International Journal of Advances in Computer Science & Its Applications
Volume 6 : Issue 2 *[ISSN 2250-3765]*
Publication Date : 31 August, 2016

A semi-automatic method for building semantic data repository

Applied for Vietnamese tourism datasets

[Dr Tuan-Dung Cao, Anh-Tien Le]

Abstract **- In this work, we propose Semantic Tourist Data Innovation System (ViSIS) - a framework to generate linked dataset of tourism such as restaurants, hotels, attractions and hospitals. ViSIS extracts and integrates information about places from various relational datasets which contain the data from many different social media or location-based websites such as Foursquare, Agoda, Mytour, etc. These enormous and heterogeneous data are converted into a semantic knowledge base, which currently consists of more than 100,000 places in Vietnam and has started to expand with the information from other famous cities, i.e. London, Barcelona and Rome. This paper will describe the architecture of ViSIS, its processes and present an improved method for categorizing places.**

Keywords **- rdf, ontology, semantic web, rdf converter, tourism, linked data.**

I. Introduction

Since the concept of Semantic Web of Tim Berners Lee was introduced in 2001, the Web has seen the growth of linked data websites that allow the creation of a global-scale interlinked data space, known as the Web of Data, by exposing datasets previously isolated as data graphs, which can be interlinked and integrated with other datasets. The purpose of Web of data is to connect object's concepts and contents to each other, instead of simply connecting documents. Thus Web of data has led to the conversion of existing documents into linked data , and to the creation of new datasets. Among them, one of the most exploited datasets is DBpedia, which is the linked data version of Wikipedia. In addition, new data models have been implemented, to represent data in a common standard way [1]. The most famous data models are the Resource Description Framework (RDF) for the description of entities and the Web Ontology Model (OWL) (W3C, 2012) for the description of concepts.

The principles of Semantic Web have been applied to different fields of knowledge, spanning from cultural heritage to health . In this paper, we focus on the field of tourism, which is one of many industries that has benefited enormously from using of the Website in the internet, and been explored in various recommender system and mobile application for instance Expedia, Yelp, etc. However, tourism datasets are mostly presented in relational database. Moreover, these data contain many duplications from different sources so they are heterogeneous and difficult for many systems to search information about a tourist object in the relationship with the others.

Dr. Tuan-Dung Cao, Anh-Tien Le
Hanoi University of Science and Technology
Vietnam

On the other hand, there are researches shown that semantic web brings many benefits if applied in tourism and heritage fields [2]. Unfortunately, the semantic data about tourist are poor and it is a costly and time-consuming process of manually building a linked dataset even with skilled engineers and experts. Therefore we have studied a method for automating most of the processes and reducing human jobs. Also we present our datasets in the ontological model called VTIO.

The remainder of the paper is organized as follows. Section 2 reviews some related works. Section 3 outlines the concept and processes of ViSIS and section 4 describes the improvement of places categorization process. Finally, Section 5 contains the conclusions and future work.

II. Related Work

Over the last years, there are many research about generating semantic data. We could learn from the framework that converts JSON to RDF by Julien Tscherrig, Philippe Cudr´e-Mauroux, Elena Mugellini, Omar Abou Khaled and Maria Sokhn [3]. Chang-Su Kim, Sung-Han Kim and Hoe-Kyung Jung also studied transferring data from CSV to RDF [4]. Most of these research have the purpose to create semantic web from the data source stored in XML, JSON or CSV.

The most well-known linked dataset, DBpedia [5] is available in different languages. Its English version contains about 4.0 million things, classified in different categories, including people, places, creative works, organizations, species and diseases. However, DBpedia, as well as Wikipedia, contains only a small number of things related to the tourism domain, such as accommodations and restaurants. In addition, to the best of our knowledge, only few linked datasets have been implemented in the field of tourism.

There are also few initiatives of tourism linked data. A concrete example is Tourpedia, which combines and aggregates data extracted from four social media: Facebook, Foursquare, Google Places and Booking.com. Tourpedia contains almost half a million places, divided in four categories: accommodations, restaurants, points of interests (POIs) and attractions. Tourpedia was developed within the OpeNER Project [6]. The main objective of OpeNER is to provide a set of ready-to-use modules for the natural language processing. More specifically, OpeNER focuses on building linguistic pipelines in six languages (English, Spanish, German, French, Italian, and Dutch) that enable the identification and disambiguation of named entities and the analysis of sentiment in opinionated texts.

The author has also developed a Semantic Tourist Action Access and Recommending System called STAAR [7]. The

main focus of this system is to help a tourist find relevant information for his trip through Web and smartphone. They can located some recommendations and suggestions about the interested location and even the best itinerary from a starting point. The system defined an ontology providing sematic primitives for describing travel resources and tourist profiles.

Our research concentrated on developing a system for generating semantic data from various source on the Internet, not just a specific data model.

III. An Overview about ViSIS

In ViSIS, a web interface was developed based on CakePHP framework that allows engineers to observe and analysis the data. The tourism datasets from the internet would be extracted and processed through four main modules presented below:

- The crawler modules will obtain the data from different websites and store them into a raw data table.
- The data normalization module receives these tables and combines them into a new table before standardization processes.
- One of the ViSIS's innovations is duplication handling module, the information would be stored into a merged data table after this stage.
- Finally, a new ontology would be generated in form of RDF file by the semantic enrichment module. This file would be upload into an AllegroGraph Server for practical applications.

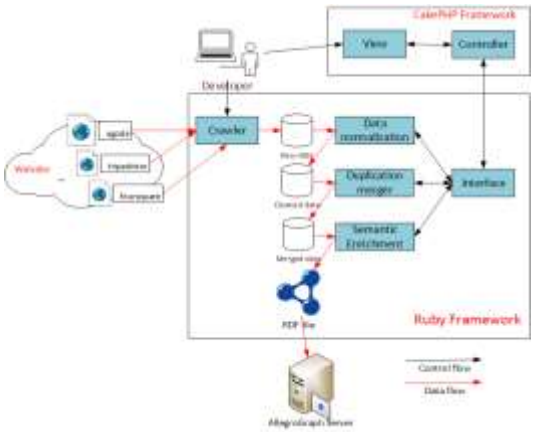

Figure 1. ViSIS Architecture

1) Crawler

The main function of this module is to find and extract data from various sources on the Internet and store in MySQL databases.

The data were extracted from a website using a crawler written in Ruby. Each websites has a corresponding distinguish crawler because they have different architectures. Our system can gather information from more than 10 website such as mytour, amthuc365, agoda, yelp, foody…

Extract data from websites can be approached by analysis their HTML structure. One of the most common method is to represent Document Object Model – DOM [8] from the HTML source. Then the data could be easily extracted by interrogate DOM tree.

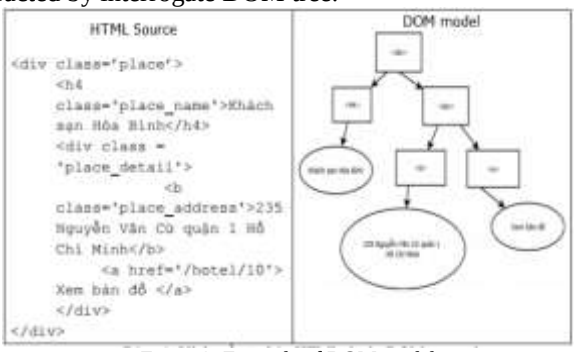

Figure 2. Example of DOM model

Before crawling the information from a website, developers need to study the website's structure. This step allows the developers to have an understanding of the site's structure, where the data are placed. Then we also choose a starting url for the crawler to begin with.

Then, we need to configure the MySQL parameter (username, password, and database) for each crawler to store the future information. The table schema is pre-defined so all the database would have a same structure. The HTML source could be easily extracted using 'rest-open-uri' gem in Ruby.

To build the DOM tree from the HTML source, we used the Nokogiri library which also helps us to extract the information from the HTML tags.

A crawler would be executed through each modules orderly:

- createTable: create the pre-defined schema table for the database
- getLink: get all the url of every places in the website and store into the link table.
- deleteDuplicate: delete all the duplicated links in the link table.
- getDetails: navigate to the urls corresponding to each places and extract needed information and store into detail tables

There are several websites that generate the HTML source using JavaScript, which makes them difficult to obtain the HTML structure. In this case, we employ a different method to obtain the HTML source rather than common ways, which is using Selenium Web driver framework [9] to extract HTML source after the JavaScript codes of the site finish executed.

2) Data normalization

This module processes the raw data extracted through the following stages:

- Join datasets collected in the crawling step.
- Delete unnecessary records base on the categories
- Process data with six steps as mentioned below:

Step 1: Check and adjust mobile numbers: convert the mobile number to international number format with country codes. For example: 09656.33.888 => +849656.33.888 (Vietnam)

Step 2: Adjust coordinates, add missing longitude and latitude: covert all coordinates to decimal degrees, get the missing coordinates from obtained address by using Geokit framework.

Step 3: Place categorization: Category must be compatible with the categories pre-defined in ontology VTIO. This steps would be described more specific in section IV of this paper.

Step 4: Address splitting: number, streets, district, and city. This step allows the system to create a table contains streets and districts of a city without manually collecting from other sources and these records will be mapped into an rdf uri or a node in our VTIO ontology.

Step 5: Check and adjust place's name.

Step 6: Check image links: find and remove the error image links from the datasets.

3) Duplication merger

The duplicate place objects are detected and deleted in this stage. The system will merge these places using specific algorithm resulting in only one detailed place.

To find duplicated resources, we consider these fields:

- Latitude, longitude and address: With the coordinates we could calculate the distance between two places. If the value calculated is smaller than a suitable parameter, we could conclude that two places are the same.
- Phone number: Even though each venue has its own mobile number, however the phone crawled sometimes doesn't belong to the place but it does belong to the website. Therefore, this field will not be considered.
- Description and name: The description of two records describing a same place contains some similar information, for example, number of rooms or rent fee in a hotel. However, we have to extract useful data from unstructured information to compare two description, it is a complicated problem and will not be mentioned in this paper.

We present our algorithm as below:

Each place P contains a set of attribute
P = {Name, Num, Street, Dist, Lat, Lng, Source}
Where
Name : place' name *Num*: address number
Street: address street *District*: address district
Lat: Latitude *Lng*: Longtitude
Source: the website contains the information
We developed these modules:
CalDistance(P1, P2): calculate the distance between two resources.
Comp2Str(N1, N2): Compare the similarity between two string. The return value is between 0 and 1, the larger the value, the higher the similarity.
CompAddr(S1, S2). In which, S1, S1 could be number, street or district. The function return TRUE if two parameter are the same or one parameter is null.
After experimenting, we conduct some conditions for determining two places are the same:

$$\begin{cases} Comp2Str(Name1, Name2) > 0.7 \\ CompAddr(Number1, Nuber2)\ AND\ CompAddr(Street1, Street2) \\ AND\ CompAddr(District1, Distric2) \\ CalDistance(P1, P2) < 1km \end{cases}$$

Figure 3. Duplication detecting algorithm

4) Semantic enrichment

After above steps, the system converts the records from relational database to RDF statements. Each places is corresponding to a triple.

The RDF model of the ontology VTIO is presents as following:

```
<!—-header part, namespaces declaration -->
<rdf:RDF xmlns="http://hust.se.vtio.owl#"
xml:base="http://hust.se.vtio.owl"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<!— Cuisine place classes construction -->
        TO DO
<!— Accommodation place classes construction -->
        TO DO
<!— Attraction place classes construction -->
        TO DO
<!— Street classes construction-->
        TO DO
<!— District classes construction -->
        TO DO
<!— Province classes construction-->
        TO DO
</rdf:RDF>
```

Figure 4. RDF file structure

Similar to other ontologies, our ontology VTIO contains of the definition part and the expression part [10].

The definition part is referred as ontology's frame. It includes the definition of each classes, their DatatypeProperty and ObjectProperty.

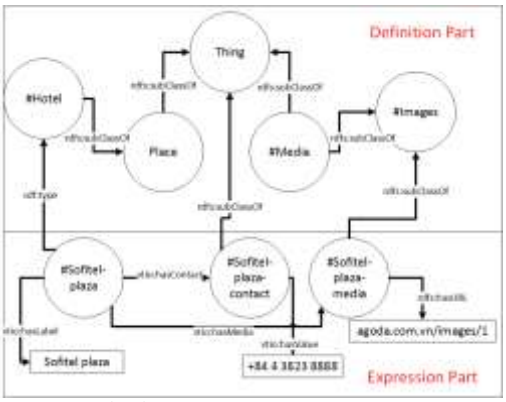

Figure 5. Example of definition and expression part in ontology VTIO

After describing a full-detailed frame, the main job of constructing a tourism knowledge datasets is developing the expression part of the ontology which is where the information stores.

One of most important steps is generating the base URI. It is the header of every URI belongs to places. Because each places is defined as an URI so there must not have any duplication. Therefore, we generate the URI based on this formula:

$$URI = name + number + street + district + datatype$$

For example: Sofitel Plaza is at 1 Thanh Nien Street, Ba Dinh District, Ha Noi, so its media uri would be: *#sofitel-plaza-1-thanh-nien-ba-dinh-ha-noi-media.*

In this module, the system also explores the information obtained and extract useful data to integrate into our semantic datasets. For instance, from the description or service value of a places, we could extract useful data to generate place attributes, for instance, vtio:relatedToTopic(for example: asian cuisine) or vtio:hasWifi = true if the service value contains word 'wifi'. The method is using comparison algorithms.

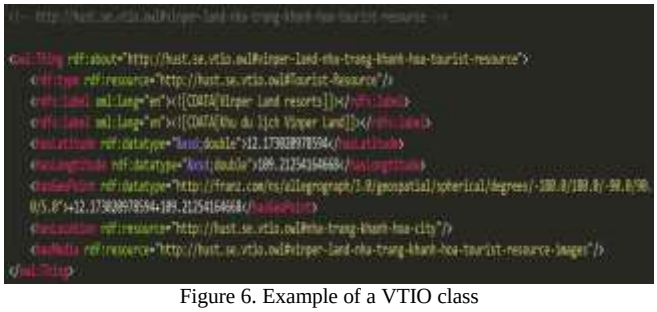

Figure 6. Example of a VTIO class

IV. Improving the accuracy of categorization method

A. *Keyword based method (KBM)*

Categorizing places has been and always be a challenge problem in data crawling. In our ontology VTIO we have utilized 70 categories and 5 levels so the problem is far more complicated.

Initially, the categorization jobs are performed by comparison method using the information about places which is the name or the former category collected previously. They are then matched with the list of given categories, and the name of the places is considered more important than its former category

The results after processing with the Thua Thien Hue database presented below:

- Number of records: **406 places**
- Number of places categorized correctly**: 328 places**
- Accuracy: **328/406 = 80.82%**

It is clear that the accuracy of this method is not really good, if the input contains thousands of records, the error would be very large and it would be a problem for manual validation.

B. *TF-IDF based method (TBM)*

As the results of the KB method, it is certain that name of the place is not enough to determine its category. Therefore, we has approached a new way to categorize the venue by using its description. The algorithm proposed in our work is based on **TF-IDF** [11]. The process is presented as following:

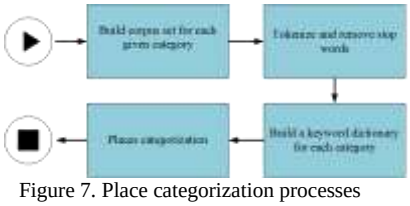

Figure 7. Place categorization processes

Step 1: Build corpus corresponding to each categories

We gather some datasets with the places that have already been categorized. Each categories is corresponding to a corpus containing descriptions of 1000-2000 places.

Step 2: Tokenize and remove stop words

In the description of every records, there are various kinds of words such as noun, verb or adjective that describes the corresponding places. However, there are many words, for example, numbers or adverbs from which descriptive information about places can hardly be extracted. Consequently, we tokenize the corpus of description and remove any stop word or unnecessary tokens before next step.

Step 3: Build a keyword dictionary for each categories

After first two steps, we calculate the meaning of each token in the corpus to found out how much its value corresponding to a specific category. We use TF-IDF algorithm for this steps.

TF-IDF: Term Frequency – Inverse Document Frequency, is a numerical statistic that describes how important a word is to a document in a collection. The more popular a word is, the less meaningful information we can extract from it.

In our system, the TF-IDF will that determines the importance of a word with a given category is calculated using the following formula:

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D) \quad (1)$$

Where:

- TF – term frequency: measures how frequently a term occurs in a document.

$$tf(t,d) = \frac{f(t,d)}{\max\{f(w,d) : w \in d\}}$$

  - tf(t,d) – Number of times term t appears in a document
  - max(f(w,d): w ϵ d} – the maximum number of appearances of term t in a document

- IDF – inverse document frequency: measures how important a term is.

$$idf(t,D) = \frac{|D|}{|\{d \in D : t \in d\}|}$$

  - |D| - Total number of documents
  - |{d ϵ D : t ϵ d}| - number of documents where the term *t* appears (i.e.$tf(t,d) \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to $1 + |\{d \in D : t \in d\}|$.

After applying TF-IDF, we have a keyword dictionary and their values with each given categories.

Step 4: Places categorization

Calculate a connection of a description with a given category to find out the corresponding category of the place.

$$C_{category} = \sum W_{category\text{-}keyword} * C_{keyword} \quad (2)$$

Where:

- $C_{category}$ – The value show the relativeness of a description with a given category
- $W_{category\text{-}keyword}$ – The weight of a "keyword" with a given category (Calculated in step 3)

- $C_{keyword}$- Number of appearances of a keyword in the description

C. Experimental Results

With ViSIS, we have gathered information of many cities in Vietnam such as Hanoi, Hochiminh or Danang from various sources on the Internet. In this paper, we have given a test case using the datasets of Binh Thuan city which contains 986 records:

- 420 records of accommodations
- 500 records of cuisines places (restaurants, coffee…)
- 66 records of attractions

All records contains information of name, description and former category and other fields. The system will automatically process the data to calculate the place's category. They had been categorized using both keyword based method (KBM) and TF-IDF based method (TBM). The results are evaluated manually by experts by reviewing the output shown on the web interface.

The results is presents in table 1:

TABLE 1. EXPERIMENTAL RESULTS

Type	Result	KBM	TBM
Accommodation	Correct	364	408
	Wrong	56	12
	Undefined	0	0
Cuisine	Correct	406	468
	Wrong	76	25
	Undefined	18	7
Attraction	Correct	51	62
	Wrong	6	1
	Undefined	9	3
All	Correct	821	938
	Wrong	138	38
	Undefined	27	10

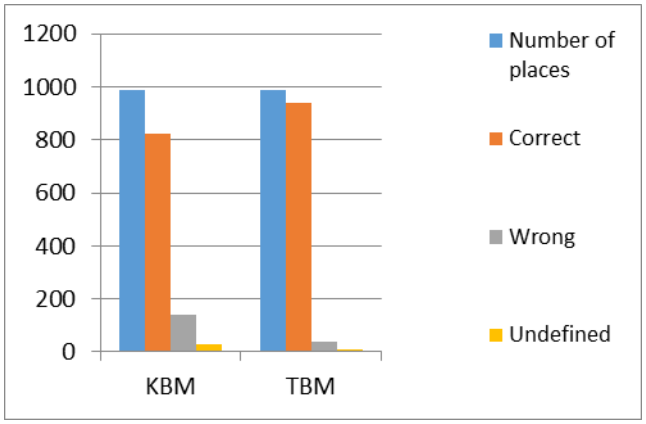

Figure 8. Experimental chart

TABLE 2. RECALL AND PRECISION

	KBM	TBM
Recall	97.26%	98.99%
Precision	88.74%	96.11%

The results show that after improving the algorithm, the accuracy of the categorization process has been increased significantly. Even though there are still several errors as the information collected are not enough and has to be categorized manually, the results have advanced the semantic data system about tourism.

V. Conclusions & Future works

This paper presents a semantic system which is one of the first research about semi-automatically generating semantic tourism data annotation from various sources on the Internet, particularly on those in Vietnam. The system not only gathers, integrates the information but also handle duplicated records and enriches the data. It also proposes an effective and accurate method for categorizing the tourism places, improves the information access for semantic applications such as STAAR or DBpedia. In the future the ViSIS would be gradually automatized every steps of synthetic the place semantic data from the whole internet to upgrade our ontology model to public and contribute to the linked datasets of the semantic web community.

References

[1] Allemang and Hendler, "Semantic web for the working ontologist", 2008

[2] Eero Hyvönen, 'Publishing and Using Cultural Heritage Linked Data on the Semantic Web'

[3] Julien Tscherrig, Philippe Cudr´e-Mauroux , Elena Mugellini, Omar Abou Khaled, and Maria Sokhn: SemantiConverter: A Flexible Framework to Convert Semi-Structured Data into RDF

[4] Chang-Su Kim, Sung-Han Kim, Hoe-Kyung Jung: A Study on Web Standard-based RDF Converter by Applying Linked Data and using RDF/XML Standard format for Data: 2015

[5] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer. Christian Bizer, "DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia"

[6] Stefano Cresci, Andrea D'Errico, Davide Gazz´e, Angelica Lo Duca, Andrea Marchetti, Maurizio Tesconi, "Towards a DBpedia of Tourism: the case of Tourpedia".

[7] Tuan-Dung Cao, Quang-Minh Nguyen, "Semantic approach to travel information search and itinerary recommendation", 2011

[8] Philippe Le Hégaret, Lauren Wood, Jonathan Robie, "What is the Document Object Model?", W3C

[9] Alan Ark, "A Selenium-WebDriver Case Study", 2015

[10] Sean Bechhofer, "OWL: Web Ontology Language"

[11] Stephen Robertson, "Understanding Inverse Document Frequency: On theoretical arguments for IDF"

About Author (s):

Dr. Tuan-Dung Cao is the Vice dean of School of Information and Communication Technology, Hanoi University of Science and Technology. He obtains his PhD in Computer Science - Research in Computer Science and Control (INRIA)

Anh-Tien Le is a student at School of Information and Communication Technology, Hanoi University of Science and Technology. He is majored in Computer Science.