A method of inquiring ontology with semantic templates

Ouyang Xin[1,2]
1.Faculty of Information Engineering and Automation
Kunming University of Science and Technology
Kunming City, P.R.China
2. Yunnan Key Lab of Computer Technology
Application, Kunming City, P.R.China
kmoyx@hotmail.com

Shuai Chunyan[3]
3. Faculty of Electric Power Engineering
Kunming University of Science and Technology
Kunming City, P.R.China
earth0806@sina.com

***Abstract*—It is always a challenge by using statistical method in corpus database to analyze semantics of natural language Sentences (NLS). This paper proposes a method of recognizing and translating ontology query in natural language, called OntoQuery-NLP. With the help of pre-create semantic templates, the OntoQuery-NLP maps NLSs matching the format of the semantic templates into formal semantic expressions. By parsing these semantic expressions, the OntoQuery-NLP recognizes the queries and gets the correct answers from ontology. Compared with other methods, the OntoQuery-NLP, without the support of any corpus, has faster retrieving speed and higher retrieving accuracy.**

Keywords-Semantics of Natural language;Ontology;Question Answering;semantic

I. INTRODUCTION

The knowledge expressed and processed by computer is usually structured data, and they are numeral data or informational data. Searching some data in a database is not an easy work for the average person, but a more highly skilled task. People hope that they can interact with computer by using natural language instead of professional language or inflexible user interface. Interacting with computer by using natural language is a research hot spot for a long time, which belongs to the fields of natural language process (NLP) and Question Answering (QA). The relevant theories of QA system in general knowledge are always challenges, and in some particular fields, the research progresses of QA system applications have got develop rapidly.

In this paper, we propose a new QA method based on ontology (OntoQuery -NLP) to deal with the queries in natural language. By predefined ontology templates, the OntoQuery-NLP translates queries expressed in natural language into corresponding semantic sentences the computer known, and gives out an accurate answer in natural language.

Generally the ontology operations including ontology establishing and concepts inquiring from ontology, we focus on concepts inquiring in natural language. The inquiring methods of ontology common properties include: interactive query/answer (such as Protégé [1]) and programming interface (such as Jena API). Both the interactive QA and the programming interface require the user to own certain domain knowledge or some trained skills (being familiar with Protégé or programming with the Jena's API). We hope that there is an easy way which can help common users without domain knowledge query questions in natural language. For example, there is a question about the price of some goods in a market: "*Could you tell me how much of X*", in which "X" can be replaced by a specific goods' name in ontology, when the computer system receives this requiring, it can easily understand the meaning of "X", as well as the user's intentions, and can give out an answer accurately and quickly.

II. RELATED WORK

Natural Language Processing (NLP), concerns with the theories and the implements of the interactions in natural language between computers and human [2].

The early research of NLP focused on how to retrieve data from the database through natural language. Literature [3] created a knowledge base including structured description and semantic description, and an expert system containing predefined rules is also established, which can help people retrieve the data that satisfied the conditions described in natural language from database. Using NLP for web information is current research hot spot and the technology of NLP combined with semantic network is in vogue. Aqualog [4] is a portable question answering system, which can translate the queries that expressed in natural language into a formal language, and Aqualog can be enriched by ontology and can be used to retrieve the answer of the queries. Text2Onto [5] was present as a framework for ontology learning from textual resources, which can translate a concrete target language into any knowledge representation formalism and calculates a confidence for each learned object through the system.

Literature [6] presents a method to identify ontology components with the help of Natural Language Processing (NLP) techniques in legal texts and the method can extract concepts and relations among the concepts.

The natural languages which are used to inquire in ontology can be divided into two classes: controlled natural language and unrestricted natural language [7]. Controlled natural language close to natural language is in essence a kind of formal language, such as ACE [8]. Superficially, the usage of ACE similar to English, in fact, ACE has more strict grammar and can be translated into logic language automatically. Unconstrained natural language is the language that can be used for communication daily, perhaps it is Chinese, English, German, French or other natural languages. The related fields of the unconstrained natural

Published by Atlantis Press, Paris, France.

language involved in are also important branches of the field of information retrieval, whose correlation models are Boolean model, vector space, latent semantic indexing, and probability model. However, these models must be trained by the corpus, and study process in the materials is similar to acquiring knowledge of corpus from these materials. After that, people can ask some questions about the materials and the answers would be retrieved through the models.

But, establishing corpus and retrieving information from corpus are complicated processes, which will consume lots of time and memories.

III. PARSING WITH SEMANTIC TEMPLATES

To deal with the above problems, we propose an ontology semantic mapping methods, which translates the natural language into a formal language based on ontology without the help of corpus. The formal language can be explained by some standard ontology languages such as RDF or OWL. By means of semantic templates mapping, the topics described by natural language are transformed into corresponding concepts in ontology.

Definition 1: Semantic Cell (SC).

In this paper, we use triples as the minimum units of semantic expression. Such as, a triple <*s, p, o*> in RDF is a SC, in which **s** means subject, **p** means property and **o** means object, and if the triple <*s, p, o*> is in accordance with the knowledge of the domain, the interpretation of the triple is true and the triple has semantic.

Definition 2: The comparing of SCs.

Suppose u1 and u2 are two SCs, u1 is <*s1, p1, o1*> and u2 is <*s2, p2, o2*>, u1 is equal to u2 if and only if s1=s2 and p1=p2 and o1=o2.

Definition 3: Semantics Block (SB). SB is a sorted set of triples in RDF.

To explain the natural language sentences, we should analysis the vocabulary of natural language (VNL) and the VNL must be classified into several categories. We simply divided VNL into VNLT, NNL, PNL, NCV; the meanings of these words should be stated in the following passages.

Definition 4: The Vocabulary of Natural Language Template (VNLT), which consists of the words in natural language, such as English, Chinese. Each word itself in VNLT has some isolated meaning and form complete semantics by combining other words in order. Such as, we can define: *VNLT= {if, then, where, how, who, what}*

Definition 5: The Nouns of Natural Language (NNL).

VNL consists of nouns of natural language in special fields, and the nouns that semantic templates used only come from VNL.

Definition 6: The Predicates of Natural Language (PNL).

PNL consists of predicates of natural language in special fields, and the predicates in semantic templates come from PNL. We extract all the predicates in the sentences of natural language in a special field and construct a set named PNL.

Definition 7: The others not covered vocabulary (NCV) above, which are the words that are not included in VNLT, NNL and PNL.

Definition 8: All the words in the vocabulary of natural language (VNL) are made up of VNLT, NNL, PNL, NCV, namely: $VNL=VNLT \cup NNL \cup PNL \cup NCV$, where, VNLT, NNL, PNL, NCV are disjoint sets. Each sentence s is a combination of NLV, $s \in NTV^*$ (* means closure).

In practical, we only need to define VNLT, NNL and PNL, the sizes of VNLT, NNL, and PNL are greater, and the level of intelligence is higher.

Similarly, the words of formal language can also be classified .In consideration of ontology description language in the future may be extended; we use formal language rather than ontology description language.

Definition 9: Formal Language Concepts (FLC) .In ontology, Ontology Concepts Vocabulary (OCV) is a FLC, FLC includes class name, instance name and other terms but attributes of ontology.

Definition 10: Formal Language Predicates (FLP) .In OWL (Ontology Web Language), FLP is the set of words for describing attributes. In RDF, FLP is similar to the Property.

Next, we will describe the relationship between the formal language and natural language, as follows.

Definition 11: A mapping between NLN and FLC:

NF: NNL→FLC

It is not a one-to-one map, but an n-to-one map.

Definition 12: Semantic Template (ST).

ST is made up of the elements in VNL and symbols: *,N and P, which is a sorted set that could express RDF semantics. The symbol * is used to represent any symbols belonging to NCV in sentence. The symbol N denotes nouns and P denotes predicates. N and P are from NNL and PNL respectively.

For example,"**where*P*N**" is a ST, suppose that NNL={beer, soda water} and PNL={have, get}. The ST covers the following sentences:"*Please tell me, where I can get beer?*" or "*Where I can get soda water?*" .

For another example, the sentence of "*How much the rice?*" can be translated by the ST of "**how much*N**".

Definition 13: The ST can be divided into First-order ST, Second-order ST. The First-order ST is the ST which only has one word of VNT, and the Second-order ST only has two words of VNT.

In this study, it is enough to using the Second-order ST to maintaining ontology.

Definition 14: Predicates Mapping (PM) defined as:

PM: PNL→FLP

The predicates in natural language sentences can be converted to attributes in ontology language through the mapping of PM.

Because of the ambiguity of the natural language, this mapping is not a one-to-one map, but n-to-one. In OWL, property consists of object property and datatype property. The property has domain and range. To meet the above conditions, we should add some constraints for mapping.

Definition 15: Predicates Mapping Restricts (PMC), refers to the position of words in natural language corresponding to the domain and range in formal language attributes. For PMC performance way, there are several plans:

Plan 1: list all the templates refer to questions. Here is an example, see table 1.

TABLE 1 THE TEMPLATES OF ALL THE QUESTIONS ASKED

Template String	***Constraint***	***Semantics***
*What*taste*haw flakes*	Null	I(<"haw flakes" taste ?o>) =true
*What*taste*cornflakes*	Null	I(<"cornflakes" taste ?o>) =true
*What*taste*chips*	Null	I(<"chips" taste ?o>) =true
*What*taste*osmanthus cake*	Null	I(<"osmanthus cake" taste ?o>) =true
*What*taste*vinegar*	Null	I(<"vinegar" taste ?o>) =true

The first column of the table 1 is template string, that describes the format of question sentences in natural language. The second column is constraint filled by null, means it is unused. The third column is semantic which indicates that the formal semantic of the sentences that match the template would be explained as true, "**?o**", in this column, means a variable when it is assigned a correct value, the triple would satisfy the condition.

The point of this plan is that sentences in natural language can be parsed directly by templates without any constraints. The plan 1 can answer the following questions: "*Please tell me, **what** the **taste** of **haw flakes**?*", the sentence is translated into a triple through the template: <"**haw flakes**" **taste ?o**>.To get the value of the "**?o**", ontology try to find the matched triple and assign the answer to the "**?o**".

Plan 2: classify all the questions in the field, and limit the usage by the constraints, just as table 2 shown.

TABLE 2 LTMS-LIB

No	***Template String***	***Constraint***	***Semantics***
1	*What*P_1*N_1*	<N_1 P_1 o> and <N_1 *rdf:type goods* > exist in ontology, and $N_1 \in P_1.domain$,$P_1 \in$ {*taste,shape,season,region*}	I(< N_1 P_1 ?o>) =true
2	*Where*have*N_1*	<N_1 P o> exist in ontology, and $N_1 \in P_1.domain$, P∈{*address,locate*}	I(< ?s P N_1>) =true
3	*Where*find*N_1*	<N_1 P o> exist in ontology, and $N_1 \in P_1.domain$, P∈{*address,locate*}	I(< ?s P N_1>) =true

In table 2, LTMS-lib means Language Template Mapping Semantic Library. The No.1 template in table 2 can answer the following natural language sentence:

"*Please tell me, what the taste of cornflakes?*"

From the above two plans, we can see that the plan 1 can answer all the problems involved in the templates in table 1, but it is a huge project for us to build numerous templates library; the plan 2 uses the concepts and relations in ontology to establish concise templates, in which each template could express more sentences than those in plan 1.So we choose the plan 2 as the way of constructing the template library.

From the point of view of comprehending natural language semantic, seen in table 2, in fact the No.2 and No.3 template have the same semantic but have different presentation. In order to reflect this in formal language, we add some asserts, for example:

The triple is: (*A address B*), whose constraint is $B \in regions$, in which ***regions*** is a class in ontology.

The semantic of above description is equivalent to the triple: (*A locate B*), and the constraint is $B \in regions$, too, and so on. We can say the interpretations of the words of ***address*** and ***locate*** are similar, that is: *I(address)=I(locate)*, **I** means interpretations.

But there is a shortage in the plan 2: if some answers must be acquired through reasoning, the results from templates directly perhaps are not correct.

Such as, "*The sugar is in 3# tank,3# tank is in the storage room*", we can express the semantic with RDF:(*sugar in "3#tank"*) and ("*3#tank" in "storage room"*).

For the question:"*where is the sugar?*", we should only get the answer "*3#tank*" under the condition of above. The answer expected is "*3#tank*" and "*storage room*".

Therefore, we can add some contents for the template "**Where*have*N1**" as following:

I(<N_1 P_1 ?o1>)=true, I(<N_2 P_2 ?o2>)=true,
I(<N_1 P_3 N_2>)=true,
And the semantic conditions:
(P_2 rdfs:subPropertyOf P_3)
(P_3 rdfs:type owl:TransitiveProperty)

Definition 16: LTMS Semantic Mapping (LSM) is defined as: *LSM: ST→SB*

This means, semantic template (ST) can be explained by semantic cells (SBs).LSM shows that the formal semantic of natural language sentence can be extracted by the templates. We use LTMS-Lib express the set of LSM. Algorithm 1 can extract the semantic from natural language sentence.

ALGORITHM 1 THE ALGORITHM OF LTMS

Input: natural language sentence; Output: SB sb

1. The words in nsl is classified by VNLT, NNL, PNL, NCV, and sorted by their positions.
2. The elements in NNL and PNL are converted into FLC and FLP by the use of NF and PM.
3. Search the templates that match the structure of sentence and meet the semantic conditions of FLP in LTMS-lib, if the template is choose, then go on to the next step.
4. Get the corresponding triples of the template from LTMS-Lib, and assign it to the variable of sb.
5. Return sb as the result.

In order to test the above theory, we design an experiment as follow.

IV. EXPERIMENT

Because the retrieval of ontology in essence belongs to the field of information retrieval (IR), a popular evaluation method is TREC QA Track [9], but TREC QA Track needs large-scale corpus according to the algorithm of the evaluation. We only focus on ontology; it is not suitable for

us to use TREC QA Track evaluation to test our algorithm. Therefore, we adopt the self-defined testing methods in which the computer generates the testing data randomly and automatically and verifies the data manually.

We construct a goods ontology, and define *VNLT= {what, where, how much, who}*.The elements in NCV come from the concepts of the pre-defined ontology, which can be extracted automatically by some algorithms. We use table 2 as LTMS-Lib for testing, and a testing plan is designed in table 3.

TABLE 3 TESTING PLAN

1. Questions sentences (named it **nls**) are generated at random and checked whether meet the following conditions: (1). There is only one word from VNLT in **nls**. (2). The **nls** has at least one word that comes from NCV.
2. Put the nls into algorithm 1 for processing, the algorithm return the result, namely, **sb**;
3. Check whether **nls** matches daily language specification and check **sb** whether correct or not by common sense.
4. If the number of the loop is up to 200 times, then algorithm halts;
5. Go to step 1 and continue.

In our testing plan:

The step 1 in table 3 prepares the data for executing algorithm 1.We have to present conditions in step 1 because it will increase the algorithm complexity when sentences contain too much words from NNL. Relying solely on this article cannot solve more complex issue. So, to decrease the complexity, we give the constraints for nls.

The step 2 gets the results from algorithm 1.

The step 3 checks whether nsl matches daily language specification (DLS) that means the sentence in nsl is recognized by person approximately. Such as, "*what are the nutritional characteristics of yogurt?*" matches DLS, but "*yogurt what the nutritional characteristics are?*" does not.

After checking DLS, people should focus on whether the answer from algorithm is correct. For the problem, "*what are the nutritional characteristics of yogurt?*" based on ontology and common sense, the correct answer should be "*high protein*".

TABLE 4 THE STATISTICS OF THE TESTING RESULTS FOR THE SENTENCES GENERATED RANDOMLY

Times	CSGR	CSMD	CSNMD	CCA	CAR
1	200	80	120	67	83.8%
2	200	86	114	69	80.2%
3	200	105	95	78	74.3%

CSGR - Count of sentences generated randomly
CSMD - Count of sentences matching DLS
CSNMD-Count of sentences not matching DLS
CCA-Count of correct answers
CAR-correct answer rate

In table 4, we generate 200 questions randomly each time and execute 200 times the algorithm 1 to get answers. After that, we check artificially the 200 answers whether they match DLS, only those sentences that match DLS whose correctness would be checked. We repeat the test three times. From the result of the test, less than one-third of the sentences match the DLS, in which about 78% of the answers are correct.

In fact, in this test, we only use a small ontology, a small LTMS-Lib and a small set named VNLT, we think that the result is satisfactory.

V. CONCLUSION

In this paper, we present an inquiring model based on ontology and natural language, called OntoQuery-NLP, to convert queries in natural language into formal languages the computer can recognized. The Ontology-NLP uses triples, as formal semantics, to express and map the natural language, which helps to use querying sentences in natural language and retrieving the data from ontology without training corpus. It is beneficial to develop the systems that use natural language as manipulation language and ontology as knowledge base. The system is easy for us to acquire knowledge in human-computer interactive mode. The experiment shows that the model and algorithm is effective.

This paper only describes the feasibility study of OntoQuery-NLP and further research needs to establish a large-scale ontology library and template library. In the future, we will verify the OntoQuery-NLP further by using WordNet [10] and improve the accuracy of the method.

REFERENCES

[1] N. F. Noy, M. Sintek, S. Decker, M. Crubézy, R. W. Fergerson, and M. A. Musen, "Creating semantic web contents with protege-2000," Intelligent Systems, IEEE, vol. 16, no. 2, pp. 60–71, 2001.

[2] D. Jurafsky, J. H. Martin, A. Kehler, K. Vander Linden, and N. Ward, Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, vol. 2. Prentice Hall New Jersey, 2000.

[3] A. M. Popescu, O. Etzioni, and H. Kautz, "Towards a theory of natural language interfaces to databases," in Proceedings of the 8th international conference on Intelligent user interfaces, 2003, pp. 327–327.

[4] V. Lopez, M. Pasin, and E. Motta, "Aqualog: An ontology-portable question answering system for the semantic web," The Semantic Web: Research and Applications, pp. 135–166, 2005.

[5] P. Cimiano and J. Völker, "Text2onto A Framework for Ontology Learning and Data-Driven Change Discovery," Natural Language Processing and Information Systems, pp. 257–271, 2005.

[6] G. Lame, "Using NLP techniques to identify legal ontology components: concepts and relations," Law and the Semantic Web, pp. 169–184, 2005.

[7] N. F. Noy, "Semantic integration: a survey of ontology-based approaches," ACM Sigmod Record, vol. 33, no. 4, pp. 65–70, 2004.

[8] N. Fuchs, U. Schwertel, and R. Schwitter, "Attempto Controlled English—not just another logic specification language," Logic-Based Program Synthesis and Transformation, pp. 1–20, 1999.

[9] E. Voorhees and D. K. Harman, TREC: Experiment and evaluation in information retrieval, vol. 63. MIT press Cambridge^ eMA MA, 2005.

[10] G. A. Miller, "WordNet: a lexical database for English," Communications of the ACM, vol. 38, no. 11, pp. 39–41, 1995.