Automatic Ontology Construction from Vietnamese text

Dai Quoc Nguyen[†], **Dat Quoc Nguyen**[†], **Khoi Trong Ma**[†], and **Son Bao Pham**[†,‡]
[†] University of Engineering and Technology
Vietnam National University, Hanoi
{dainq, datnq, khoimt_52, sonpb}@vnu.edu.vn
[‡] Information Technology Institute
Vietnam National University, Hanoi

***Abstract*—Ontologies have served as a knowledge representation about the whole world or some part of it. Building ontologies is a challenging and active research area. Manually constructed Ontologies often have higher quality than the ones created by automatic or semi-automatic approaches but they tend to be more applicable to small domains. Automatic approaches are considered more suitable for building large scale Ontologies where time and efforts of human experts become a bottleneck. For both paradigms, approaches to building Ontologies from Vietnamese texts are still very limited. In this paper, we propose a system that automatically builds Ontology from Vietnamese texts using cascades of annotation-based grammars. Obtained experimental results on a university organizational structure domain are very promising.**

***Keywords*-Ontology construction system; Information extraction;**

I. Introduction

In the field of Artificial Intelligent, an Ontology is defined as a formal, explicit specification of a shared conceptualization [1][2]. Ontologies have served as a knowledge representation about the whole world or some part of it. With the capabilities of representing information as well as supporting inference, Ontologies have been applied in a number of fields, including artificial intelligent, question answering, Semantic Web, biomedical informatics, software engineering, systems engineering etc.

Ontology construction is an active research area yet challenging. Approaches to building ontologies can be broadly categorized into two types namely manual and (semi)-automatic ones. In general, manually constructed Ontologies have higher quality than their automatic counterpart but they tend to be more suitable for tasks with small-size knowledge base. To scale to larger knowledge bases, automatic or semi-automatic tools are more promising as they bring the power of computers to save time and effort of human experts.

In this paper, we propose a system that automatically builds Ontology from texts for Vietnamese. Our system is implemented using the GATE annotation-based framework [3] with the front-end component performs syntactic analysis to automatically detect noun phrases and relation phrases from input documents. Subsequently, phrases indicating classes, individuals, relationships and properties in Ontology are extracted. Information identified by the front end component will be processed by the back-end component to create an OWL Ontology using Text2Onto tool [4].

The rest of the paper is organized as follows: in section II, we provide some related works including existing approaches to building Ontologies and Text2Onto. We then describe our system as well as our experiments in section III and section IV respectively. Finally, conclusions and future works will be presented in section V.

II. Related works

Protégé [5][6] is one of the most widely used platforms to manually build Ontologies. This software has an extensible architecture to possibly integrate additional plug-ins. One of the popular plug-ins integrated into the Protégé platform is the OWL plug-in that is a Semantic Web extension of the platform. It provides a library of Java methods to manage the open-source ontology formats for OWL (Web Ontology Language) and RDF (Resource Description Language).

There are many different methods that have been proposed to automatically generate Ontology from texts. Hu and Liu [7] introduced a system creating Ontology from texts by using WordNet. They detect concepts from text and then search concepts from WordNet [8] corresponding with identified concepts. Kong et al. [9] described a WordNet-based method where concepts are determined by using returned results via querying WordNet, and then classes are created based on these concepts to build Ontology. Users can extend the Ontology by adding new concepts and export to OWL format.

Some methods utilize Ontology learning process to automatically create Ontology as presented in [10][11]. These methods extract the information from a wide range of input document formats including UML, XML, text and web to acquire Ontology concepts and provide a graphical interface for modifying the generated Ontology.

Text2Onto [4] is a framework for Ontology learning. It automatically or semi-automatically generates Ontology from textual resources. Text2Onto contains two modules of a Probabilistic Ontology Models (POMs) used to calculate probabilities for the concepts, and a data-driven change

discovery module responsively detecting changes in the corpus to improve the accuracy of the Ontology. Furthermore, Text2Onto facilitates the interaction with users to manually modify existing Ontologies.

III. Ontology construction system for Vietnamse

In this section, we describe our system for automatically building Ontology from Vietnamese text. Our system consists of a Syntactic analysis component and an Ontology extraction component as shown in figure 1.

A. Syntactic analysis component

The syntactic analysis component includes four main modules. The first two modules are used to detect noun phrases and phrases capturing relations between noun phrases from texts. Based on the detected phrases, the last two modules automatically identify candidate-phrases representing classes, individuals, relationships and properties in the Ontology.

This component is developed using the GATE [3] framework to detect potential phrases as semantic annotations. We wrapped existing linguistic processing modules for Vietnamese [12] such as Word Segmentation, Part-of-speech tagger GATE as plug-ins. Returned results of these modules are annotations capturing information such as sentences, words, nouns and verbs. Each annotation has a set of feature-value pairs. For example, a word corresponds a *TokenVn* annotation which has a feature *category* storing the word's part-of-speech tag. This information can then be reused for further processing in subsequent modules. Noun phrases, relation phrases and the information about classes, individuals and properties are identified by using patterns over existing linguistic annotations. Our modules are structured as JAPE (Java Annotation Pattern Engine) transducers in GATE, a set of cascaded JAPE grammars. A JAPE grammar allows one to specify regular expression patterns based on semantic annotations.

1) Noun phrases detection module: Ontology's classes and Ontology's individuals are normally expressed as noun phrases. Therefore, it is important that we can reliably detect noun phrases. Following [13][14], in noun phrases detection module, we determine noun phrases by utilizing JAPE grammars over *TokenVn* annotations. When a noun phrase is matched, an annotation *NounPhrase* is created to mark up the noun phrase. In addition, in order to identify the *Concept* or *Object* class that the noun phrase belongs to, we use the following heuristic:

If a noun phrase contains a single noun (not including numeral nouns) and does not contain a proper noun, it is a *Concept*. If a noun phrase contains a proper noun or contains at least three single nouns, it is an *Object*. Otherwise, concept or object class is determined using a manually constructed domain-dependent dictionary. This information is stored in the *NounPhrase* annotation's *type* feature.

2) Relation phrases detection module: This module is used to detect semantic relations between noun phrases. This information captures the relationships between potential individuals or classes in the Ontology. We exploit the following patterns to identify relation phrases:

((“có$_{have\|has}$”) (Noun Phrase$_{type==Concept}$) (“là$_{is}$”) | (“có$_{have\|has}$”) (Adjective) (“là$_{is}$”))$_{Relation-has}$
((Verb)+ (NounPhrase$_{type==Concept}$) (Preposition) (Verb)**?**)$_{Relation-noun}$
((Verb)+ ((Preposition) (Verb)**?**)**?**)$_{Relation-verb}$

When a relation phrase is matched by one of the patterns, an annotation is created to mark up this relation phrase. There are three types of relation annotations named *Relation-has*, *Relation-noun* and *Relation-verb*. Furthermore, we utilize the *hasNoun* feature of *Relation-has* and *Relation-noun* annotations to capture the noun phrases within the relation phrases.

3) Classes and Individuals detection module: In this module, we firstly identify the phrases indicating the classes. If a noun phrase is annotated by a *NounPhrase*$_{type==Concept}$ annotation, it contains a concept corresponding a class in Ontology. Moreover, for single noun within phrase covered by a *NounPhrase*$_{type==Object}$ annotation, if the single noun is not a proper noun and is not a numeral noun, it may also be a concept representing a class in the Ontology. A *Class* annotation is created to capture these *candidate*-concepts.

In the next step, we determine the noun phrases representing individuals in the Ontology. In general, noun phrases marked by *Nounphrase*$_{type==Object}$ annotations are candidates for individuals in the Ontology. In addition, in a noun phrase annotated by a *Nounphrase*$_{type==Object}$ annotation, the *Class* annotation is occasionally followed by a phrase which points to an individual.

4) Relationships and Properties detection module: Each individual in the Ontology has some properties connecting with other individuals. A property contains *domain* and *range* features which effectively represent the relationship between the range and corresponding domain. This module is used to determine the phrases that can be properties in the Ontology. From the detected relation phrases, the phrases annotated by *Relation-has*, *Relation-noun* or *Relation-verb* annotations are potential properties in the Ontology. The *hasNoun* feature in these annotations can be utilized to identify the domain and range of the corresponding property.

Following examples are patterns for detecting property phrases:

(Class)$_{Domain}$ (Individual)**?** (Relation-verb)$_{Property}$ (Class)$_{Range}$ (Individual)**?**
(Class)$_{Domain}$ (Individual)**?** (Relation-has$_{hasNoun \rightarrow Range})_{Property}$
(Relation-noun$_{hasNoun \rightarrow Domain})_{Property}$ (Class)$_{Range}$

A *Property* annotation is created with two features *Domain* and *Range* capturing its corresponding domain and

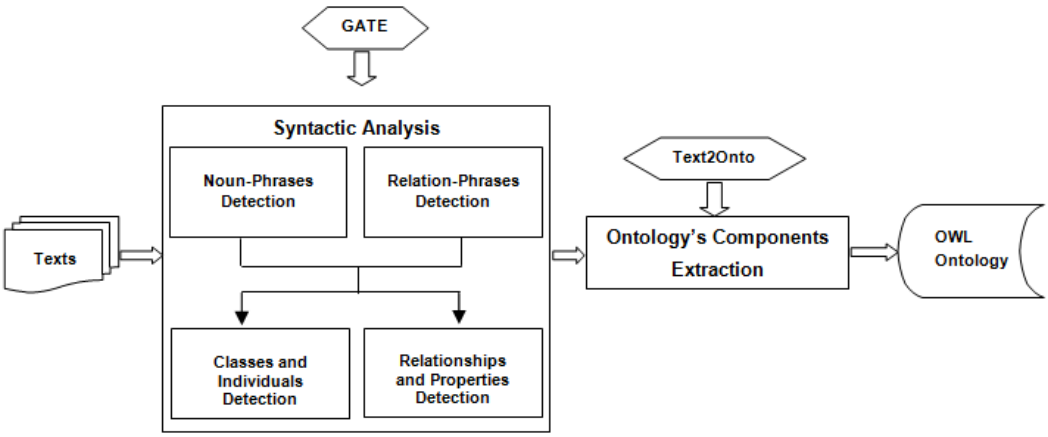

Figure 1. Architecture of our system to automatically build Vietnamese Ontology.

range.

We also determine phrases indicating the relationships between classes or between individuals and classes in the Ontology. Specifically, we need to identify that if a class is a subclass of another class, or whether an individual is an instance of a class. We create *SubClassOf* annotations to capture this subclass-attribution via features *SuperClass* and *SubClass*, and *InstanceOf* annotations to capture this instance-attribution via the *Class* feature.

The following patterns are used to identify the relationships:

((Class)$_{SuperClass}$ ({TokenVn.category == "An"}) | (Class)$_{SuperClass}$ ({TokenVn.category == "Aa"}))$_{SubClass}$
(Class)$_{SuperClass}$ (Individual)**?** (Relation-has$_{hasNoun \rightarrow SubClass}$)
(Relation-noun$_{hasNoun \rightarrow SubClass}$) (Class)$_{SuperClass}$
((Class) (Individual))$_{InstanceOf}$
((Relation-has$_{hasNoun \rightarrow Class}$) (Individual))$_{InstanceOf}$
((Individual) (Relation-noun$_{hasNoun \rightarrow Class}$))$_{InstanceOf}$

B. Ontology extraction Component

The first component generates *Class*, *Individual*, *SubClassOf*, *InstanceOf* and *Property* annotations from the input text documents. These annotations indicate the phrases that could potentially be classes, individuals and their relationships and properties in the Ontology. Based on these informations, the Ontology extraction component utilizes Text2Onto tool [4] to create the corresponding Ontology and export the Ontology into the OWL format.

Various Ontology learning algorithms of Text2Onto are used in our system to compute probabilities of output Ontology' elements. These probabilities are very useful when the users interact with the system to further improve the quality of the output Ontology. However, this semi-automatic approach with the intervention of expert users is beyond the focus of this paper.

IV. Experiments

In our experiment, we collected a Vietnamese text corpus of 434 sentences in the Vietnam Nation University organizational structure domain. This corpus is divided into 2 parts: the training corpus of 300 sentences and the test corpus of 134 sentences. We developed our system by manually creating the grammars based on a training corpus only. The quality of the system is evaluated by judging the quality of the generated Ontology based on the test corpus.

Using our system, Ontology is generated from the test corpus which we called *Auto-Ontology*. The *Auto-Ontology* includes 31 concepts, 29 properties, and 75 individuals. To evaluate the quality of the automatically generated Ontology, a goal standard ontology called *Manual-Ontology* is built manually by two people using the Protégé tool [6]. The *Manual-Ontology* contains 19 concepts, 17 properties, and 46 individuals. The *Auto-Ontology* is evaluated based on five factors:

- *Class factor*: The classes in *Auto-Ontology* are compared against the classes in *Manual-Ontology*. It is considered correct if the two classes cover the same phrase.
- *Individual factor*: The individuals in *Auto-Ontology* are compared against the individuals in *Manual-Ontology*. It is considered correct if the two individuals cover the same phrase.
- *Property factor*: The properties in *Auto-Ontology* are compared against the properties in *Manual-Ontology*. It is considered correct if the Property annotation and its domain and range features in both Ontologies cover the same phrases.
- *Instance factor*: The InstanceOf relationship between an individual and a class in *Auto-Ontology* are compared against its counterpart in *Manual-Ontology*. It is considered correct if the InstanceOf annotation and its features in both Ontologies cover the same phrases.

- *Subclass factor*: The SubClassOf relationship between classes in *Auto-Ontology* is compared against its counterpart in *Manual-Ontology*. It is considered correct if the SubClassOf annotation and its features in both Ontologies cover the same phrases.

We use $F_{measure}$ as a metric to measure the accuracy:

$$F_{measure} = \frac{2 * Recall * Precision}{Recall + Precision}$$

where *Precision* is defined as the ratio between the number of correct achieved results and the actual number of achieved results in *Auto-Ontology* while *Recall* is defined as the ratio between the number of correct achieved results and the total number of results in *Manual-Ontology*.

Table I
EXPERIMENT RESULTS FOR CLASSES, INDIVIDUALS, RELATIONSHIPS AND PROPERTIES

Cases	**Precision**	**Recall**	$\mathbf{F}_{measure}$ **(%)**
Class factor	19/31	19/19	76.00
Individual factor	46/75	46/46	76.03
Property factor	17/29	17/17	73.91
Instance factor	43/75	43/46	71.07
Subclass factor	13/31	13/19	52.00

Table I give the evaluation for classes, individuals, relationships and properties factors of the automatically constructed Ontology based on the test data. Because all of training and testing sentences are from narrow domain, the created JAPE grammars [3] with high generalization embedded engineer's knowledge are able to cover all phrases indicating the Ontology's components. This is the reason that the Recall results of Class, Individual and Property equal to 1.00.

V. CONCLUSION

In this paper, we describe an approach for automatically constructing Ontology from Vietnamese texts. Our system includes two components: syntactic analysis and Ontology construction components. Based on Gate framework [3], the syntactic analysis component detects phrases which capture the information about the classes, individuals, relationships and properties in the input texts. Subsequently, the Ontology extraction component uses Text2Onto [4] to generate the output Ontology.

Experimental results of the system are promising when evaluating the accuracy of the extracted classes, individuals and their relationships and properties in the Ontology. In the future, we will extend our grammars to provide better coverage for our system.

ACKNOWLEDGEMENTS

This work is partially supported by the Research Grant from Vietnam National University, Hanoi No. QG.10.23.

REFERENCES

[1] W. N. Borst, "Construction of engineering ontologies for knowledge sharing and reuse," Ph.D. dissertation, Enschede, September 1997. [Online]. Available: http://doc.utwente.nl/17864/

[2] T. R. Gruber, "Towards principles for the design of ontologies used for knowledge sharing," *Int. Journal Human-Computer Studies*, vol. 43, no. 5/6, 1995.

[3] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications," in *Proc. of ACL 2002*, pp. 168–175.

[4] P. Cimiano and J. Vlker, "Text2onto - a framework for ontology learning and data-driven change discovery," in *Proc. of NLDB 2005*.

[5] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubzy, H. Eriksson, N. F. Noy, and S. W. Tu, "The evolution of protégé: An environment for knowledge-based systems development," *International Journal of Human-Computer Studies*, vol. 58, pp. 89–123, 2002.

[6] H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen, "The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications," in *The Semantic Web – ISWC 2004*, 2004, pp. 229–243.

[7] H. Hu and D.-Y. Liu, "Learning owl ontologies from free texts," in *Proc. of 2004 International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 1233–1237.

[8] C. D. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[9] H. Kong, M. Hwang, and P. Kim, "Design of the automatic ontology building system about the specific domain knowledge," in *Proc. of ICACT 2006*, vol. 2.

[10] A. Maedche and S. Staab, "Ontology learning for the semantic web," *IEEE Intelligent Systems*, vol. 16, pp. 72–79, 2001.

[11] E. Maedche and S. Staab, "The text-to-onto ontology learning environment," in *Software Demonstration at ICCS-2000 - Eight International Conference on Conceptual Structures*, 2000.

[12] D. D. Pham, G. B. Tran, and S. B. Pham, "A hybrid approach to vietnamese word segmentation using part of speech tags," in *Proc. of KSE 2009*, pp. 154–161.

[13] D. Q. Nguyen, D. Q. Nguyen, and S. B. Pham, "A vietnamese question answering system," in *Proc. of KSE 2009*, pp. 26–32.

[14] D. Q. Nguyen, D. Q. Nguyen, and S. B. Pham, "Systematic knowledge acquisition for question analysis," in *Proc. of RANLP 2011*, pp. 406–412.