# Towards A Comprehensive Semantic Annotation Method for Knowledge Acquisition from Classical Chinese Poetry

Tianyong Hao, Yinsheng Zhang, Fang Xia, and Chunshen Zhu

***Abstract*—In this paper, we propose a comprehensive semantic annotation method supported by a user-oriented markup language named *Olan* to facilitate semantic annotation for the purpose of acquisition of knowledge from classical Chinese poetry so as to build a high quality knowledge base. *Olan* is a language readable and operable by human annotators and transformable to formal knowledge representation languages such as OWL (Web Ontology Language) for knowledge reasoning. To ensure the effectiveness of the method, we develop a multi-language semantic annotation tool. With the features of online and offline searching, ontology visualizing, knowledge transforming and reasoning, the method is applicable of knowledge acquisition for semantic annotation of classical Chinese poetry.**

***Index Terms*—Semantic annotation tool, ontology, knowledge acquisition, Olan, OWL**

## I. Introduction

In recent years, more and more systems tend to use accumulated and represented knowledge in an effort to provide information services with improved precision. For example, major web search services such as Google and Yahoo are using ontology-based approaches to find and organize content on the Web. "Google's acquisition of Applied Semantics, Inc. - one of the leading vendors of semantic extraction tools - portends an active role for ontologies in their technology solutions" as observed by Denny [1]. Consequently, knowledge acquisition, formalization, presentation and sharing have attracted increasing attention from the field [2], [3].

As a large amount of knowledge is hidden in unstructured texts, knowledge acquisition from texts becomes one of the very important research areas in artificial intelligence [3]. Conventional knowledge acquisition approaches manually annotate and extract knowledge from documents and then formalize the knowledge at the conceptual level. This acquisition procedure relies mainly on a large number of knowledge engineers' manual processing, which can be an extremely labor intensive, time consuming, and often troublesome task. Thus, automated or semi-automated knowledge acquisition techniques are more preferred [4], usually at the expense of annotation quality.

Therefore, because of 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, acquisition of high quality knowledge for the purpose of fine-grained data processing still relies mainly on costly manual labour at present. Classical Chinese poetry is a case in point. Apart from features common to Chinese poetry in general, such as the absence of articles, grammatical genders, cases, tenses, and the scarcity of pronouns and prepositions [5], it is subject to strict formal restrictions in terms of the number of syllables, tonal variations and rhyming patterns, representing textual formations that appear to be concise in diction yet rich in implication, highly rhetorical, and thus linguistically complex. Even for human readers, specific training is needed to capture the nuance of such texts [6]. Probably that explains why few attempts have been made to process classical Chinese poetry for automatic knowledge acquisition.

To assist with knowledge acquisition from classical Chinese poetry, a user-oriented annotation method is proposed in this paper. To support the method, we design a markup language and name it *Olan*. Annotations written in this language can be easily and conveniently transformed into a target formal knowledge representation language such as OWL (Ontology Web Language) for further processing. To ensure its effectiveness, we also developed a multi-language semantic annotation tool with such features as online and offline searching, ontology visualizing, and knowledge expressing, creating, transforming and reasoning.

## II. A User-oriented Semantic Annotation Method

Human-operated semantic annotation is important for constructing a fine-grained high quality knowledge base. But human errors, such as those caused by lack of consistency among individual annotators can give rise to various kinds of complications, and coordination and integration of annotations will be in serious jeopardy. For example, the potential relations in a procedural text are extremely complicated and difficult to be formalized by knowledge engineers. It is partly due to the difficulties of separating procedural knowledge from other types of knowledge and transforming the procedural knowledge to a target language which computers can understand. Therefore, a user-oriented conventional annotation language, with terms clearly and structurally defined, is essential for group annotation exercises.

To that end, we design a user-oriented semantic annotation language named *Olan*, which is easily readable and usable for annotators, meanwhile can be easily interpreted by an OWL-transforming system. It consists of five components: symbols, relation definitions, attribute

Manuscript received March 14, 2012; revised April 16, 2012.

T. Hao is with the Department of Chinese, Translation and Linguistics, City University of Hong Kong, Hong Kong, China. (e-mail: haotianyong@gmail.com.)

Y. Zhang, Fang Xia and Chunshen Zhu are with the Department of Chinese, Translation and Linguistics, City University of Hong Kong, Hong Kong, China

definitions, transforming rules, and logic, which can be used jointly to represent the intermediate form of knowledge. Compared with the "Frame-slot" method [7], this language can help dramatically increase knowledge engineers' efficiency in manual labelling, by improving the readability of the knowledge snippets extracted, among other things. Furthermore, *Olan* has a complete solution to information transformation in the sense that annotations in this language can be transformed onto any commonly used knowledge representation languages such as RDF or OWL.

In the symbol component of *Olan*, "Def" means the start of a new annotation item, which could be a Chinese character or a word. This symbol identify a cluster of annotation units, in which a single annotation unit, starts with "{" and end with "}", is a complete annotation of this item in a particular context. An annotation element is the annotation on a particular aspect of the item and is separated by ":". It contains a slot, which is a certain description of either attributes or relations and consists of a slot name and a slot value, and the constraint of the slot. The "Loc" indicates the location of an item in the original text. "Cons" is the constraints of a certain slot in an annotation element, and ";" is the split tag between a constraint value and its aspect.

TABLE I: EXAMPLES OF RELATION DEFINITIONS WITH EXPLANATION IN OLAN

| Symbols | Relations | Examples |
|---|---|---|
| IS_A | is a | Wang Wei is a poet |
| HAS_INSTANCES | has a | Poets has a instance Wang Wei |
| IS_KIND | is a kind of | Car is a kind of motor vehicle |
| HAS_KINDS | has types of | Car has a type of cab |
| IS_PART | part whole | Wheels are parts of cars |
| HAS_PARTS | whole part | A car has part wheels |

TABLE II: EXAMPLES OF ATTRIBUTES AND VALUES IN OLAN FORMAT

| Attribute | Attribute values |
|---|---|
| Form | Formed; unformed; square; round; angular ... |
| Dimensionality | dot; linear; planar; cubic; flat; protruding; dented; layered ... |
| Sharpness | blunt; sharp; pointed; pointless ... |
| Fatness | Fat; bony ... |
| Measurement→Length | Long; Short ... |
| Quantity→Amount | many; few; single; double; mass; fragment; some; sufficient; insufficient; half ... |

In addition to the basic symbols, *Olan* has also included a group of relation symbols to represent relations between a particular item and other items. More specifically, "IS_A" indicates the relation of an item, *item1*, and its upper concept, *item2*, represented as IS_A(*item1*, *item2*), as well as "HAS_INSTANCES" as opposite. "IS_KIND" means the relation of being a kind of while "HAS_KINDS" means the relation of having types of. The part-whole relation is respectively tagged as "IS_PART" and "HAS_PARTS". On these basic symbols, the user may add symbols to designate other types of relations. For example, the user may define a relation as "HAS_OTHER_NAMES" and use the symbol to indicate that the annotated item has been referred to by different names elsewhere in the corpus. Table I gives a list of exemplary relations.

The attributes of an annotated item need to be specified. And in *Olan* an attribute is presented in a format of slot with splitter ":", in which constraint is optional. For example, the attribute "Sharpness" have the values such as "blunt", "sharp", "pointed", and "pointless" thus can be represented as, for example, "Sharpness: sharp", in which "sharp" is the slot value.

With the above symbols covering relations and attributes and their values, knowledge engineers can annotate Chinese poetry using *Olan* language to represent intermediate knowledge. To make the annotation in *Olan* transformable onto common knowledge representation languages such as OWL, a conversion schema was developed to do with *Olan*, the workings of which are described in the following:

By this conversion schema, an annotated item to be transformed is tagged as *s*, and its pattern as $p_s$, while the target re-presentation of the item in, say, OWL is tagged as *o*, and its pattern as $p_o$. As to the semantic content of the item, it is divided by slots and each slot is defined as "[SLOT *id*]". The slotting can be used in both the source text in *Olan* and the target text in a knowledge representation language such as OWL for ease of transformation. An example from *Olan* to OWL is shown in the table below:

TABLE III: AN EXAMPLE OF PATTERN TRANSFORMATION ON OLAN

| Source pattern | Target pattern |
|---|---|
| Def [ITEM]<br>{<br><SLOT *id*=1>[SLOT name]: [SLOT value]<br><SLOT *id*=2>[SLOT name]: [SLOT value]<br>} | <owl:Class rdf:ID="[ITEM]"><br><owl:Restriction><br><owl:onProperty rdf:resource="#[SLOT *id*=1].name" /><br><owl:hasValue rdf:resource="#[SLOT *id*=1].value" /><br></owl:Restriction><br><owl:Restriction><br><owl:onProperty rdf:resource="#[SLOT *id*=2].name" /><br><owl:hasValue rdf:resource="#[SLOT *id*=2].value" /><br></owl:Restriction><br></owl:Class> |

In addition to the basic slot-to-slot transformation, in more complex cases where, for instance, the slot number is not fixed, a flow control mechanism is included in the conversion schema to increase the flexibility of slot number by using a "<WHILE>" tag. The logic judgment, in the flow control mechanism for patterns transforming, further extend the expression capability to solve the complicated annotation transformation.

A transformation can become even more complicated when a key slot is found empty, that is, there is no semantic content (missing parts) to fill in the slot, because of, say, the incompleteness of the annotation item. In such cases, the system needs either to find the related parts (attributes or relations) for the item elsewhere in the corpus or to treat it

as an incomplete data in preprocessing, with the slot remaining empty. For example, the annotators may leave out some parts since the same parts in previous or the next annotation elements already exist. To enable the system to detect and confirm the existence of a related annotation element, two labels are introduced, namely {#PRE [SLOT *id*]} and {#NEX [SLOT *id*]}, to identify the annotation elements concerned in a previous or subsequent occurrence of the same element in the corpus.

## III. Logic Reasoning

An important function of the system is to retrieve or acquire new knowledge by automated reasoning, which is a logic operation in essence. The reasoning not only relies on the premises or rules determined by users, but also largely on the four basic databases the system is based on, of which the first three are in OWL. The remainder of this section explains how this logic and reasoning function works on the basis of the structure of expressions of the data (as propositions).

**1)** Database **D1** is a domain ontology describing concept nodes in a tree structure, and relations between the nodes. The relations include "IS_A" ("is a") and other relations such as "PART_WHOLE". The relations also exists in other database **D2**, and **D3**. We define

**D1**=$\{N, R_{D1} \mid N$ is a set of nodes of the ontology, $R_{D1}$ is a set of the relations$\}$, and

$R_{D1}=\{r_{D1}(n_i, n_{i+1}) \mid n_i, n_{i+1} \in N, i=1,2, ..|N|-1\}$,

where, $n_i$ and $n_{i+1}$ are two nodes connected directly in a descending order and $r_{D1}(n_i, n_{i+1})$ is the relations of the two nodes of the same type. For example, the moon IS_A celestial body represents a $r_{D1}$ relation. Here, the word/phrase in a box indicates a concept node in the ontology. The reasoning based on **D1** runs as follows:

$$r_{D1}(n_i, n_{i+1}) \wedge r_{D1}(n_{i+1}, n_{i+2}) \rightarrow r_{D1}(n_i, n_{i+2})$$

An example of reasoning the relation between "reptile", "cold blooded vertebrate", and "vertebrate" is given as follows:

reptile ISA cold blooded vertebrate $\wedge$ cold blooded vertebrate ISA vertebrate $\rightarrow$ reptile ISA vertebrate

The transitivity exists in the above relation reasoning is applicable to all the suitable nodes in the ontology automatically.

**2)** Database **D2** is a set of attributes and their values which are generalized from human annotation. This database is characterized by two directly linked attribute nodes in the subordinate relation of "HAS", as defined below:

**D2** $=\{<AN, AV>, r_{D2}(an_i, an_{i+1}) \mid i=1,2, ..m\}$,

In which $AN$ refers to a set of attributes and $AV$ its corresponding value set. $<AN, AV>$ is a semantic annotation structure such as < fullness, full|empty>, in which the "|" is the *nand* relation. The $an_i$ and $an_{i+1}$, belonging to $AN$, are two linked nodes in **D2**, while The $r_{D2}(an_i, an_{i+1})$ means a set of "HAS (such-and-such attributes) relations between the two nodes, as seen for example, in the morality HAS immoral.

The reasoning based on **D2** thus runs as follows:

$$r_{D2}(an_i, an_{i+1}) \wedge <an_{i+1}, av_{i+1}> \rightarrow r_{D2}(an_i, av_{i+1})$$
$$r_{D2}(an_i, an_{i+1}) \wedge r_{D2}(an_{i+1}, an_{i+2}) \rightarrow r_{D2}(an_i, an_{i+2})$$

The below are an actual example:

situation HAS circumstances, circumstances HAS urgency $\rightarrow$ situation HAS urgency

**3)** Database **D3** is the annotated words/phrases in the form of a set of atom propositions drawn from the annotated domain texts, such as poems. Propositions in this database are tagged as $P$ and defined as follows:

$P=\{p(n_i, n_j)\}$,
（Predicate type 3-1）

where, $n_i$ and $n_j$ are two nodes of $N_i$, or *AN or AV*. For example, $p$=HAS (moon，brightness), while in

$P=\{p(wn_i, wn_j)\}$,
（Predicate type 3-2）

where $wn_i$ and $wn_j$ are two annotated words/phrases showing an instance-class relation to the corresponding nodes $n_i$ and $n_j$ , in which the relation are determined by the annotators:

$P=\{p(wn_i, n_j)\}$
（Predicate type 3-3）
$P=\{p(n_i, wn_j)\}$
（Predicate type 3-4）

We thus can define $r_{D2}$ as the reasoning based on Predicate types from 3-1 to 3-4, for instance as seen below, which is based on Predicate type 3-3 on **D3**:

$$p(wn_i, n_j) \wedge (r(n_i, n_{i+1}) \vee r(n_j, n_{j+1})) \rightarrow p(n_{i+1}, n_j) \vee p(n_{j+1}, n_j),$$

where the $r$ is a relation either in $r_{D1}$ or in $r_{D2}$. Following the reasoning, two new propositions $p(n_{i+1}, n_j)$ and $p(n_{j+1}, n_j)$ may be extracted. We can see the concept $n_{i+1}/n_{j+1}$ is in relation to its super concept $n_i/n_j$. An actual example of the reasoning is shown below:

HAS_ATTR (fluid, wide) $\wedge$ tide IS_A fluid $\rightarrow$ HAS_ATTR (tide, wide))

**4)** Database **D4** is a set of reasoning rules produced by the operations $(\neg, \rightarrow, \wedge, \vee)$ of the propositions on **D1**, **D2**, and **D3**, or of the propositions by the system. **D4** are labeled as $\{rule_i\}$, which refers to a reasoning rule. Three rules are shown below as examples:

HAS_ATTR ($a$，cool) $\wedge$ LEAD(cool, tactile sense) $\rightarrow$ *Convey*($a$, tactile sense);

IS_A ($b$，lunar ) $\wedge$ LEAD(lunar, light perception) $\rightarrow$ *Convey*($b$, light perception);

IS_A ($a$，lunar ) $\wedge$ LEAD(lunar, light perception) $\wedge$ HAS_ATTR ($a$，cool) $\wedge$ LEAD(cool, tactile

sense)→*Convey*(*a*, tactile sense)∧*Convey*(*a*, light perception);

The first example means that if an object *a* has the attribute of "cool", by whose lead to the tactile sense, the computer can by deduction know that *a* has an tactile aspect. For example, if *a* is "cold water", then <*Convey* (cold water, tactile sense)> is established, which means if a "cold water" shows up in a poem, it can convey a sense of touch.

The reasoning based on **D4** in mapping format thus runs as follows:

$$<\{Rule_i\}, \neg, \wedge, \vee> \rightarrow <\{Rule_i\}, \neg, \wedge, \vee>$$

## IV. Summary

This paper proposes a user-oriented semantic annotation method by proposing a markup language, *Olan*, to assistant annotators for high quality knowledge acquisition. This language consists of five components: symbols, relation definitions, attribute definitions, transforming rules, and logic, which can be used jointly in a concerted manner to represent the intermediate form of knowledge. By way of online and offline searching and ontology visualization, annotated content can be transformed onto a common knowledge representation language such as OWL easily and conveniently. The transformation is further supported by a multi-language-based semantic annotation tool to unify and integrate annotations done by more than one human annotator.

## References


[1] M. Denny, "Ontology Tools Survey, Revisited," http://www.xml.com/pub/a/2004/07/14/onto.html, 2004.

[2] P. Hendriks, "Why Share Knowledge? The Influence of ICT on the Motivation for Knowledge Sharing," *Journal of Knowledge and Process Management,* vol.6, no.2, pp.91-100, 1999.

[3] P. R. Bowden, P. Halstead, and T. G. Rose, "Extracting Conceptual Knowledge from Text Using Explicit Relation Markers," *Advances in Knowledge Acquisition. Lecture Notes in Artificial Intelligence*, Vol. 1076. Springer-Verlag, Berlin, pp.147-162, 1996.

[4] Y. M. Wang, V. Johanna, and P. Haase, "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, 2006.

[5] C. Yee, "Introduction to 300 T'ang Poems". The University of Iowa. http://www.chinapage.com/poem/300poem/introduction.html, 1999.

[6] A. C. Fang, F. J. Lo, and C. Chinn, "A Computational Framework for Syntax-Driven Structured Analysis of Imageries in Tang and Song Poems," In *Proc. of the 4th China International Symposium on Tang and Song Poetry*, pp.289-300, Hangzhou, China, 2009.

[7] C. Orasan, "PALinkA: a highly customizable tool for discourse annotation," In *Proc. of the 4th SIGdial Workshop on Discourse and Dialog*, pp. 39 - 43, Sapporo, Japan, 2003.