Postprint: Construction of a Question Answering Knowledge Graph Ontology Model Integrating Multi-level Data

Authors: Zhou Yi, Liu Zheng, Su Xiaoqing, Jin Ticheng

Date: 2023-04-01T15:51:24+00:00

Abstract

[Purpose/Significance] To address the issues of low accuracy rate, low resolution rate, and poor user satisfaction in question-answering pair-based intelligent question answering, this study constructs a knowledge graph ontology model and develops intelligent question answering based on knowledge graphs to resolve the problems faced by question-answering pair-based intelligent question answering systems.

[Method/Process] First, we analyze the current problems and their root causes in intelligent question answering, and propose a solution for constructing knowledge graphs to support intelligent question answering. Second, building upon existing ontology model construction methods, we propose a multi-round iterative method that integrates multi-level data. This method utilizes multi-level data sources including business data, user data, and dynamic data from business systems, with core steps comprising three iterative cycles: establishing a basic framework, refining the knowledge structure, and aligning the knowledge structure. Finally, taking the return and exchange domain as an example, we elaborate on the specific steps for ontology model construction, building a knowledge graph ontology model from scratch through incremental accumulation.

[Results/Conclusion] The knowledge graph with the return and exchange ontology model as its schema layer was deployed in an intelligent question-answering system for experimentation. The experimental results demonstrate that after the deployment of the return and exchange knowledge graph, the accuracy rate of intelligent question answering increased by 50%, and the resolution rate increased by 300%. Accuracy rate is defined as the ratio of correctly answered questions to the total number of answered questions, while resolution rate is defined as the ratio of answers that precisely solved user problems to the total number of answered questions. The ontology model construction method proposed in this paper organizes a complete and fine-grained domain knowledge

structure from scattered domain knowledge, enabling intelligent question answering to provide precise answers to users and effectively resolving the dilemma of question-answering pair-based intelligent question answering.

Full Text

Preamble

Ontology Model Construction for Question-Answering Knowledge Graphs Integrating Multi-Level Data

Zhou Yi[1], **Liu Zheng**[1,2], **Su Xiaoqing**[3], **Jin Ticheng**[3]

[1] National Science Library, Chinese Academy of Sciences, Beijing 100190
[2] Department of Library, Information and Archives Management, School of Economics and Management, University of Chinese Academy of Sciences, Beijing 100190
[3] Huawei Device Co., Ltd., Shenzhen 518129

Abstract: [Purpose/Significance] To address the low accuracy, resolution rate, and user satisfaction of intelligent question answering (QA) systems based on question-answer pairs, this study constructs a knowledge graph (KG) ontology model to enable KG-based intelligent QA, thereby solving the problems faced by QA pair-based systems. [Method/Process] First, we analyze current challenges in intelligent QA and their causes, proposing a solution that leverages knowledge graphs to support intelligent QA. Building on existing ontology construction methods, we propose a multi-round iterative approach that integrates multi-level data—including business data, user data, and dynamic business system data—as data sources. The core steps involve three iterative cycles: building a basic framework, improving knowledge structure, and aligning knowledge structure. Finally, using the return and exchange domain as an example, we elaborate the specific steps for ontology model construction, creating a KG ontology model from scratch through incremental 叠加. [Result/Conclusion] Deploying the return/exchange knowledge graph with this ontology model as the schema layer in an intelligent QA system yielded significant improvements: QA accuracy increased by 50% and resolution rate by 300%. Accuracy is defined as the ratio of correctly answered questions to total answered questions, while resolution rate refers to the proportion of answers that precisely solve user problems. The proposed ontology construction method systematically organizes complete, fine-grained domain knowledge structures from scattered domain knowledge, enabling intelligent QA to provide precise answers and effectively resolving the dilemmas of QA pair-based systems.

Keywords: knowledge graph; ontology model; precise question answering; multi-level data
Classification Number: G250
DOI: 10.13266/j.issn.0252-3116.2022.05.013

In recent years, many enterprises have gradually applied intelligent QA in customer service to improve efficiency and reduce costs. Intelligent QA can be categorized into three types: task-oriented, knowledge acquisition, and chatbot. This paper focuses on knowledge acquisition QA. From the perspective of data organization, there are three approaches: (1) QA pair-based systems that pre-establish a set of question-answer pairs and use keyword matching to find the best match; (2) KG-based QA that leverages structured knowledge in graphs for semantic understanding and precise answers; and (3) reading comprehension-based QA that extracts answer fragments from given documents without pre-existing knowledge extraction. Current enterprise customer service systems predominantly use QA pair-based approaches, which suffer from low problem resolution rates and poor user satisfaction due to limited richness in QA pairs and manual configuration of similar questions, necessitating transformation and breakthroughs. Reading comprehension-based QA remains in the research stage. KG-based QA supports semantic matching at multiple granularities and enables contextual session identification and reasoning between entities, making it an effective path to overcome the limitations of QA pair-based systems and achieve intelligent customer service transformation.

A knowledge graph consists of a schema layer and an instance layer. While open-domain KGs may have only an instance layer, vertical domain KGs require high-quality, accurate knowledge and a complete schema layer to abstract important concepts and relationships for inter-domain reuse and integration. Therefore, the ontology model is critical for domain KG construction. This paper focuses on structuring domain knowledge based on multi-level data to explore building a well-structured and comprehensively covered KG ontology model.

This research is supported by the National Key R&D Program of China "Advanced Manufacturing Distributed Technology Service Integration and Demonstration" (Project No.: 2019YFB1405100).

Authors: Zhou Yi, Assistant Librarian, Master's degree; Liu Zheng, Research Librarian, PhD, Corresponding author, E-mail: liuz@mail.las.ac.cn; Su Xiaoqing, Knowledge Management Expert, Master's degree; Jin Ticheng, Knowledge Management Engineer.
Received: 2021-07-06; **Revised:** 2021-11-05; **Pages:** 125-132; **Responsible Editor:** Wang Chuanqing

2. The Dilemma of QA Pair-Based Intelligent Question Answering

QA pair-based intelligent QA faces challenges in knowledge coverage, granularity, and quality.

2.1 Difficulty in Improving Knowledge Coverage

Numerous QA pairs exist in isolation without connections, leading to redundancy and knowledge gaps. On one hand, operators must continuously add new QA pairs and similar questions, creating heavy knowledge management burdens that make it impossible to cover all potential questions. On the other hand, without a complete and standardized domain knowledge architecture, even continuous addition of individual knowledge points cannot define the knowledge boundaries required to meet user needs.

2.2 Difficulty in Refining Knowledge Granularity

In practice, user questions vary in generality, making it difficult for QA pairs to maintain consistent granularity. For example, users might ask "What is the return policy?" or "Can I return a newly purchased product after seven days?" Setting QA pair granularity too fine requires excessive manual effort, while setting it too coarse yields overly general answers that fail to solve actual problems. For instance, when users ask for nearby store addresses, providing only a method to query store addresses, or when asking about warranty coverage for a phone battery, receiving the entire warranty policy instead of specific information.

2.3 Difficulty in Ensuring Knowledge Quality

QA pair-based systems require manually writing numerous QA pairs and similar questions, which cannot be finely controlled for quality. Understanding biases, staff turnover, and other factors can lead to redundant, inconsistent, or even contradictory QA pairs. When domain knowledge updates, synchronizing related QA pairs becomes difficult, creating maintenance challenges.

---

3. Solutions to the QA Pair-Based Intelligent QA Dilemma

To address these issues, this paper explores constructing a well-structured and comprehensively covered KG ontology model based on multi-level domain data. The goal is to transition from QA pair-based to KG-based intelligent QA, improving knowledge coverage and quality while supporting multi-granularity user questions.

3.1 Using Ontology Models to Depict the Full Domain Knowledge Structure

A knowledge graph is a technical approach that uses graph models to describe knowledge and model relationships between entities. Its standardized description of domain knowledge maintains consistent knowledge granularity, supports synchronized updates and semantic reasoning across multiple application scenarios, and thereby improves knowledge quality while reducing maintenance difficulty.

The schema layer serves as the conceptual model and logical foundation of a KG, constraining the instance layer. In the schema layer, nodes represent concepts and edges represent conceptual relationships. The instance layer stores instance data of classes or relationships from the schema layer, with facts stored as "entity-relationship-entity" or "entity-attribute-value" RDF graphs or property graphs. Building an ontology model as the schema layer facilitates the integration and reuse of knowledge across business sub-domains while avoiding redundancy, forming a reasonably structured and comprehensively covered knowledge architecture.

3.2 Integrating Multi-Level Data to Optimize Coverage and Granularity

With digitalization deepening, domain knowledge exists in more diverse forms, providing richer potential data sources for ontology model construction. This paper explores methods for integrating multi-level data to accurately depict the full landscape of domain knowledge.

The key to building a KG ontology model lies in determining the horizontal boundaries and vertical granularity of domain knowledge. As early as 1995, M. Grüninger et al. proposed using competency questions (CQs) to define ontology domain and scope. Subsequently, various ontology construction methods have used CQs to determine scope, such as: What domain will the ontology cover? What will we use the ontology for? What types of questions should the ontology information answer? Who will use and maintain the ontology? While CQs can help define the general domain and scope during requirement analysis, their limited number and manual setting by few experts cannot determine domain boundaries and granularity in detail. Recent work by D. Wiśniewski et al. proposed discovering more CQ patterns through lexical and syntactic analysis to support ontology construction, but this still cannot meet the needs of KG ontology construction for intelligent QA.

In terms of data sources, previous research on enterprise domain KG construction has predominantly used static knowledge such as encyclopedia data, basic enterprise information, news, listing data, and business documents. While important, such static knowledge from business operators often diverges from real user needs and is insufficient for building comprehensively covered, appropriately granular KG ontology models.

With diversified domain data, using multi-level data for KG ontology model construction becomes feasible. User-provided questions and search queries constitute important domain knowledge components. Incorporating user data enables accurate delineation of domain knowledge boundaries based on objective data. Furthermore, since users typically interact with business systems when obtaining services, integrating dynamic data from business systems can locate users' specific situations in real-time. Therefore, constructing KG ontology models requires integrating data from both business operators and users, combining static and dynamic knowledge.

---

4. Ontology Model Construction Integrating Multi-Level Data

The multi-level data integration method follows the UPONLite approach, employing a lightweight, multi-round iterative ontology construction process that uses business operation data, user data, and business system dynamic data as sources.

The construction process is divided into preparation and construction phases. The preparation phase defines the domain scope and boundaries, collecting domain knowledge guided by application objectives. The construction phase comprises multiple iterative cycles, each using different levels of domain knowledge as input to incrementally complete the ontology model. The first cycle uses concise, understandable FAQ data to build a basic framework. The second cycle integrates knowledge documents and user data to improve the knowledge structure. The third cycle integrates business system data for further refinement, supporting dynamic interaction between the KG and business systems. Each cycle includes the same detailed steps: building a domain terminology table, defining classes and class hierarchies, defining properties, and representing the ontology model—all completable in spreadsheets.

As shown in Figure 1 [Figure 1: see original paper], the process includes: (1) Domain and scope definition: defining the KG's domain and scope, clarifying reasons for creation, intended use, and user types. (2) Data preparation: investigating and collecting all knowledge relevant to ontology construction, including reusable ontologies and other knowledge organization resources. (3) Basic framework construction: the first cycle uses FAQ as starting data to build a basic framework through four steps—building a domain terminology table by identifying terms from FAQs; defining classes and hierarchies by identifying independent objects and their relationships; defining properties describing class internal structures; and ontology representation documenting the complete structure. (4) Knowledge structure improvement: the second cycle integrates knowledge documents and user data to supplement classes and properties. (5) Knowledge structure alignment: the third cycle aligns business system data structures, adding classes and properties.

The three cycles of basic framework construction, knowledge structure improvement, and alignment may involve multiple adjustments until an application-compliant ontology model is built. We now elaborate the construction method using the return/exchange service domain.

4.1 Domain and Scope Definition

Ontology construction should always target application requirements without needing to include all domain information. Before construction, the KG's basic domain and scope must be determined, expressed in the form of a KG requirements specification.

QA pair-based intelligent QA performs poorly in the return/exchange domain. User feedback shows many irrelevant answers, and even when matching preset QA pairs, users remain dissatisfied. To address these issues, this study plans to construct a return/exchange domain KG to improve accuracy and resolution rates through KG-based intelligent QA.

4.2 Data Preparation

The principle is to reuse existing ontologies or patterns to avoid "reinventing the wheel." This research investigates and collects all knowledge relevant to domain KG construction, including existing ontologies, vocabularies, terminology lists, classification schemes, and multi-level domain data such as FAQs, knowledge documents, real user questions, and business system data structures.

In this case study, the OPAL (Object, Process, Actor modeling Language) ontology design model supports business ontology construction. As a knowledge organization resource, it is collected for reuse. Domain knowledge includes return/exchange-related FAQs, knowledge documents, real user questions, and business system data structures.

4.3 Building the Basic Framework with FAQ Data

Using FAQ data to build the basic framework involves constructing a domain terminology table, identifying terms as classes or properties, and defining them in detail.

4.3.1 Building the Domain Terminology Table Extract domain terms from FAQs—these may be objects to describe, object properties, or property values. When identifying terms, explore horizontally to find ontology scope boundaries and vertically to consider appropriate detail levels or granularity.

After obtaining terms, add semantic descriptions to build the terminology table, including synonyms, textual descriptions, and term categories (distinguishing classes from properties). This preliminary organization prepares for subsequent class and property definition.

From return/exchange FAQs, we extract terms like "packaging box," "audit," "warranty card," and "invoice." The OPAL model classifies classes into three types: object, process, and actor, with property types including atomic property (AP), reference property (RP), and complex property (CP). Return/exchange business fits the OPAL model, so term categories follow these distinctions.

4.3.2 Defining Classes and Class Hierarchies The terminology table cannot represent rich term structures. This step classifies independent objects based on specialization relationships to define class hierarchies through isA and partOf relationships.

Approaches include top-down, bottom-up, or combined methods. This study uses a combined approach: first defining the most common and certain individual classes, then specializing and generalizing them.

The return/exchange domain applies OPAL's high-level concept templates—object, process, and actor—to build an abstract framework. Under the root "Thing" class, three branches are created as first-level classes: "Object," "Actor," and "Process," representing abstract roles in return/exchange business, with subclasses added beneath each branch.

4.3.3 Defining Properties Properties describe classes. After defining class hierarchies, properties are extracted from the terminology table and connected to their classes. For example, the "Product" class has properties like "price" and "model."

Properties have facets including type, definition, aliases, value type, cardinality constraints, and single/multi-value characteristics. Return/exchange ontology properties follow OPAL's definitions: atomic properties with literal values (e.g., "definition" as a string), reference properties with instances as values, and complex properties with internal structure. We extend complex properties with two structures: (1) Key-Value (KV) structure where single conditions determine property values, such as "operation instructions" for the "Application" class varying by entry point; (2) Compound Value Type (CVT) structure where multiple condition combinations determine values, such as determining product return eligibility based on return reason, product, and validity period.

4.3.4 Ontology Model Representation The results from the first three steps are consolidated into a complete ontology model record table, including class tables documenting class hierarchies and property tables documenting property definitions and their associated classes.

4.4 Improving Knowledge Structure with Documents and User Data

The basic framework derived from FAQ needs refinement by integrating other knowledge layers.

4.4.1 Integrating Knowledge Documents Knowledge documents contain more comprehensive domain knowledge that may overlap with FAQs. This step abstracts classes and properties not covered in FAQs to improve the knowledge structure. After basic framework construction, ontology builders have accumulated domain knowledge, making it feasible to identify missing classes and properties from lengthy documents using manual or automated methods.

Return/exchange process and policy documents are selected to identify classes and properties absent from the FAQ-based framework for addition.

4.4.2 Integrating User Data Real user questions directly reflect user needs, aligning ontology model scope and granularity with actual applications to ensure coverage and refine granularity. User questions from intelligent QA or other channels often have large volumes. Machine learning methods like the Bert model can classify these questions and mark representative data in each category. Human reviewers then examine representative data to identify new classes and properties for addition to the basic framework.

In the return/exchange domain, historical data is classified using the Bert model, with representative data marked in each category. Questions about canceling or modifying applications reveal the need to add "Cancel Application" and "Modify Application" subclasses under the "Application" class, as shown in Table 1 .

4.5 Aligning Knowledge Structure with Business System Data

Dynamic data in business systems records users' latest and most detailed information. Real-time interaction between intelligent QA and business systems enables responses based on current information, providing granular answers or solutions. Therefore, integrating business system data is necessary.

During alignment, first identify whether terms in data structures need to be added as new classes or properties. Then define and integrate them with existing ontology elements. Mapping between ontology models and business system data structures involves four scenarios: (1) one-to-one mapping where identical fields exist in both systems, requiring simple mapping like order numbers; (2) many-to-one mapping where one business system field contains multiple ontology properties, requiring value splitting, such as a "return status" field corresponding to multiple process states; (3) one-to-many mapping where multiple system fields correspond to one ontology property, requiring field combination, such as "year, month, day" mapping to a single "date" property; (4) fields not yet covered in the ontology, requiring new class or property additions.

Relevant systems include enterprise return/exchange systems and e-commerce platforms. Aligning the ontology model with their data structures adds classes like "Order" and "Return Application Form" to support real-time data access during user queries, enabling answers based on specific user contexts. For example, when users ask "What stage is my return at?", the system can query and return the current status from business systems.

4.6 Return/Exchange Knowledge Graph Ontology Model

After completing the multi-level data integration cycles, the return/exchange ontology model is obtained as shown in Figure 2 [Figure 2: see original paper].

5. Evaluation of Return/Exchange Ontology Model Application

Using the return/exchange ontology model as the schema layer, we constructed the instance layer to form a complete KG. To evaluate its effectiveness in addressing existing dilemmas, we conducted statistical analysis and experimental evaluation focusing on knowledge coverage, granularity, and QA accuracy/resolution rates.

Knowledge Coverage: Before KG construction, the return/exchange domain had only about 50 QA pairs. After KG construction, the ontology had nearly 250 properties covering thousands of questions through property values or combinations, dramatically improving coverage.

Granularity: The KG flexibly supports answers at different granularities. By integrating user questions and business system data, the ontology's classes and properties align with user question granularity. KG-based QA provides specific answers tailored to user contexts rather than generic responses. The multi-level structure also enables answering both fine-grained and coarse-grained questions. For instance, the "Return" class has subclasses "Online Return" and "Offline Return"—when users ask about returning physical store purchases, the system targets "Offline Return," while general return questions map to the parent "Return" class.

Accuracy and Resolution Rate: After deploying the return/exchange KG in the intelligent QA system, we randomly sampled over 400 original human customer service questions for manual evaluation. Results showed accuracy increased by 50% and resolution rate by 300%, as detailed in Table 2 .

Table 2. Experimental Results of Knowledge Graph-Based Intelligent QA

Metric	Before KG Application	After KG Application
Accuracy	[value]	+50%
Resolution Rate	[value]	+300%

Statistical analysis demonstrates that integrating multi-level data—including user data and business system dynamic data—significantly improves KG-based QA capabilities. User data alignment ensures ontology structure granularity matches real user needs, making coverage nearly complete. Business system alignment enables real-time dynamic data access to locate users' specific situations, providing precise answers rather than generic clues. This multi-level integration substantially enhances QA accuracy and resolution rates.

This study proposes a multi-level data integration approach for KG ontology construction to transition from QA pair-based to KG-based intelligent QA. The return/exchange case study shows that the constructed ontology model effectively ensures knowledge coverage and granularity, dramatically improving QA accuracy and resolution rates while enabling capability upgrades. Beyond KG construction, the multi-level data integration ontology can also identify gaps in enterprise-provided knowledge, optimize business data, and support domain knowledge classification and intent classification systems.

This research focuses on enterprise business domain ontology construction. While the multi-level data types are analyzed based on enterprise contexts, application to other domain types requires further validation in future work.

References

[1] SINGHAL A. Official Google blog: introducing the knowledge graph: things, not strings [EB/OL]. [2021-07-02]. https://www.blog.google/products/search/introducing-knowledge-graph-things-not/.

[2] IRENE P. Knowledge graphs vs property graphs: a brief overview and comparison [EB/OL]. [2021-05-21]. https://www.topquadrant.com/wp-content/uploads/2020/08/Knowledge-Graphs-vs.-Property-Graphs-A-Brief-Overview-and-Comparison.pdf.

[3] GRUBER T R. A translation approach to portable ontology specifications [J]. Knowledge acquisition, 1993, 5(2): 199-220.

[4] GRÜNINGER M, FOX M S. Methodology for the design and evaluation of ontologies [EB/OL]. [2021-12-21]. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.44.8723&rep=rep1

[5] USCHOLD M, KING M. Towards a methodology for building ontologies [M]. Edinburgh: University of Edinburgh, 1995.

[6] FERNÁNDEZ-LÓPEZ M, GÓMEZ-PÉREZ A, JURISTO N. Methontology: from ontological art towards ontological engineering [M]// Engineering workshop on ontological engineering. 1997 (Mar.): 33-40.

[7] SURE Y, STAAB S, STUDER R. On-to-knowledge methodology (OTKM) [M]// Handbook on ontologies. Berlin: Springer, 2004: 117-132.

[8] SUÁREZ-FIGUEROA M C, GÓMEZ-PÉREZ A, FERNÁNDEZ-LÓPEZ M. The NeOn methodology for ontology engineering [M]// Ontology engineering in a networked world. Berlin: Springer, 2012: 9-34.

[9] DENICOLA A, MISSIKOFF M, NAVIGLI R, et al. A software engineering approach to ontology building [J]. Information systems, 2009, 34(2): 258-275.

[10] NOY N F, MCGUINNESS D L. Ontology development 101: a guide to creating your first ontology [EB/OL]. [2021-12-21]. https://corais.org/sites/default/files/ontology_{{{development}

[11] GANGEMI A, PRESUTTI V. Ontology design patterns [M]// Handbook on ontologies. Berlin: Springer, 2009: 221-243.

[12] PRESUTTI V, DAGA E, GANGEMI A, et al. eXtreme design with content ontology design patterns [C]// Proceedings of workshop on ontology patterns. Washington D.C.: CEUR-WS.Org., 2009: 83-97.

[13] Zhu Yanxin, Xu Wei, Wang Xia, et al. Construction of evidence-based medicine literature ontology based on FAIR principles—taking asthma drug treatment literature ontology as an example [J/OL]. Information Studies: Theory & Application: 1-12 [2021-09-30]. http://kns.cnki.net/kcms/detail/11.1762.G3.20210819.1201.002.html.

[14] Jia Junzhi. Thoughts on converting the Chinese Thesaurus into an ontology [J]. Journal of Library Science in China, 2007(4): 41-44.

[15] Duan Ruilong, Song Wen. Review of thesaurus-to-ontology conversion methods [J]. Journal of Intelligence, 2012, 31(7): 66-71.

[16] ZHEN Z, TANG A, SHEN J Y, et al. Thesaurus-based approach for building domain ontology with a case study of military aircraft prototype ontology construction [J]. Journal of Southeast University (English Edition), 2006(3): 353-356.

[17] WIŚNIEWSKI D, POTONIEC J, ŁAWKYNOWICZ A, et al. Analysis of ontology competency questions and their formalizations in SPARQL-OWL [J]. Journal of Web Semantics, 2020(29): 1-20.

[18] Yang Bo, Liao Yiming. Research on enterprise dynamic risk-oriented knowledge graph construction and application [J]. Modern Information, 2021, 41(3): 110-120.

[19] Zhang Su, Xu Hui. Research on enterprise knowledge service model construction based on knowledge graphs [J]. Information Science, 2020, 38(8): 68-73.

[20] DE NICOLA A, MISSIKOFF M. A lightweight methodology for rapid ontology engineering [J]. Communications of the ACM, 2016, 59(3): 79-86.

[21] SUÁREZ-FIGUEROA M C, GÓMEZ-PÉREZ A. Ontology requirement specification [M]// Ontology engineering in a networked world. Berlin: Springer, 2012: 93-106.

[22] D'ANTONIO F, MISSIKOFF M, TAGLINO F. Formalizing the OPAL business ontology design patterns with OWL [M]// Enterprise interoperability II. London: Springer, 2007: 345-356.

Author Contributions

Zhou Yi: Framework design, model construction, paper writing and revision.
Liu Zheng: Provided topic selection and framework design, model construction, paper revision.
Su Xiaoqing: Participated in model construction and review, paper revision.

Jin Ticheng: Responsible for data collection, participated in model construction, conducted application experiments.

Note: Figure translations are in progress. See original paper for figures.