A BOOTSTRAPPING METHOD FOR AUTOMATIC CONSTRUCTING OF THE WEB ONTOLOGY INSTANCES AND PROPERTIES

Song-il CHA [1] and Myong-jin HAN [2]

[1] Information Science College, University of Sciences, Pyongyang, DPR Korea
[2] Department of Computer Systems, Pyongsong College of Technology, Pyongsong, DPR Korea

ABSTRACT

With the phenomenal growth of the Web resources, to construct ontologies by using existing resources structured in the Web has gotten more and more attention. Previous studies for constructing ontologies from the Web have not carefully considered all the semantic features of the Web documents. Hereby it is difficult to correctly construct ontology elements from the Web documents that are increasing daily. The machine learning methods play an important role in automatic constructing of the Web ontology. Bootstrapping technique is a semi-supervised learning method that can automatically generate many terms from the few seed terms entered by human. This paper proposes bootstrapping method that can automatically construct instances and data type properties of the Web ontology, taking proper noun as semantic core element of the Web table. Experimental result shows that proposed method can rapidly and effectually construct instances and its properties of the Web ontology.

KEYWORDS

Ontology, Bootstrapping, Instance, Property, Seed, Web tables.

1. INTRODUCTION

Today, the Semantic Web comprises techniques that promise to dramatically improve the current WWW and its use. With the emergence of the Semantic Web and the growing number of heterogeneous data sources, the benefits of ontologies are becoming widely accepted [1]. Accordingly, researches for using ontologies on the Web have become active [2]. Currently, in many cases, most Web ontologies are simpler than previous ontologies used in the design and diagnosis. The Web ontologies define terms used as data (metadata) for explaining things of a special domain. Manually setting ontology up would entail a lot of time, not to mention that there are only a handful of experts available. For this reason, researchers are paying attention to automatic transformation of the Web resources in the areas into ontologies [3, 4]. In order to provide the necessary means to widely apply ontologies to various fields there are today many proposals for using ontology learning and machine learning, and until now the study on domain ontology learning has been flourishing.

Rupasingha et al. [5] proposed a Web service clustering method through calculating the semantic similarity of Web services using ontology learning method. El Asikri et al. [6] described the commonalities of the areas, such as the semantic web and data mining, in order to resolve problem of extracting useful and shared knowledge, as well as solve the problem of the interoperability between Web systems by using the ontology learning from Web content. Rupasingha et al. [7] presented a method for calculating Web service similarity using both ontology learning and machine learning that uses a support vector machine for similarity calculation in generated ontology instead of edge count base method. Kumara et al. [8] proposed clustering approach that considers the complex data type as well as the simple type in measuring

the service similarity. This approach used hybrid term similarity method which proposed in their previous work to measure the similarity. Song et al. [9] reviewed the related concepts and methods of ontology construction and extension, proposed an automatic ontology extension method based on supervised learning and text clustering. This method used the K-means clustering algorithm to separate the domain knowledge, and to guide the creation of training set for Naive Bayes classifier.

Jupp et al. [10] presented Webulous that is an application suite for supporting ontology creation by design patterns, and provided simple mechanisms for the addition of new content in order to reduce the overall cost and effort required to develop ontologies. Peng et al. [11] proposed a method which can learn a heavy-weighted medical ontology based on medical glossaries and Web resources, in order to deal with heterogeneous knowledge in the medical field. Wei et al. [12] presented a semi-automatic construction method for agricultural professional ontology from web resources. For semi-structured web pages, this method automatically extracted and stored structured data through a program, built pattern mapping between relational database and ontology through human-computer interaction, and automatically generated a preliminary ontology, finally completed checking and refining by domain experts

The Web is an enormous resource of information contained in billions of individual pages. Most information resource on the Web is presented in the form of semi-structured or unstructured documents, encoded as a mixture of loosely structured natural language text and template units. Yu et al. [13] proposed a modified hierarchical concepts tree building method by applying pruning algorithm on the graph. They used the clue words to product queries containing hierarchical relation to get corpus rich in concepts hierarchical relation through the search engine from Web. Vasilateanu et al. [14] proposed a semantic search engine for relevant documents in an enterprise, based on automatic generated domain ontologies, with observing on the component for ontology learning and population. Manvi et al. [15] focused on generating domain specific ontology for retrieving hidden web contents. In this paper a knowledge base used in automatically filling up search interfaces for retrieving hidden web data.

The Web tables are used mainly for structuring information, and they are the strongest means of presenting structured information. The Table structures represent relations between data in the table. Therefore, ontologies can be easily extracted from a table by using structural features of the table [16]. However, understanding of table contents requires table structure comprehension and semantic interpretation, which exceed the complexity of corresponding linguistic tasks. Previous studies for constructing domain ontologies from the Web table are centralized to interpret table structure.

The comparatively comprehensive and complete model for the analysis and transformation of the tables is Hurst's [17]. This model analyzes the tables along graphical, physical, structural, functional, and semantic dimensions. Jung et al. [18] suggested a method for extracting table-schemata based on table structure and heuristics. Using this method, a table is converted into a table-schema and a triple. Chen et al. [19] employ heuristic rules to filter out non-genuine tables from their test set and make assumptions about cell content similarity for the table recognition and interpretation. Wang et al. [20] proposed a machine learning based approach to classify given table entity as either genuine or non genuine. Pivk et al. [21] focused on understanding table-like structures only due to their structural dimension and transforming the most relevant table types into F-logic frames. Tijerino et al. [22] described the automatic generation of ontologies from the normalized tables, which is a structure they got after normalizing table-equivalent data. Tanaka et al. [23] proposed a method for extracting relations based on interpretations given by humans, in order to interpret structures of each tables correctly. This method is easy to apply to tables in various domains because it uses interpretations given by humans and generalized table structures instead of a domain-specific knowledge base. Jung et al. [24] detected that, generally, a table

provides a semantic core element in a HEAD, and proposed a method for automatically extracting domain ontology using heuristics for extracting table schemata based on semantic core element.

As specified above, most research endeavors to interpret the table by using structural characteristics of the table. But, most Web tables are designed by humans, thus, it has a certain limit to automatically interpret table using only structural information of the table. Though Jung et al. [24] proposed heuristics for detecting semantic characteristics based on the location of the table cells, they did not mention which becomes semantic core element.

Through the observation about semantic features of the table, it is found that if there are proper nouns on the table, then they can become a semantic core element. So this paper focuses on the proper noun extraction method, which is a pre-requirement for interpretation of table structure based on proper nouns. That is, this paper proposes an automatic extraction method of the instance composed of proper nouns. Bootstrapping-based semi-supervised learning method aims to rapidly and accurately obtain brief domain ontology from the table cells consisting of proper nouns [25, 26].

Bootstrapping method, which aims at automatically generating instances and their relations in a given domain, is a promising technique for ontology creation. H. Davulcu et al. [27] proposed the OntoMiner system which offers automated techniques for creating ontologies based on a small collection of relevant Web sites. The work presented an approach for bootstrapping and populating large, rich, and up-to-date domain ontologies that organize the most relevant concepts, their relationships, and instances (which correspond to members of concepts). W. S. Wu et al. [28] presented the DeepMiner system which learns domain ontologies from the source Web sites. Given a set of sources in a domain of interest, DeepMiner first learns a base ontology from their query interfaces. It then grows the current ontology by probing the sources and discovering additional concepts and instances from the data pages retrieved from the sources. A. Segev et al. [29] proposed an ontology bootstrapping process for web services. The proposed ontology bootstrapping process integrates the results of two methods, namely Term Frequency/Inverse Document Frequency (TF/IDF) and web context generation, and applies a method to validate the concepts using the service free text descriptor, thereby offering a more accurate definition of ontologies. F. Keshtkar et al. [30] presented a novel semantic bootstrapping framework that uses semantic information of patterns and flexible match method. The work considerably enhance based on iterative bootstrapping model which generally implies semantic drift or low recall problem.

Through the experimental observation about semantic features of the Web tables, it is found that if there are proper nouns on the table, then they can become a semantic core element. The author proposes algorithms to automatically construct all instances or properties belonging to a given class, taking few terms belonging to class composed of proper noun as the seed. A bootstrapping method is proposed to construct ontologies with the instances and properties. The paper focuses on the extracting instances and properties based on interpreting the table contents by using structural and semantic characteristics of the table.

The paper is structured as follows: Section 2 presents an automatic generation method of the instance belonging to given class from Web tables; Section 3 describes an automatic property generation method based on proper noun extraction; Section 4 evaluates our method according to the experimental result; Finally, Section 5 provides conclusions to our work.

2. AUTOMATIC INSTANCE CONSTRUCTION

The knowledge on the domain terminology is required in order to manually construct ontology about products that are increasing daily such as CD / DVD, software etc., are not known already.

Thus, it is highly regarded to automatically build an ontology based on Internet resources of a given area. But, in order to automatically generate semantically correct ontology, certainly, must to be based on some clues. If to extract instances from the Web table does not depend on any clue and uses only structural information of the table, it is difficult to determine quality of the extracted instance. Through experimental interpreting of Web table structure, it is found that if there are proper nouns on the table, then they can be as the semantic core element, that is, as the instance. The semantic core element is a head cell that plays the role of the 'pivot' in understanding table structure [24]. Once proper nouns are extracted from a table, table structures can more accurately interpret focusing on proper nouns that are semantic core element of the table. Therefore, this section describes method generating automatically other instances in the table, taking some proper nouns such as already familiar product name as a clue. That is, this method is an approach which extracts the rest instances in the same class by using the proper noun extraction method, having some instances given by the user as a seed.

2.1. PROPER NOUN EXTRACTION MODEL BASED ON BOOTSTRAPPING

In this section, the proper noun extraction method employs for automatic extraction of the instance. This method is an approach, that if there is the row or the column composed by the proper noun in the Web table, then, considering it as instance, the proper nouns are extracted. That is, the proper noun extraction means to extract the other terms guessed belong to the class which the entered word (proper noun) belongs to. This section presents the proper noun extraction method using bootstrapping.

Bootstrapping is shown as follows: firstly, generate a pattern from the document in accordance with a small amount of seed terms, then using this pattern again extract other words from the document, and lastly using the extracted terms create another word. A large amount of terms can be extracted from a small amount of seed terms by a repeat of this process. To begin with we define the fundamental notions.

A proper noun p is a noun that is the name of a specific individual, place, or object. For example, personal name, country name, denomination name, organization name, and so on. A proper noun set P is a set of proper nouns, i.e. $p \in P$.

A seed term s is a proper noun specified artificially before proceeding with automatic learning. A seed term set S is a set of the seed terms, i.e. $s \in S$.

From the above definition, we can know that $s \in P$ and $S \subset P$. For example, Table 1 shows the seed term set which belongs to each class. Herein, first column is class name and second column is the proper nouns which belong to the class. For example, desktop is a name of desktop class, and Acer, Asus and Compaq are the instance (manufacturer brand) which belongs to the desktop class.

A domain table set T is a set of genuine tables which is chosen in the Web tables of a given domain, collected using search engine such as Google or Yahoo. In this paper, in order to obtain genuine tables of given domain, the algorithm proposed in the previous study is used [18]. Withal, search keys for obtaining of the domain table set are a domain name (that is, a class name) and the seed term set selected by user in the beginning.

Table 1. An example of the seed term belonging to a given class.

Desktop	Acer, Asus, Compaq, Dell, eMachines, Everex, Gateway, HASEE, HP, Lenovo, Panasonic, Samsung, Sony, TCL, Toshiba, etc.
Digital Camera	Agfa, Canon, Casio, Contax, Epson, FujiFilm, HP, Kodak, Konica Minolta, Kyocera, Leica, Nikon, Olympus, Panasonic, Pentax, Ricoh, Samsung, Sanyo, Sigma, Sony, Toshiba, etc.
LCD TV	Akai, AOC, Axion, Benq, Casio, Dell, Diamond, Epson, Gateway, GPX, Haier, Hewlett Packard, Hitachi, Honeywell, Hyundai, JVC, Konka, LG, Mitsubishi, NEC, Nikon, Panasonic, Philips, Samsung, Sanyo-Fisher, Sharp, Skayworth, Sony, Toshiba, etc.
Publishing Company	Pearson, Reed Elsevier, ThomsonReuters, Wolters Kluwer, Bertelsmann, Hachette Livre, McGraw-Hill Education, Grupo Planeta, De Agostini Editore, Scholastic, Houghton Mifflin Harcourt, Holtzbrinck, Cengage Learning, Wiley, Informa, HarperCollins, Shogakukan, Shueisha, Kodansha, Springer Science and Business Media, etc.
Chinese Province	Anhui, Shandong, Guangdong, Jiangsu, Hunan, Hubei, Liaoning, Shanxi, Inner Mongolia, Tianjin, Ningxia, etc.
Country	Brazil, Canada, China, France, Germany, India, Indonesia, Italy, Poland, Spain, Thailand, UK, Ukraine, etc.

The proper noun extraction model based on bootstrapping is shown below.

This model extracts automatically new proper nouns based on the few seed terms from the domain table set. In this work, the model is named as IC-Model (Instance Construction-Model). Figure 1 shows IC-Model. In the model, the dotted line arrow denotes a process for extracting a pattern that contain the initial seed term. The extraction of the pattern from the domain table set only needs to be determined once.

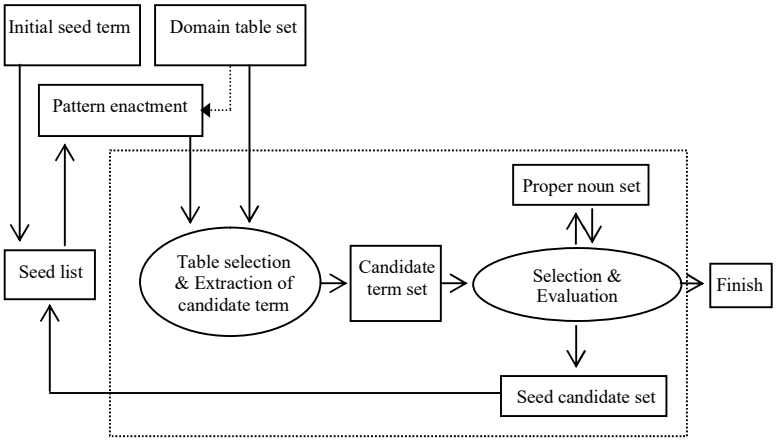

Figure 1. IC-Model

The input of the IC-Model is the domain table set T and the initial seed term set S, the output of this model is the proper noun set. A quality of the seed term set greatly affects the accuracy of the extracted proper nouns. Therefore, users must select more obvious and important terms for the initial seed term set. In addition, through experimental study, it is confirmed that can increase the accuracy of the proper noun extraction in the case of that N_s (the number of the seed terms) is

three or more. The number of the seed list used in this model is always N_s. A threshold value N of the seed list is the number of an old seed term among the seed terms entered in the each loop of bootstrapping. In the beginning, N=N_s. Constitute the threshold value N of the seed list as greater one than two. This means that it must contain at least two initial seed term in the seed terms used at the every loop. That is, can effectively extract terms when N≥2. But if not, can't guarantee accuracy of the extracted terms. The seed list is possible combinations of initial seed terms composed of greater one than two.

At the first loop period of bootstrapping, this model extracts terms taking N_s initial seed terms. But, at the second loop period, it extracts possible candidate terms taking N_s-1 initial seed terms and a new seed candidate. Likewise, at the third loop period, it extracts possible candidate terms taking N_s-2 initial seed terms and two new seed candidates. The loop period finishes when N<2.

In the pattern production process, are instituted the TABLE tags that wrap each of the seed term from the table. In the table selection and extraction process of the candidate term, model selects tables that contain the seed term pattern from the domain table set and extract all rest cells which appears with the seed terms in same row or same column of the table. In the selection and evaluation process, model firstly, only adds the obtained candidate terms which do not overlap to the proper noun set. Secondly, selects the seed candidates for next loop among the proper noun set and add to a seed candidate set. Finally, evaluates the threshold value N of the seed list, and finishes when N<2, if not repeats the loop period.

2.2. INSTANCE CONSTRUCTION ALGORITHM

This subsection proposes the detailed algorithm for acquiring the instances based on IC-Model. The input of this algorithm is the domain table set T and three initial seeds s_1, s_2, s_3. The output of this algorithm is the acquired instance set P. Our goal is to rapidly and accurately generate the domain ontology. Therefore, three seed terms are used, i.e. N_s=3. Algorithm 1 shows the instance acquisition algorithm.

Algorithm 1. Instance construction algorithm

1: Input S,T
2: Choose s_1,s_2,s_3∈S
3: X←{s_1,s_2,s_3}, X_0←{s_1,s_2,s_3}, P←s_1 ∪s_2 ∪s_3, C_0←Ø
4: Create pattern which wraps S from T
5: Construct candidate term set C from the table in T by using pattern that wraps X_0
6: //Evaluation and selection
7: // Evaluation
8: From C remove terms that don't correspond to evaluation condition
9: From C remove all c that c∈C ∧ c∈P , it's result also remains to C
10: P←P∪C
11: **if** isn't $|X| \geq 2$ **then**
12: output P, and finish
13: **end if**
14: // Selection
15: **if** processed all elements in C_0 **then**
16: from S select new element pair and replace X by it;
17: **If** C_0=Ø **then**
18: C_0←C
19: **end if**
20: **end if**
21: Select new seed candidate set X_0 from X and C_0

22: **go to** 5

In algorithm 1, X is an element pair which consisted by arbitrary combination of elements in the seed term set S, for example $\{s_1, s_2, s_3\}$,$\{s_1, s_2\}$,$\{s_1, s_3\}$,$\{s_2, s_3\}$,$\{ s_1\}$,$\{ s_2\}$,$\{ s_3\}$, and so on. C_0 is a candidate term set acquired when the seed term is $\{s_1, s_2, s_3\}$. C is a candidate term set which is consisted at the rest every loop period. X_0 is a seed term set that used as input of every loop period. In the beginning, the algorithm enters the bootstrapping process, taking $\{s_1, s_2, s_3\}$ which consist of three terms. Then add the new acquired terms to the candidate term set C. In the second loop, takes two old (before used) terms and a newly acquired term as a seed term. Because it can't be considered that the newly acquired terms are certainly right instance. This will prevent that the accuracy is lower, and take another new elements as a seed. Fundamental steps of algorithm are shown as follows:

Pattern enactment. In order to produce pattern, it must be found TABLE tags which wraps each of the seeds from arbitrary table of the domain table set. For example, when {Hunan, Hubei, Shanxi} is a seed term, it must be taken <td>Hunan</td>, <td>Hubei</td>, <td>Shanxi</td> which is wrapped by TABLE tag as a pattern. That is, can be constituted the <td>term</td> used for defining the table cells as a pattern.

Every table's creator is using TABLE tag of different description form, but this algorithm employs only <td>…</td> tag which denotes the table cell for the purpose of pattern enactment. Withal, for a brief descriptive purpose, is considered only the case of nonuse of <td> *tag attribute* such as ALIGN, VLIGN, EIDTH, HIGHT, BACKGROUND, BTCOLOR, and so on. Figure 2 shows the example of the Web table that contains above three seeds.

```
<table border>
  <tr>
<th>Chinese provinces</th>
<th><p>Total population</p>
      <p>(million)</th>
<th><p>Populstion density</p>
      <p>(people/Km²)</th>
  </tr>
  <tr>
<td>Shandong</td>
<td>86.2</td>
<td>562</td>
  </tr>
  <tr>
<td>Guangdong</td>
<td>72.6</td>
<td>343</td>
  </tr>
  <tr>
<td>Jiangsu</td>
<td>68.0</td>
<td>663</td>
  </tr>
<tr>
  <td>Hunan</td>
  <td>62.5</td>
  <td>306</td>
</tr>
<tr>
  <td>Anhui</td>
  <td>58.7</td>
  <td>420</td>
</tr>
<tr>
  <td>Hubei</td>
  <td>55.9</td>
  <td>301</td>
</tr>
<tr>
  <td>Liaoning</td>
  <td>39.8</td>
  <td>273</td>
</tr>
<tr>
      <td>Shanxi</td>
      <td>29.6</td>
      <td>189</td>
</tr>
<tr>
      <td>Inner Mongolia</td>
      <td>22.0</td>
      <td>19</td>
</tr>
<tr>
      <td>Tianjin</td>
      <td>8.9</td>
      <td>787</td>
</tr>
<tr>
      <td>Ningxia</td>
      <td>4.9</td>
      <td>95</td>
</tr>
  </table>
```

Figure 2. An example of a table contained the proper nouns

Acquiring the candidate term. In order to construct the candidate term set, we assume as follows: If certain term appears with the proper noun in same row or same column of a table, then this term also is a proper noun. Therefore, are taken the row or column of table as object of candidate term extraction. For example, the second column of Figure 3 contains the proper noun "Wuhan" that is known already. Withal, another terms on this column are proper nouns belonging to "Chinese cities" class. Before extracting the candidate term, must be determined table reading orientation, i.e. *row wise* or *column wise* [19]. For example, if the first row of the table consists of the attribute cells, and the others are value cells, then this table is the column wise. If the table is row wise, then take each cell of the row in which exists proper noun as candidate term, and if the

table is column wise, then take each cell of the column in which exists proper noun as candidate term.

Evaluation. In order to extract more accurate candidate term, must be resolved table HEAD. Withal, must be determined candidate term extraction range in the row (or column) in which exists proper noun, that is, must be resolved value region corresponding to the table HEAD. Figure 4 is an example in which exists two different HEADs (1st row and 13rd row). For the purpose of extracting candidate term taking above proper noun as a seed, must be restricted range of value region, i.e. there is candidate extraction object between 2nd row and 12nd row. But if not, it has possibility to extract irrelevant terms. In this example, "Mexico", "Germany", and so on are irrelevant term. Therefore, in this step it is evaluated whether each term *c* of the candidate term set *C* is a term belonging to a given class, and choose only fit terms. Withal, in the evaluation step, evaluate whether newly extracted proper noun exists in the proper noun set *P*, and only add proper noun which does not overlap to *P*.

No	Chinese cities	Total population (million)	Land area (sq km)
1	Shanghai	12.89	5,299
2	Beijing	10.98	12,484
3	Chongqing	10.18	7,154
4	Wuhan	7.86	8,494
5	Tianjin	7.64	7,418
6	Guangzhou	6.00	3,718
7	Xi'an	5.16	3,547
8	Nanjing	5.01	4,723
9	Shenyang	4.92	3,495
10	Shantou	4.80	1,954
11	Chengdu	4.65	2,177
12	Hangzhou	4.02	3,068
13	Harbin	3.95	4,272
14	Foshan	3.51	3,848
15	Jinan	3.42	3,257
16	Changchun	3.15	3,603
17	Tangshan	2.97	1,230

Figure 3. An example ofa table which proper nouns are located in the coloum wise

Chinese provinces	Total population (million)	Population density (people/Km²)
Shandong	86.2	562
Guangdong	72.6	343
Jiangsu	68.0	663
Hunan	62.5	306
Anhui	58.7	420
Hubei	55.9	301
Liaoning	39.8	273
Shanxi	29.6	189
Inner Mongolia	22.0	19
Tianjin	.9	787
Ningxia	4.9	787
Countries of comparable population size	**Total population (million)**	**Population density (people/Km²)**
Mexico	84.5	43
Germany	79.4	222
Iran	58.9	36
Italy	57.0	189
France	56.7	103
Spain	39.3	78
Canada	27.8	3
Australia	16.9	2

Figure 4. An example of the table with two head

3. AUTOMATIC PROPERTY CONSTRUCTION

3.1. OBSERVATION ON THE LOCATION OF THE PROPERTIES

It is difficult to extract inclusively, if property also as well as instance does not depend on any clue. Therefore, in this section is proposed a method for automatically constructing properties belonging to the class using proper noun extraction approach based on bootstrapping.

In order to extract the property from the Web table, firstly, based on the proper noun extraction method mentioned above, extract the instance belonging to the class, and take a set of these instances as P. Then, determine three initial seed property belonging to the class, and denote them as a_1, a_2, a_3. Denote a set of the properties as A, i.e. a_1, a_2, $a_3 \in A$. For example, in the first row of the table on Figure 5, attributes such as Type, Standard, Resolution, and so on, are property belonging to a camera class, and cells (but the first cell is excepted) of the first column are instances of a camera Model class.

l	Type	Standard	Resolution	Sensor	Frame/ Line Rate	Interface	Light Spectrum
CV-S3200N	1-CCD Color Interlaced	NTSC	758 X 486	1/2”	30	Analog	Visible
CV-S300P	1-CCD Color Interlaced	PAL	737 X 575	1/2”	25	Analog	Visible
CV-S3300N	1-CCD Color Interlaced	NTSC	758 X 486	1/3”	30	Analog	Visible
CV-S3300P	1-CCD Color Interlaced	PAL	737 X 575	1/3”	25	Analog	Visible
CB-040MCL	1-CCD Color Progressive	SVGA	776 X 582	1/2”	60	Mini-CL	Visible
CB-040GE	1-CCD Color Progressive	SVGA	782 X 582	1/2”	60	GigE Vision	Visible
CB-080GE	1-CCD Color Progressive	XGA	782 X 582	1/3”	30	GigE Vision	Visible
BB-500CL	1-CCD Color Progressive	QSXGA	1024 X 768	2/3”	15	Camera Link	Visible

Figure 5. An example of the table that contains the several attributes

As shown in the figure, when the table is column wise, generally, the properties are constructed by the cells which lie at the first row of the table. When extracting the property using the proper noun extraction method, the seed term contains three properties and one instance which is selected from instances collected already. In each product there are special and common attributes. For example, not only digital camera but also computer has "resolution" attribute, that is, this is the common attribute. When are taken only three properties as the seed property, it has possibility which can extract even properties belonging to other class. Thus, is added one instance to the seed term set *S*, in order to extract rightly the property belonging to given class.

3.2. PROPERTY CONSTRUCTION ALGORITHM BASED ON THE IC-MODEL

The property construction algorithm based on the IC-Model similar to the instance construction algorithm (algorithm 1). The properties discussed in this subsection are *Owl:DatatypeProperty* [31, 32].

Algorithm 2 shows the property construction algorithm.

Algorithm 2. Property construction algorithm

1: Input A,P,T
2: $R \leftarrow s_1 \cup s_2 \cup s_3$
3: Create pattern which wraps A and P from T
4: $X \leftarrow \{s_1,s_2,s_3\}$; Take arbitrary $p \in P$, $X_0 \leftarrow \{p,s_1,s_2, s_3\}$, $C_0 \leftarrow \emptyset$
5: Construct candidate property set C from the table in T by using pattern that wraps X_0
6: // Evaluation and selection
7: // Evaluation
8: From C remove properties that isn't DatatypeProperty
9: From C remove all c that $c \in C \wedge c \in R$, it's result also remains to C
10: $R \leftarrow R \cup C$
11: **if** isn't $|X| \geq 2$ **then go to** 21
12: // Selection
13: **if** processed all elements in C_0 **then**
14: from A select new element pair and replace X by it
15: **if** $C_0=\emptyset$ **then**
16: $C_0 \leftarrow C$
17: **end if**
18: **end if**
19: Select new seed property set X_0 from P, X, C_0
20: **go to** 5
21: **if** processed all elements in P **then**
22: output R, and finish
23: **else go to** 4
24: **end if**

In algorithm 2, X is a element pair which consisted by arbitrary combination of elements in the seed property set A. C is a candidate property set which consisted at the every loop period. R is a property set. X_0 is a seed term set that it used as input of every loop period.

In the beginning, the algorithm enters the bootstrapping process, taking $\{s_1, s_2, s_3\}$ which consist of three properties and an instance of P. In the second loop, the algorithm takes two old (before used) properties, a newly extracted property, and an instance of P as a seed term. If all elements of property set were processed, then replace the instance by a new element and repeat process. In the subsection below, stepwise instantiate major steps of algorithm.

Pattern enactment. In order to produce pattern, must be looked for TABLE tags which wrap each of the properties from arbitrary table of the domain table set. The HEAD tag of the table is often denoted by <td>term</td>, <td>term</td>, <td *tag attribute*>term</td>, <th>term</th>, <th *tag attribute* >term</th>, and so on. At the HEAD of the table, *tag attribute* is used for the purpose of standing out. In order to denote HEAD cell, every table's creator uses table tag of different description form, but for a brief descriptive purpose, is employed only <th *tag attribute* >term</th> pattern (see Figure 2).

Acquiring the candidate property. In order to construct the candidate property set, we assume as follows: If certain term appears with the seed property in HEAD row or HEAD column of a table, then this term also is a property. Therefore, are taken the HEAD row or HEAD column of the table as the object of candidate property acquisition.

Evaluation. In this step, firstly, remove properties that are not DatatypeProperty from the candidate property set. That is, if "total" or "other" appears at the HEAD of the table, then remove it from the property candidate. Denote the term set which isn't DatatypeProperty by "*NonwordList*", i.e. it is needed $c \notin$NonwordList. Then, from HEAD row cell remove the element that <th> tag's attribute is COLSPAN. That HEAD element generally has hierarchical structure, thus this is not DatatypeProperty that must be obtained. Withal, in this step, evaluate whether newly acquired property exists in the property set R, and only add property which does not overlap to R. In the case of property acquisition, also institute threshold value N of seed property set A as greater one than two.

4. EMPIRICAL EVALUATION

4.1. EMPIRICAL EVALUATION OF THE ALGORITHM 1

Automatic instance construction method (algorithm 1) mentioned above is suited to obtain instances, which are severely changed, such as new products. In this experiment, in order to correctly construct the seed term set, we selected twenty publishing companies based on revenue rank of the world's largest publishing companies in 2008 from the Web pages. Figure 6 shows a Web table in which second column contains the instances belonging to a "publishing company" class, i.e. the seed term set is consisted of cells in the second column of this table.

Firstly, in order to obtain an experimental data, must be chosen initial seed belonging to "publishing company" class. If number of the initial seed is many, then accuracy is enhanced. Through experimental observations, it is confirmed that when take two initial seed terms, a large amount of irrelevant terms are extracted and the accuracy was decreased. Hence, are selected already known three initial seeds, i.e. {McGraw-Hill Education, Wiley, Reed Elsevier}. Next, in order to obtain the domain table set, the method collects arbitrary two hundred Web tables among around twenty thousand Web pages of the "publishing" domain acquired using search key "publishing company", "McGraw-Hill Education", "Wiley" and "Reed Elsevier" of Google search engine. After that, according to the algorithm 1 proposed in the section 2, the method acquires the instances belonging to the "publishing company" class. In this case, must be taken <td> McGraw-Hill Education </td>, <td> Wiley </td>, <td> Reed Elsevier </td> which iswrapped by <td> tag and </td> tag used for defining the table cells as a pattern.

Previous researches endeavored to interpret the tables using structural characteristic, hence, it is difficult to experimentally compare with our method. Withal, it is troublesome to rightly measure recall and precision of the experimental results obtained by our method. Therefore, in this experiment, we evaluated that the proposed instance construction algorithm how to effectively

increase the instances about every time operation. For the above purpose, we compared with twenty instances extracted from Figure 6, i.e. watch obtaining process of correct twenty instances.

RANK	COPANY NAME	PARENT COMPANY	NATIONALITY	REVENUES
1	Pearson	Pearson	UK	5,044
2	Reed Elsevier	Reed Elsevier	UK/NL/US	4,586
3	ThomsonReuters	The Woodbridge Company Ltd	Canada	3,485
4	Wolters Kluwer	Wolters Kluwer	NL	3,374
5	Bertelsmann	Bertelsmann	Germany	2,980
6	Hachette Livre	Lagardere	France	2,159
7	McGraw-Hill Education	McGraw-Hill	US	1,794
8	Grupo Planeta	Grupo Planeta	Spain	1,760
9	De Agostini Editore	Gruppo De Agostini	Italy	-
10	Scholastic	Scholastic Corp	US	1,499
11	Houghton Mifflin Harcourt	Education Media and Publishing Group	US/Cayman Islands	1,712
12	Holtzbrinck	Verlagsgruppe Georg von Holtzbrinck	Germany	-
13	Cengage Learning	Apax Partners et al.	UK	1,172
14	Wiley	John Wiley & Sons	US	1,139
15	Informa	Informa	UK	1,028
16	HarperCollina	News Corp	US/AUS	944
17	Shogakukan	Shogakukan	Japan	927
18	Shueisha	Shueisha	Japan	902
19	Kodansha	Kodansha	Japan	886
20	Springer Science And Business Media	Cinven and Candover	UK/Germany/Italy/ France	880

Figure 6. An example of the table of the publishing domain extracted from the Web page

In order to evaluate the experimental results, were used the *recall* and *precision* metrics as follows:

$$Recall = \frac{Number\ of\ acquired\ correct\ instance}{Total\ number\ of\ correct\ instances} \quad (1)$$

$$Precision = \frac{Number\ of\ acquired\ correct\ instance}{Total\ number\ of\ acquired\ instances} \quad (2)$$

The recall and precision metrics can be interpreted as follows:

Firstly, we discuss the recall. In the case of the simple input, the system registers singly the correct instance depending on the input sequence. Therefore, the recall is proportional to input time, in the case of seventeenth input only arrive at 100 percent. In the case of our proper noun extraction, is extracted a large amount of candidate instances at first time, at second time the recall arrives at 64 percent. However, at next time the candidate term is overlapped, and rising speed is slow, at sixth time it arrives at 100 percent

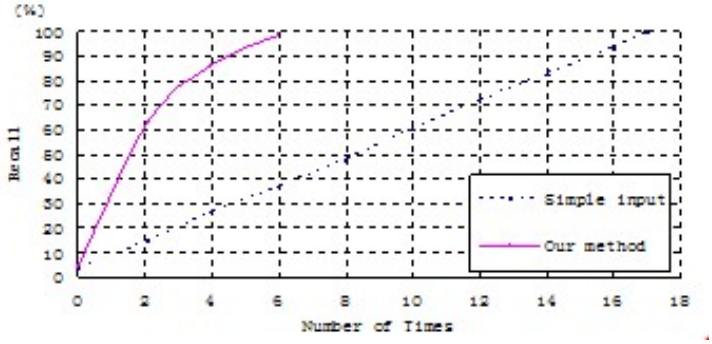

Figure 7. The comparison of the experimental results

Through this, it is confirmed that the proper noun extraction method can increase effectively the instance from a small amount of user input.

Next, we discuss the precision. In the case of the simple input, the precision is 100 percent. But, in the case of our instance extraction based on IC-Model, the precision decreases and at seventeenth time almost arrives at 94 percent. A cause that the precision does not arrive at 100 percent is that was included the irrelevant terms.

4.2. EMPIRICAL EVALUATION OF THE ALGORITHM 2

The property construction method (algorithm 2) employs the instances which extracted by the instance construction algorithm (algorithm 1). For an experiment, are selected products which have comparatively many attributes such as mobile phone, digital camera, LCD TV, LCD monitor and CRT monitor. Next, in order to obtain the domain table set, we collected around ten thousand Web pages acquired using individually search key “mobile phone”, “digital camera”, “LCD TV”, “LCD monitor” and “CRT monitor” of Google search engine. Table 2 shows the representative attributes belonging to each product class.

In order to illustrate property acquisition process, the experiment takes the mobile phone as an example. Firstly, the experiment selects three attributes belonging to the mobile phone as the seed property, i.e. {dimensions, weight, display size}. Next, extract the instances belonging to mobile phone using the algorithm 1. Then, according to the property construction algorithm (algorithm 2), extract the properties taking {Nokia, dimensions, weight, display size} as the seed term. In this seed term, Nokia is an instance of the “mobile phone” class.

Next, in order to obtain more correct table set, the experiment choose again the Web pages which contain “Nokia”, “dimensions”, “weight” and “display size” from around two thousand Web pages acquired by search key “Mobile phone”.
When {Nokia, dimensions, weight, display size} is a seed term, it can be taken <td>Nokia</td> and <th>dimensions</th>, <th>weight</th>, <th>display size</th> which is wrapped by HEAD tag as a pattern. Then, according to the algorithm 2 proposed in the section 3, the experiment acquires the properties belonging to the “Mobile phone” class.
Likewise, can be extracted properties belonging to other class. Table 3 shows total experimental result.

Table 2. An example of the correct attribute set.

Mobile phone	2G Network, 3G Network, Dimensions, Weight, Display type, display size, Alert types, Internal memory, Card slot, GPRS, EDGE, WLAN, Bluetooth, USB, Camera, OS, CPU, Messaging, Browser, Games, Colors, GPS, Battery, etc.
Digital camera	Resolution, Bit depth, Lens system, Picture mode, Image control functions, Memory card compatibility, Shutter speed, Zoom capability, Flash, Focus, Shooting mode, LCD display, Sound capture, Connectivity, Power supply, System requirements, Dimensions, Weight, Model, etc.
LCD TV	Display type, Resolution, Screen size, Aspect ratio, Dynamic contrast ratio, Response time, Viewing angle, Audio output power, USB, HDMI ports, TV size, etc.
LCD monitor	Screen type, Viewable size, Resolution, Backlight, Color depth, Contrast ratio, Viewing angle, Response time, Luminance, Video input, Horizontal/vertical frequency, Power consumption(On), Power consumption(Off), On screen display, Controls, Power, Dimensions, Weight, Safety and EMI, VESA compliant, Kensington lock ready, Integrated audio, Miscellaneous, etc.
CRT monitor	Screen type, Viewable size, Resolution, Dot pitch, Face, Video input, Horizontal/vertical frequency, Controls, Dimensions, Safety and EMI, Miscellaneous, etc.

Table 3. Recall and precision for acquired properties.

Product class	No. of web tables	Experimental result	
		Recall (%)	Precision (%)
Mobile phone	156	95.4	91.8
Digital camera	139	93.0	89.6
LCD TV	112	89.2	88.7
LCD monitor	134	94.7	91.2
CRT monitor	97	90.8	90.3

As shown in the table 3, the average recall of our method arrives at 93 percent. Through this, it is confirmed that proposed method can obtain almost the properties belonging to a given class. Withal, the average precision arrives at 90 percent. To extract the properties from the Web pages which describe only actual attributes of the product is the most ideal. But, in this experiment, did not choose from the Web pages applied like that.

5. CONCLUSIONS

This paper has presented an automatic construction method of instances and properties belonging to a given class from the Web table resources based on bootstrapping. Our approach was able to acquire fast a great amount of instances and properties using bootstrapping from a small amount of seed terms entered by human. When interpreting and straightening out table structure according to ontology schemata, to determine accurately class, instance and property gives a decisive impact on the use of ontologies in the future. Though this paper briefly extracted only instances composed of proper nouns and DatatypeProperty, we consider that this approach offer a good basis not for construction of arbitrary domain ontology fit with our intent, but for right semantic interpretation of table structure. In our future work, a more precise method for extracting ObjectProperty and other relationships will be developed based on extracted proper nouns.

ACKNOWLEDGEMENTS

This work was supported in part by the science & technology development fund of University of Sciences (100/425-63053-243/1). The authors are grateful to our colleagues at Centre for Natural Science Research and Information Science College for their help in collecting the Web pages and for discussing. They also would like to thank the anonymous reviewers for their careful reading and in-depth criticisms and suggestions.

REFERENCES

[1] G. Antoniou, van F. Harmelen, (2008) A semantic web primer, (Second Edition). Cambridge, MA: MIT Press.

[2] J. M. Serrano, (2012) Applied Ontology Engineering in Cloud Services, Networks and Management Systems. Springer.

[3] F. Zhang, J. W. Cheng, Z. M. Ma, (2016) “A survey on fuzzy ontologies for the Semantic Web”. Knowledge Engineering Review, Vol. 31, No 3,pp278-321.

[4] K. D. Mogotlane, J. V. Fonou-Dombeu, (2016) "Automatic conversion of relational databases into ontologies: a comparative analysis of protégé plug-ins performances". International Journal of Web & Semantic Technology (IJWesT), Vol.7, No.3/4, pp21-40.

[5] R. A. H. M. Rupasingha, I. Paik, B. T. G. S. Kumara, T. H. A. S. Siriweera, (2016) "Domain-aware Web Service Clustering based on Ontology Generation by Text Mining". In Proc. of the 7th IEEE Annual Information Technology, Electronics & Mobile Communication Conference.

[6] M. El Asikri, J. Laassiri, S. D.Krit, H. Chaib, (2016) "Contribution To Ontologies Building Using the Semantic Web and Web Mining". In Proc. of the International Conference on Engineering & Mis.

[7] R. A. H. M. Rupasingha, I. Paik, B. T. G. S. Kumara (2015) "Calculating Web Service Similarity using Ontology Learning with Machine Learning". In Proc. of the IEEE International Conference on Computational Intelligence and Computing Research, , pp. 201-208.

[8] B. T. G. S. Kumara, I. Paik, K. R. C. Koswatte, W. H. Chen, (2014) "Ontology Learning with Complex Data Type for Web Service Clustering". In Proc. of the IEEE Symposium on Computational Intelligence and Data Mining, pp. 129-136.

[9] Q. X. Song, J. Liu, X. F. Wang, J. Wang, (2014) "A Novel Automatic Ontology Construction Method Based on Web Data". In Proc. of the 10th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 762-765.

[10] S. Jupp, T. Burdett, D. Welter, S. Sarntivijai, H. Parkinson, J. Malone, (2016) "Webulous and the Webulous Google Add-On - a web service and application for ontology building from templates". Journal of Biomedical Semantics, Vol. 7.

[11] J. Peng, Y. R. Du, Y. Chen, M. Zhao, B. Pei, (2015) "Medical ontology learning based on Web resources". In Proc. of the 12th Web Information System and Application Conference, pp. 116-119.

[12] Y. Y. Wei, R. J. Wang, Y. M. Hu, X. Wang, (2012) "From Web Resources to Agricultural Ontology: a Method for Semi-Automatic Construction". Journal of Integrative Agriculture, Vol. 11, No. 5, pp. 775-783.

[13] H. Yu, X. Q. Lv, L. P. Xu, (2015) "Use web resources to construct ontology concept hierarchy". In Proc. of the International Conference on Applied Science and Engineering Innovation, Vol. 12, pp. 1006-1011.

[14] A. Vasilateanu, N. Goga, E. A. Tanase, I. Marin, (2015) "Enterprise Domain Ontology Learning from Web-Based Corpus". In Proc. of the 6th International Conference on Computing, Communication and Networking Technologies, pp. 112-117.

[15] Manvi, A. Dixit, K. K. Bhatia, B. Wadhwa, (2014) "Generating Domain Specific Ontology for Retrieving Hidden Web Contents". In Proc. of the International Conference on Information Systems and Computer Networks, pp. 66-71.

[16] W.Gatterbauer, P. Bohunsky, M. Herzog, B. Krüpl, B. Pollak, (2007) "Towards Domain-Independent Information Extraction from Web Tables". In Proc. of 16th International World Wide Web Conference (WWW2007), pp.71-80.

[17] M. Hurst, (1999) "Layout and language: Beyond simple text for information interaction - modelling the table". In Proc. of 2nd International Conference on Multimodal Interfaces, pp. 243–249.

[18] S.W. Jung, M.Y. Kang, H.C. Kwon, (2006) "Hybrid Approach to Extracting Information from Web-Tables". In Proc. of 21st International Conference on the Computer Processing of Oriental Languages (ICCPOL2006), LNAI, vol. 4285, pp. 109-119.

[19] H.H. Chen, S.C. Tsai, J.H. Tsai, (2000) "Mining Tables from Large Scale HTML Texts". In Proc. of 18th International Conference on Computational Linguistics (COLING2000), Saarbrucken, Germany, pp.166-172.

[20] Y. Wang, J. Hu, (2002) "A Machine Learning Based Approach for Table Detection on the Web": In Proc. of 11th International World Wide Web Conference (WWW2002), pp. 7-11.

[21] A. Pivk, P. Cimiano, Y. Sure, (2004) “From Tables to Frames”. In Proc. of 3rd International Semantic Web Conference (ISWC2004), LNCS, Hiroshima, Japan, Vol. 3298, pp.166-181.

[22] Y. A. Tijerino, D. W. Embley, D. W. Lonsdale, G. Nagy, (2003) “Ontology Generation From Tables”. In Proc. of 4th International Conference on Web Information Systems Engineering (WISE'03), Rome, Italy, pp.242-249.

[23] M. Tanaka, T. Ishida, (2006) “Ontology Extraction from Table on the Web”. In Proc. of 2006 International Symposium on Applications and the Internet (SAINT'06), pp.284-290.

[24] S.W. Jung, M.Y. Kang, H.C. Kwon, (2007) “Constructing Domain Ontology Using Structural and Semantic Characteristics of Web-Table Head”. In Okuno, H.G., Ali, M. (eds.) IEA/AIE 2007, LNAI, Vol. 4570, pp. 665-674.

[25] S. Abney, (2002) “Bootstrapping”. In Proc. of 40th Annual Meeting of the Association for Computational Linguistics (ACL'02), Philadelphia, pp. 360-367.

[26] S. I. Cha, M. J. Han, (2015) “A Method to Extract Automatically Ontology Instances Using Bootstrapping”. Bulletin of Academy of Sciences, the DPR Korea, Vol. 367, No. 1, pp.19-20, ISSN 1680-4223.(in Korean)

[27] H. Davulcu, S. Vadrevu, S. Nagarajan, I. V. Ramakrishnan, (2003) “OntoMiner: Bootstrapping and populating ontologies from domain-specific Web sites”. IEEE Intelligent Systems, Vol. 18, No. 5, pp.24-33.

[28] W. S. Wu, A. H. Doan, C. Yu, W. Y. Meng, (2006) “Bootstrapping domain ontology for semantic web services from source web sites, Technologies for E-Services”. LNCS, Vol. 3811, pp.11-22.

[29] A. Segev, Q. Z.Sheng, (2012) “Bootstrapping Ontologies for Web Services”. IEEE Transactions on Services Computing, Vol. 5, No. 1, pp.33-44.

[30] C. Y. Zhang, W. R. Xu, Z. Y. Ma, S. Gao, Q. Li, J. Guo, (2015) “Construction of semantic bootstrapping models for relation extraction”. Knowledge-Based Systems, Vol .83.

[31] OWL Reference, (2003) “OWL Web Ontology Language Reference”. http://www.w3.org/ TR/owl-ref/.

[32] OWL 2 Overview, (2012) “OWL 2 Web Ontology Language Document Overview (Second Edition)”. http://www.w3.org/TR/owl2-overview/.