Pertanika J. Sci. & Technol. 21 (1): 169 - 182 (2013)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Toward Automatic Semantic Annotating and Pattern Mining for Domain Knowledge Acquisition

Tianyong Hao[1]* and Yingying Qu[2]

[1]*School of Computer Science and Engineering, University of New South Wales, Australia*

[2]*Faculty of the Built Environment, University of New South Wales, Australia*

ABSTRACT

Due to the high complexity of natural language, acquisition of high quality knowledge for the purpose of fine-grained data processing still mainly relies on manual labour at present, which is extremely laborious and time consuming. In this paper, a new automatic approach using semantic annotating and pattern mining is proposed to assist engineers for domain knowledge acquisition. This approach uses Minipar to label sentences processed from domain texts. Based on the dependency relations, structural patterns are extracted and semantic bank is applied to annotate and represent concepts with semantic labels considering sentence contexts. The approach further learns and assigns relations to previously extracted concepts by pattern matching. The involved concepts and semantic labels with learned relations together, as extracted knowledge, enrich domain knowledge base. Preliminary experiments on Yahoo! Data in "heart diseases" category showed that the proposed approach is feasible for automatic domain knowledge acquisition.

Keywords: Knowledge acquisition, semantic annotation, semantic bank, structural pattern, transformation rule

INTRODUCTION

With the mass production of digital information over the web, extracting and snatching knowledge with effectiveness, efficiency and accuracy have become an urgent and challenging tasks (Wang *et al.*, 2006). In recent years, knowledge acquisition from texts has become one of hot research fields in Artificial Intelligence (Tanaka & Jatowt, 2010; Tanaka & Jatowt, 2010; Deng & Han, 2012; Carayannis, 2012; Fan *et al.*, 2012). The task of knowledge acquisition is to extract knowledge for an information system or expert system by setting up sound, perfect, effective knowledge base.

Article history:
Received: 31 March 2012
Accepted: 31 August 2012

E-mail addresses:
haotianyong@gmail.com (Tianyong Hao),
yingyinqu2@gmail.com (Yingying Qu)
*Corresponding Author

ISSN: 0128-7680

In the past several decades, a number of these methods were followed with Cognitive Science and other disciplines.

Due to the high complexity of natural language, i.e. with the huge amount of concepts and relations in free texts being too complicated to be formalized by automatic methods, the acquisition of high quality knowledge for the purpose of fine-grained data processing is still mainly relying on the costly manual labour at present.

In the field of medical informatics, there is a high demand for domain knowledge within a formally specified framework and knowledge based is getting more and more accepted (Campbell *et al.*, 1994; Hull & Gomez, 1999; Cao, 2001; Dawoud *et al.*, 2012). It requires medical knowledge to be expressed in a sound and consistent manner. Knowledge engineering in an ontologically coherent (sub) domain makes use of existing knowledge compilations, such as ICD-10, SNOMED, MESH, UMLS, as much as possible. Due to the insufficiency of formal specification and levels of granularity, only a limited amount of the knowledge can be reused even if they are judged transferable in principle (Carenini & Moore, 1993; Firdaus *et al.*, 2012).

Most knowledge is contained in a large collection of unstructured textual documents. Ordinary approaches are to acquire knowledge from the documents manually and then formalize the knowledge at the conceptual level. However, this acquisition procedure mainly relies on a large number of knowledge engineers' manual efforts, which is rather labour intensive, time consuming and troublesome. Moreover, the huge amounts of concepts and relations in domain texts are extremely complicated to be formalized by knowledge engineers. Thus, automated or semi-automated knowledge acquisition techniques are urgently required (Wang *et al.*, 2006; Firdaus *et al.*, 2012).

In this paper, a novel automatic method is proposed for domain knowledge acquisition from domain text corpus by semantic annotating and pattern mining. After pre-processing the corpus by splitting into sentences and removing the noise data, the proposed method analyzes the corpus by using Minipar to label sentences. The nouns and noun phrases, regarded as the concepts, are extracted along with structural patterns. In order to improve matching and learning efficiency, frequent patterns are mined based on these structural patterns using a sequence-based Apriori algorithm. The concepts are then annotated with semantic labels by WordNet and a further defined semantic bank containing unit annotations and context annotations.

With regard to training on annotated data, the method can generate structural patterns to match with a group of frequent patterns previously extracted. The concepts and their annotated relations in each pattern can be applied into the sentences with the same matched frequent pattern to derive more concepts with similar relations, or to learn more relations with the same concepts. The automatically extracted concepts, semantic labels and relations can thus dramatically enrich domain knowledge bases.

We used Yahoo! Answers (2012) corpus as of 10/25/2007 as an experimental dataset. It contained 4,483,032 questions and their correct answers, which are reliable domain text sources as the answers were rated by human users. Furthermore, the dataset contained a small amount of metadata such as best answers on human selection. Finally, 19,622 questions were extracted with their best answers in the "heart diseases" category as the domain corpus. By proposing a number of knowledge transformation rules, the preliminary experiments showed that the annotations could be transformed to standard knowledge representation formats such as OWL

accurately and stably. Therefore, the method is feasible for domain knowledge acquisition aiming at knowledge sharing.

The rest of this paper is organized as follows: Section 2 introduces the proposed framework of automatic knowledge acquisition. Section 3 presents sentence labelling and pattern mining. Section 4 describes semantic annotating of the concepts and relation learning by pattern matching. In Section 5, preliminary experiments with knowledge transformation rules are discussed, and Section 6 summarizes this paper and discusses future work.

THE FRAMEWORK OF DOMAIN KNOWLEDGE ACQUISITION

Admittedly, human-based knowledge acquisition can ensure the high quality of extracted knowledge. However, it is a long-term laborious work with high cost. Thus, automatic acquisition methods are more preferable, especially for quick acquisition demand. In this paper, an automatic method was proposed to extract domain knowledge based on semantic annotating and frequent pattern mining.

Through this method, domain corpus was firstly pre-processed to separate the text paragraphs into sentences and to remove noisy data such as Mathematics formulas. Sentences were then analyzed and labelled by Minipar to extract nouns, noun phrases and structural patterns. By using the sequence-based Apriori algorithm, the frequent patterns were mined from those extracted structural patterns to improve learning efficiency. Regarding those nouns or noun phrases as concepts, these concepts were then annotated by WordNet and a semantic bank which contained unit annotations and context annotations, as a semantic label annotation technique. After that, the patterns with their conceptual relations, learned from annotated training resources, were matched with the mined frequent patterns. According to the matched parts, the corresponding relations could be applied to the concepts associated with the mined frequent patterns. With defined knowledge transformation rules, these concepts and relations can be further transformed into standard knowledge representation formats such as OWL thus to enrich domain knowledge bases. The related framework is shown Fig.1.

The automatic knowledge acquisition method contains seven main steps, which are shown in the fig.1:

Step 1: Pre-processing a domain corpus by formatting, removing noisy data, and splitting text paragraphs into sentences;

Step 2: Labelling all sentences to extract all nouns, noun phrases, and structural patterns;

Step 3: Mining frequent patterns from the structural patterns using a sequence-based frequent pattern mining algorithm;

Step 4: Annotating all the concepts with semantic labels based on WordNet and a semantic bank-based annotation method;

Step 5: Training annotated resources to match with the frequent patterns;

Step 6: Assigning relations extracted from matched patterns to the previously extracted concepts;

Step 7: Presenting all the concepts and relations using OWL or RDF, aided by knowledge transformation rules to enrich domain knowledge bases.

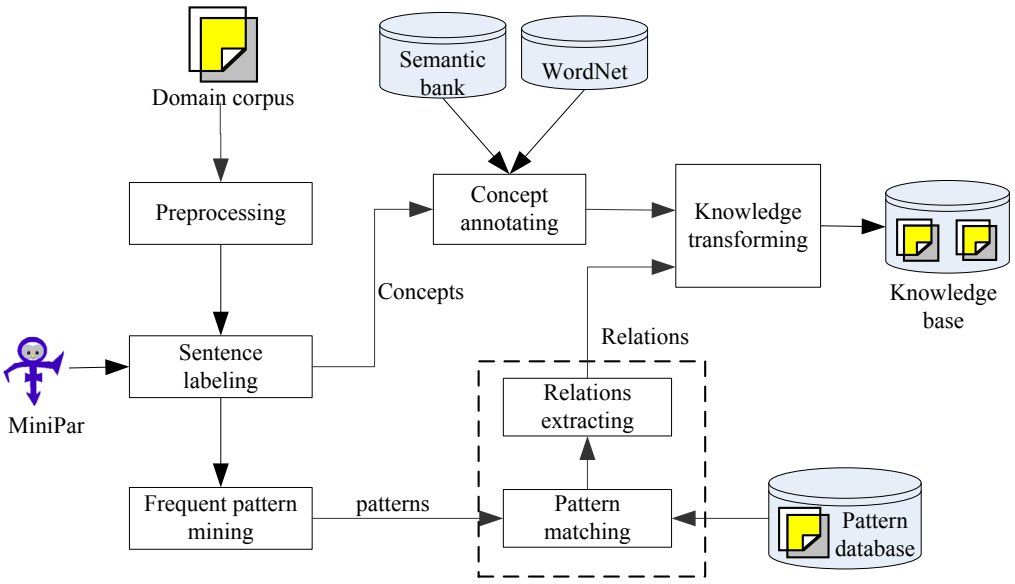

Fig.1: A framework of the automatic domain knowledge acquisition

With the previous processing steps, the framework can discover concepts with their relations, which are then transformed to be formal knowledge representations. Furthermore, this framework can be extended to more corpora or web documents to extract a large amount of knowledge automatically.

SENTENCE LABELING AND FREQUENT PATTERN MINING

The pre-processed sentences from domain corpus are used for sentence structure labelling and frequent structural pattern mining. A pattern is defined as the core structure of a sentence, which can then be extended as the generalization of similar sentences that shares the similar structure.

Sentence Labeling

There are many syntax analysis tools, such as TreeTagger (2012) and OpenNLP (2012), to label sentences by identifying nouns and verbs. However, most of these tools parse sentences by part-of-speech tagging without considering the relations between the objects in the sentences, making it difficult to find the core structures. Dependency Grammar (2012) is a class of syntactic theories developed by Lucien Tesnière. Sentence structure is determined by the relations between a word (a head) and its dependents, which are distinct from phrase structure grammars. The dependency relationship in this model is an asymmetric relationship between a word called head (governor) and another one called modifier. This kind of relationship can be used to analyze the dependency thus to acquire the main structure and nouns of a sentence effectively.

In this study, Minpar was used as a dependency parser tool to analyze sentence relationship in the paper. Minipar is a principle-based broad-coverage parser for the English language (Berwick *et al.*, 1991). Like Principar, Minipar represents its grammar as a network where nodes represent grammatical categories and links represent types of syntactic (dependency) relationships. The output of Minipar is a dependency tree. The links in the diagram represent dependency relationships. The direction of a link is from the head to the modifier in the relationship. Labels associated with the links represent the types of dependency relations. Furthermore, according to the evaluation with the SUSANNE corpus, Minpar achieves about 88% precision and 80% recall, in relation to dependency relationships (2012). The reliable performance ensures the quality of sentence labelling work.

By using Minipar, the sentences in corpus can be labelled with part-of-speech and dependency relations. Therefore, the core sentence structure, which includes nouns, verbs and other meaningful terms, could be extracted. For example, the sentence "what are the risks of plugging a hole in the heart?" was analyzed by using Minipar and the result is shown in Table 1 below.

TABLE 1: Minipar output on the example sentence

E1	(()	fin	C	*	*)*	
1	(what	~	N	E1	*whn*	(gov fin))
2	(are	be	VBE	E1	*i*	(gov fin))
E4	(()	what	N	4	*subj*	(gov risk)
3	(the	~	Det	4	*det*	(gov risk))
4	(risks	risk	N	2	*pred*	(gov be))
5	(of	~	Prep	4	*mod*	(gov risk))
E0	(()	vpsc	C	5	*pcomp-c*	(gov of))
E2	(()	~	N	E0	*s*	(gov vpsc))
6	(plugging	plug	V	E0	*i*	(gov vpsc))
E5	(()	~	N	6	*subj*	(gov plug)
7	(a	~	Det	8	*det*	(gov hole))
8	(hole	~	N	6	*obj*	(gov plug))
9	(in	~	Prep	8	*mod*	(gov hole))
10	(the	~	Det	11	*det*	(gov heart))
11	(heart	~	N	9	*pcomp-n*	(gov in))
12	(?	~	U	*	*punc)*	

In order to facilitate high speed pattern extraction, the output of Minipar is firstly converted into XML format. The nodes in XML are aligned with hierarchical structure, which can dramatically speed up pattern extraction. The XML format of the example is shown in Fig.2.

After the analysis, a group of nouns such as "risk", "hole" and "heart", as well as noun phrases, could be obtained. With the tagged relations, "risk" has the relation of "gov" to "be", "hole" has the relation of "gov" to "plug", and "heart" has the relation of "gov" to "in". Since "gov" means the actual word that it refers to, the structural patterns can be acquired level by

```
<?xml version="1.0" encoding="utf-8" ?>
- <node label="E3" category="U">
  - <node label="E1" category="C" word="" root="fin">
      <node label="1" category="N" word="what" root="what" relation="whn" />
    - <node label="2" category="VBE" word="are" root="be" relation="i">
      - <node label="4" category="N" word="risks" root="risk" relation="pred">
          <node label="E4" category="N" word="" root="what" relation="subj" />
          <node label="3" category="Det" word="the" root="the" relation="det" />
        - <node label="5" category="Prep" word="of" root="of" relation="mod">
          - <node label="E0" category="C" word="" root="vpsc" relation="pcomp-c">
              <node label="E2" category="N" word="" root="~" relation="s" />
            - <node label="6" category="V" word="plugging" root="plug" relation="i">
                <node label="E5" category="N" word="" root="~" relation="subj" />
              - <node label="8" category="N" word="hole" root="hole" relation="obj">
                  <node label="7" category="Det" word="a" root="a" relation="det" />
                - <node label="9" category="Prep" word="in" root="in" relation="mod">
                  - <node label="11" category="N" word="heart" root="heart" relation="pcomp-n">
                      <node label="10" category="Det" word="the" root="the" relation="det" />
                    </node>
                  </node>
                </node>
              </node>
            </node>
          </node>
        </node>
      </node>
    </node>
    <node label="12" category="U" word="?" root="?" relation="punc" />
  </node>
```

Fig.2: The Minipar output of the example in XML format

level. In each level, incremental part, which is the content extended compared with the previous pattern, is judged if it contains a noun or noun phrase. Only the patterns that fulfil constraints are regarded as qualified structural patterns. On the same example, all the extracted structural patterns and their related concepts are shown in Table 2.

TABLE 2: The extracted patterns and their related concepts on the same example

Structural patterns	**Concepts**
what(*gov* fin) be(*gov* fin) [*N1*](*gov* be)	risk
what(*gov* fin) be(*gov* fin) [*N1*](*gov* be) of(*gov N1*) plug(*gov* vpsc) [*N2*](*gov* plug)	risk; hole
what(*gov* fin) be(*gov* fin) [*N1*](*gov* be) of(*gov N1*) plug(*gov* vpsc) [*N2*](*gov* plug) in(*gov N2*) [*N3*](gov in)	risk; hole; heart

Frequent Pattern Mining

Since a single structural pattern may be specific, to improve the performance of matching and learning, frequent patterns are more appropriate to be used for efficient knowledge acquisition. Apriori, as a classic association rule mining method, is applied to mine possible frequent patterns. It is an algorithm for learning association rules and was designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation).

Given a set of item sets (for instance, sets of retail transactions, each listing individual items purchased), Apriori algorithm attempts to find subsets which are common to at least a

minimum number *C* of the item sets. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length *k* from the item sets of length *k*-1. Then, it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent *k*-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates. In this paper, threshold *support* for a pattern candidate *p* is defined as the number of pattern candidates that contain *p*.

In frequent pattern learning, those candidates whose occurrences are lower the *support* threshold are removed. Furthermore, two corollaries are applied to reduce calculation complexity. Let *X*, *Y*, *Z* be any three pattern candidates, and observe that if $X \subseteq Y$ and $Y \subseteq Z$, then $support(X) \geq support(Y) \geq support(Z)$, which leads to the following two corollaries: 1) If *Z* is frequent, then any subset *X* or *Y* is also frequent; 2) If *X* is not frequent, then any superset *Y* or *Z* cannot be frequent.

Regarding a structural pattern as an item set, a pattern can be split into several items thus can be sent to Apriori for mining. However, the sequences, as order information between items, need be fully considered. After sequence-based Apriori algorithm, the patterns whose recorded occurrence values larger than a threshold are mined as frequent patterns, which are further used in the next section.

SEMANTIC ANNOTATING AND RELATIONS LEARNING

Semantic Annotating

There are a large amount of nouns and noun phrases acquired from corpus labelling. Regarding them as concepts here, it is an essential step to annotate these concepts for semantic representation. In order to annotate them, semantic labels were used as annotation labels based on WordNet (2012), which provides a large database of English lexical items available online.

A semantic label here is defined as [*Concept1*] \ [*Concept2*], where these two concepts *(Concept1* and *Concept2)* have a relation with *Sub*(*Concept2*, *Concept1*), which means *Concept2* is the sub concept of *Concept1*. The purpose is to assign a label with a super concept label as constraint to reduce potential ambiguities. In this semantic label definition, concepts can be retrieved and labelled using WordNet. For instance, in the following sentence "What is the exact relationship between pulse rate and systolic pressure?", "pulse rate" have two senses, and thus the super concepts are: 1) "vital sign", "sign", "evidence", "information", and so on; and 2) "rate", "magnitude relation", "relation", and so on. Hence, the semantic labels of the "pulse rate" are tagged as "[sign\ vital sign]" and "[magnitude relation \rate]" finally.

The annotated concepts with semantic labels are important sources to enrich or even construct the basis of domain knowledge base. To improve annotation efficiency and label concepts in a complex context, these annotated concepts are represented in the form of semantic bank, which can be transferred into knowledge format directly as described in the following section. The idea of semantic bank is from one of our research work (Hao *et al.*, 2009). The

semantic bank is a kind of database to store concepts, corresponding semantic labels, and context annotation data. It consists of two parts: 1) unit annotation; and 2) context annotation.

The unit annotation part contains single concepts with their semantic labels. For example, the concept "pulse rate" is tagged as "[sign\ vital sign]" and "[magnitude relation \rate]". The format of the unit concept is represented as follows:

[*Concept***] HAVING [***Semantic labels***]**

The context annotation is a type of representation of unit annotation and occurrences in a specific context. For example, the concept "pulse rate" has two semantic labels. However, in the example sentence, the semantic label of "systolic pressure" was identified as "[vital sign\ blood pressure]". In this context, the "pulse rate" was assigned with the semantic label "[sign\ vital sign]". The format of context annotation is represented as follows:

([*Term1***] HAVING [***Semantic labels 1***] WITH [***Term2***] HAVING**

[*Semantic labels 2***]):** *Occurrence*

The *Semantic labels* in [*Semantic labels*] can be modified and the *Occurrence* can be increased and updated when there are new semantic labels used for the current concepts. An example of the context annotation is shown in Table 3, in which "pulse rate" has two different labels in different sentence contexts. From the context annotation, the probability model can be used to calculate the possible semantic senses in certain context.

TABLE 3: An example of context annotation

Concept$_1$ with label	*Concept$_2$ with label*	*Occurrence*
pulse rate **HAVING** [sign\vital sign]	systolic pressure **HAVING** [vital sign\blood pressure]	1
pulse rate **HAVING** [magnitude relation\rate]	valve **HAVING** [body part\structure]	2

With the semantic bank, the concordance of concepts and their semantic labels are recorded with occurrences. When the data are large enough, they can be used to determine the semantic labels for a given new sentence. A naïve Bayesian formulation was used with the hypothesis that each word in a sentence was thought to be independently distributed to determine the semantic label. The probability of the semantic label for each concept can be calculated by using the following equation:

$$P_{c_i}(label_n \mid \prod_{j \neq i} c_j) = \frac{P_{c_i}(label_n) \prod_{j \neq i} P_{c_i}(c_j \mid label_n)}{P_{c_i}(\prod_{j \neq i} c_j)} \quad (2)$$

where the left part denotes that the probability of concept c_i is distinguished by the semantic label $label_n$ on the condition that c_i co-occurs with c_j. $P_{c_i}(label_n)$ is the prior probability of c_i, which is distinguished by semantic label $label_n$. The denominator means the probability of c_i co-occurs with each c_j and $P_{c_i}(c_j \mid label_n)$ gives the probability of appearing c_j when c_i is labeled by $label_n$. Noting that the denominator remains constant when c_i and $label_n$ are fixed, as a result, we only need to calculate the product of each $P_{c_i}(c_j \mid label_n)$ and $P_{c_i}(label_n)$ to determine the semantic label of c_i using the following equation:

$$label = Max \arg_n [P_{c_i}(label_n) \prod_{j \neq i} P_{c_i}(c_j \mid label_n)] \quad (3)$$

The concept annotation method based on the semantic bank is further implemented in our system, which can analyze a sentence and annotate concepts with semantic labels automatically.

Relation Learning

Since the acquired frequent patterns have a larger coverage of sentences than normal structural patterns, and they can be used to extract more concepts having similar relationship theoretically. With annotated and validated training data, our method can learn the patterns and their relations so as to be applied into the extraction of more concepts extracted from the original domain corpus. The detailed method is as follows:

Given an annotated sentence with knowledge representation as a training sentence s_t, the method first analyzes the sentence by Minipar to acquire structural patterns p_t and corresponding relation r_t. After that, the pattern is matched with each pattern in the frequent pattern set. If there is a pattern p_f matched, the r_t is then applied to the relation representation of all nouns extracted from p_f. The detailed equation for this purpose is shown in the following:

$$MScore(p_t) = \frac{2 \times \left| MS_{p_t} \cap MS_{p_f} \right|}{\left| MS_{p_t} \right| + \left| MS_{p_f} \right|} \quad (4)$$

p_t and p_f are firstly split into items MS_{p_t} and MS_{p_f} and these items are then used to match with each other. The matching score, as *MScore* for the pattern p_t, is calculated based on the counting matched items with all the items.

From Equation (4), the frequent pattern matched best to the current structural pattern was obtained compared with a matching threshold. If there is no frequent pattern fulfils the current structural pattern still can be applied to pattern learning though its question matching coverage is less than the frequent patterns.

For example, there is training sentence "the signs of a heart attack are chest discomfort, discomfort in other areas of the upper body, shortness of breath, and other symptoms". Its annotation is shown in the following:

<heart attack>
<Has_attribute_Signs rdf:resource="# *chest discomfort, discomfort in other areas of the upper body, shortness of breath, and other symptoms* "/>
</ heart attack>

The structural pattern of the example sentence is "the(*gov* [*N1*]) [*N1*](*gov* be) of(*gov N1*) [*N2*](gov *N1*) be(*gov* fin) [*N3*](*gov* be)". Therefore, the pattern can be matched with the whole pattern set. Suppose there is a frequent pattern matched and its covered sentences include "The benefits of red wine are reducing coronary heart diseases, maintains the immune system, polyphenols, resveratrol, flavonoids, anti-bacterial activity, anti-stress". The relation is then applied in the sentence, and thus, a relation is learned automatically, as follows:

<red wine>
<Has_attribute_benefits rdf:resource="# *reducing coronary heart diseases, maintains the immune system, polyphenols, resveratrol, flavonoids, anti-bacterial activity, anti-stress* "/>
</red wine>

By this way, all the possible relations related to the concepts in our domain sentences could be acquired, especially using the previous semantic bank. Therefore, the semantic annotations of concepts with their relations can be used to enrich or built a domain specific knowledge base.

PRELIMINARY EXPERIMENTS

In this study, the automatic knowledge acquisition method was implemented in a Windows-form application. The domain was selected as "heart diseases" from Yahoo! Answers (2012) as of 10/25/2007, which was used as corpus since it contained huge amount of data (4,483,032 items) and also tagged with rated answers. These human-based answers are favourable domain text source. Furthermore, the dataset contained a large amount of metadata, i.e., which answer was selected as the best answer, as well as the category and sub-category that were assigned to this question. Therefore, the questions with their best answers were selected in the "Heart Diseases" category as our domain dataset, containing 19,622 items in total. The example data in the domain are shown in Fig.3.

After pre-processing the corpus into sentences, Minipar was applied to automatically identify the concepts and dependency relations. In this experiment, the focus was mainly placed on recognizing nouns and noun phrases, in which the latter has higher priority since the noun phrases can convey more specific semantic information. After that, the system extracted all the possible structural patterns. With the sequence-based Apriori algorithm, the frequent patterns are mined and extracted. WordNet is further applied to annotate the nouns and nouns phrases as the format of unit annotation and context annotation. In order to annotate a sentence more accurately, all the semantic labels and items were recorded with their occurrences into a semantic bank. These statistical data were further used to assist annotation on specific sentences.

```
12993  What is the name of the blood vessels which carry blood to the heart?
12994  I have had a stroke,diabetis and now parkinsons. Which one is keeping
       me awake at nights?
12995  How can you cure SVT ( supra-ventricular tachycardia) ?
12996  What is the Medical Term for an Enlarged Heart?
12997  What is a septal infarct and what is the maximum period of getting
       cardiac enzymes for infarct diagnosis?
12998  Has any one ever heard of Athlete's heart? I was told by a doctor I
       have it but offered no treatment or any a
12999  What do ththe meaning of Ballooning?When it is needed to do ballooning
       in our heart?
13000  What could cause extreme pounding of heart while at sleep?
```

Fig. 3: Question Examples in "Heart Diseases" category

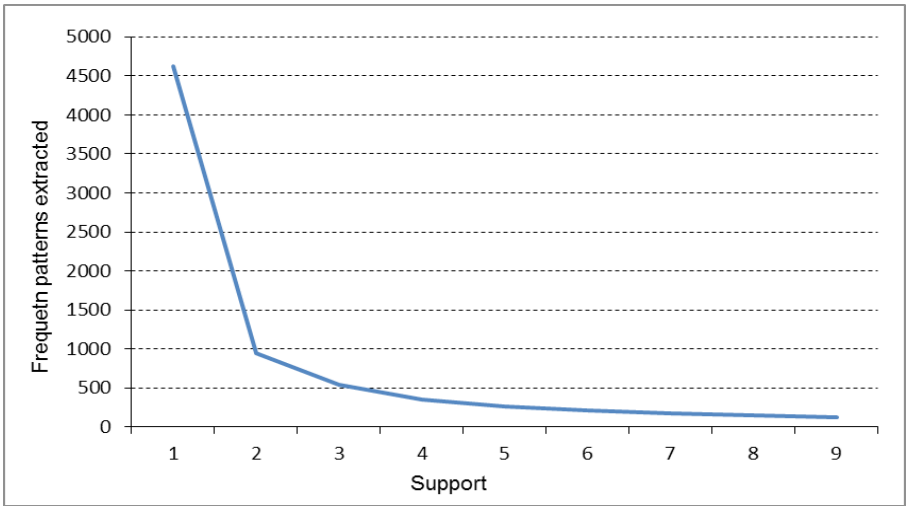

Fig.4: The extracted frequent patterns by different support values

From a total of 19,622 question items, 10.3 words were allocated for each question on average. Using the proposed pattern extraction method, 56,118 structural patterns were finally extracted in total, indicating that 2.86 patterns for each sentence on average. With different values of the *support* threshold, the number of the extracted frequent patterns changed dramatically, and the distribution is shown in Fig. 4.

Since it was difficult to find annotated knowledge resources in this domain as the training data, 100 sentences were randomly selected from the domain dataset and they were also manually annotated for this preliminary experiment. On these training data, the structural patterns and concepts were firstly extracted automatically by sentence labelling. After that, all the concepts and related semantic labels that are automatically annotated by the semantic bank and WordNet were randomly selected. The relations in the training data were also annotated and applied into the sentences where the matched patterns had been generated from. Based on the training data, there were a total of 272 structural patterns extracted. With regard to the pattern matching on all the other sentences in the whole dataset, 1,929 new concepts and 632 relations were finally acquired in total, and the results are shown in Table 4. The preliminary

experiment showed that the proposed method could effectively extract knowledge by using the proposed semantic annotating and pattern mining method. With more training sentences involved, the number of knowledge pieces extracted from domain dataset could dramatically increase so as to enrich or even build a new domain knowledge base.

TABLE 4: Extracted concepts and relations based on 100 training data

Patterns extracted	Sentences matched	Concepts extracted	Relations extracted
272	1,297	1,929	632

With relation learning based on the training data, the annotated relations and semantic bank are potentially desirable to be added into knowledge base. To represent them in a standard knowledge format, two formal knowledge representations - Resource Description Framework (RDF) and Web Ontology Language (OWL) - are applied. RDF is a family of World Wide Web Consortium (W3C) specifications, which were originally designed as a metadata model. It has come to be used as a general method for conceptual description or modelling of information. OWL is a family of knowledge representation languages for authoring ontologies, and is endorsed by W3C (Smith *et al.*, 2004). In this study, the OWL Full was used as our knowledge representation method since it provides compatibility with RDF Schema.

The annotated data, either in semantic bank or other relation representation ways, need to be transformed into OWL format. Therefore, a set of knowledge transformation rules (KTR) was defined for this purpose. KTR is an intermediate description logic language, which can be transformed into first-order logic. The involved semantic labels are converted using the following rule and the *SubClassOf* relation is then represented in OWL, as follows:

KTR	$\forall x, there\ exist\ [x / concept_1] \rightarrow SubClassOf\ (concept_1, x)$
OWL format	*<owl:Class rdf:ID=" #[x]">* *<rdfs:subClassOf rdf:resource="#[concept$_1$]"/>* *</owl:Class>*

In the same way, the unit annotation in semantic bank can be converted by the following rule into OWL. It is important to mention that the relation inside a semantic label is already converted using the previous rule. Therefore, the following rule only considers the sub-class outside the semantic labels.

KTR	$\forall y, there\ exist\ [y\ HAVING\ \ concept_1 / concept_2]$ $\rightarrow SubClassOf\ (y, concept_2)$
OWL format	*<owl:Class rdf:ID=" #[y]">* *<rdfs:subClassOf rdf:resource="#[concept$_2$]"/>* *</owl:Class>*

The context annotation in semantic bank is more complicated since the words inside a sentence are concordant. Thus, "restriction" was used to represent and the corresponding KTR was defined as follows:

KTR	$\forall x \forall y, there\ exist\ [x\ HAVING\ \ concept_1 / concept_2]$ $WITH\ [y\ HAVING\ \ concept_3 / concept_4]$ $\rightarrow SubClassOf\ (x, concept_2)\ with\ constraint : SubClassOf\ (y, concept_4)$ ...
OWL format	*<owl:Class rdf:ID="#[x]">* *<rdfs:subClassOf rdf:resource="#[concept$_2$]" />* *<owl:Restriction>* *<owl:onCondition rdf:resource="#[y]" />* *<rdfs:subClassOf rdf:resource="#[concept$_4$] " />* *</owl:Restriction>* *</owl:Class>*

Similarly, the relations can also be transformed into knowledge representation format. In this way, the extracted knowledge can be represented in a standard way automatically and can be applied to enrich domain knowledge base directly.

CONCLUSION AND FUTURE WORK

A novel method for automatic knowledge acquisition from domain texts is proposed in this paper. Taking "heart disperses" from Yahoo! Answers as the source domain corpus, this method makes use of Minipar to label sentences and extract structural patterns. A semantic bank is further proposed to annotate concepts regarding to sentence context based on WordNet to enhance the semantic representation. With regard to the pattern matching, the relations extracted from training data were applied to the previously extracted concepts. Thus, the concepts, annotated labels, and relations can be converted by knowledge transformation rules to enrich domain knowledge base automatically. As for further development, more experiments with larger dataset and evaluation/comparison of the performances of knowledge extraction will be taken into consideration in the future.

REFERENCES

Berwick, R. C., Abney, S. P., & Tenny, C. (1991). *Principle-Based Parsing: Computation and Psycholinguistics*. Kluwer Academic Publishers.

Campbell, K., Das, A., & Musen, M. (1994). A Logical Foundation for Representation of Clinical Data. *Journal of the American Medical Informatics Association*, *1*(3), 218-232.

Cao, C. G. (2001). Medical Knowledge Acquisition From Encyclopedic Texts. *LNCS*, *2101*, 268-271. Berlin.

Carayannis, E. G. (2012). Knowledge Arbitrage, Serendipity, and Acquisition Formality: Their Effects on Sustainable Entrepreneurial Activity in Regions. *IEEE Transactions on Engineering Management*, *58*(3), 546-577.

Carenini, G., & Moore, J. (1993). Using the UMLS Semantic Network as a Basis for Constructing a Terminological Knowledge Base: a preliminary report. *In Proceedings of Annual Symposium on Computer Applica-tions in Medical Care*, pp. 725-729.

Dawoud, K., Qabaja, A., Shang Gao, Alhajj, R., & Rokne, J. (2012). Identifying Cancer Biomarkers by Knowledge Discovery From Medical Literature. In *Proceeedings of IEEE International Conference on Computational Advances in Bio and Medical Sciences*.

Deng, H., & Han, J. (2012). Uncertainty Reduction for Knowledge Discovery and Information Extraction on the World Wide Web. *Proceedings of the IEEE*, Vol. 100, Issue: 9, pp. 2658-2674.

Dependency grammar. (2012). Retrieved from http://en.wikipedia.org/wiki/Dependency_grammar/.

Fan, J., Kalyanpur, A., Gondek, D. C., & Ferrucci, D. A. (2012). Automatic Knowledge Extraction From Documents. *IBM Journal of Research and Development*, *56*(3.4), 5:1- 5:10.

Firdaus, O. M., Suryadi, K., Govindaraju, R., & Samadhi, T. M. A. A. (2012). Medical Knowledge Sharing Guideline: A Conceptual Model. In *Proceedings of International Conference on ICT and Knowledge Engineering*, pp. 22- 26.

Hao, T. Y., Ni, X. L., Quan, X. J., & Liu, W. Y. (2009). Automatic Construction of Semantic Dictionary for Question Categorization. *Journal of Systemics, Cybernetics and Informatics*, *7*(6), 86-90.

Hull, R., & Gomez, F. (1999). Automatic Acquisition of Biographic Knowledge From Encyclopedic Texts. *Expert Systems with Applications*, *16*(3), 261-270.

Minpar Evaluation. (2012). http://www.cs.ualberta.ca/~lindek/downloads.htm.

OpenNLP. (2012). http://opennlp.sourceforge.net/projects.html.

Smith, M. K., Welty, C., & McGuinness, D. L. (2004). OWL Web Ontology Language Guide. *W3C*.

Tanaka, K., & Jatowt, A. (2010). Automatic Knowledge Acquisition from Historical Document Archives: Historiographical Perspective. *LNCS*, *6259*, 161-172.

TreeTagger. (2012). Retrieved from http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/.

Wang, Y. M., Johanna, V., & Haase, P. (2006). Towards Semi-automatic Ontology Building Supported by Large-scale Knowledge Acquisition. In *AAAI Fall Symposium On Semantic Web for Collaborative Knowledge Acquisition*, Vol. FS-06-06, pp.70-77.

WordNet. (2012). Retrieved from http://wordnet.princeton.edu/.

Yahoo! Answers. (2012). Retrieved from http://answers.yahoo.com/.