BUILDING QUESTION ANSWERING SYSTEM BASED ON COMPUTING DOMAIN ONTOLOGY

TẠ DUY CÔNG CHIẾN

Khoa Công Nghệ Thông Tin, Trường Đại học Công nghiệp Thành phố Hồ Chí Minh;
taduycongchien@iuh.edu.vn

Abstract. Question answering systems are applied to many different fields in recent years, such as education, business, and surveys. The purpose of these systems is to answer automatically the questions or queries of users about some problems. This paper introduces a question answering system is built based on a domain specific ontology. This ontology, which contains the data and the vocabularies related to the computing domain are built from text documents of the ACM Digital Libraries. Consequently, the system only answers the problems pertaining to the information technology domains such as database, network, machine learning, etc. We use the methodologies of Natural Language Processing and domain ontology to build this system. In order to increase performance, I use a graph database to store the computing ontology and apply no-SQL database for querying data of computing ontology.
Keywords. Ontology, Question answering, Graph databases.

1 INTRODUCTION

Domain ontology, including of the concepts and the relations among the concepts, is applied in a variety of applications. The Question Answering (QA) system in Information retrieval is one of the applications to be applied to the domain specific ontology. In other words, The QA systems enable asking questions and retrieving an answer using natural language queries [1]. The QA systems play an important role in the science and the life. There are a lot of algorithms relating to Natural Language Processing, Machine Learning, Deep Learning which are applied to develop the QA systems [2-3]. QA systems are a growing research field worldwide [4]. The demand for this kind of system increases day by day since it delivers short, precise and question-specific answers [5]. In the life, the QA systems help the business companies understanding clearly the needs of their customers to develop their business. In the colleges or universities, the QA systems are constructed to serve pupils in their training. Unlike Information Retrieval, where full documents are returned from user requests, QA systems usually return the precise short answers instead of full documents [6]. Therefore, the QA systems are developed for restricted domain and have limited capabilities.

With good domain ontology, we can determine the answers to the any questions of users. My idea is to use the keywords in the questions or the queries of users to understand the subject of the questions or the queries. After that, I will use the computing ontology to find out the answers based on the subject of the queries.

My key contributions are as follows: (i) a large-scale dataset from the ACM Digital Library, Wikipedia and WordNet focus on computing domain have been crawled; (ii) I propose a novel method for obtaining the list of keywords from questions or queries of users; (iii) the algorithm for generating automatically the Cypher query based on the list of keywords to answer the questions.

The rest of this paper is organized as follows: section 2 - related works; section 3 - automatic subject labeling of text document; section 4 - experimental results and discussion; section 5 - conclusions and future works

2 RELATED WORKS

In recent years, QA systems are interested in the researchers specifically for information extraction

researchers [7]. As outline from Athenikos [8] and Kolomiyets [9], they built a QA system based on the knowledge base. G. Suresh kumar et al [10] proposed a methodology to extract concept relations from unstructured text using a syntactic and semantic probability-based Naïve Bayes classifier for building a QA system based on domain ontology. Their QA system includes two modules, (1) a dynamic concept relational (CR) ontology construction module capable of extracting new concepts from the web and incorporating the extracted knowledge into the CR Ontology knowledge base, and (2) an answer extraction module that formulates the query string from the natural language question according to the expected answer and retrieves the information from the ontology for answer formation. Y. Cao et al [11] built an online question answer system for clinical questions. They proposed a methodology based on Natural Language Processing to extract information from text corpus. M. Sarrouti et al [12] proposed a methodology for building the QA system. They use the text document retrieval system based on PubMed search engine and UMLS similarity to retrieve relevant documents to a given biomedical question. Then they take the abstracts from the retrieved documents and use Stanford CoreNLP for sentence splitter to make a set of sentences, i.e., candidate passages. Using stemmed words and UMLS concepts as features for the BM25 model, they finally compute the similarity scores between the biomedical question and each of the candidate passages and keep the N top-ranked ones. J. Fukumoto et al [13] built a QA system using NLP. In order to answer the given questions, they firstly extract modifying words to query words of an input question because modifying words will be constraints to query words. Then, the system classifies the extracted modifying words according to their Named Entity types such as city names, sports names and so on and the most frequent type will be selected. The words in this type will be used for narrowing down and re-retrieving documents. In order to select an appropriate word, the system retrieves documents using each word in this type and chooses the document set which includes the largest number of answer candidates. QA system provides the word as the best clue word which can retrieve the largest number of answer candidates using user interaction. QA system will continue to the above interaction until a user gets an appropriate answer to the given question. M. Spranger et al [14] also built a QA system in the field of criminal proceeding. They proposed an integrated computational solution for supporting the evaluation process of forensic text using computer linguistic technologies. Their QA- system can solve a specific criminal issue and visualize issue-centred case-relevant relationships. For this purpose, several state-of-the-art techniques in the fields of text categorization and information/event extraction are analyzed with respect to their suitability for the peculiarities of the considered domain. Subsequently, several approaches for solving domain- specific problems are introduced. Z. Hu et al [15] proposed a deep learning approach for predicting the quality of online health expert question-answering services.

Generally, there is a lot of methods to build QA systems. The researches can apply approaches related to NLP, Machine Learning, Deep learning or hybrid approaches. In this paper, I use NLP and specific domain ontology to build the QA system in computing domain.

3 BUILDING THE QUESTION ANSWERING SYSTEM BASED ON COMPUTING ONTOLOGY

Definition 1. Questions or queries are the natural language sentences which are given as input by the users

According to Kolomiyets [9], questions can be divided some types as: Factoid, list, definition and complex questions. Factoid questions are the ones that ask about a simple fact and can be answered in a few words [16], for instance, How far is it from Earth to Mars?; list questions demand as an answer a set of entities that satisfies a given criteria [16], for instance, When did Brazil win Soccer World Cups?; definition questions expect a summary or a short passage in return [17], for instance, How does the mitosis of a cell work?; complex Question is about information in a context. Usually, the answer is a merge of retrieved passages. This merge is implemented using algorithms, such as: Normalized Raw-Scoring, Logistic Regression, Round-Robin, Raw Scoring and 2-step RSV [18].

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.

3.1 Architectural Layout of the Question Answering System

Figure 1 illustrates the simple architectural layout of the proposed QA system. There are three main modules in this layout including **Computing Domain Ontology** (CDO), **Question Analysis** and **Interpreter**. The CDO is built on Computing domain, which is the knowledge base storing the computing concepts. The question analysis phase consists of question analysis, semantic analysis in order to identify the list of keywords. Interpreter phase will generate automatically the Cypher query from the list of keywords to access CDO for selecting the best answer of the given questions.

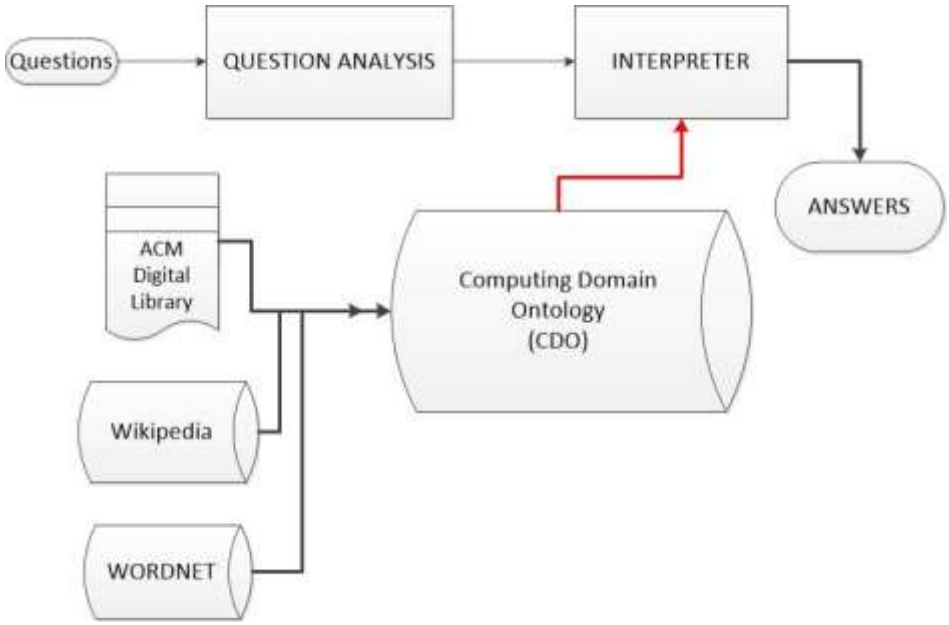

Figure 1: Architecture of QA system

3.2 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.

Formally, an ontology can be defined as a tuple:

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

Where,

C, is set of classes, i.e., Concepts represent categories of the computing domain (for example, "Artificial Intelligence, hardware devices, NLP" (C)

I, is a set of instances belong to categories. Set I consists of vocabulary of computing (for example, "robotic, Random Access Memory "$\in$ I)

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

$N = N^S$ is the set of synonyms of instances of set I.

$H = H^H$ is the set of hyponyms of instances of set I.

$Y = Y^H$ is the 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 $\in$ I, c $\in$ C} is the set of semantic relationships between concepts of set C and instances of set I and are denoted by {belong_to (i, c) | i $\in$ I, c $\in$ C} mean that i belong to category c. (e.g., belong_to ("robotic", "Artificial Intelligence")

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 {rel (s, i) | s ∈ S, i ∈ I} and mean that s has a semantic relationship with i.

The semantic 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 computing 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. In order to build it, we extract categories from ACM Classification Categories [19]. 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 vocabulary about computing domain, e.g., "robot", "Super vector machine", "Local Area network", "wireless", "UML", etc. In order to setup this layer, we use Wikipedia focus on English language and computing domain.

The third layer of CDO is known as the Synset layer. To set up this layer, we use the WordNet ontology. Like Wikipedia, we only concentrate on computing domain. This layer includes synonyms, hyponyms, and hypernyms of instances of the Ingredient layer. As we combine two ontologies, which are Wikipedia and WordNet. Therefore, CDO has an advantage that is the instances can belong to many different categories of computational domain.

The last layer of CDO 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. The overall hierarchy of CDO is shown in Fig. 2 [20].

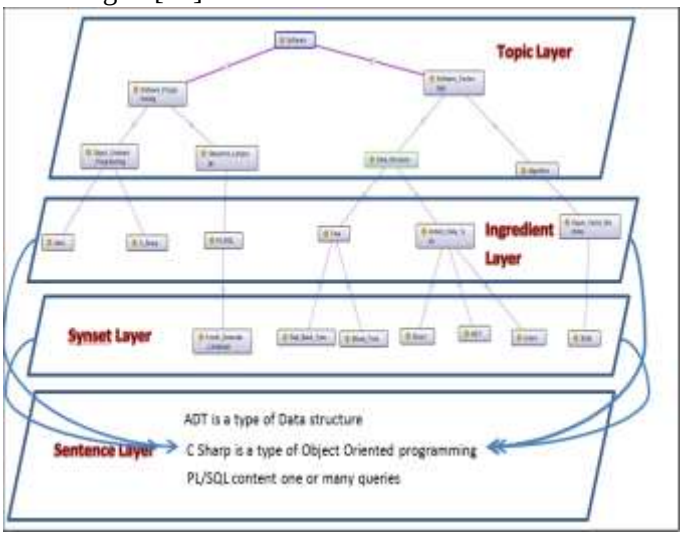

Figure 2: The hierarchy of CDO

An approach for building a QA system based on CDO includes two phases:

- Question analysis.
- Interpreter and answering the queries based on CDO.

3.3 Question Analysis

In question analysis phase the user gives a sentence of natural language as input and it is sent to Tokenizer. The Tokenizer split the sentences into words based on whitespace character. The tokenized words are taken to extractor for stemming process. In stemming process, the extractor maintains the

collection of predefined words which is used for comparison with the incoming new words. Predefined words are most used words in the document for querying. It compares the tokenized words with the predefined and extracts the main keywords. I.e., the keywords are words that are present in the predefined list of words. Then from the extracted words, the root words are identified. In the semantic analysis, the identified set of words will be given as input. The parse tree is generated through the parser and subject, object & verb present in the set of words is identified. The output of this analysis will be the collection of identifying words. We use OpenNLP [21] to tokenize, stemming the queries. We also use Stanford Dependency Parser (SDP) [22] in order to generate parse tree and Part-Of-Speech tagging (POS tag).

3.4 Interpreter and answering the queries based on CDO

Definition 3. An interpretation of the list of keywords query K= {k_1, k_2, …, k_n} on a graph database D is a no-SQL query such as "match (A:R_1), (B:R_2),...(N:R_n) where A_1=k_1 AND A_2=k_2 AND… AND A_n=k_n", where A, B,…, N are instances of the relations R_1, R_2,…, R_n and A_1, A_2,..., A_n are the properties of R_1, R_2, R_n respectively (A is difference of A_1, A_2, A_n)

After question analysis, the system will generate the Cypher Query Language of Neo4j graph database based of the list of keywords. This procedure is called Interpreter.

Since each keyword in a query can be mapped to relation name, there are $\sum_{k=1}^{n} R_k$ (n=5 because there are 5 relations including ingredient, synonym, hyponym, hypernym and other relations belong to sentence layer such as: PART-OF, IS-A) different interpretations. Therefore, there is a total of N*$\sum_{k=1}^{n} R_k$ different interpretations, where N represents keywords and n represents the relations. My work is that how different kinds of meta-information and interdependencies between the mapping of keywords to database terms can be exploited in order to choose only one interpretation. We propose an adaptable approach for generating the Cypher query from the list of semantic keywords, as shown in Fig 3 [20].

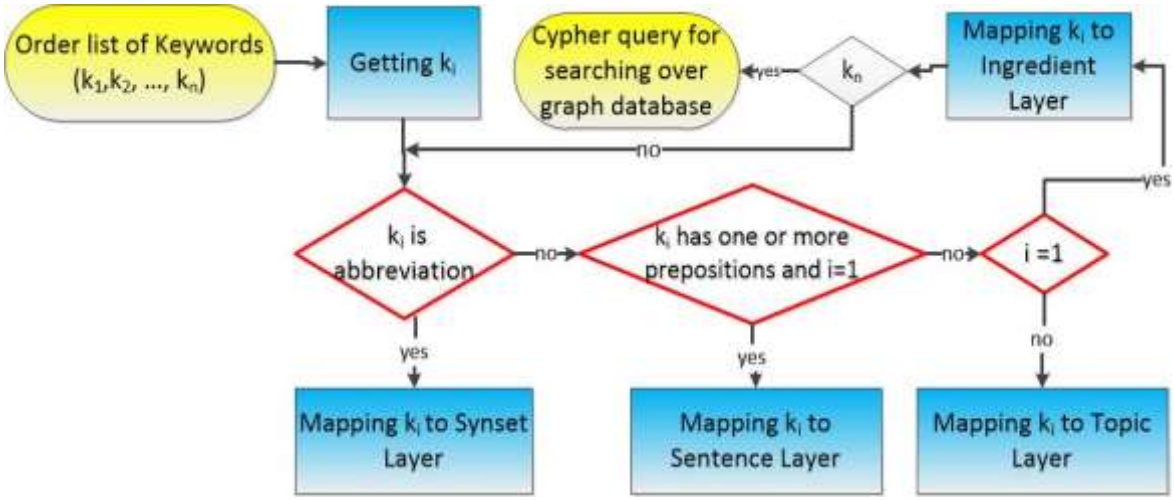

Figure 3: Algorithm for generating the Cypher query based on the list of semantic keywords

We propose an algorithm for generating the Cypher query based on the list of semantic keywords to search answers in graph database as follows.

Algorithm 1

Input: Order List of Keywords (K)

Output: The Cypher query using for graph database (CQ)

For each keyword k_i in K

If (k_i is abbreviation) then

Mapping k_i to Synset Layer
CQ ← instance of Synset layer = k_i
Else
If (k_i has one or many prepositions and i=1) /* k_i is the first keyword in K */
Mapping k_i to Sentence Layer
CQ ← instances of Sentence layer like k_i
Else
If (i=1) then
Mapping k_i to ingredient relation
CQ ← instance of Ingredient layer = k_i
Else
Mapping k_i to topic layer
CQ ← instance of Topic layer = k_i
End if
End if
End if
End For
Return CQ

The Cypher query, which is generated from the algorithm 1, will be used to access graph database for searching the best answers based on CDO. Additionally, there are approximately 170 terms in Topic layer, 200000 relations in Synset, Sentence layers, and 400000 terms in Ingredient layer [20]

4 EXPERIMENTAL RESULT AND DISCUSSION

We implement numerous experiments for studying the efficiency of the proposed approach. We select two real corpora for testing.

The first corpus is a fraction of papers belonging to ACM Digital Libraries. We pick a random 300 papers as below:

- 100 papers belong to Software category;
- 100 papers belong to Database category;
- 100 papers belong to Artificial Intelligent category

We use three measures: Precision (P), Recall ® and F-measure for experimental evaluation.

$$P(C_i) = \frac{Correct(C_i)}{Correct(C_i) + Wrong(C_i)} \quad (1)$$

$$R(C_i) = \frac{Correct(C_i)}{Correct(C_i) + Missing(C_i)} \quad (2)$$

$$F - measure(C_i) = 2\frac{Precision(C_i) * Recall(C_i)}{Precision(C_i) + Recall(C_i)} \quad (3)$$

Where:

C_i denotes a category in CDO; Correct (C_i) denotes a number of the sentences which are found in CDO and they accurately belong to the category C_i; Wrong (C_i) denotes a number of the sentences which are found in CDO, but they do not belong to category C_i; Missing (C_i) denotes a number of the sentences which are not found in CDO. The experimental evaluation is shown as table 1.

Table 1. The experimental evaluation of the QA system with the first corpus

Categories	Precision	Recall	F-measure
Software	91,03%	87,62%	89,29%
Database	89,74%	85,32%	87,45%
Artificial Intelligent	93,27%	90,18%	92,15%

The second corpus is the queries, which are input directly by users. This corpus includes 320 queries having the 4 types of sentence structure as below

- 80 queries are only noun phrases, e.g., "Java language", Relational database", "Support vector machine", etc.
- 80 queries are simple sentences, which contain only simple subject and simple predicate [23]. Simple subject is a noun or noun phrase and the simple predicate is always a verb, verb string or compound verb, e.g., "What is Java language", "How is Relational database", "What is robot", "What is router", etc.
- 80 queries are simple sentences, which contain subject and complex predicate [23]. The complex predicate consists of the verb and all accompanying modifiers and other words that receive the action of a transitive verb or complete its meaning, e.g., "Java is programming language", "What is indexed in SQL Server", "How is Robot changing many things in life", etc.
- 80 queries contain complex sentences, wrong grammar sentences, and unfinished sentences, i.e. they do not contain a complete thought, e.g., "mining database", "How warehouse database", "What SQL Server is database belongs to relational database", etc.

The experimental results are shown as table 2

Table 2. The experimental evaluation of the QA system with the second corpus

Kinds of queries / Parameters	Noun phrases	Simple sentences (Subject + simple predicate)	Simple sentences (Subject + complex predicate)	Wrong / Complex sentences
Number of queries	80	80	80	80
Correct answer results	100%	90%	82%	67%
Wrong answer results	0%	10%	18%	32%

The scores reported in table 2 reveals that the ratio of the correct answers of the QA system is higher than the wrong answers with the simple queries because the cypher queries which are generated in these cases are very well.

5 CONCLUSIONS AND FUTURE WORKS

Our experiment tried to answer automatically the queries of users based on specific domain ontology. We proposed an approach has two phases: the first is to analysis the queries of users in order to identify the list of keywords of queries and the second phase is to automatically generate the Cypher Query Language, which is the no-SQL type for accessing data form Neo4j graph database. We use the algorithms of Natural Language Processing, domain specific ontology and the SDP tool in order to solve a proposed approach. The experimental results are evaluated by the precision and recall measures. Results generated by such experiments show that the proposed algorithm yields performance respectably. On the future, we will upgrade the Question answering system in order to increase ratio of the correct answers with complex sentences

REFERENCES

[1] P. Aarabi, "Virtual cardiologist—a conversational system for medical diagnosis", in Proceedings 26th Annual IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), IEEE, pp. 1–4, 2013
[2] A. Mishra, S.K Jain, "A survey on question answering systems with classification", Journal of King Saud University – Computer and Information Sciences, 2016
[3] M.A. Calijorne, F.S. Parreiras, "A literature review on question answering techniques, paradigms and systems", Journal of King Saud University – Computer and Information Sciences, 2018
[4] E.M. Voorhees, D.M. Tice, "Building a question answering test collection", in Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp. 200–207, 2000
[5] S. Pudaruth, K. Boodhoo, L. Goolbudun, "An intelligent question answering system for ict", In Proceedings of The International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT). IEEE, pp. 2895–2899, 2016
[6] M.A. Bauer, D. Berleant, "Usability survey of biomedical question answering systems", Human Genom, 2012
[7] M. Maybury, "New directions in question answering. Advances in open domain question answering", Springer, pp. 533–558, 2008
[8] S.J. Athenikos, H.Han, A.D. Brooks, "A framework of a logic-based question answering system for the medical domain (loqas-med)", in Proceedings of the 2009, ACM symposium on Applied Computing. ACM, pp. 847–851, 2009.
[9] O.Kolomiyets, M.-F. Moens, "A survey on question answering technology from an information retrieval perspective", Journal of King Saud University, Computer and Information Sciences, p.p 5412–5434, 2011.
[10] G. Suresh kumar et al, "Concept relation extraction using Naive Bayes classifier for ontology-based question answering systems", Journal of King Saud University, Computer and Information Sciences, 2015.
[11] Y. Cao et al, "An online question answering system for complex clinical questions", Journal of Biomedical Informatics, vol 44, pp. 277–288, 2011.
[12] M. Sarrouti, S.O.E. Alaoui, "A passage retrieval method based on probabilistic information retrieval model and UMLS concepts in biomedical question answering", Journal of Biomedical Informatics, pp. 96-103, 2017.
[13] J. Fukumoto, N. Aburai, R. Yamanishi, "Interactive Document Expansion for Answer Extraction of Question Answering System", in Proceedings of the 7th International Conference in Knowledge Based and Intelligent Information and Engineering Systems (KES2013), 2013
[14] M. Spranger, D. Labudde, "Establishing a Question Answering System for Forensic Texts", Journal of Procedia - Social and Behavioral Sciences, pp. 197-205, 2014
[15] Z. Hu et al, "A deep learning approach for predicting the quality of online health expert question-answering services", Journal of Biomedical Informatics, vol 71, pp. 241-253, 2017
[16] M.H. Heie, E.W. Whittaker, S. Furui, "Question answering using statistical language modelling", Journal of Computer Speech Language, vol 26, 193–209, 2012
[17] M. Neves, U. Leser, "Question answering for biology", Methods 74, 36–46, 2015
[18] M. Garc´a-Cumbreras, F. Martínez-Santiago, L. Ureña-López, "Architecture and evaluation of bruja, a multilingual question answering system", Information Retrieval Journal, vol 15, 413–432, 2012.
[19] ACM. [Online]. Available: http://www.acm.org/about/class/ccs98-html.
[20] Chien Ta Duy Cong. "Building information extraction system based on computing domain ontology", Doctoral thesis of the University of Technology HCM, Vietnam, 2016
[21] OpenNLP. [Online]. Available: https://opennlp.apache.org/.
[22] The Stanford Natural Language Processing Group," [Online]. Available: http://nlp.stanford.edu/software/lex-parser.shtml
[23] Capital Community College Foundation. [Online]. http://grammar.ccc.commnet.edu/grammar/objects.htm

XÂY DỰNG HỆ THỐNG TRẢ LỜI CÂU HỎI DỰA TRÊN BẢN THỂ HỌC TRÊN MIỀN CHUYÊN BIỆT TIN HỌC

Tóm tắt: Trong những năm gần đây, các hệ thống trả lời câu hỏi đã được áp dụng cho nhiều lĩnh vực khác nhau như trong giáo dục, kinh doanh và trong việc khảo sát điều tra. Mục đích của các hệ thống này được là trả lời tự động các câu hỏi hay các câu truy vấn của người dùng về một số các vấn đề. Bài báo này

giới thiệu một hệ thống trả lời câu hỏi được xây dựng dựa trên Bản thể học trên miền chuyên biệt. Bản thể học này bao gồm dữ liệu và từ vựng liên quan đến lĩnh vực Tin học được xây dựng từ các tập tin văn bản lấy từ thư viện ACM. Do đó, hệ thống chỉ có thể trả lời các vấn đề liên quan đến lĩnh vực Công nghệ thông tin chẳng hạn như cơ sở dữ liệu, mạng máy tính, học máy, v.v... Để xây dựng hệ thống, chúng tôi đã dùng phương pháp Xử lý ngôn ngữ tự nhiên và Bản thể học . Để giúp cho hệ thống đạt hiệu quả cao, chúng tôi đã sử dụng cơ sở dữ liệu đồ thị để lưu trữ Bản thể học và áp dụng no-SQL của cơ sở dữ liệu trong việc truy vấn Bản thể học trên miền Tin học.

Ngày nhận bài:18/07/2019

Ngày chấp nhận đăng:11/11/2019