Semi-automated construction of a knowledge graph with template

Tangzhao Wei, Ying Jiang, Yue Wang, Yuming Luo, Weibin Lin and Zihao Chen

School of Beijing Normal University, Zhuhai 519000, China

*Corresponding author e-mail: jpz6311whu@bnuz.edu.cn, a13631236487m0@sian.cn, b310181909@qq.com, c458398582@qq.com, d549288160@qq.com, echenzima0@163.com

Abstract. The Internet has massive data and a large number of relationships describing various entities and entities. HTML pages with semi-structured and structured data are important data sources for building knowledge graph. This paper proposes to construct a knowledge graph based on a template semi-automatically and construct a lure template by filling the template as a data source to realize the domain knowledge graph by extracting the triple data in the multi-data source through the attracting template. And build an application platform to provide a domain knowledge graph, reduce human participation to achieve efficient construction of knowledge graph.

1. Introduction
Knowledge graph is a large-scale semantic network that connects knowledge or concepts through semantic relations and shows its development process and structural relationship. Knowledge graph can provide practical and valuable reference for disciplinary research and is widely used in securities, banking, finance, taxation, court, logistics, and other fields. The main task of knowledge graph construction is to integrate structured, semi-structured, and unstructured data, and construct target knowledge graph by mining entities and analyzing relationships between entities. There are three main ways to construct knowledge graph: manual build, semi-automated build, and fully automated build. The manually constructed knowledge graph has the highest accuracy, but the labor and time cost is the highest among the three construction schemes. Although the fully automated construction is the lowest in labor cost compared with the other two construction methods, its technical requirements are high and currently difficult to implement in most fields. The semi-automated construction method is between the manual construction and the fully automated construction, and the current application field more extensive, and the technical requirements are relatively low, and the automation is relatively low. In this context, the semi-automated construction of knowledge graph has significant research value. This paper describes a semi-automated construction process for the website knowledge graph and constructs a refillable template to construct domain knowledge graph by filling templates. By analyzing the system output data, it prove that this construction method can efficiently build domain knowledge graph.
2. Research status at home and abroad
The concept of knowledge graph was first proposed by Google in 2012, initially to enhance the function of search engines and improve the quality of search results. At present, there are many knowledge graphs at home and abroad. The more famous knowledge graphs in foreign countries include Google's Knowledge Vault [1], Apple's Wolfram AlphaAno [2], and Microsoft's Probase [3]. The main knowledge graphs in China are Baidu's intimate [4], Sogou's knowledge cube [5].

In the construction of the knowledge graph, the ontology construction of a knowledge graph is especially important, which determines the data pattern of the knowledge graph, so the results obtained from ontology construction play a decisive role in the overall construction of the knowledge graph. At present, most of the research shows that the construction of ontology is mainly divided into three types of construction methods: manual construction, automatic construction, and semi-automatic construction.

The artificial construction method, as the name implies, is constructed by a large number of humans to work, and the artificial in this construction usually refers to a large number of sub-domain experts to cooperate. At present, the two most representative human-edited ontology at home and abroad are WordNet [6] and Cyc [7]. The knowledge graph of artificially constructed ontology can ensure the quality of ontology construction well, but at the same time, its disadvantages are apparent. It takes a lot of human resources and time, and it needs to define each attribute relationship from scratch and determine the constraints between them. Even at the same time, to facilitate the artificial construction of the knowledge graph ontology, many ontology editing tools and construction methods have appeared so far, such as Mike Uschold & King skeleton method, TOVE ontology development method, CACTUS method and Methontology method [8]. Web Onto [9], OntoEit [10] and Protégé [11] are more famous for tools; these tools help builders build with artificial more efficiently, saving users time and effort. However, the artificially constructed ontology is also facing issue. It is still challenging to keep up with the speed of changes in Internet information, and the information of the knowledge graph gradually lags. In today's Internet-scale, the amount of ontology content edited by domain experts is difficult to meet people's needs. At the same time, the talent resources of domain experts are gradually lacking. The ontology of leader workers is limited by conditions and will fall behind in time and quality. Manually built tools are not well integrated.

The way of automatic construction, usually called ontology learning [12], the main advantage of automatic construction is to use knowledge acquisition technology, machine learning technology to automatically obtain the information of the ontology from various existing data. Compared with the artificial construction method, this method dramatically improves the efficiency of knowledge graph construction and reduces the time cost and human resource cost of ontology construction. But at the same time, in the process of automatic construction, it is tough to achieve fully automated knowledge acquisition, the problem of information accuracy, and the complexity of building the model. At present, the methods of automatic construction of knowledge graph at home and abroad mainly include Jean method, H. Waste method, TextOnt method, domestic Yang Zhengku method, Wang Lei et al.

The semi-automatic build method is a kind of construction method between manual build an automatic build. It focuses on the technology of automated build, and also requires the builder to participate in the guidance. Such a build process is done in a semi-automatic way. Because of the shortcomings of manual construction and the difficulty of achieving fully automated construction, the semi-automatic construction method has attracted more and more scholars' attention and research. At present, some relatively good results can be seen; the semi-automatic construction of ontology is beginning to mature and deep. For example, the semi-automated construction research [14] of the Chinese medical knowledge graph, using the EARES system to obtain the entity attribute relationship from the medical field website, store the data in the Neo4j graph database, and then use the graph database to display. The research results propose a semi-automatic extraction system for entity attribute separation, which acquires the attribute relationship of the entity and realizes the process of semi-automatic construction of ontology.

In this context, this paper proposes a semi-automated solution for building ontology, which extracts the associated data on the Internet through the template after filling. In the process of extraction, it is
adaptive to the depth and breadth of the network, and Build associated data based on the template identifying the corresponding entity. Therefore, the domain knowledge graph is constructed efficiently.

3. The Design and Construction of the System

As shown in FIG.1, the process architecture of the knowledge grape semi-automatic construction system is constructed under the guidance of humans. At the beginning of the construction, the builder fills the template, which mainly fills the target website information and the entity extraction rule group and the configuration information of the extractor. After the system successfully verifies the template information, the system persists to the database to provide subsequent step calls. Wherein the verification content is mainly for whether the extraction rule is reasonable and valid and whether the extractor configuration information is verified within the scope. After the system successfully verifies the template information, the system persists to the database to provide subsequent step calls. The verification content is mainly for whether the extraction rule is reasonable and valid and whether the extractor configuration information is verified within the scope. The acquired knowledge data is deduplicated and stored in a local database, enabling us to process the data further. The local knowledge data is accurately modelled using a script and then imported into the Allegro Graph database. Finally, the user is provided with a search and relationship mining operation through a visual interface.

![Figure 1. Construction process architecture diagram](image-url)
3.1. Obtaining Entities from Data Sources

![Figure 2. Code of dynamic web page Downloader](image)

A large number of pages are processed during the data source acquisition entity. This paper divides the pages to be processed into three types: list page, detail page, and irrelevant page. The list page is the page containing the URL of the detail page; the detail page is the page containing the entity data; the irrelevant page refers to the page not in the range of the template processing page, including Ad page and repeat page. Web page URL has unique characteristics. And web pages of the same type also have similar URL formats. This feature can be used to distinguish different types of web pages. This paper uses URL regular expressions to distinguish between list pages and detail pages. For example, use the "http://s.askci.com/stock/summary/[^s]*" regular expression to match "http://s.askci.com/stock/summary/000001/" and "http://s.askci.com/stock/summary/000002/". These details page URLs are used to extract knowledge data. The extractor performs different operations on different pages. It performs a URL extraction operation on the list page to get detailed pages and other list pages. The detail page is an operation of extracting an entity based on a triplet in the template information and constructing the initial triplet data by the attribute. The irrelevant page skips the operation and does not process the page.

3.2. Design of the extraction rule of triples

In a data source, the data is usually presented as a detail page. For example, in the details column on the left side of FIG.3 "Ping An Bank", there is a URL of the details page of "company profile", "company executives", "share holding company" and other data. As shown in FIG.3, the details page in a data source, the target triples extracted in the collection domain are <Ping An Bank, Territory, Guangdong Province>, <Ping An Bank, chairman, Xie Yonglin>, <Ping An Bank, Secretary of the Board, Zhou Qiang>, <Ping An Bank, legal representative, Xie Yonglin> and so on.
Given the difficulty in determining the number of triples in the detail page, this paper proposes a method for dynamically constructing the triplet extraction rules. In the stage of filling the template, the filler dynamically adds the extraction rule group and establishes an equal extraction group according to the actual number of triples on the detail page. One extraction group contains a <Subject, Predicate, and Object>. The implementation of dynamically added actions is based on a JavaScript scripting language. Each data node in the Dom tree of an HTML page can be represented by a root path uniquely tagging an XPath path to it. Such a markup method is also standard in the same template page. For example, to analyze the entities and attributes in the details page above. The following table shows some of the triplet XPath paths:

| Subject         | Predicate          | Object                      |
|-----------------|--------------------|-----------------------------|
| /html/body/div[2]/div[3]/div[2]/div[2]/table/tbody/tr[1]/td[2]/text() | /html/body/div[2]/div[3]/div[2]/div[2]/table/tbody/tr[2]/td[1]/text() | /html/body/div[2]/div[3]/div[2]/div[2]/table/tbody/tr[2]/td[2]/text() |
| /html/body/div[2]/div[3]/div[2]/div[2]/table/tbody/tr[4]/td[1]/text() | /html/body/div[2]/div[3]/div[2]/div[2]/table/tbody/tr[4]/td[2]/text() | /html/body/div[2]/div[3]/div[2]/div[2]/table/tbody/tr[4]/td[3]/text() |
| /html/body/div[2]/div[3]/div[2]/div[2]/table/tbody/tr[12]/td[1]/text() | /html/body/div[2]/div[3]/div[2]/div[2]/table/tbody/tr[12]/td[2]/text() | /html/body/div[2]/div[3]/div[2]/div[2]/table/tbody/tr[12]/td[3]/text() |
3.3. Template development

The knowledge data extractor extracts the entities on the page. Read the triplet configuration information of the template and extract it to generate a ternary set. They are stored in the local database in the later step.

Designing a reusable fill template is the core of this paper. This paper uses the template driver configuration to guide the extractor to the entity extraction of the data source. The fill template contains data source description information, triple extraction rules, and extractor configuration. The extractor configuration information system has been populated with default attributes, and the default attributes can be modified according to actual needs. As shown in FIG.4, Area.1 is used to detect the depth and process the list page. It needs to fill in the URL regularity of the list page and the XPath of the grab area. This paper sets the support processing details page depth to 1 to 4 layers. The Area.2 is the use of filled with the regularity and triplet extraction rules of the judgment detail page URL, and triples can be added as needed. The Area.3 is used to modify the default configuration information of the decimator.

![Figure 4. template fill interface](image-url)
4. Knowledge Graph Application System

4.1. The semi-automatic construction system of the knowledge Graph

This Building system is based on Springboot and Thymeleaf. Springboot achieves the logic of the back end, and Thyme leaf achieve the interactive function of the front end. As shown in FIG.5, Template Creation Class provides the ability to create the domain to be built; Extraction starter provides template driver to obtain data to produce triples and data; Retrieve provides the graph visualization of keyword being retrieval.

We establish two domain using this knowledge Graph: financial listed company information and Anhui tourist attractions. Financial listed company information creates four templates for obtaining triples, and the data source comes from the website of China Commercial Industry Research Institute. There were 128,944 triples ultimately Imported into the AllegroGrap database. Anhui tourist attractions create two templates for obtaining triples, and the data source comes from Ctrip, a tourism website. There were 6,116 triples imported into Allegro rap database.

4.2. The Retrieval System based on Knowledge Graph

The Retrieval System of Knowledge Graph is based on ECharts and SpringBoot. ECharts provides relationships diagrams for the visualization of data and interacting with the back end through Restful style data. We use ECharts obtained the background data through the retrieval key then renders these data to show the relationship between the nodes related to the retrieval node.

5. Summary

This paper elaborates a semi-automated construction method of knowledge graph for domain-oriented websites. The method is based on a highly reliable data source, filling the data source information into a fillable template, and then using the template to drive the knowledge data under the massive data of the Internet. Collecting, extracting triple data, and semi-automatically building knowledge graphs. Use the Java language to build a semi-automated construction system of knowledge graphs, and create knowledge graphs of financial listed companies and Anhui tourist attractions. The practical results show that the construction method has individual practicability under the massive data of the Internet.

Acknowledgments

This work is supported by a project from 2019 Guangdong College Students Science and Technology Innovation Cultivation Special Funds (Project No. pdjh2019b0583). It is supported by Science and Technology Plan Project of Guangdong Province (Project No.2017A040405004 and 2019A101002095), a grant from Provincial Key Platform and Major Scientific Research Project of Universities in Guangdong Province (Project No. 2017GXJK215), and a grant from National Statistics Science Research Project (Project No. 2019LZ30). It’s also supported by Guangzhou Social Science Planning Project (Project No. 2019GZY26) and Zhuhai Social Science Planning Project (Project NO. 2019YB051). The corresponding author is Ying Jiang (jpz6311whu@bnuz.edu.cn).
References

[1] Anonymous. Google's Knowledge Vault [J]. Computer Active, 2014 (432).
[2] Meng Liu. Research on Knowledge-Based Knowledge Search Engine Wolfram Alpha [J]. Library Science Research, 2010 (04): 58 - 61+64.
[3] W. Wu, H. Li, H. Wang, and K. Q. Zhu. Probase: a probabilistic taxonomy for text understanding. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, ACM New York, USA, 2012: 481 - 492.
[4] Baidu knows. https://baike.baidu.com/item/ Baidu Zhixin, Baidu Encyclopedia
[5] Zhicuo. https://baike.sogou.com/v66616234.htm, Sogou Encyclopedia
[6] C. Fellbaum, WordNet: an electronic lexical database, MIT Press, London, 1998.
[7] C. Matuszek, J. Cabrai, M. Witbrock, and J. DeOliveira: An introduction to the syntax and content of Cyc. In: Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, 2006: 44 - 49.
[8] Qiu Xin. Analysis of Artificial Ontology Construction Method [J]. China Water Transport (the second half of the month), 2009, 9 (03): 132 - 133.
[9] A. J. Duineveld, R. Stoter, M. R. Weiden, B. Kenepa, and V. R. Benjamins. Wonder tools A comparative study of ontological engineering tools. Int'l Journal of Human-Computer Studies, 2000, 52 (6): 1111 - 1133.
[10] Y. Sure, J. Angele, M. Erdrmann, S. Staab, R. Studer, and D. Wenke. OntoEdit: Collaborative ontology engineering for the semantic Web. In: Proc.of the ISWC 2002. 2002: 221 - 235.
[11] N. F. Noy, R. W. Fergerson, and M. A. Musen. The knowledge model of protege-2000: Combining interoperability and flexibility. In: Proc. of the EKAW 2000, 2000: 17 - 32.
[12] A. Maedche and S. Staab. Ontology Learning for the Semantic Web. IEEE Intelligent Systems.2001, 16 (2): 72 - 79.
[13] Xie Wei, Wang Panqing, Peng Cheng. Automatic construction method of ontology [J]. Electronic Design Engineering, 2015, 23 (15): 39 - 41.
[14] Zou Yuwei. Semi-automated construction of Chinese medical knowledge map [D]. Wuhan University of Science and Technology, 2016.