International Journal of Pure and Applied Mathematics
Volume 118 No. 19 2018, 1869-1883
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

Architecture of Knowledge Graph Construction Techniques

Zhanfang Zhao[1], Sung-Kook Han[2],In-Mi So*[3]
[1]College of Information Engineering,
Hebei GEO University,
Hebei, 050031, China
[2]Department of Computer Engineering,
WonKwang University,
City Iksan, JeonBuk, 54538, Korea,
[3]College of Liberal Arts, WonKwang University,
City Iksan, JeonBuk, 54538, Korea
zzfsjz@gmail.com[1], skhan@wku.ac.kr[2],
s0301@wku.ac.kr[3]
Corresponding author*
Phone:+82-010-4947-1007

February 4, 2018

Abstract

Background/Objectives: Some large-scale knowledge graphs (KGs) have been released to enhance the application of Web of Data. This paper analyzes the development procedures of KGs and their related technologies to provide the practical information.

Methods/Statistical analysis: This paper surveys and analyzesthe architectures of the well-known KGs such as DBpedia, Google Vault, YAGO, and NELL to derive the core system components. The development approaches of KGs are categorized by means of the properties of their core components. The paper also investigates the general

system architecture and procedures of KG development and analyzes their core technologies used in each phase. The recent trends in KGs will be reviewed.

Findings: There are two types of the approaches in the development of KGs: top-down approach focusing on knowledge schema such as the domain ontologies and bottom-up approach focusing on knowledge instances such as Linked Open Data (LOD) datasets. In the top-down approach emphasizing the well-defined domain ontologies, the domain ontologies and their schema should be defined at first, and then and then knowledge instances are added into knowledge base. The bottom-up approach extracts knowledge instances from knowledge resources. After knowledge fusing the populated instances, the top-level ontologies are built by means of knowledge instances to create the whole KGs.

The general procedures of KG development consist of three phases: knowledge extraction, knowledge construction and knowledge management. In the knowledge extraction, text mining and other text analytics are important technologies. Machine Learning (ML) gets attention in knowledge extraction recently. For knowledge construction, LOD is commonly used as the basic knowledge model. Some KGs try to use NoSQL databases and their query languages. Although knowledge management is relatively behind the development, versioning and refinement is important to the continuous quality improvement.

Improvements/Applications: The presented is paper presents the general architecture of KGs and the development approaches. The related important technologies applied in each phase of KG development are also discussed. This will provide the clear understanding of the state-of-the-art in knowledge graph development.

***Key Words*:** Knowledge graph, Knowledge acquisition, Knowledge fusion, Visualization, Machine learning, Linked Open Data

1 Introduction

The research on knowledge graph is booming, sincein 2012 the Google announced that using knowledge graph technology to improve the search engine's capability in order to enhance the user's

search quality and search experience. Nowadays, along with the continuous development of intelligent information service applications, the knowledge graph has been widely applied to many fields. However, it is still difficult to understand the construction and development of knowledge graph due to the limited public disclosure about the technology details. Therefore, this paper provides a survey on the knowledge graph construction, as well as analyses techniques and challenges during construction process [1,2].

This paper providesarchitecture of knowledge graphs, which describes the detailed construction procedure. Base on the architecture, the relative techniques are discussed, which include knowledge acquisition, knowledge fusion, knowledge storage, knowledge query and visual display. In addition, this paper also presents the problems and challenges of constructing knowledge graphsand discusses the future perspectives of relative techniques.

In the last few years,some large-scale knowledge graph products have been released, such as Google Knowledge Vault, YAGO, NELL and WordNet. These knowledge graph products have greatly promoted the development of semantic technique. However few details have been published about knowledge graph construction, since the technique competition between enterprises.

Some reports and papers focused on techniques of knowledge graph construction have been published in recent years. Nevertheless, most of papers have not given clear construction procedures. The following will review previously research on techniques of knowledge graph construction.

Due to the diversity of knowledge sources in knowledge graph, knowledge extraction and fusion techniques based on different data sources have been widely studied [3,4,5].Storage of knowledge graph still doesnt have a wildly accepted scheme until now, even though many NoSQL storage scheme have been proposed in the past years[6]. Some storage schemes are based on RDF, also known as triple store. Some storage schemes are based on graph database such as Neo4j and MongoDB.At present, query and visualization is a critical problem to promote the development of knowledge graph. A lot of researches about knowledge visualization have been conducted [7,8].

2 Architecture of Knowledge Graphs

In general, the architecture of knowledge graphs can be derived as shown in Figure 1. From the perspective of knowledge graphs based on ontology, there are two main approaches to create knowledge graph. One is top-down, and the other is bottom-up. Top-down approach means that ontology and schema should be defined, and then knowledge instances are added into knowledge base. This approach emphasizes the well-defined domain ontologies to represent the actual instances of knowledge graphs. The bottom-up approach extracts knowledge instances from the Linked Open Data (LOD) or other knowledge resources. After knowledge fusing the populated instances, the top-level ontologism are built by means of knowledge instances to create the whole KGs. This papermainly discuss bottom-up approach and related techniques. Figure 1 shows bottom-up approach of knowledge graph construction. The part of the dashed box is the creation process that needs to constantly iterative.

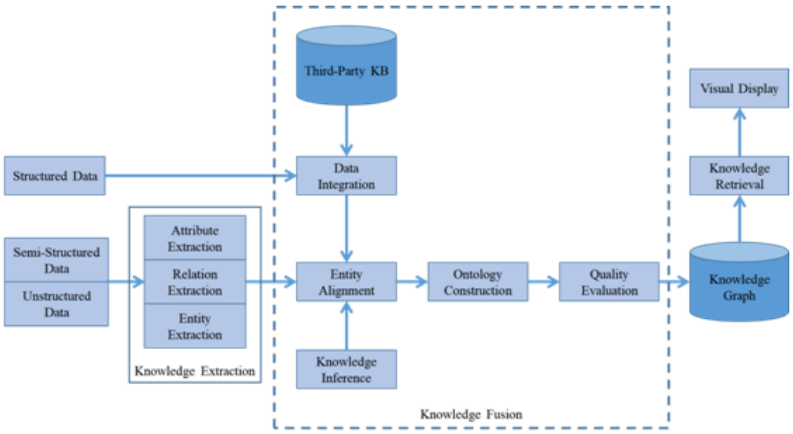

Fig. 1. The Architecture of knowledge Graph

Knowledge construction of the bottom-up approach is an iterative update process, including knowledge acquisition, knowledge fusion, knowledge storage and retrieval. The primary sources of knowledge acquisition include structured data, semi-structured data and unstructured data. Knowledge extraction consists of entity extraction, attribute extraction and relation extraction. Knowl-

edge fusion is an iterative process, which need to constantly construct ontology and evaluate quality of ontology. Currently, knowledge storage is usually based on NoSQL databases.

3 Procedures of Bottom-up Approachof Knowledge Graphs

This section describes the overallprocedures of the bottom-upapproach to construct knowledge graphs. Figure 1 shows the whole architecture of this method. Each of following subsections presents the technologies used in each phase in this architecture.

3.1 *Knowledge Extraction*

Knowledge extraction is analyzed from four aspects: data sources, types, approaches and tools.

3.1.1 *Data sources*

For knowledge extraction, there are mainly three kinds of data sources: structured data likerelational databases, semi-structured data like HTML, XML and JSON data, unstructured data like free text, image and documents.Different data sources result in different methods and strategies of knowledge extraction. The extracted knowledge is usually represented in a machine-readable format, such as RDF and JSON-LDformat.

Most ofthe early knowledge graphs only extract knowledge from specific and single data source. For example, the early knowledge graphs of medical industry usually extract knowledge from electronic medical record or medical paper on the line. Contrast to industry knowledge graphs, large-scale comprehensive knowledge graphs extract knowledge from a great variety of data sources. For example, Google Knowledge Vault gains knowledge from text documents, HTML trees, HTML tables and human annotated pages.

Beyond that, some datasets and websites provide some high quality knowledgeinstances for knowledge graphs. For example, Wikipedia, the typical Web-based encyclopedia, has become the most important instances source for knowledge graph.At present,most

of knowledge graphs extract the instances from Wikipedia such as YAGO and DBpedia. Some industry knowledge graphs also get knowledge from Wikipedia. For example, the medical CRF (Conditional Random Fields) system captures medical knowledge from Wikipedia [9].In addition, WordNet, GeoNames, ConceptNet, IMDB (Internet Movie Database) and MusicBrainz are also good knowledge resources.

Heterogeneous, cross-domain and multilingual are typical characteristics of current knowledge resources for large-scale knowledge graph, which bring great challenges for techniques of knowledge extraction.

3.1.2 *Types of Knowledge Extraction*

The types of knowledge extraction are roughly divided into three types: entity extraction, relation extraction and attribute extraction. In fact, the attribute extraction can be thought as a kind of special relation extraction.

Entity extraction, including named-entity recognition (NER), is to discover entities from a wide variety of knowledge resource and try to classify them into pre-defined categories such as person, location, organization, news title, service,time, date, and so on. The quality of entity extraction usually greatly influences the efficiency and quality of subsequent knowledge acquisition, so it is one of the most fundamental and important part of knowledge extraction.

After entity extraction, the relationships among the entities are analyzed to the conceptual extract relations. Relation extraction is to find the relations between entities and obtain semantic information in order to construct knowledge graphs.

The attribute extraction is to define the intentional semantics of the entities while the relation extraction is to specify the denotational semantics of the entities. The attribute extraction is important to define the concept of the entity more clearly.

3.1.3 *Approaches of Knowledge Extraction*

Approaches of knowledge extraction involve in Natural Language Processing (NLP), text mining, and machine learning.Early knowledge extraction usually uses manually annotated corpus.At that time, knowledge extraction approaches mainly used rule-based and

dictionary-based methods. Machine learning approaches mainly use the supervised learning algorithms that build learning model from manually annotated training dataset.However, manually annotated corpus need large number of domain experts and need to spend much time to generate. The scale of resulting corpus is usually small, which is hard to meet the need of knowledge extraction.Moreover, the supervised approaches have significant limitation even though they can get high accuracy. The supervised approaches rely too heavily on the original corpus because theycannot identify new named entitiesand cannot generalize to different relations.

At present, some semi-supervised and unsupervised algorithms have been proposed to avoid part of the annotation effort. Many different classifier types of machine learning have been applied to knowledge extraction, such as Hidden Markov Models (HMM) [10], Conditional Random Fields (CRF) [11], k-Nearest-Neighbors (KNN) [12], Maximum Entropy Models [13]and Support Vector Machines (SVM)[14].The performances of these machine learning algorithms depend on the features that are used.

For the evaluation of the quality of extraction algorithms, precision, recall and F-measure are usually used. Precision, also called as positive predictive value, is the proportion of correct classifications where the instances are judged to be positives. Recall, also known as sensitivity, is the proportion where the positives are judged to be positives.Precision is used to measure the quality of classification results and recall is used to measure the ability of the classifier to find the correct match. F-measure is the harmonic mean of precision and recall,also is a comprehensive evaluating indicator.The definition of these criteria factors are as follows:

$$\text{precision}=\frac{TP}{TP+FF} \quad \text{recall}=\frac{TP}{TP+FN}$$

$$\text{F-measure}= 2\times(\frac{precision\times recall}{precision+recall})$$

TP (True Positive): a positive instance that is also predicted to be positive.

FP (False Positive): a negative instance that is predicted to be positive.

FN (False Negative): a positive instance that is predicated to be negative.

3.1.4 *Tools of Knowledge Extraction*

Over years many tools for knowledge extraction have been released, which greatly facilitated the development of relatedtechnologies.The following table 1 lists some knowledge extraction tools and their usages.Different tools show different performances and functions.It is required to select suitable tool according to the tasks of knowledge extraction and the features of knowledge resources.

TABLE 1.Comparison of Knowledge extraction tools

Name	usages
Stanford NER [15]	Entity extraction
OpenNLP [16]	Entity extraction
AIDA [17]	Entity extraction
CiceroLite [18]	Entity extraction, sense tagging, relation extraction, and semantic role labeling.
FOX [19]	Entity extraction, sense tagging, term extraction, and relation extraction.
Open Calais [20]	Entity extraction, relation and fact extraction
ReVerb [21]	Identifies and extracts binary relationships
Wikimeta [22]	Multilingual named entity recognition and sense tagging.

3.2 *Knowledge Fusion*

The goal of knowledge fusion is to realize entity alignment and ontology construction, which is an iterative process. Ontology construction will not end until the results of quality evaluation meet requirements.

3.2.1 *Entity Alignment*

Entity alignment, also known as entity resolution or entity matching, is the process to judge whether or not different entities refer to the same objects of the real world. Entity alignment usually uses a variety of entity matching techniques combined with the characteristics of knowledge graphs to identify and align the entities that refer to the same objects. Figure 2 shows the process of entity alignment in detail.

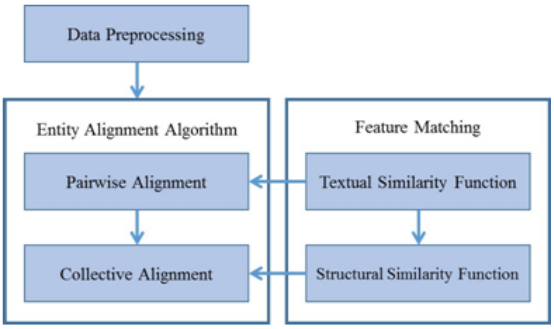

Fig. 2. The Process of Entity Alignment

Data preprocessing is also called standardization of data, which is important step of entity alignment task. Data preprocessing must deal with the integrity and consistency issues, such as multi-source heterogeneous, inconsistency of data definitions and diversity of data representations. Data integration in Figure 1 also belongs to data preprocessing.

Entity alignment has two steps: pairwise align and then collective entities align.The feature matching is usually used to realize entity alignment.The feature matching is originated from NLP applications that consists of attribute similarity function and relation similarity function.While the textual similarity function is used to match and compare the attributes, the structural similarity function is used to match and compare the relationships.The process of entity alignment usually relies on external information such as Wikipedia links or entity information in manually developed corpus. In recent years, knowledge inferencehas been proposed to apply to entity alignment. The main idea is to use logic rules obtained from the third knowledge graph or corpus in order to identify and align entity[15].

3.2.2 *Ontology Construction and Evaluation*

The goal of knowledge fusion is create ontology and construct knowledge graph. Except aligned entity and relation, creating ontology and knowledge graph need more works: constructing taxonomy and hierarchical structure, adding metadata and other data source.In order to assure the quality of knowledge graph, using general ontology like FOAF and general metadata from schema.org are nec-

essary. If the quality evaluations of ontology and knowledge graph do not meet requirement, the process of constructing and fusion of knowledge graph will be iterate.

3.3 *Storage of Knowledge Graph*

Knowledge graphs isusually stored in NoSQL databases. There are two main storage types: one is RDF (Resource Description Framework) based store, and the other is to use graph database store.

RDF is a representation of knowledge graphs, which uses triple (subject, predicate and object) and IRI/URIto describe the graph structure. Early RDF storage is non-native: DBMS based approaches which use triple table, property table and vertical partitioning to store and query RDF data. Later, a series of native storage systems have been proposed such as Jena2, 3store, RDFStore, 4Store, TripleT, RDF3X, Virtuoso, and so on. Most of native storage systems provide SPARQL or SPARQL-like query. At present, some new storage approaches combined NoSQL databases with RDF storage systemshave been proposed. For example, the hybrid storage approaches such as Jena+HBase, Hive+HBase, Cassandra+Sesame have been test and applied for the large volume of knowledge [16, 17].Some NoSQL databases like Couchbasehavebeen also applied to store RDF-modeled knowledge.The advantage of RDF-based knowledge graph storage is that the efficiency of query and merge-join of triple patterns is good. However, the query efficiency is improved by indexing,the better query resultsrequest huge cost of storage space.

Graph databasesare another important way to store knowledge graph, which stores nodes, edges and properties of graphs. The advantages of this approach are that the graph databasesthemselves provide the perfect graph query languages and support a variety of graph mining algorithms. However, the distributed storage of the graph databasescause some management problems: slow update of knowledge, high maintenances cost and inconsistence of distributed knowledge. The typical graph database Neo4j is popular so that it is an open source project and provides native graph storage.

There is no appropriate storage scheme proposed for knowledge graphs. The followings are the primary requirements for the storage

of large-scale knowledge graphs.

- The underlying storage should be guaranteed to be scalable and highly available, which can use relational databases, NoSQL databases, and in-memory databases.
- Data segmentation can be done as required.
- Cache and index are used timely.
- The storage systems can handle the large volume of knowledge graphs effectively. .

3.4 *Retrieval and Visualization of Knowledge Graphs*

SPARQL is widely used as the standard query language of knowledge graph. Almost all large-scale knowledge graph systems provide SPARQL query endpoint. There are many kinds of output formats of SPQRAL query result such as JSON, JSON-LD, XML, RDF/XML, RDF/N3, CSV, TSV and HTML. However, almost all the output formats are machine-readable, not human-readable. Therefore, the visualization of knowledge graphs is one of the hot research topics [18]. Some formats of query results of knowledge graphs are based on text. The visualization using the browsers are the most common approaches as shown in IsaViz, RDF Gravity, DBpedia Mobile, Fenfire and OpenLink Data Explorer. These two types of approaches have their own pros and cons. The query result formatsbased on text provide fine grained analysis of knowledge graph. The graphical visualization of knowledge graphs allows the bigger picture to navigate and discovery the related knowledge.

Knowledge retrieval is also semantic retrieval. Knowledge retrieval is no longer simple character matching. It usuallyuses logic rules under semantic model and inference model to realize retrieval since ontology is based on description logic. Consequently, knowledge retrieval has the capability of reasoning.At present, knowledge graph has been widely applied to smart search, Q/A systems and recommendation systems.

4 Conclusion

This paper discusses two approaches of knowledge graph construction: top-down and bottom-up. The architecture of bottom-up approach is described in detail, which consists of knowledge acquisition, knowledge fusion, knowledge storage and retrieval.

This paper introduces four aspects of knowledge extraction with their concepts, related approaches, and development tools. Currently, the heterogeneous,multilingual and cross-domainknowledge resources bring great challenge for the techniques of knowledge extraction.Entities alignment is critical step during knowledge fusion, which can use not only NLP methods but also knowledge inference methods. Two types of storage schemas of knowledge graph are discussed: RDF-based and graph database. Their pros and cons are analyzed. This paper discusses the storage principles for large-scale knowledge graphs and knowledge retrieval and visualization of knowledge graph.

This paper presents a general procedure of knowledge graph construction and reviews the related technologies and the research issues. This will help to understand the state-of-the-art in knowledge graph construction and development.

Acknowledgment

This paper was supported by Wonkwang University in 2017

References

[1] Sudeepthi G, Anuradha G, Babu M S P. A survey on semantic web search engine. IJCSI International Journal of Computer Science Issues, 2012, pp. 241-245.

[2] Piskorski J, Yangarber R. Information extraction: past, present and future, multilingual information extraction and summarization. Springer Berlin Heidelberg, 2013, pp. 23-49.

[3] Heng Ji, Challenges from Information Extraction to Information Fusion. Proceedings of the 23rd International Conference on Computational Linguistics, 2010. pp. 507-515.

[4] Han Xianpei, Zhao Jun. Named entity disambiguation by leveraging wikipedia semantic knowledge. Proc of the 18th

ACM Conf on Information and Knowledge Management. New York: ACM, 2009, pp. 215-224.

[5] Mendes P N, Muhleisen H, Bizer C. Sieve: Linked data quality assessment and fusion. Proc of the 2nd Int Workshop on Linked Web Data Management at Extending Database Technology. New York: ACM, 2012, pp. 116-123.

[6] Blin G, Cur O, Faye D C. A survey of RDF storage approaches[J]. REVUE AFRICAINE DE LA RECHERCHE EN INFORMATIQUE ET MATHMATIQUES APPLIQUES, 2016, 15.

[7] Dadzie A S, Rowe M. Approaches to visualising linked data: A survey[J]. Semantic Web, 2011, 2(2): 89-124.

[8] Bikakis N, Sellis T. Exploration and visualization in the web of big linked data: A survey of the state of the art[J]. arXiv preprint arXiv:1601.08059, 2016.

[9] Bodnari A, Deleger L, Lavergne T, et al. A Supervised Named-Entity Extraction System for Medical Text[C]//CLEF (Working Notes). 2013.

[10] Scheffer T, Decomain C, Wrobel S. Active hidden markov models for information extraction[C]//International Symposium on Intelligent Data Analysis. Springer, Berlin, Heidelberg, 2001: 309-318.

[11] Zhang H, Liu C, Yang C, et al. An improved scene text extraction method using conditional random field and optical character recognition[C]//Document Analysis and Recognition (ICDAR), 2011 International Conference on. IEEE, 2011: 708-712.

[12] Garcia V, Debreuve E, Nielsen F, et al. K-nearest neighbor search: Fast GPU-based implementations and application to high-dimensional feature matching[C]//Image Processing (ICIP), 2010 17th IEEE International Conference on. IEEE, 2010: 3757-3760.

[13] Kambhatla N. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations[C]//Proceedings of the ACL 2004 on Interactive poster and demonstration sessions. Association for Computational Linguistics, 2004: 22.

[14] Zhang K, Xu H, Tang J, et al. Keyword extraction using support vector machine[J]. Advances in Web-Age Information Management, 2006: 85-96.

[15] Rinser D, Lange D, Naumann F. Cross-lingual entity matching and infobox alignment in Wikipedia[J]. Information Systems, 2013, 38(6): 887-907.

[16] Khadilkar V, Kantarcioglu M, Thuraisingham B, et al. Jena-HBase: A distributed, scalable and efficient RDF triple store[C]//Proceedings of the 2012th International Conference on Posters and Demonstrations Track-Volume 914. CEUR-WS.org, 2012: 85-88.

[17] Cudr-Mauroux P, Enchev I, Fundatureanu S, et al. NoSQL databases for RDF: an empirical evaluation[C]//International Semantic Web Conference. Springer, Berlin, Heidelberg, 2013: 310-325.

[18] Sun K, Liu Y, Guo Z, et al. Visualization for Knowledge Graph Based on Education Data[J]. International Journal of Software and Informatics, 2016, 10(3).