한 CoreOnto API에 대해서 설명하였고, 시맨틱 기술을 지원하기 위해 구축되는 시스템에 대한 구축 결과에 대해서 제시했다.

앞으로 사용자의 급격한 증가를 고려하고 공개된 네트워크의 보안을 위해 인증 Key 및 사용자 관리에 대한 개발이 필요하다. 이것을 통해 인증되지 않은 접속자에 대해서 사전에 위험요소를 제거할 필요성이 있기 때문이다

시맨틱 기술에 필요한 모듈의 확장과 온톨로지 관련 API 및 리소스 자원에 대한 API의 추가적인 개발이 필요하다. 시맨틱 기술 개발을 위한 API로 구성하여 핵심적인 역할을 하는 CoreOnto API를 위해 연구가 계속 진행 중이다

감사의 글

본 논문은 한국산업기술평가관리원의 정보 통신 선도기반 기술개발 사업의 연구 결과로 수행 되었습니다

참고문헌

- [1] Davies, J., Fensel, D., and van Harmelen, F., ed. (2002) Towards the Semantic Web - Ontology-Driven Knowledge Management. Wiley.
- [2] Gruber, T. R. (1993) A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, Vol. 5.
- [3] McIlraith, S. A., Son, T. C., and Zeng, H. (2001) Semantic Web Services, IEEE Intelligent Systems, Vol. 16, Num. 2, pp. 46-53.
- [4] Noy, N. F., M. Crubezy, et al. (2003) Protege-2000: an open-source ontology-development and knowledgeacquisition environment. AMIA Annual Symposium Proceedings, pp. 953
- [5] Philipp, C. and Volker, J. (2005) Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery, In Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems - NLDB'05 (pp. 227-238), June 15-17
- [6] Carroll, J. J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., and Wilkinson, K. (2004) Jena: Implementing the Semantic Web Recommendations, In the proceeding of 13th World Wide Web Conference, pp 74-83
- [7] Wikipedia, <http://en.wikipedia.org/wiki/Json>
- [8] DBPedia, <http://dbpedia.org/About>
- [9] McGuinness, D. L. and van Harmelen, F. (2003) Owl web ontology language overview, <http://www.w3.org/TR/2003/WD-owl-features-20030331/>, March 2003
- [10] Choi, D. H. and Choi, K. S. (2008) Automatic Triplet Relation Extraction by Dependency Parse Tree Traversing, EKAW 2008 Poster, Catania, ItalySmith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195--197 (1981)