

# Semantic Pattern for User-Interactive Question Answering

Tianyong Hao<sup>1</sup> Qingtian Zeng<sup>1,2</sup> Liu Wenyin<sup>1</sup>

<sup>1</sup>*Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Hong Kong, China*

<sup>2</sup>*Department of Computer Science and Technology, Shandong University of Science and Technology, Qingdao 266510, China*  
{tianyong, qtzeng, csluwy}@cityu.edu.hk

## Abstract

*A new semantic pattern is proposed in this paper, which can be used by users to post questions and answers in user-interactive question answering (QA) system. The necessary procedures of using semantic pattern in a QA system are also presented, which include question structure analysis, pattern matching, pattern generation, pattern classification and answer extraction. A user interface of using semantic pattern is also implemented in our QA system, which allows users to effectively post and answer questions. It gains good overall user satisfaction.*

## 1. Introduction

The interest is increasing very fast in the research of question answering (QA) recently [1]. Many open domain QA systems analyze the input question to determine the desired type of the answer [2]. However, among 65 participating systems in the TREC-10 Question Answering competition, the winning system used just one resource: a fairly extensive list of surface patterns [3]. With the development of natural language processing, more and more QA research works employ semantic technique to obtain more accurate answers.

Many online QA systems claim that they use pattern to tackle user's questions and extract answers. Deepak and Eduard propose a surface text pattern to extract answers [3]. Ion gives three different linguistic patterns to extract relevant information [4]. However, all of these patterns do not include semantic information and are therefore called "poor-knowledge approaches" [5]. Hence, they cannot extract precise answer or relevant information without semantically analyze questions and answers.

There are a few progresses in adding semantic representation into linguistic pattern to improve the accuracy of answers. Maximiliano proposes a type of se-

matic pattern [5], which uses EuroWordNet as a lexical database. However, this pattern cannot provide semantic information itself and cannot represent constraints of part of a sentence. Steffen and Michael propose a different kind of semantic pattern, which is used for communication between semantic web developers and for mapping and reusing different target languages [6]. However, it is not easy for question generation and representation, and furthermore, it is mainly designed for professional and therefore is difficult for end-users to use.

In our research, we have proposed a new kind of pattern with semantic description based on surface pattern. This kind of pattern allows QA systems to locate answers in the desire type and filter out irrelevant answers by matching the semantic part of question patterns. These semantic patterns can be used in both automatic and interactive QA systems. In order to fully utilize semantic patterns, associated research work including semantic structure analysis of question, pattern matching, pattern generation, pattern classification and answer extraction should also be done. These semantic patterns can be both obtained manually by engineers and users themselves or extracted automatically from questions in free text. We have also designed a user interface for users to interactively ask and answer questions. User can simply fill in the blank areas in the patterns to complete a question or answer, instead of writing all the question or answer content in the traditional way. All of these features can facilitate users to post and answer questions easily and rapidly, and thus increase the usability.

We have developed a user-interactive QA system—*CuteAid* as a platform for users to ask and answer questions using semantic patterns. The major interactions of using pattern during the QA procedure are listed below:

- 1) When a user wants to ask a question, he can select a category related to his question.

- 2) The user can then select an appropriate pattern for his question.
- 3) The user can fill in all the blank areas in the pattern before submitting the question.
- 4) The user can create a semantic pattern by using the semantic pattern generation guide (SPG-Guide) if he cannot find his desire pattern.
- 5) When a user answers a question, he can just fill in the blank areas of the associated answer pattern.

This paper is organized as follows: Section 2 introduces the formal definition for semantic pattern. Section 3 presents all necessary procedures of using semantic patterns, including structure analysis and pattern matching, generation and classification and answer extraction. Section 4 shows the user interface for using semantic pattern. Section 5 summarizes our present work and preview of our future work.

## 2. Definition of semantic pattern

Our Semantic pattern is defined based on surface pattern, in which semantic information is added for knowledge acquisition and machine understanding.

A semantic pattern is mainly composed by five components:

- **Question\_Target**: used to represent the target of question and the answer pattern. For example, "Entity/Product" means that the target of a question is a product; and the answer pattern is also "[Entity/Product]".
- **Question\_Type**: used to represent the type of question. For example, when, what and why.
- **Concept**: used to label meaningful nouns in the question. It is represented by a concept hierarchy to express its super class in our classification ontology. The content of this part can be filled in with an instance of this class in a real question.
- **Event**: used to represent something happening or any specific behavior. We also represent it using concept hierarchy. The content of this part can also be filled in with an instance of this class in a real question.
- **Constraint**: used to represent the constraints of concepts or events. It is useful in answer extraction and knowledge acquisition.

Among these five parts, Question\_Target, Question\_Type, Concept and Event are four essential components, while Constraint is optional and can be added if necessary. The formal definition for semantic pattern is given in Figure 1.

```

Semantic Pattern ::= (<Question_Target>, <Question_type>,
[Concept], [Event], {Constraint})
(1) <Question_Target>
<Question_Target> ::= <Target: Class1\Class2>
Where "<" and ">" are two labels and must occur
synchronously. "Class1\Class2" represents two level classes
of the question target, which satisfy the relation SubClass
(Class1, Class2).
(2) <Question_Type>
<Question_Type> ::= <Type: Question Type >
Where "<" and ">" are two labels and must occur
synchronously. "Question Type" represents the type of
question, such as "What", "Where", "When" and so on.
(3) [Concept]
[Concept] ::= [Concept1\Concept2]
Where "[" and "]" are two labels and must occur
synchronously. "Concept1\Concept2" represents the concept
relationship in the concept hierarchy, where they satisfy the
relation Subcategory (Concept1, Concept2).
(4) [Event]
[Event] ::= [Event1\Event2]
Where "[" and "]" are two labels and must occur
synchronously. "Event1\Event2" represents the concept
relationship in the event hierarchy, where they satisfy the
relation Subcategory (Event1, Event2).
(5) {Constraint}
{Constraint} ::= {O: Concept or Event; F: Facet}
Where "{", ":", "}" , "O" and "F" are five labels which must
occur in one constraint. "O" represents an object of constraint
and "F" represents the aspect of the constraint.

```

**Figure 1.** The formal definition of semantic pattern

According to the above definition of semantic pattern, we can define patterns for question representation and analysis. Figure 2 shows a simple example for semantic pattern. In which, answer pattern has the same value as question target.

```

Question: "What book did Rachel Carson write in 1962?"
Semantic Pattern: <Target: Entity\Product> <Type: What>
[Physical_Object\Product] did [Physical_Object\Human]
[Event\Action]?
Answer Pattern: [Entity\Product]
Constraint: {O: write F: Date=1962}

```

**Figure 2.** An example for semantic pattern

## 3. Workflow of using semantic patterns

Semantic pattern is used in several parts within CuteAid [17]. When a user posts a question in free text, the system analyzes the structure and keywords of the question, and selects a relevant pattern from the pattern base. If the pattern is found, the system uses its existing pattern; otherwise the system generates a suitable question pattern automatically. Alternatively, the user or administrator can create a pattern himself using the semantic pattern generation guide (SPG-Guide). The newly generated pattern can be classified based on the classification ontology for pattern management and utilization. Particularly, the generated pattern is useful

for matching with new questions' patterns. Once the pattern for the question is known, it can be used by the system to match with existing questions' patterns and then extract their answers and send them to the user automatically. The workflow of semantic pattern system is shown in Figure 3. It mainly includes three parts: (1) question structure analysis and pattern matching, (2) semantic pattern generation and classification and (3) Answer extraction.

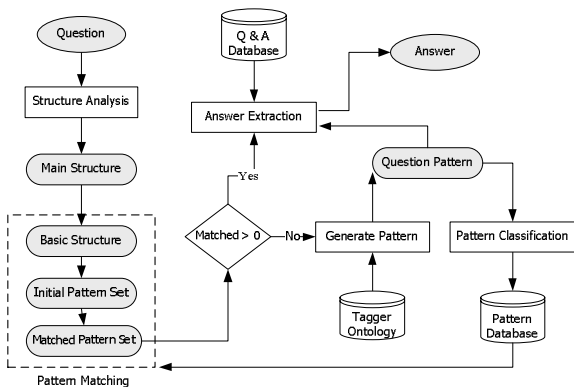


Figure 3. Workflow of using semantic patterns

### 3.1. Structure analysis and pattern matching

Given a free text question, the system can analyze and obtain the main structure of the question in three steps:

Step 1: Acquire the question type by a basic keyword search.

Step 2: Acquire the useful nouns and verbs by natural language processing.

Step 3: Generate the main structure of the question.

The system then calls the pattern matching procedure to find an appropriate question pattern for the main structure. Firstly, nouns and verbs in the main structure are replaced by 'string' (which is a symbol representing a character string) to form the basic structure, which includes the question type and the structure of the question. Secondly, the system returns some patterns by basic structure matching, which are referred to as the initial pattern set (IPS). For each noun or verb in the main structure, the system retrieves its super concepts from WordNet [18], which are used to match the suitable tagger in the Tagger Ontology (see Figure 4) so as to tag these concepts. The Tagger Ontology contains upper concept hierarchy of WordNet. For better understanding and using by users, it only includes two-level concepts. Matching is to build the relation between WordNet and the Tagger Ontology. The concepts matched will be filtered in the IPS to find the matched pattern set (MPS). If MPS is empty, the

system should generate a suitable semantic pattern (see pattern generation).

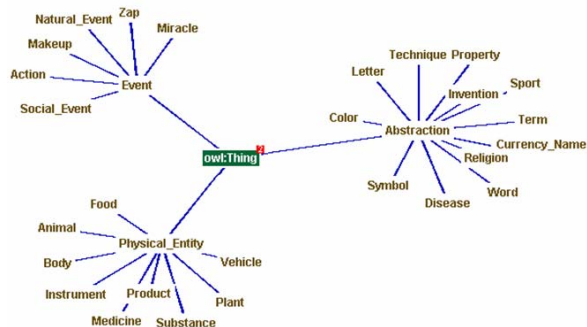


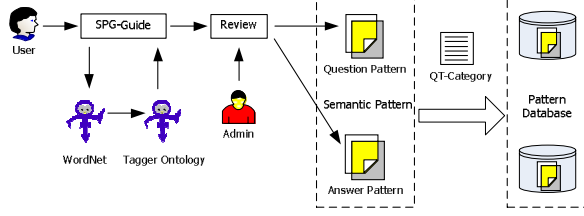
Figure 4. Visualization of Tagger Ontology

For example, when a user posts a question “What is the color of rose?”, the system first analyzes the question type from the essential word “What”, and then obtains two nouns “color” and “rose” by simple syntax-analysis. Finally, the system generates a main structure “<Type:What> is the [noun=color] of [noun=rose]?”, which is a pattern structure without semantic information.

Replacing the nouns in the main structure with strings, the system can obtain the basic structure of “<Type:What> is the [string] of [string]?”, which can be used to obtain IPS by querying this structure from the pattern base using keyword matching. If IPS is not empty, the system searches nouns of “color” and “rose” in WordNet to find their super concepts. For example, the super concept of “rose” in WordNet is “bush, woody plant, vascular plant, plant, organism, living thing, object, physical entity, entity”. The system then matches these concepts from the Tagger Ontology and found ‘plant, physical entity’ as the super concept of “rose”. Therefore, ‘[Physical\_EntityPlant]’ can be used to label the concept of rose. These labeled concepts in the pattern are used as keywords to filter IPS to MPS. MPS can then be used to extract answers from the QA database.

### 3.2. Pattern generation and classification

In our system, ordinary end users or domain experts can create semantic patterns for specific questions using the semantic pattern generation guide (SPG-Guide). The administrators of the system can review these patterns. If they are correct, the administrator can ask the system to classify them according to the question target category (QT-Category) and add them into the pattern database. The collaborative process of generating a pattern is shown in Figure 5.



**Figure 5.** Semantic pattern generation and classification by user or domain expert

Given a specific question, SPG-Guide analyzes the question to obtain its main structure (see Structure Analysis). Then, we can get MPS according to pattern matching. If MPS is empty, SPG-Guide needs to generate a new semantic pattern. The key step of SPG-Guide is matching between WordNet and Tagger Ontology.

The most important role of the Tagger Ontology is to tag nouns and verbs in the main structure. These tagged keywords contain semantic description and relationship of the hierarchical concepts in the ontology and can therefore be understood by machine easily. It is also useful for question or answering. Users can understand the content of the blanks by tips of the concept hierarchy.

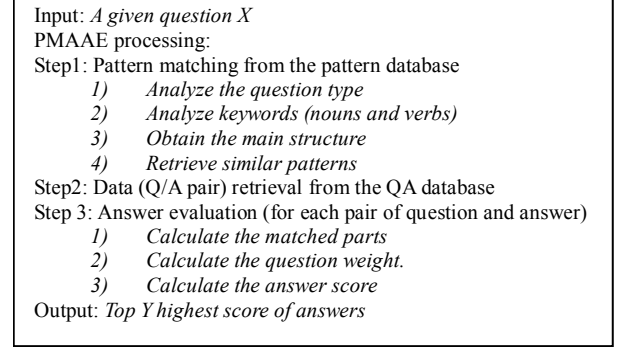
When a new pattern is created, the administrator has to check the pattern and revise it if it is not correct. Since many QA systems associate specific matching information (indicative words, surface word patterns, etc.) with their question targets [1], it is useful to create question targets for answer extraction. Hence, we design a question target category (QT-Category) to classify and manage different patterns, which mainly includes six categories (Abbreviation, Entity, Description, Human, Location and Numeric).

For the example mentioned above, if the system cannot find BPS, it will generate a question pattern automatically. From pattern matching, we can obtain the nouns of “color”, “rose”, and their labels “[Abstraction\Property]” and “[Physical\_Entity\Plant]”. Then, SPG-Guide requires the user to specify the question target (which is “Entity\Color”) manually from the QT-Category. Finally, the administrator can modify and add it into the pattern database according its QT-Category.

### 3.3. Answer extraction

We propose a new approach called Pattern Matching and Answer Evaluation (PMAAE) to extract answers from QA database based on semantic pattern matching. This method firstly finds questions and answers using similar patterns by pattern matching. The result containing questions and answers can be evaluated by an answer evaluation algorithm developed by

us. Finally, some best answers will be sent back to the user. The main steps of PMAAE are shown in Figure 6.



**Figure 6.** Main steps of answer extraction

The answer evaluation algorithm is mainly used to evaluate the answer score according to the weight of questions and the quality of answers. The higher the score, the better the answer. The quality of answer is calculated using the method of traditional bisecting K-means algorithm with new feature vectors [1] in our system. Given an answer set  $AS_1 = \{A_1, A_2 \dots\}$ , we can calculate the score for each  $A_i$  ( $1 \leq i \leq \text{Number\_of\_answers}$ ) using the following formula.

$$\text{Score}(A_i) = \text{Weight}(A_i) * \text{Quality}(A_i) \quad (1)$$

Where,  $\text{Weight}(A_i)$  is the weight of  $A_i$  and  $\text{Quality}(A_i)$  is the quality of  $A_i$ .  $\text{Weight}(A_i)$  can be calculated using the following formula:

$$\text{Weight}(A_i) = \text{Weight}(Q_i\_T) + \text{Weight}(Q_i\_Y) + \text{Weight}(Q_i\_C) \quad (2)$$

Where,  $\text{Weight}(Q_i\_T)$  means the weight of question target of  $Q_i$ ,  $\text{Weight}(Q_i\_Y)$  means the weight of question type of  $Q_i$ , and  $\text{Weight}(Q_i\_C)$  means weight of concept and event of  $Q_i$ . Assume the total number of all components of the pattern is  $M$ , i.e.,  $M = |T| + |Y| + |C|$ , the weight of each component is calculated as follows:

$$\text{Weight}(Q_i\_T) = \alpha * \text{Match}(T) * \frac{1}{|M|} \quad (0 \leq \alpha \leq 1) \quad (3)$$

$$\text{Weight}(Q_i\_Y) = \beta * \text{Match}(Y) * \frac{1}{|M|} \quad (0 \leq \beta \leq 1) \quad (4)$$

$$\text{Weight}(Q_i\_C) = \delta * \sum_{i=1}^{M-2} (\text{Match}(C_i) * \frac{1}{|M|}) \quad (1 \leq M, 0 \leq \delta \leq 1) \quad (5)$$

Where,  $\text{Match}(T)$  represents whether the question target is matched or not, which has two values, 0 means NOT matched, and 1 means matched,  $\alpha$  represents the importance of matching the question target.  $\text{Match}(Y)$  and  $\text{Match}(C)$ ,  $\beta$  and  $\delta$  are similarly defined.

For example, given the question “What is the color of rose?” Its question pattern is “<Target:Entity\Color> <Type:What> is the [Abstraction\Property] of [Physical\_Entity\Plant]?”. We can acquire all historical questions and answers using this pattern. For each pair of question and corresponding answer, the system calculates its weight to obtain the score of the answer. The user can obtain the best answers by ranking the scores of all answers.

For a question “What is the shape of rose?” the number of concepts in the question pattern is 2. Hence,  $M = 1+1 +2 =4$ . Assume  $\alpha=0.77$ ,  $\beta =0.36$ ,  $\delta =0.84$  and the original quality of answer is 0.7, we can obtain the following:

$$\begin{aligned} \text{Weight}(Q_i\_T) &= 0.77 * 0 = 0 \\ \text{Weight}(Q_i\_Y) &= 0.36 / 4 = 0.09 \\ \text{Weight}(Q_i\_C) &= 0.84 * (0+1 / 4) = 0.21 \\ \text{Weight}(A_i) &= 0+0.09+0.21 = 0.3 \\ \text{Score}(A_i) &= 0.3 * 0.7 = 0.21 \end{aligned}$$

#### 4. User interface for user interactive QA using semantic patterns

A user interface has been developed for users to use semantic patterns in CuteAid, in which a user can ask and answer questions based on patterns. Moreover, users can define question or answer semantic patterns by themselves. The interface is easy to use and can improve efficiency of asking and answering.

When asking a question, a user can select a question pattern and fill in all the blanks to finish it. There are three main steps:

##### Step 1: Select category

The system has categories about question targets, which include two level concepts. The user has to select the first-level category and then its sub-category. For example, when the user wants to ask a question about the color of rose, he should select the category of “Entity” and the subcategory of “Color”.

##### Step 2: Select pattern

After category selection, the system shows all of patterns in this category. The user has to select a pattern relevant to his question. For example, when the question is “What is color of rose?” he can select the pattern of “What is [Abstraction\Property] of [Physical\_Object\Plant]?”

##### Step 3: Fill in pattern

After the user chooses the pattern, the system can automatically form the question pattern filling in area according to the structure of the pattern, and the user only needs to fill in the blank parts of the pattern instead of writing the full question. For the question mentioned above, the user just needs to fill in two blanks, with the tips of “Abstraction\Property” and

“Physical\_object\Plant”, respectively, as shown in Figure 7.

Figure 7. UI of using semantic pattern to ask

When the user wants to reply a question, the system creates the blank areas according to the answer pattern. The user can just fill in the blank and submit the answer (see Figure 8). Therefore, it is very convenient to answer a question by using a semantic pattern.

Figure 8. Answer a question by using semantic pattern

If the pattern required by the user is not included in this system, he can use SPG-Guide to create pattern himself. SPG-Guide analyzes the user’s question to obtain main structure and then generate a semantic pattern using the method of pattern generation after the user sets the question target. The user interface of SPG-Guide is shown in Figure 9.

Figure 9. UI of the semantic pattern generation guide

#### 5. Conclusion and future work

In order to obtain high accuracy answers, this paper proposes a new kind of semantic pattern with a formal definition. It includes semantic descriptions of different

parts, which can be used to analyze semantic relationship between the question/answer and the pattern. We also illustrate how to use semantic patterns in a QA system, including structure analysis, pattern matching, pattern generation, pattern classification and answer extraction. We also apply it in a real QA system—CuteAid, in which we design the user interface to facilitate the usage of semantic patterns. The design allows end users to post and answer questions effectively. In our study of user satisfaction by a survey, it yields good overall user satisfaction.

However, the relationships between different parts of “concept” and “event” are not yet specified in this version. Therefore, it is not easy to represent and acquire knowledge from the QA database though it is convenient for end users. We need to work more on analysis of semantic relationships in patterns and knowledge acquisition. Though we have done some experiments such as structure analysis and automatic pattern generation and the result shows that 93.6% questions can be generated and 78.5% is accurate in a set of 5000 questions [16], more experiments are needed to test its advantage on answer extraction and other tasks.

## 6. Acknowledgement

The work described in this paper was fully supported by a grant from City University of Hong Kong (Project No. 7001462) and the China Semantic Grid Research Plan (National Grand Fundamental Research 973 Program, Project No. 2003CB317002).

## 7. References

- [1] N.F. Xie, M. Feng, and W.Y. Liu, “Answer Clustering and Fusion in a User-Interactive QA System”, *Technical Report*, City University of Hong Kong, 2005.
- [2] H. Eduard, H. Uif, and R. Deepak, “A Question/Answer Typology with Surface Text Patterns”, *Proceedings of the 40th ACL conference*, Philadelphia, 2002.
- [3] R. Deepak, and H. Eduard, “Learning Surface Text Patterns for a Question Answering System”, *Proceedings of the 40th ACL Conference*, Philadelphia, 2002.
- [4] M. Ion, “Extraction Patterns for Information Extraction Tasks: a Survey”, *Workshop on Machine Learning for Information Extraction*, Orlando, 1999.
- [5] S.N. Maximiliano, S. Armando, and P. Manuel, “Semantic Pattern Learning Through Maximum Entropy-based WSD Technique”, *Proceedings of 5th Workshop on Computational Language Learning*, Toulouse, 2001.
- [6] S. Steffen, and E. Michael, “Engineering Ontologies Using Semantic Patterns”, *Proceedings of the IJCAI-01 Workshop on E-Business & the Intelligent Web*, Seattle, 2001.
- [7] H. Zhuge, *The Knowledge Grid*, World Scientific Publishing Co., Singapore, 2004.
- [8] H. Zhuge, China’s E-Science Knowledge Grid Environment, *IEEE Intelligent Systems*, 19 (1) (2004) 13-17.
- [9] E. Sneider, “Automated Question Answering: Pattern-Based Approach”, *Ph.D. thesis*, Department of Computer and Systems Sciences, Stockholm University and the Royal Institute of Technology, Sweden, 2002.
- [10] H. Uif, “Parsing and Question Classification for Question Answering”, *Proceedings of the Workshop on Open-Domain Question Answering at the ACL-01 Conference*, Toulouse, 2001.
- [11] M. Wu, X.Y. Zheng, and D. Michelle, “Question Answering By Pattern Matching, Web-Proofing, Semantic Form Proofing”, *The 12th Text Retrieval Conference*, Washington, 2003.
- [12] C. Chad, X. Li, and M. Paul, “Question-Answering via Enhanced Understanding of Questions”, *Proceedings of the 11th Text Retrieval Conference*, Gaithersburg, 2002.
- [13] R. Dmitri, and R. Jose, “Web Question Answering Through Automatically Learned Patterns”, *Proceedings of the Joint Conference on Digital Libraries*, Tucson, 2004.
- [14] N. Srin, H. Sanda, “Question Answering Based on Semantic Structures”, *International Conference on Computational Linguistics*, Geneva, 2004.
- [15] S. Eriks, “Automated Question Answering Using Question Templates that Cover the Conceptual Model of the Database”, *Proceedings of the 6th International Conference on Applications of Natural Language to Information Systems*, Stockholm, 2002.
- [16] T.Y. Hao and W.Y. Liu, “Semantic pattern: definition and generation”, *Technique report*, City University of Hong Kong, 2006
- [17] CuteAid, <http://www.cs.cityu.edu.hk/QA/>
- [18] WordNet, <http://wordnet.princeton.edu/>