Tạp chí Khoa học và Công nghệ, Số 50, 2021

AN APPROACH TO EXTENDING QUERY SENTENCE FOR SEMANTIC ORIENTED SEARCH ON KNOWLEDGE GRAPH

TẠ DUY CÔNG CHIẾN
Khoa Công Nghệ Thông Tin,Trường Đại học Công nghiệp thành phố Hồ Chí Minh,
taduycongchien@iuh.edu.vn

Abstract. There are many applications related to semantic web, information retrieval, information extraction, and question answering applying ontologies in recent years. To avoid the conceptual and terminological confusion, an ontology is built as a taxonomy ontology which identifies and distinguishes concepts as well as terminology. It accomplishes this by specifying a set of generic concepts that characterizes the domain as well as their definitions and interrelationships. There are some methods to represent ontologies, such as Resource Description Framework (RDF), Web Ontology Language (OWL), databases etc. depending on the characteristic of data. RDF, OWL usually is used the cases when data structure is objects which the relationship among the objects is simple. But if the relationship among the objects is more complex, using databases for storing ontologies is an approach to be better. However, using relational databases do not sufficiently support the semantic orientated search by Structured Query Language (SQL) and the searching speed is slow. Therefore, this paper introduces an approach to extending query sentences for semantic oriented search on knowledge graph.

Keywords. Knowledge graph; Semantic search; Extending query.

1 INTRODUCTION

Applying databases for Semantic approach to keyword search has become an active field of research in recent years. Depending on different applications and the structure of databases, semantic orientation search over relational databases applies in many ways. There is a lot of research relevant to this field. Atkinson et al [1] proposed a new approach to automatic metadata extraction and semantic indexing for educational purposes is proposed to identify learning objects that may assist educators to prepare pedagogical materials from the Web. M. Saleh [2] proposed an approach for semantic query in traditional relational database based on ontological layer. Firstly, this technique starts by wrapping the relational database with a schema ontology extracted from the relational database schema and adapted with global domain ontology. Secondly, the user issues semantic query which is mapped using the schema ontology into SQL statements to the relational database repository. Finally, the results were mapped into semantic knowledge and appear to the user. In general, there are many researches relevant to semantic orientation extraction and semantic search over relational databases. However, the most of above research focus on relational databases and therefore the searching speed is slow if data is enough big. In this paper, we introduce an adaptable approach for searching semantic-based keywords on Neo4J - graph database. This approach can be applied to any simple or complex query and any graph databases.

Our key contributions are as follows: (i) we propose a novel method for obtaining the keyword list from input queries by the Stanford Lexical Dependency Parser (SDLP) considering syntactic grammar of sentences; (ii) the extending queries for semantic search over graph database is generated automatically considering the taxonomy of a domain specific ontology; (iii) the graph database in this case only focuses on Computer Domain with over 300,000 items, which covers 170 distinct categories.

The rest of this paper is organized as follows: section 2 - related works; section 3 – approach to extending query sentence for semantic oriented search based on the knowledge graph; section 4 - experimental results and discussion; section 5 - conclusions and future works.

2 RELATED WORKS

As outline from Bergamaschi et al [3], they showcased QUEST (QUEry generator for STructured sources), a search engine for relational databases that combines semantic and machine learning techniques for transforming keyword queries into meaningful SQL queries. The search engine relies on two approaches:

the forward, providing mappings of keywords in database terms (names of tables and attributes, and domains of attributes), and the backward, computing the paths joining the data structures identified in the forward step. QUEST is able to compute high quality results even with little training data and/or with hidden data sources such as those found in the deep Web. Elsayed et al [4] provided an easy way for casual users to access relational databases using a set of keywords. Their system extends the existing schema-free Keyword Search over relational database systems with semantic match features. This system exploits domain ontology to progressively return related terms that can be used to retrieve more relevant answers to end user. In the Oracle database with 12c release [5] it allowed users to store semantic data and ontologies, to query semantic data and to perform ontology-assisted query of enterprise relational data, and to use supplied or user-defined inference to expand the power of querying on semantic data. Bergamaschi et al [6] proposed a metadata approach of keyword search over relational databases. Their approach offers significant improvements in the identification of the semantically meaningful SQL queries that describe the intended keyword query semantics. They extend and exploit the Hungarian (a.k.a., Munkres) algorithm [7] to develop a technique for the systematic computation of the contextual weights that leads into the generation and ranking of the different interpretations of a keyword query in terms of SQL. They considered the order of keywords and the correlated keywords in the user's queries. Hannah et al [8] provided a comprehensive overview of the broad area of semantic search on text and knowledge bases. They classify their work according to two dimensions: the type of data test, knowledge bases, combination of these and the kind of search keyword, structured and natural language. Son T.C et al [9] provided a method, called QSQN-WF, for evaluating queries to Dialog databases under the well-founded semantics. In this paper, we propose an approach to extending queries for semantic oriented search based on keywords, which obtain from user's queries based on semantics and syntactics of keywords relevant to the computing domain on graph database. We also provide an extensive experimental evaluation. In addition, the graph database in this case represents to our computing domain ontology and we use Neo4J for demo purpose only.

3 AN APPROACH FOR EXTENDING QUERY SENTENCES TO SEARCH ON GRAPH DATABASE

3.1 Overview of the Computing Domain Ontology (CDO)

Ontology is a formal and explicit specification of a shared conceptualization of a domain of interest. Their classes, relationships, constraints, and axioms define a common vocabulary to share knowledge. Conceptualization refers to an abstract model of some phenomenon in the world. Explicit, means that the type of concepts used, and the limitations of their use are explicitly defined. Formal, refers to the fact that the ontology should be machine-readable. Shared, reflects the notion that ontology captures consensual knowledge, that is, it is not private to some individual but accepted by a group.

Formally, an ontology can be defined as the tuple [10]:

$$O = (C, I, S, N, H, Y, B, R)$$

Where,

C, is set of classes, i.e., concepts represent categories of computer domain (for example, "Artificial Intelligent, hardware devices, NLP" ∈ C)

I is set of instances belong to categories. Set I consists of vocabulary of computer (for example, "robotic, Random Access Memory "∈ I)

$S = N^S \cup H^H \cup Y^H$ is the set of synonyms, hyponyms and hypernyms of instances of set I.

N = NS is set of synonyms of instances of set I.

H = HH is set of hyponyms of instances of set I.

Y = YH is set of hypernyms of instances of set I. (e.g., "ADT", "data structure", "ADT is a kind of data structure that is defined by programmer" are synonymous, hyponymous and hypernymous of "Abstract data type")

B = {belong to (i, c) | i ∈ I, c ∈ C} is set of semantic relationships between concepts of set C and instances of set I and are denoted by {belong to (i, c) | i ∈ I, c ∈ C} mean that i belong to category c. (e.g., belong to ("robotic", "Artificial Intelligent")

R = {rel (s, i) | s ∈ S, i ∈ I} is the set of relationships between terms of set S and instances of set I and are denoted by hierarchy and are denoted by {rel (s, i) | s ∈ S, i ∈ I} mean that s has a relationship with i. The relationships can be synonymous, hyponymous or hypernymous. (e.g., synonym (“ADT”, “Abstract data type”), hyponym (“data structure”, “Abstract data type”), hypernym (“ADT is a kind of data structure that is defined by programmer”, “Abstract data type”).

In addition, all concepts, instances of this ontology focus on computer domain; therefore, this ontology is known as Computing Domain Ontology (CDO). The structure of CDO is separated into four layers:

The first layer is known as the Topic layer. To build it, we extract vocabularies from ACM Categories [10]. We obtain over 170 different categories from this site and rearrange them in this layer.

Next layer is known as the ingredient layer. In this layer, there are many different instances, which are defined as nouns or compound nouns from vocabularies about computer domain, e.g., “robot”, “Super vector machine”, “Local Area network”, “wireless”, “UML”, etc. To setup this layer, we use Wikipedia to focus on English language and computer domain.

The third layer is known as the Synset layer. To set up this layer, we use the WordNet ontology. Like Wikipedia, we only focus on computer domain. This layer encloses a set of synset. A synset includes synonyms, hyponyms, and hypernyms of instances of the ingredient layer.

The last layer is known as the Sentence layer. Instances of this layer are sentences that represent syntactic relations extracted from preprocessing stage. Hence, these sentences are linked to one or many terms of the Ingredient layer. This layer also includes sentences that represent semantic relations between terms of Ingredient layer, such as, IS-A, PART-OF, MADE-OF, RESULT-OF, etc.

We use Neo4J to store CDO. Neo4J is graph database. The graph database representing for Computing domain is shown in Fig. 1.

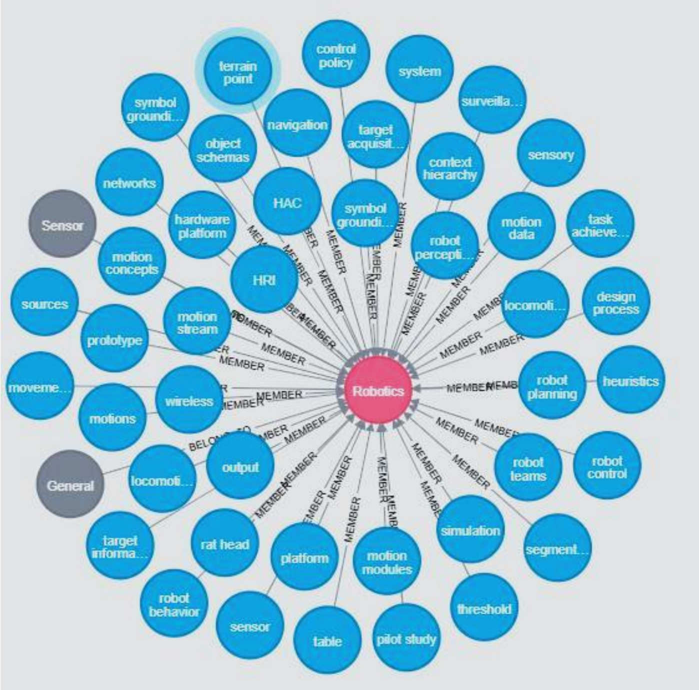

Figure 1. The hierarchy of CDO is represented by graph database (Neo4J)

As the Figure 1, the node is labeled “Robotics” symbolled a category in computing domain and other nodes are presented the instances of this category.

3.2 An approach to extending query sentences for semantic oriented search on Graph Database.

Definition 1. A query Q by natural language includes some words (w_1, w_2, …, w_n), which can be nouns, compound nouns, adjectives, verbs and adverb phrases. Query Q can be

- Nouns, compound nouns or simple sentence, e.g., “Relational database”, “Java is programming language”.

– Complex sentence, e.g., “Oracle database is a relational database system, which is usually used for business”.

Definition 2. A list of keywords is an ordered list of words (k_1, k_2, …, k_n), which obtained from the query Q by eliminating the unnecessary words.

In order to get the list of keywords, we use Stanford Lexical Dependency Parser (SLDP) [12]. The Stanford typed dependencies representation was designed to provide a simple description of the grammatical relationships in a sentence that can easily be understood and effectively used by people. SLDP generates a dependency graph, which maps straightforwardly onto a directed graph. For example, considering query “Robot is tell a lot of this conference”, dependency graph of this sentence is shown in Fig 2.

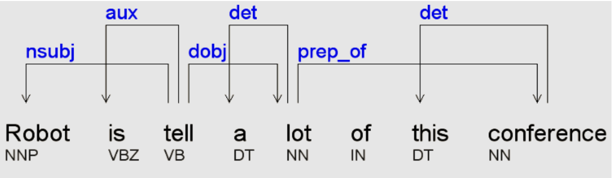

Figure 2. The dependency graph is generated by SLDP

We propose the algorithm for extracting keywords from query Q by using SLDP as follows.

Algorithm 1. Extracting keywords from query Q

```
Input: Query Q
Output: List of keywords K
List of keywords K ←∅
Dependency graph G ← SLDP(Query) /*SLDP generate dependency graph */
for each node of P
   if (Existing Subject in G) then /* Stanford dependency representation is nsubj) */
      keyword ← Subject
   else
         if (Existing noun/noun phrase in G) then
            keyword ← noun/noun phrase
         end if
   end if
  if (Subject has modifiers being adjective, adverb) then
      keyword ← adverb + adjective+ Subject
      K ← keyword
  end if
end for
K ← Filter (K) /* remove the keywords are not necessary*/
Return K
```

According to algorithm 1, we use the SLDP tool for collecting the keywords from a user’s query. Firstly, we select the subject of the query. Next, the nouns, noun phrases or subject modifiers will be considered. In the case of complex sentences, Filter (K) function will remove the keywords are not necessary to narrow down the list of keywords based on the context and syntax of these sentences.

Definition 3. A graph database D is a collection of entities which have relationship each other. An entity in graph database is denoted as E (A_1, A_2, …, A_n), where E is the name of the entity and A_1, A_2, . . ., A_n are attributes of the entity. The vocabulary of the database E, denoted as VE, is the set VE= $\{X \mid \exists E (A_1, A_2, \ldots, A_n) \in E\}$ [6]

Definition 4. An interpretation of the list of keywords query K={k_1, k_2, …, k_n} on a graph database D is an Cipher query in Neo4J such as: MATCH (E1)-[A[r_1:X1] ← (E_2)-[A2:X2] .... ← (E_n)-[An:Xn]) where $E_1.A_1=k_1$, $E_2.A_2 = k_2$ …. $E_n.A_n = k_n$ return $E_1.A_1$, $E_2.A_2$, ...$E_n.A_n$

Example 3.1. Consider other query “Detecting keywords of sentences in text files”. The dependency map straight forwardly onto a directed graph, as shown in Fig 3.

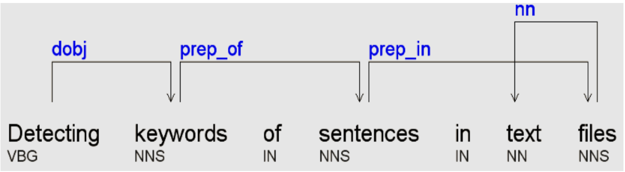

Figure 3. The dependency graph represents to example 3.1

Applying algorithm 1 for this graph (Fig.4), the list of keywords includes two keywords, “keywords of sentences” and “text files”. However, the keyword “text files” is removed from the list because this one is the noun phrase of place. At least two different interpretations can be generated from the keyword, “keywords of sentences”. One is the “MATCH (E1)-[r1:MEMBER] ← (E2)-[r2:Hyponym] ← (E3)-[r3:Hypernym] ← (E4)-[r4:Synonym] where E1.content CONTAINS 'keyword' return E1.Content, E2.Content, E3.Content, E4.Content” and the other is “MATCH (E1)-[r1:MEMBER] ← (E2)-[r2:Hyponym] ← (E3)-[r3:Hypernym] ← (E4)-[r4:Synonym] where E4.content='keyword of sentences’ return E1.Content, E2.Content, E3.Content, E4.Content”.

We propose an algorithm for processing the semantic-based keyword search as follows.

Algorithm 2. The algorithm for processing the semantic-based keyword search.

```
Input: Order List of Keywords K
Output: The Cipher query (C) for searching information on graph database after mapping keywords of the order list
for each keyword $k_i$ in the order list of keywords K
  if ($k_i$ is abbreviation word) then
     Search $k_i$ on Synnonym relation
      C ← Entity having an attribute = $k_i$
  else
      if ($k_i$ has one or many prepositions and i=1) /* $k_i$ is the first keyword in the
                                  order list K */
      Search $k_i$ to other relation
       C ← Entity having an attribute like $k_i$
      else
        if (i=1) then
          $k_i$ is an entity which has a relationship “MEMBER” with “root”
          C ← Entity = $k_i$
        else
            $k_i$ is an attribute of entity which has a relationship “MEMBER”
            C ← Attribute of Entity = $k_i$
        end if
     end if
  end if
end for
Return Cipher Query C
```

Example 3.2. The other query “CPU Pentium dual core I5”. The dependency map straightforwardly onto a directed graph, as shown in Fig 4.

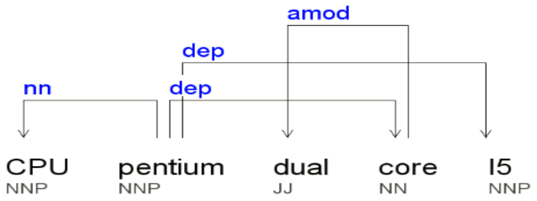

Figure 4.The graph represents to query by example 3.2.

Applying algorithm 1, the list of keywords in this case includes three keywords, “CPU”, “Pentium” and “dual core I5”. Then applying algorithm 2, the interpretation of this query is like that “MATCH (E1)-[r1: MEMBER] 🡐 (E2)-[r2:Hyponym] 🡐 (E3)-[r3:Hypernym] 🡐 (E4)-[r4:Synonym] 🡐 (E5)-(r5:BELONG_TO) where E4.content='CPU’ and (E5.content='pentium' Or E5.content='dual core I5') return A.Content, B.Content, C.Content, D.Content, E.Content “.

4 EXPERIMENTAL RESULT AND DISCUSSION

We implement numerous experiments for studying the efficiency of the proposed approach. We select for our experiments with two data sets. The first set is the abstracts of papers, which we get from ACM Digital Library as following.

- 150 abstracts in Software category. All these abstracts are focus on Software category based on keywords of papers.
- 150 abstracts in Database category. All these abstracts are focus on Database category based on keywords of papers.
- 150 abstracts in Artificial Intelligent (AI) category. All these abstracts are focus on Artificial Intelligent category based on keywords of papers.

All these abstracts of ACM Digital Library include the simple sentences and complex sentences.

The simple sentences have only one main clause, for example “CPU is Central Processing Unit” and complex sentences have a main clause and one or more subordinate clauses, introduced by a subordinating conjunction, for example “Artificial Intelligent is applied to many fields but the number of research is limited until now.”

The other set is the queries, which are manually input made directly by end users. These queries of end users are also categorized to 3 subset including Software, Database and Artificial Intelligent. Besides the category on dataset, in each subset of queries of end users is also including the different structure of sentences such as simple sentences and complex sentences. The data set of end users as following:

- 100 sentences including simple and complex sentences in Software category.
- 100 sentences including simple and complex sentences in Database category.
- 100 sentences including simple and complex sentences in Artificial Intelligent category.

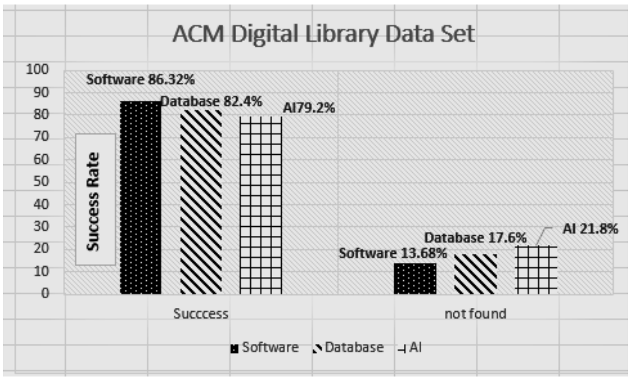

Figure 5. Experiment with the ACM Digital Library data set.

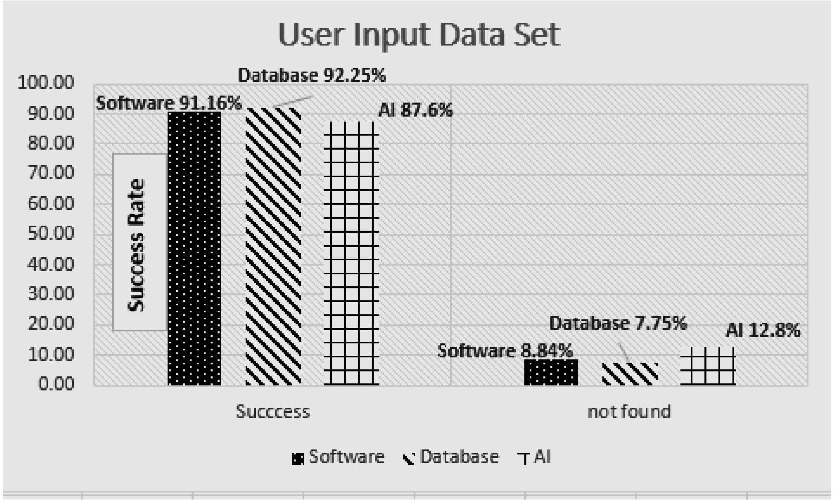

Figure 6. Experiment with the user input data set.

The scores reported in Figure 5 reveal that the success rate for interpreting the keyword queries in the first data set (Fig 5) is lower than that in the second set (Fig 6) in the same category, e.g., 86.32% vs. 91.16% in software category, 82.4% vs. 92.25% in database category, and 79.2% vs. 87.6% in AI category. The success rate means that generating Cipher query commands are successful and searching data on knowledge graph with these commands returns exactly. In a contrast, the "not found" means that the generating Cipher query command are not successful and searching data on knowledge graph also do not return anything.

5 CONCLUSIONS AND FUTURE WORKS

A novel approach is proposed in this paper has been for extending queries to search semantic-based keyword on knowledge graph. Efforts were also invested to reduce the overall processing time while interpreting keyword queries to Cipher queries in graph database. The process of interpretation occurs automatically considering meta-information and syntactics of queries. We have applied Natural Language Processing with supporting by SLDP in our approach. We have implemented and evaluated the proposed approach for two data sets related to Computer domain, ACM Digital Library, and input queries by users. The results are good for simple queries but not very good for complex queries, especially to the data set of the ACM Digital Library. In the future, we will optimize the algorithms to solve these problems.

REFERENCES

[1]. J. Atkinson, A. Gonzalez, M. Munoz, H. Astudillo, *Web Metadata Extraction and Semantic Indexing for Learning Objects Extraction, on The 26th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE 2013), vol. 7906, 2013, pp. 131-140.*

[2]. M. Saleh, *Semantic-Based Query in Relational Database Using Ontology*, *Canadian Journal on Data, Information and Knowledge Engineering*, vol. 2, no. No. 1, January 2011

[3]. S. Bergamaschi, F. Guerra, M. Interlandi, "*QUEST: A Keyword Search System for Relational Data based on Semantic and Machine Learning Techniques*, in *Proc of the VLDB Endowment*, vol. 6, Trento, Italy, 2013, pp. 1222-1225.

[4]. A. Elsayed, A. Eldin, D. E. Zanfaly, *Enhancing Keyword Search over Relational Databases Using Ontologies*, in *Proc. Int. Conf on the Third International Conference on Advances in Computing & Information Technology* (ACITY 2013), Chennai, India., 2013, pp. 147-154.

[5]. Oracle. [Online]. Avalable : *https://docs.oracle.com/database/121/RDFRM/toc.htm*

[6]. S. Bergamaschi, E. Domnori, F. Guerra, *Keyword Search over Relational Databases: A Metadata Approach*, in *Proc. Int. Conf on the 2011 ACM SIGMOD International Conference on Management of data (SIGMOD'11)*, Athens, Greece, 2011.

[7]. F. Bourgeois, J. C. Lassalle, *An extension of the Munkres algorithm for the assignment problem to rectangular matrices*, *Communications of ACM*, vol. 4, no. 12, pp. 802-804, December 1971.

[8]. H. Bast, E. Haussmann, B. Bjorn, *Semantic Search on Text and Knowledge Bases*, *Journal of Foundation and Trends in Information Retrieval*, Vol 10, Issue 2-3, 2016

[9]. Son. T. C, Linh Anh Nguyen, Ngoc Thanh Nguyen, *Extending Query-Subquery Nets for Deductive Databases under the Well-Founded Semantics*, *International Journal of Information and Experience Engineering in Semantic Society: Some Challenges, Approaches, and Case Studies*, vol. 48, 2017

[10]. L. Zhang, *Ontology Based Partial Building Information Model Extraction*, *Journal of Computing in Civil Engineering*, pp. 1-44, Mar 2012.

[11]. *ACM Computing Classification System*. [Online]. Avalable : https://www.acm.org/publications/class-2012

[12]. *The Stanford Natural Language Processing Group*. [Online]. Avalable : http://nlp.stanford.edu/software/lex-parser.shtml.

MỘT GIẢI PHÁP MỞ RỘNG CÂU TRUY VẤN CHO VIỆC TÌM KIẾM HƯỚNG ĐẾN NGỮ NGHĨA TRÊN ĐỒ THỊ TRI THỨC

Tóm tắt. Trong những năm gần đây Bản thể học được áp dụng trong nhiều ứng dụng khác nhau, đặc biệt là trong lãnh vực Web ngữ nghĩa, Truy xuất thông tin, Khai thác thông tin và các Hệ thống trả lời câu hỏi. Mục đích của Bản thể học là để loại bỏ sự nhầm lẫn về các khái niệm và thuật ngữ. Bản thể học được hình thành dựa trên một tập các khái niệm đặc trưng cho miền chuyên biệt mà bản thể học đề cập cũng như các các định nghĩa và các quan hệ trong bản thể học. Phụ thuộc vào các đặc tính của dữ liệu mà có một số phương pháp để biểu diễn Bản thể học như: Khung mô tả tài nguyên (RDF), Ngôn ngữ bản thể học Web (OWL) hay các cơ sở dữ liệu. RDF, OWL phù hợp cho các đối tượng dữ liệu có mối quan hệ đơn giản. Nhưng nếu mối quan hệ giữ các đối tượng dữ liệu phức tạp hơn thì dùng các cơ sở dữ liệu để biểu diễn là phù hợp hơn. Tuy nhiên ngôn ngữ truy vấn SQL trong các cơ sở dữ liệu quan hệ không hỗ trợ việc tìm kiếm hướng đến ngữ nghĩa và tốc độ tìm kiếm thường chậm. Do đó bài báo này đề nghị một phương pháp mở rộng câu truy vấn để tìm kiếm hướng đến ngữ nghĩa trên đồ thị tri thức.

Từ khóa. Đồ thị tri thức; Tìm kiếm ngữ nghĩa; Mở rộng truy vấn.

Ngày nhận bài: 05/05/2021

Ngày chấp nhận đăng:17/07/2021