# Semi-automatic Annotation System for OWL-based Semantic Search\*

C.-H. Liu<sup>1</sup>, S.-C. Hung<sup>2</sup>, J.-L. Jain<sup>3</sup>, and J.-Y. Chen<sup>4</sup>

<sup>1, 3, 4</sup> Department of Computer Science & Information Engineering National Central University Jhong-Li, Taiwan 32099 Email: jasonjychen@gmail.com

> <sup>2</sup> Identification and Security Technology Center Industrial Technology Research Institute
>  Rm. 212, Bldg.52, 195, Sec.4, Chung Hsing Rd. Chutung, Hsinchu, Taiwan, ROC sc\_hung@itri.org.tw

Abstract-Current keyword search by Google, Yahoo, and so on gives enormous unsuitable results. A solution to this perhaps is to annotate semantics to textual web data to enable semantic search, rather than keyword search. However, pure manual annotation is very time-consuming. Further, searching high level concept such as metaphor cannot be done if the annotation is done at a low abstraction level. We, thus, present a semi-automatic annotation system, i.e. an automatic annotator and a manual annotator. Against the web ontology language (OWL) terms defined by Protégé, the former annotates the textual web data using the Knuth-Morris-Pratt (KMP) algorithm, while the latter allows a user to use the terms to annotate metaphors with high abstraction. The resulting semantically-enhanced textual web document can be semantically processed by other web services such as the information retrieval system and the recommendation system shown in our example.

Keywords- semi-automatic annotation system, semantic search, web ontology language (OWL)

# I. INTRODUCTION

The current keyword search by Google, Yahoo, and so on gives inaccurate results because two keywords may have the same string, but with different semantics. For instance, a user wants to find the blaze news. He/she searches for "祝融" (God of fire, which stands for a blaze in Chinese) by Google. The user would find much information about characters of the God of fire, but not about blaze.

Ontology is a knowledge description technology, which could description semantics of textual web data such as string and article. The web ontology language [2] (OWL) is a popular language used to describe ontology. An OWL term annotated by user is a concept of the world, which is used to improve search accuracy. However, for most users the annotation seems difficult. Further, as different users may annotate different terms to the same data, some management scheme is needed. We thus propose a semi-automatic annotation system, which assists user to annotate textual web data and manages the terms defined by users. There are three issues below to be solved:

- 1. The current information retrieval (IR) is a keyword search, not a semantic search, which gives inaccurate results. We figure that it would be necessary to improve search accuracy by annotating semantics to textual web data.
- 2. Because many people do not understand high abstraction concepts of a domain, they simply annotate some low-abstraction keyword (string in the textual web data) of the domain ontology. For example, the title of news is "祝融" (God of fire). For educated people, they know that this is a metaphor (high abstraction concept) for "火 災" (blaze), so they annotate "火災" (blaze) to the news. On the other hand, ordinary people may just annotate the string "祝融" (God of fire) to the news, which makes semantic search essentially equivalent to the old keyword (string) search.
- 3. Most textual web data are updated very frequently. Pure manual annotation of them is extremely time-consuming.

We address the three issues above:

- 1. We propose the semantically-enhanced textual web document that is annotated with semantic terms for IR service to give more accurate results than otherwise.
- 2. We allow various users to share terms in annotation, which enhances the abstraction level. The user could be expert or general user, and they share the terms among all users. The terms include low-level and high-level abstraction information, which is helpful for semantic search. For example, an expert can annotate "祝融" (God of fire), a high abstraction concept of blaze, to a

<sup>\*</sup>A previous version of this paper was published in the Third Workshop on Engineering Complex Distributed Systems (ECDS 2009) March 16-19, 2009, Fukuoka, Japan, pp. 475-480.

poem containing blaze. While a general user can annotate "火" (fire), a low abstraction concept to it.

3. The linear time complexity of the Knuth-Morris-Pratt (KMP) [4] algorithm used in the automatic annotation saves a lot of time. This reduces the load of manual annotation.

This paper is organized as follows. Firstly, we compare our approach with other researches in section 2. Secondly, we introduce the architecture of this system in section 3. Next, we describe an example in section 4. Finally, we draw conclusions in section 5.

## II. RELATED WORK

This section compares related research from two perspectives: 1) abstraction level of annotation, and 2) search capability.

## A. Abstract level of annotation

Samhaa [5] thinks that the meaning of header of article must be clear or well defined, if it is annotated through automatic way. However, the meaning of header normally cannot stand for the article, thus the terms annotated by automatic way are low-level abstraction, even are mistake or ambiguous.

JIM's [6] ACE system let users input a free text into parser, and then it compare the free text with ontology to do term replacement. However, ACE system cannot annotate the whole article, and it is also difficult to find out results through the high-level abstraction.

In our system, the terms that contains low and high level abstraction information is annotated by various users to the same textual web data, which is shared among users to enhance search accuracy.

# B. Search capability

National Digital Archives Program in Taiwan [7] had developed a series of the poetry retrieval system. However, it is only keyword search.

The class with Parent-child relationship is defined in Samhaa's approach, which is described in XML markup language. However, it is still difficult to describe property or to define the relation between different classes. Further, it is not easy to expand.

In our system, we use OWL to describe semantic information to form the semantically-enhanced textual web data. The IR service can use it to find out more accurate results. For example, we input the below string:

"長城邊塞的戰場人事" (Man in Battles at Great Wall)

It will find out five poems about border, such as "王昌齡 出塞" (Wang T.-L, Out of Border). Nevertheless, the poetry retrieval system cannot find them out.

## III. A SEMI-AUTOMATE ANNOTATION SYSTEM

This section presents the Semi-automatic Annotation System architecture and how to implement it.

## A. Architecture of a Semi-Automate Annotation System

A lot of semantic annotation systems has been developed, which is divided to two kinds: 1) Pattern-based and 2) Machine learning-based according to the taxonomy described by Lawrence [10]. In this paper, we utilize Pattern-based to build a Semi-automatic Annotation System. Its architecture is shown in figure 1.

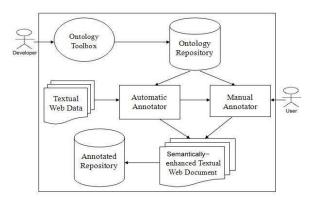


Figure 1 Architecture of Semi-automatic Annotation System

- 1. Ontology Toolbox: By using Protégé graphical tree structure tool, a developer can establish terms and properties in tree structure. After established, the ontology will be transformed into Java code. And then the developer can use it in Java program.
- 2. Ontology Repository: We use SESAME RDF repository [9] and OWL plug-in [10] to manage ontologies. There are two ways to use ontologies to annotate textual web data: 1) Automatic Annotator and 2) Manual Annotator.
- 3. Automatic Annotator: This provides an automatic annotator interface. By using KMP algorithm, the terms in the Ontology Repository will be matched with textual web data. If there are terms matched between them, these will be saved as OWL files into Annotated Repository.
- 4. Manual Annotator: This provides graphical editor interface. User can use terms defined by developer to annotate the textual web data, which the annotation information will be saved as OWL file into Annotate Repository.
- 5. Annotated Repository: By using SESAME RDF repository, this manages the ontology and OWL file generated by Automatic or Manual Annotator.
- 6. Semantically-enhanced Textual Web Document: This contains two parts textual web data and OWL file. And, it provides the semantic information for others web services. For example, the IR service will retrieve semantic information from semantically-enhanced textual web document, and then send to Recommendation Service (RS). After that, it will

recommend the relative information according to the semantic information.

7. Textual Web Data: This is textual type information on the Web such as string, news, message, and article.

#### B. Implementation of a Semi-Automate Annotation System

In our architecture, the core of ontology toolbox is the Protégé graphical tree structure tool, which is responsible for editing domain ontology. In ontology repository, we use SESAME to store domain ontologies and annotation terms in OWL files.

There are two annotators: 1) automatic annotator, and 2) manual annotator. In the former, the KMP compares the strings in textual web date against the terms in domain ontology. If matched, they will be automatically annotated to form an OWL file into SESAME. In the latter, a user can explore the domain ontology by hierarchical structure. First, he/she will see the top-level terms of the domain ontology, and then he/she could select a term such as  $\Lambda$  (people) to view its subclasses such

as 述懷 (memory), 思考 (think), etc. Then, he/she could select appropriate terms as annotation terms.

In semantic search, our system will compare the string user inputted with annotated terms. If matched, the system will return textual web data according to the terms. And, the terms will be shown in different font size as tag cloud through calculating the number of annotations.

In the domain ontology, we use Protégé graphical tree structure tool to build the poem ontology we defined, and then we can quickly revise tree structure and properties straightly (Fig 2.). After that, we transform the poem ontology into Java code, and user can get the terms by declaring object to access them. The poem ontology include 1116 classes, the first level includes 40 classes such as "京都" (capital), "人事" (people) and "儒家" (Confucian), the second level includes 1076 classes such as "留別" (stay and leave), "嘲戲" (ridicule) and "尋訪" (visiting). Each of the 1116 classes stands for a concept of real world.

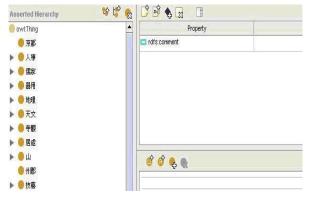


Figure 2 The Ontology developed by Protégé

A user can select appropriate terms to annotate poem, and then he/she will get one semantically-enhanced poem document. Next, the user writes the corresponding Java code by using the ClassifyFactory API provided by Protégé. Furthermore, the user can save terms as an OWL file.

Figure 3 shows an OWL file, in which the poem is "出塞 (out of border) 王昌齡 (Wang T.-L.)" and the OWL terms are " 述懷" (memory), "人事" (people), "長城" (great wall) and "戰 場" (battlefield).

#### IV. AN EXAMPLE

This section illustrates the example of OWL-based poem semantic search system, which we develop based on our architecture as shown in figure 4:

xmins:xsd="http://www.w3.org/2001/XMLSchema	wl#" xmlns:rdf="http://www.w8.org/1999/02/22-rdf- #" xmlns:rdfs="http://www.w3.org/2000/01/rdf-sch 06952901718#" xml:base="http://140.115.155.45/m"
<ul> <li><owl:ontology 140.115.155.45<="" http:="" rdf:about="&gt;&lt;/li&gt; &lt;/ul&gt;&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;owl:imports rdf:resource=" td=""><td>i/classify.owl" /&gt;</td></owl:ontology></li></ul>	i/classify.owl" />
- <c:classification rdf:id="C1"></c:classification>	
<c:hasname rdf:datatype="http://www.w3.org/20&lt;/td&gt;&lt;td&gt;JO1/XMLSchema#string">人事</c:hasname>	
- <c:classification rdf:id="c3"></c:classification>	
<c:hasname rdf:datatype="http://www.w3.org/20&lt;/td&gt;&lt;td&gt;D&lt;b&gt;01/XMLSchema#string&lt;/b&gt;"><b>長城</b></c:hasname>	
- <c:poem rdf:id="p1"></c:poem>	
- <c:classifiedby></c:classifiedby>	
- <c: classification="" id="c4" rdf:=""></c:>	
<pre><c:hasname rdf:datatype="http://www.w3.org&lt;/pre&gt;&lt;/td&gt;&lt;td&gt;g/2001/XMLSchema#string">神信</c:hasname></pre>	
<c:classifiedby rdf:resource="#c3"></c:classifiedby>	
- <c:classifiedby></c:classifiedby>	
- <r: classification="" id="c?" rdf:=""></r:>	
	q/2001/XMLSchema#string">戰場
	J/ 2001/ Mileochema String Strag (remesteries
<pre></pre>	
	DO1/XMLSchema#string">出業_王昌齡
	JU1/XMLSChemd#Schny >出基_土自爵
Figure	3 OWL file
	$\bigcirc$
	Apported use Poem Information
	Repository Retrieval System
Doom	Repository

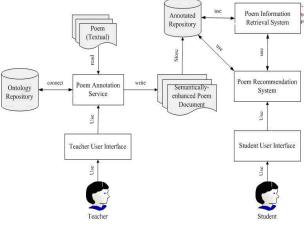


Figure 4 OWL-based Poem Semantic Search System

In figure 4, we add two user interfaces: 1) Teachers User Interface and 2) Student User Interface. And the semanticallyenhanced poem document is used by IR and RS to do semantic search. Firstly, the system will automatically annotate the poems by using KMP algorithm. Next, by using Teacher User Interface, a teacher selects a poem 出塞\_王昌齡 (Wang T.-L, Out of Border) in two words from the poem classification on the top of figure 5, and then he/she selects 述懷 (memory), 人事 (people), 長城 (great wall) and 戰場 (battlefield) as high-level abstraction information to 出塞\_王昌齡 (Wang T.-L, Out of Border) from the poem keywords list at left side of figure 5. After that, he/she enters the "send" bottom to store the poem and the high-level abstraction information (terms). And then, the web services can use these terms in the semanticallyenhanced poem documents through SESAME RDF repository's API.

(2) (1) (1) (1) (1) (1) (1) (1) (1) (1) (1		Condition of the local division of the local
時名間率 1時名三字 Mieroso	oft Internet Explorer 🛛 🔀	1. 静名七述 1. 静名八字 1. 静名九步
歴辺 出版_工品的 出版。 田 液花 波思 絶型 無数 東 建心 月衣 杜相 要な 単度合 東西 素玄 英語	是否將所選的關鍵字存入系統 確定 取消	# 相思 接持 王相 直接 白短岛 意思 本白 这里 局資 () 因之二、未規模 插出 () 第十 古曼 多 () 包之 接持 () 化和 發現 夏之 未品種 黑袋 () 日, 和方子 查約 開盟 直社, 評准 奉兵 已没 山石
9出 <b>來_王昌齡</b> 敗開建字標記 已標記的唐約曉建字列表:	出憲_王昌翰 王昌翰	
1. <u>這價</u> 2. <u>人事</u> 3. <u>長城</u> 4.戰년	☆85明月:萬時間, 萬風到	4征人未遭。但使難城飛將在,不数
(TX-30)	473.M5.0905e4111 -	
建立物田 余能重切		
可供標記的唐時關鍵字列表:		
<u>草</u> ,玉帛,道		
家 京都 重		
事. 儒家. 程		

Figure 5(a) Teacher User Interface in Chinese

Two word	Three word	Four word	Five word
	Store key	word ?	
	Enter	Cancel	
	g TL, Out of Bord eyword in the below		e: Wang TL, Out of Border

Figure 5(b) Teacher User Interface in English

When students are learning poem, they can input high-level abstraction information to find out the poem. For example, a student inputs the keyword "長城邊塞的戰場人事" (Men in Battles at Great Wall) to find related poems. When system receives the keyword, it will compare it with the terms in the ontology. If there are two or less terms matched, then the system will deliver the terms to IR, and IR will invoke the RS to recommend five poems. If there are more than three terms matched, the system will deliver these terms to RS, and then RS will recommend the most appropriate five poems according to the semantically-enhanced poem documents.

In figure 6, the system finds out the "長城" (great wall), " 人事" (people) and "戰場" (battlefield) are the same term between keyword and ontology, and then the RS will recommend the five poems, such as 王昌齡's 出塞 (Wang T.-L, Out of Border).

After students get the five poems, they can also select the terms which is annotated by system in the poems to search related poem, for instance 月 (moon) in 出塞\_王昌齡 (Wang T.-L, Out of Border) as shown in figure 7. After that, the system will return back another five poems such as 宿建德江 (live in the J. D. River) according to 月 (moon) in 出塞\_王昌 齡 (Wang T.-L, Out of Border).

唐詩學習輔助系統	
手動輸入開設字: 加入	推薦系統所推薦出來的唐詩:
已加入的重時間鍵字列表:	1. <u>出臺_王昌齡</u>
唐時間鍵字列表向未加入任何關鍵字	2. <u>登</u> 柳州城樓寄演了封連四州
出憲_王昌齡 王昌齡	3. <u>獨不見</u>
秦時明 <u>月</u> 漢時關,萬里長征人未還。但使龍城飛將在,不	- 4. <u>塞下曲</u>
教胡馬度陰山。	5. 涼州曲

Figure 6(a) Student User Interface in Chinese

Input the keyword: Enter	Poem List of Recommendation System
Additional Keyword list:	1. Wang TL, Out of Border
No keyword	2
	3
Poem Name: Wang TL, Out of Border	4
Poem Content:	5

Figure 6(b) Student User Interface in English



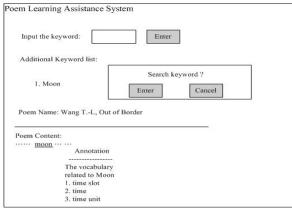


Figure 7(b) Related Term Search in English

#### V. CONCLUSIONS

We propose a semi-automatic annotation system, which assists user to annotate textual web data and manages the terms defined by user. Its advantages are:

- 1. The traditional information trivial (IR) search is combined with semantic information through the semantically-enhanced textual web document. This gives more accurate results than the old keyword search does.
- 2. The annotation terms are saved in the Annotated Repository, which is shared among all the users. This allows a user to manually annotate terms with abstract concepts. Thus, the search is improved.
- 3. In automatic annotation, using Knuth-Morris-Pratt (KMP) algorithm with linear time complexity saves a lot of time. Thus, the load of manual annotation is reduced.

#### ACKNOWLEDGMENT

The authors would like to thank the Industrial Technology Research Institute (ITRI) in Taiwan for their supports under the project "Ontology-based database management technology for surveillance data" in 2008.

#### REFERENCES

- [1] Stanford University. [Online]. Available: http://protege.stanford.edu/
- [2] World Wide Web Consortium. OWL Web Ontology Language Overview. [Online]. Available: http://www.w3.org/TR/owl-features/
- [3] Wikipedia. Information Retrieval. [Online]. Available: http://tinyurl.com/3w5qs2
- [4] Wikipedia. KMP Algorithm. [Online]. Available: http://tinyurl.com/34wneo
- [5] Samhaa R, Maryam H., and Ahmed R.. Ontology Based Annotation of Text Segments. March 2007 SAC '07: Proceedings of the 2007 ACM symposium on Applied computing.
- [6] Blythe J. and Gil Y. Incremental Formalization of Document Annotations through Ontology-Based Paraphrasing. In Proceedings of the 13th International World Wide Web Conference (New York, New York, May 2004), pp. 455-461.
- [7] National Digital Archives Program, Taiwan. [Online]. Available: http://tinyurl.com/3ocjjj, http://cls.hs.yzu.edu.tw/tang/Database/index.html
- [8] Reeve L. and Han H. Survey of Semantic Annotation Platforms, In Proceedings of SAC'05 (Santa Fe, New Mexico, USA, March 2005).
- [9] Sesame repository. [Online]. Available: http://www.openrdf.org/
- [10] OWLIM Semantic Repository. [Online]. Available: http://ontotext.com/owlim/

#### AUTHORS PROFILE

**Chih-Hao Liu** received his Master degree of Information Engineering from Chaoyang University of Technology. He is currently a PhD candidate in the National Central University in Taiwan. He joined the software engineering laboratory in 2005. He also participated the SIM (Service-oriented Information Marketplace) project from 2005 to 2007. And, his current research interests focus on Semantic Web and Agent.

Shang-Chih Hung received his Master degree of Control Engineering from the National Chiao-Tung University. He is currently with ISTC (Identification and Security Technology Center) of ITRI (Industrial Technology Research Institute) in Taiwan. ISTC focuses on developing next generation video surveillance technologies. And, his current research interests include data fusion and situation awareness.

**Jhih-Liang. Jain** received his Master degree of Information Engineering from the National Central University. He joined the software engineering laboratory in 2006. He also participated the ITRI project "Ontology-based database management technology for surveillance data" in 2008.

**Jason Jen Yen Chen** is with the Department of Computer Science and Information Engineering in the National Central University in Taiwan. He earned international recognition by winning Top, Third, and Fifth Scholar in the world in the field of System and Software Engineering in 1995, 1996, and 1997, respectively. The ranking is based on cumulative publication of six leading journals in that field. His current research interests include agile method and agent technology.