Extract Semantic Web Knowledge from Wikipedia Tables and Lists

Julthep Nandakwang
Department of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
julthep@nandakwang.com

Prabhas Chongstitvatana
Department of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
prabhas@chula.ac.th

***Abstract*—Wikipedia is currently one of the most significant knowledge base for human but it is still hard to access automatically by machines. Nowadays, there are much efforts to transform the knowledge resides in Wikipedia to machine readable information in the form of Linked Data to be used in many intelligent applications with Semantic Web technology. However, the knowledge extracted from Wikipedia focused on some structured elements such as Infoboxes and Categories. There are many important semi-structured and unstructured elements that have not yet been extracted to Linked Data. This work proposes a method to extract two elements of the rich informative part of Wikipedia which are Tables and Lists. Some sets of newly created schema and ontology are provided to support the extracted information.**

Keywords—knowledge extraction; Linked Data; list; Semantic Web; table; Wikipedia

I. Introduction

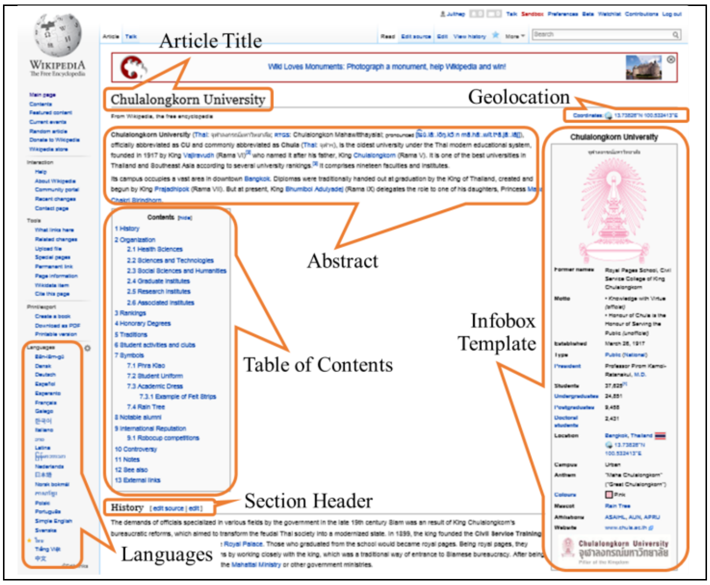
Wikipedia has been used widely as a platform of Knowledge Management. Wikipedia use wiki concept to collect all the knowledge in nearly all domain by letting its users contribute and review the contents in the system. The big issue of Wikipedia is its contents (called “articles”). They are in the form of human readable and editable. It is hard for a machine to extract, analyze or infer the knowledge from the repository for further processing. Wikipedia content have to be transformed to the machine readable and “understandable” format by incorporate “meaning” to the textual content in Wikipedia.

This paper proposes a method to extract tables and lists data. The aim is to provide more Linked Data that are currently missing. First of all, we illustrate the structure of a Wikipedia article. We will review related studies of the methods, techniques and mechanisms used to transform semi-structured or unstructured data to structured data.

II. Wikipedia Article Anatomy

Wikipedia articles have the same common elements as illustrated in Fig. 1, nothing more nothing less, with little different only in the length of the contents. These elements are described as followings:

- Article Title – the name of the article
- Abstract – the brief description of the article
- Infobox Template – the at-a-glance data
- Geolocation – the absolute position of subject
- Table of Contents – automatically generate from...
- Section Headers – use for content segmentation
- Languages – to switch between Wikipedia sites
- Wikitext – the textual content of the article
- Wikilink – the intra-link between articles
- Pictures – the thumbnails to the full-sized pictures
- Lists – listed items in bulleted or numbered
- Tables – values arrange in tabular form
- References – citations of the information
- Related Articles – list of related articles
- External Links – links to external websites
- Navbox Template – navigator to articles collection
- Categories – classification of the article



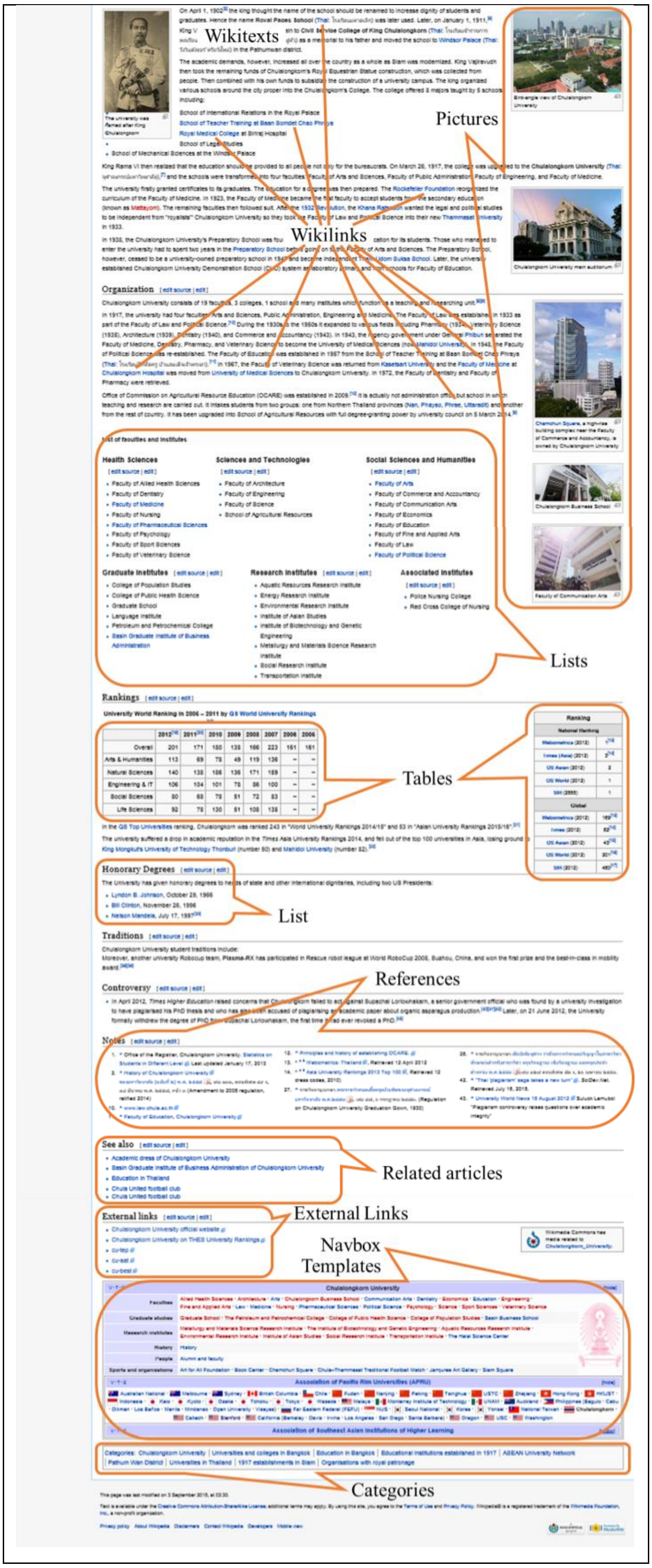


Fig. 1. Sample of a Wikipedia article entitled “Chulalongkorn University”

III. Related Researches and Methodology

There are several studies on information extraction from Wikipedia to Linked Data. Some works related to the extraction of table-type data such as spreadsheet to RDF triples. These works can be categorized into group hierarchy as shown in Fig. 2. However, we focus on DBpedia and YAGO due to their widespread use.

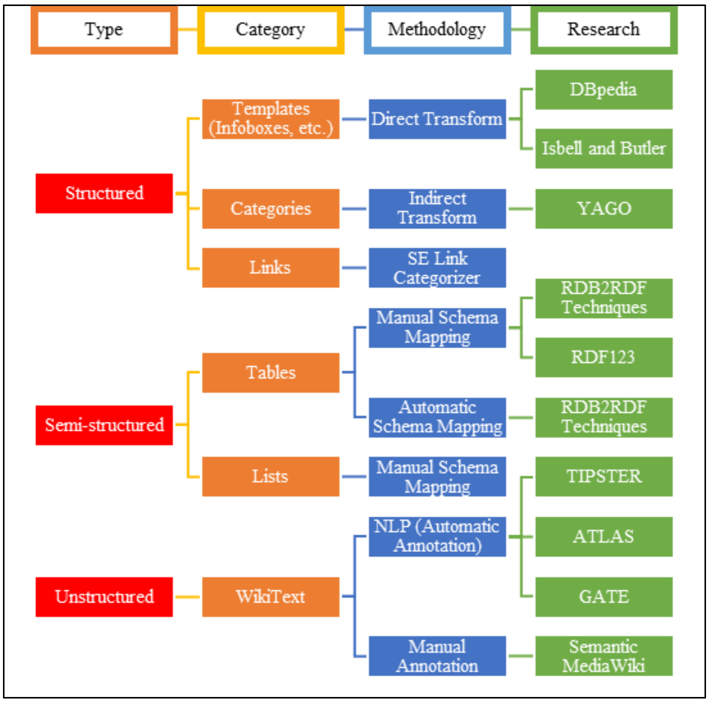


Fig. 2. Wikipedia textual content extraction approaches

DBpedia [1], is a research of the Free University of Berlin and Leipzig University, intended to extract Wikipedia’s structured information (Abstract, Infoboxes and Categories) to Linked Data. It also supports multiple languages. The goal of the project is to be the nucleus of Linked Data. Isbell and Butler [2] at Digital Media Systems Laboratory of HP studies the transformation mainly from Infoboxes but cover wide aspects of Wikipedia semi-structured and unstructured data with generalized methodology. YAGO [3] is a research of the Max Planck Institute for Informatics to extract structured information from Wikipedia by using the Categories data together with the Synsets data from WordNet project of Princeton University. RDB2RDF survey [4] is a report of W3 Incubator Group covers many research projects related to Relational Database to RDF conversion method. Although, it does not cover Wikipedia contents, it can be applied to transform article tables. RDF123 project [5] intended to create offline and online tools to convert the data in spreadsheet to RDF triples. TIPSTER, ATLAS and GATE are Natural Language Processing projects that could be used to extract and transform unstructured Wikitext to structured graph-based information. Semantic MediaWiki [6] is an extension to MediaWiki (the system is used as the core of Wikipedia) to let Wikipedian manually annotate a semantic information directly to Wikitext.

IV. Extract Information to Linked Data

Compared with the whole structure of Wikipedia, only some elements have been transformed to Linked Data such as Abstract, Infoboxes and Categories. Many of remaining elements have not been transformed yet. In this paper, we focus on two of the most important and rich informative elements of Wikipedia which are Tables and Lists. Let us start with an illustration of kinds of information that we would like to extract from Wikipedia articles.

To retrieve Abstract information from Linked Data that transform by DBpedia from Wikipedia article entitled "*Chulalongkorn University*". A SPARQL query is shown below:

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?value
WHERE {
 dbr:Chulalongkorn_University dbo:abstract ?value.
}
```

The output is shown below:

"Chulalongkorn University, officially abbreviated as CU and commonly abbreviated as Chula, is the oldest university under the Thai modern educational system, founded in 1917 by King Vajiravudh (Rama VI) who named it after his father, King Chulalongkorn (Rama V). It is one of the best universities in Thailand and Southeast Asia according to several university rankings. It comprises nineteen faculties and institutes. Its campus occupies a vast area in downtown Bangkok. Diplomas were traditionally handed out at graduation by the King of Thailand, created and begun by King Prajadhipok (Rama VII). But at present, King Bhumibol Adulyadej (Rama IX) delegates the role to one of his daughters, Princess Maha Chakri Sirindhorn."

Suppose we are interested in the university ranking. To find the ranking of this university, we need to look further from the tables in the original Wikipedia article, such as in TABLE I.

TABLE I. Ranking table in Chulalongkorn University article

University World Ranking in 2005 – 2012 by QS World University Rankings

	2012	2011	2010	2009	2008	2007	2006	2005
Overall	201	171	180	138	166	223	161	161
Arts & Humanities	113	69	78	49	119	136	–	–
Natural Sciences	140	138	186	136	171	159	–	–
Engineering & IT	106	104	101	78	86	100	–	–
Social Sciences	80	68	78	51	72	83	–	–
Life Sciences	92	78	130	51	108	138	–	–

If we are interested in the organization of this university, we need to look back to the same Wikipedia article. They are in the form of multi-level list, as in Fig. 3

List of faculties and institutes

Health Sciences

- Faculty of Allied Health Sciences
- Faculty of Dentistry
- Faculty of Medicine
- Faculty of Nursing
- Faculty of Pharmaceutical Sciences
- Faculty of Psychology
- Faculty of Sport Sciences
- Faculty of Veterinary Science

Sciences and Technologies

- Faculty of Architecture
- Faculty of Engineering
- Faculty of Science

Social Sciences and Humanities

- Faculty of Arts
- Chulalongkorn Business School
- Faculty of Communication Arts
- Faculty of Economics
- Faculty of Education
- Faculty of Fine and Applied Arts
- Faculty of Law
- Faculty of Political Science

Fig. 3. Institutes list in Chulalongkorn University article from Wikipedia

It is important that there should be a way to extract information from Wikipedia tables and lists to provide more data to fill-up the missing information in Linked Data.

V. Schema, Ontology and Algorithm

To accommodate data extract from Wikipedia tables and lists, we need to create a set of schema and ontology to provide additional vocabularies for creating triples derived from the extraction. And we will demonstrate the examples of algorithm used to extract tables and lists out as RDF triples.

A. *Tables*

The RDF schema to represent table is created, as shown below:

```
tlo:lastRow rdfs:domain rdfs:Resource .
tlo:lastRow rdfs:range rdfs:Literal .
tlo:lastColumn rdfs:domain rdfs:Resource .
tlo:lastColumn rdfs:range rdfs:Literal .
tlo:hasIndex rdfs:domain rdfs:Resource .
tlo:hasIndex rdfs:range rdfs:Literal .
tlo:hasColumn# rdfs:domain rdfs:Resource .
tlo:hasColumn# rdfs:range rdfs:Literal .
tlo:Table rdfs:subClassOf rdfs:Class .
tlo:ColumnHeader rdfs:subClassOf rdfs:Class .
tlo:TableCell rdfs:subClassOf rdfs:Class .
```

The example of the structure to represent a 2 × 2 table with RDF triples using the above schema is presented below:

```
_:Table1 rdf:type tlo:Table .
_:Table1 tlo:lastRow 2 .
_:Table1 tlo:lastColumn 2 .
_:Row0 tlo:hasIndex 0 .
_:Row0 rdf:type tlo:ColumnHeader .
_:Row0 tlo:hasColumn1 "ColumnHeader1" .
_:Row0 tlo:hasColumn2 "ColumnHeader2" .
_:Row1 tlo:hasIndex 1 .
_:Row1 rdfs:label "RowHeader1" .
_:Row1 rdf:type tlo:TableCell .
_:Row1 tlo:hasColumn1 "Row1Cell1" .
_:Row1 tlo:hasColumn2 "Row1Cell2" .
_:Row2 tlo:hasIndex 2 .
_:Row2 rdfs:label "RowHeader2" .
_:Row2 rdf:type tlo:TableCell .
_:Row2 tlo:hasColumn1 "Row2Cell1" .
_:Row2 tlo:hasColumn2 "Row2Cell2" .
```

An algorithm to generate a set of RDF triples from all tables in a Wikipedia article is shown in Fig. 4. The algorithm consists of three nested loops: the outer loop scans all tables in the article and generates a set of triples to represent the header of each table. The next loop scans all rows of the table to generate the triples for each row. The innermost loop scans all columns in the row to generate the triples to represent a cell value.

```
Algorithm: Transform Wikipedia tables to RDF triples
Output() is an RDF triple serializer
ItemType[] is one of {tlo:Table, tlo:ColumnHeader, tlo:TableCell}
GenTable()
For i = 1 to LastTable of current article
  Output( _:TableNode%i rdfs:label %TableLabel[i] . )
  Output( _:TableNode%i rdf:type %TableType[i] . )
  Output( _:TableNode%i tlo:lastRow %LastRow . )
  Output( _:TableNode%i tlo:lastColumn %LastColumn . )
  For j = 0 to LastRow of current table
    Output( _:TableNode%i-%j tlo:hasIndex %j . )
    Output( _:TableNode%i-%j rdfs:label %RowLabel[i][j] . )
    If j = 0
      Output( _:TableNode%i-%j rdf:type tlo:ColumnHeader . )
    Else
      Output( _:TableNode%i-%j rdf:type tlo:TableCell . )
    For k = 1 to LastColumn of current row
      Output( _:TableNode%i-%j tlo:hasColumn%k %Cell[i][j][k] . )
```

Fig. 4. The algorithm to generate RDF from tables

An example of Wikipedia table from TABLE I. being transformed into a series of RDF triples is shown in Fig. 5.

```
_:TableNode1 rdfs:label "University World Ranking in
    2005 – 2012 by QS World University Rankings" .
_:TableNode1 rdf:type tlo:Table .
_:TableNode1 tlo:lastRow 7 .
_:TableNode1 tlo:lastColumn 9 .
_:TableNode1-0 tlo:hasIndex 0 .
_:TableNode1-0 rdfs:label "" .
_:TableNode1-0 rdf:type tlo:ColumnHeader .
_:TableNode1-0 tlo:hasColumn1 "2012" .
_:TableNode1-0 tlo:hasColumn2 "2011" .
_:TableNode1-0 tlo:hasColumn3 "2010" .
_:TableNode1-0 tlo:hasColumn4 "2009" .
_:TableNode1-0 tlo:hasColumn5 "2008" .
_:TableNode1-0 tlo:hasColumn6 "2007" .
_:TableNode1-0 tlo:hasColumn7 "2006" .
_:TableNode1-0 tlo:hasColumn8 "2005" .
_:TableNode1-1 tlo:hasIndex 1 .
_:TableNode1-1 rdfs:label "Overall" .
_:TableNode1-1 rdf:type tlo:TableCell .
_:TableNode1-1 tlo:hasColumn1 "201" .
_:TableNode1-1 tlo:hasColumn2 "171" .
_:TableNode1-1 tlo:hasColumn3 "180" .
_:TableNode1-1 tlo:hasColumn4 "138" .
_:TableNode1-1 tlo:hasColumn5 "166" .
_:TableNode1-1 tlo:hasColumn6 "223" .
_:TableNode1-1 tlo:hasColumn7 "161" .
_:TableNode1-1 tlo:hasColumn8 "161" .
. . .
_:TableNode1-6 tlo:hasIndex 6 .
_:TableNode1-6 rdfs:label "Life Sciences" .
_:TableNode1-6 rdf:type tlo:TableCell .
_:TableNode1-6 tlo:hasColumn1 "92" .
_:TableNode1-6 tlo:hasColumn2 "78" .
_:TableNode1-6 tlo:hasColumn3 "130" .
_:TableNode1-6 tlo:hasColumn4 "51" .
_:TableNode1-6 tlo:hasColumn5 "108" .
_:TableNode1-6 tlo:hasColumn6 "138" .
_:TableNode1-6 tlo:hasColumn7 "-" .
_:TableNode1-6 tlo:hasColumn8 "-" .
```

Fig. 5. Sample of RDF triples output generated from table in TABLE I.

The RDF triples will be connected to the Linked Data with the owl:sameAs semantic link as shown below:

```
dbr:Chulalongkorn_University owl:sameAs
     tlr:Chulalongkorn_University .
tlr:Chulalongkorn_University tlo:hasItem
     _:TableNode1 .
```

We could use SPARQL query to get the data in any cell of table by just knowing the name of row and column of that cell. For example, if we want to know the value of "*Engineering & IT*" row and "*2009*" column of the "*University World Ranking in 2005 – 2012 by QS World University Rankings*" table, the query will be like this:

```
PREFIX rdfs: <http://w3.org/2000/01/rdf-schema#>
PREFIX tlo: <http://tulip.cba.net/ontology/>
SELECT ?value
WHERE {
     ?column tlo:hasIndex 0 .
     ?column ?hasColumn "2009" .
     ?row rdfs:label "Engineering & IT" .
     ?row ?hasColumn ?value .
}
```

The result is *78*. We can also present the transformed RDF triples in various ways. Fig. 6 shows the presentation of results in Fig. 5 as a line chart.

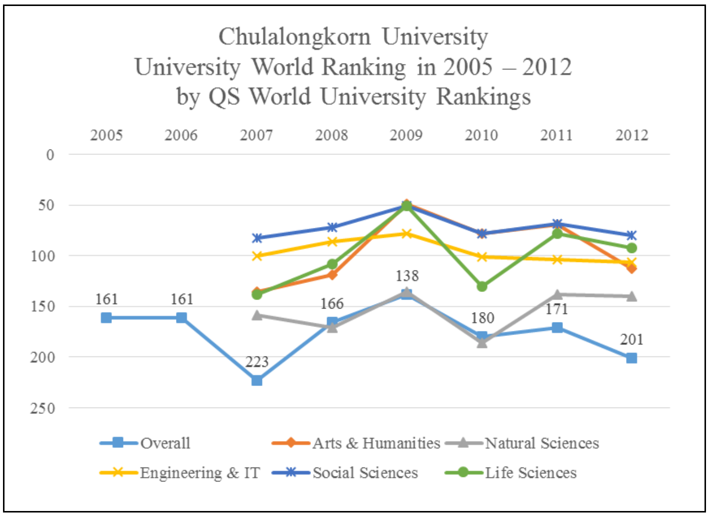


Fig. 6. A trend line chart generate from RDF triples in Fig. 5

B. Lists

To represent Wikipedia Lists with RDF triples, it is important to preserves the ordering of List items whether the List is ordinal (numbered) or nominal (bulleted). (in this section, we differentiate the "article List" from the "list data structure" by initial capital letter, except the word ItemList which is the camel case, and `rdf:List` which is an RDF class). According to W3 specification of the RDF, the preservation of the ordering of objects is implement as recursive lists terminated with an empty list. The RDF schema are defined by the W3 as follows:

```
rdf:first rdfs:domain rdf:List .
rdf:first rdfs:range rdfs:Resource .
rdf:rest rdfs:domain rdf:List .
rdf:rest rdfs:range rdf:List .
rdf:nil rdf:type rdf:List . (empty list instance)
```

We need an additional schema defined as follows:

```
tlo:hasItemList rdfs:domain rdfs:Resource .
tlo:hasItemList rdfs:range rdf:List .
```

An example of the structure to represent a 3-item List by RDF triples is shown below, and illustrated as RDF Graph in Fig. 7.

```
_:List1 tlo:hasItemList _:ItemList1 .
_:ItemList1 rdf:first _:Item1 .
_:ItemList1 rdf:rest _:ItemList2 .
_:ItemList2 rdf:first _:Item2 .
_:ItemList2 rdf:rest _:ItemList3 .
_:ItemList3 rdf:first _:Item3 .
_:ItemList3 rdf:rest rdf:nil .
```

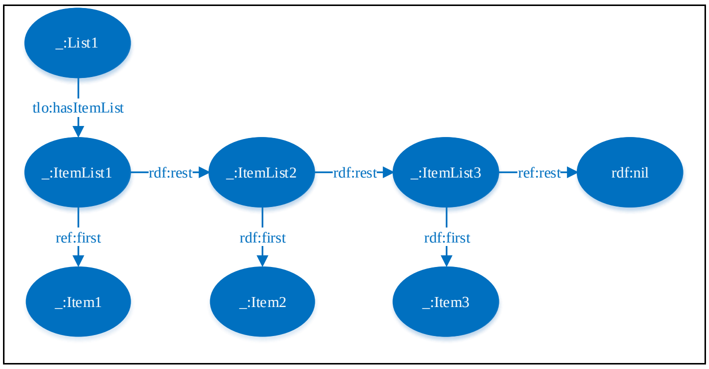


Fig. 7. RDF graph represented the sample list triples

The example shows that `_:List1` has `_:ItemList1` as a list of items which is `rdf:List` class. The `_:ItemList1` consists of the first element `_:Item1` which is `rdfs:Resource` class and the remaining members of the list are `_:ItemList2` and so on which are again `rdf:List` class. The final item is the list is `_:ItemList3`. It is followed by `rdf:nil` which is the empty list which finally complete this list. The `_:List1` can be presented as a recursive list as shown here.

```
_:List1 = {_:Item1, {_:Item2, {_:Item3, {} }}}
```

To reduce the cost of accessing last items in the list sequence, the generation of RDF triples based on standard definition of W3 are considered (Allemang and Hendler, p.49) [7]. It is necessary to create an additional set of schema to help retrieving ordered information. In the creation of RDF triples, we can generate both forms of representation and include them together in the same graph. Our approach is to represent ordered List by adding a predicate that directly point to individual elements of a List and adding another predicate that specifies the index of each element, as follows:

```
_:List1 = {{_:Item1,1}, {_:Item2,2}, {_:Item3,3}}
```

This approach enables us to access individual elements in an ordered List just the same way as unordered List. Now there are indexes of each element, which directly reflect the order of that item. These indexes are optional when represent the unordered data. The additionally RDF schema is defined as follows:

```
tlo:hasItem rdfs:domain rdfs:Resource .
tlo:hasItem rdfs:range rdfs:Resource .
tlo:hasIndex rdfs:domain rdfs:Resource .
tlo:hasIndex rdfs:range rdfs:Literal .
tlo:NumberedList rdfs:subClassOf rdfs:Class .
tlo:BulletedList rdfs:subClassOf rdfs:Class .
tlo:NoBulletList rdfs:subClassOf rdfs:Class .
tlo:Terminal rdfs:subClassOf rdfs:Class .
```

The previous example updated with additional information is shown below, and illustrated as graph in Fig. 8.

```
_:List1 tlo:hasItemList _:ItemList1 .
_:List1 tlo:hasItem _:Item1 .             ← Additional
_:ItemList1 rdf:first _:Item1 .
_:Item1 tlo:hasIndex 1 .                  ← Additional
_:ItemList1 rdf:rest _:ItemList2 .
_:List1 tlo:hasItem _:Item2 .             ← Additional
_:ItemList2 rdf:first _:Item2 .
_:Item2 tlo:hasIndex 2 .                  ← Additional
_:ItemList2 rdf:rest _:ItemList3 .
_:List1 tlo:hasItem _:Item3 .             ← Additional
_:ItemList3 rdf:first _:Item3 .
_:Item3 tlo:hasIndex 3 .                  ← Additional
_:ItemList3 rdf:rest rdf:nil .
```

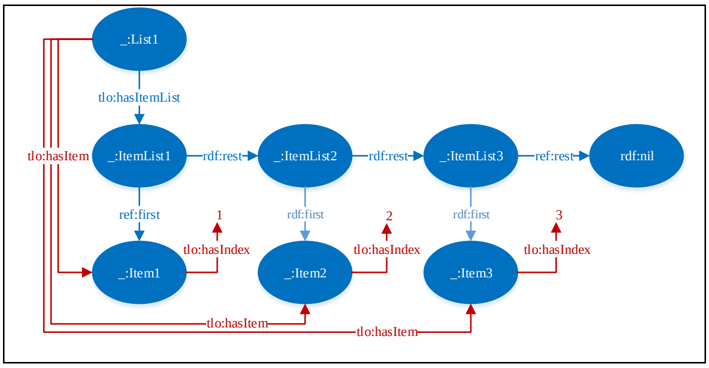


Fig. 8. RDF graph represents the sample list triples with indexes

The algorithm to transform all Lists in a Wikipedia article to RDF triples is shown in Fig. 9. The algorithm starts from GenList() which contain the main loop to scan all lists in an article to generate the list header. GenItem() scans all items in each list. However, any item in the list could be an inner list (or list of lists), so GenItem() is called recursively to handle this issue. It generates a set of triples repeatedly until the terminal item is found.

```
Algorithm: Transform Wikipedia lists to RDF triples
Output() is an RDF triple serializer
ItemType[] is one of {tlo:NumberedList, tlo:BulletedList,
    tlo:NoBulletList, tlo:Terminal}
---------------------------------------------------------------
GenList()
For i = 1 to LastList of current article
    Output( _:ListNode%i rdfs:label %ItemLabel[i] . )
    Output( _:ListNode%i rdf:type %ItemType[i] . )
    GenItem( i )
---------------------------------------------------------------
GenItem( i as string )
For j = 1 to LastItem of current list
    If j = 1
        Output( _:ListNode%i tlo:hasItemList _:ItemList%i-%j . )
    Output( _:ListNode%i tlo:hasItem _:ListNode%i-%j . )
    Output( _:ItemList%i-%j rdf:first _:ListNode%i-%j . )
    Output( _:ListNode%i-%j tlo:hasIndex %j . )
    Output( _:ListNode%i-%j rdfs:label %ItemLabel[i][j] . )
    Output( _:ListNode%i-%j rdf:type %ItemType[i][j] . )
    If ItemType[j] != Terminal
        GenItem( i & "-" & j )
    If j != LastItem
        Output( _:ItemList%i-%j rdf:rest _:ItemList%i-%(j+1) . )
    Else
        Output( _:ItemList%i-%j rdf:rest rdf:nil . )
```

Fig. 9. Lists to RDF triples generator algorithm

An example of Wikipedia Lists in Fig. 3 has been transformed by the proposed algorithm to a series of represented RDF triples shown in Fig. 10.

```
_:ListNode1 rdfs:label "List of faculties and institutes" .
_:ListNode1 rdf:type tlo:NoBulletList .
_:ListNode1 tlo:hasItemList _:ItemList1-1 .
_:ListNode1 tlo:hasItem _:ListNode1-1 .
_:ItemList1-1 rdf:first _:ListNode1-1 .
_:ListNode1-1 tlo:hasIndex 1 .
_:ListNode1-1 rdfs:label "Health Sciences" .
_:ListNode1-1 rdf:type tlo:BulletedList .
_:ListNode1-1 tlo:hasItemList _:ItemList1-1-1 .
_:ListNode1-1 tlo:hasItem _:ListNode1-1-1 .
_:ItemList1-1-1 rdf:first _:ListNode1-1-1 .
_:ListNode1-1-1 tlo:hasIndex 1 .
_:ListNode1-1-1 rdfs:label "Faculty of Allied Health Science".
_:ListNode1-1-1 rdf:type tlo:Terminal .
_:ItemList1-1-1 rdf:rest _:ItemList1-1-2 .
_:ListNode1-1 tlo:hasItem _:ListNode1-1-2 .
_:ItemList1-1-2 rdf:first _:ListNode1-1-2 .
_:ListNode1-1-2 tlo:hasIndex 2 .
_:ListNode1-1-2 rdfs:label "Faculty of Dentistry" .
_:ListNode1-1-2 rdf:type tlo:Terminal .
_:ItemList1-1-2 rdf:rest _:ItemList1-1-3 .
. . .
_:ListNode1-1 tlo:hasItem _:ListNode1-1-8 .
_:ItemList1-1-8 rdf:first _:ListNode1-1-8 .
_:ListNode1-1-8 tlo:hasIndex 8 .
_:ListNode1-1-8 rdfs:label "Faculty of Veterinary Science".
_:ListNode1-1-8 rdf:type tlo:Terminal .
_:ItemList1-1-8 rdf:rest rdf:nil .
_:ItemList1-1 rdf:rest _:ItemList1-2 .
_:ListNode1 tlo:hasItem _:ListNode1-2 .
_:ItemList1-2 rdf:first _:ListNode1-2 .
_:ListNode1-2 tlo:hasIndex 2 .
_:ListNode1-2 rdfs:label "Sciences and Technologies" .
_:ListNode1-2 rdf:type tlo:BulletedList .
. . .
_:ListNode1-3 tlo:hasItem _:ListNode1-3-8 .
_:ItemList1-3-8 rdf:first _:ListNode1-3-8 .
_:ListNode1-3-8 tlo:hasIndex 8 .
_:ListNode1-3-8 rdfs:label "Faculty of Political Science" .
_:ListNode1-3-8 rdf:type tlo:Terminal .
_:ItemList1-3-8 rdf:rest rdf:nil .
_:ItemList1-3 rdf:rest rdf:nil .
```

Fig. 10. Sample RDF triples output generated from the Lists in Fig. 3

The result RDF triples will be connected to the Linked Data with the `owl:sameAs` semantic link as shown below:

```
dbr:Chulalongkorn_University owl:sameAs
     tlr:Chulalongkorn_University .
tlr:Chulalongkorn_University tlo:hasItem
     _:ListNode1 .
```

The transformed RDF triples can be presented on many Semantic Web browsers in various ways. Fig. 11 shows the results from Fig. 10 in a mind map using SWIMM (Semantic Web Inspired Mind Map) [ref] tool.

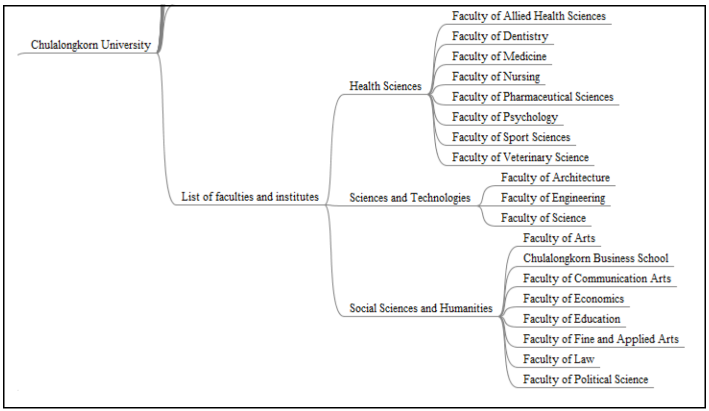

Chulalongkorn University
List of faculties and institutes
Health Sciences
Faculty of Allied Health Sciences
Faculty of Dentistry
Faculty of Medicine
Faculty of Nursing
Faculty of Pharmaceutical Sciences
Faculty of Psychology
Faculty of Sport Sciences
Faculty of Veterinary Science
Sciences and Technologies
Faculty of Architecture
Faculty of Engineering
Faculty of Science
Social Sciences and Humanities
Faculty of Arts
Chulalongkorn Business School
Faculty of Communication Arts
Faculty of Economics
Faculty of Education
Faculty of Fine and Applied Arts
Faculty of Law
Faculty of Political Science

Fig. 11. SWIMM mind map output from the RDF triples in Fig. 10

VI. CONCLUSION

Many of information in Wikipedia are in the form of tables and lists. Several important information was not been written (or could not be written) in a descriptive essay format, but instead arranged as tables and lists. In addition, some articles are consisted of only tables or lists. Wikipedia also have a special set of articles which the title begins with "List of …" obviously, contents within these articles are only the lists. Therefore, the Linked Data extracted from Wikipedia tables and lists could provide additional information to the various Semantic Web applications, and allow them to be used more efficiently. This work proposed an algorithm to extract these information. The ultimate goal is to provide a tool to convert large body of knowledge presented in the web into "structured" machine searchable form.

REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A Nucleus for a Web of Open Data," in Proceedings of the 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, Busan, Korea, 2007, pp. 722-735.

[2] J. Isbell, and M. H. Butler, *Extracting and re-using structured data from wikis*, HPL-2007-182, Digital Media Systems Laboratory of Hewlett-Packard Development Company, Bristol, 2007.

[3] F. M. Suchanek, G. Kasneci, and G. Weikum, "YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia," in Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada, 2007, pp. 697-706.

[4] S. S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. Thibodeau Jr, S. Auer, J. Sequeda, and A. Ezzat, *A survey of current approaches for mapping of relational databases to RDF*, W3C RDB2RDF Incubator Group, 2009.

[5] L. Han, T. Finin, C. Parr, J. Sachs, and A. Joshi, "RDF123: From Spreadsheets to RDF," The Semantic Web - ISWC 2008, Lecture Notes in Computer Science A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin and K. Thirunarayan, eds., pp. 451-466: Springer Berlin Heidelberg, 2008.

[6] M. Völkel, M. Krötzsch, D. Vrandečić, H. Haller, and R. Studer, "Semantic Wikipedia," in Proceedings of the 15th international conference on World Wide Web, Edinburgh, Scotland, 2006, pp. 585-594.

[7] D. Allemang, and J. Hendler, *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*: Morgan Kaufmann, 2008.