ISSN 1598-0170 (Print)
ISSN 2287-1136 (Online)
http://www.jksii.or.kr

An effective approach to generate Wikipedia infobox of movie domain using semi-structured data

Hanif Bhuiyan[1] Kyeong-Jin Oh[1] Myung-Duk Hong[1] Geun-Sik Jo[1*]

ABSTRACT

Wikipedia infoboxes have emerged as an important structured information source on the web. To compose infobox for an article, considerable amount of manual effort is required from an author. Due to this manual involvement, infobox suffers from inconsistency, data heterogeneity, incompleteness, schema drift etc. Prior works attempted to solve those problems by generating infobox automatically based on the corresponding article text. However, there are many articles in Wikipedia that do not have enough text content to generate infobox. In this paper, we present an automated approach to generate infobox for movie domain of Wikipedia by extracting information from several sources of the web instead of relying on article text only. The proposed methodology has been developed using semantic relations of article content and available semi-structured information of the web. It processes the article text through some classification processes to identify the template from the large pool of template list. Finally, it extracts the information for the corresponding template attributes from web and thus generates infobox. Through a comprehensive experimental evaluation the proposed scheme was demonstrated as an effective and efficient approach to generate Wikipedia infobox.

☞ Keyword: Wikipedia, Semantic relation, Identification, Infobox Template, Information extraction, Semi-structured data.

1. Introduction

Wikipedia is a free, multilingual, collaborative and community-based largest online encyclopedia. It is growing rapidly and has gained immense popularity since its beginning in 2001, 18 billion page views and each month nearly 500 million unique visitors (February 2014) [1]. Most importantly, it is a constantly evolving tapestry of richly interlinked textual information, which indicates its immense source of collective knowledge and semantic concepts [4], where infoboxes[1)] function as the structural information hub of the web.

The prime element of Wikipedia is its article. Usually, each article in Wikipedia describes a single topic and its title is a succinct composed of well-formed phrases that resembles a term in a conventional thesaurus [5]. But representations of information in these articles are different from each other. Articles information in Wikipedia split into two section: 1) detail explanation in a conventional unstructured text about the article's subject and 2) summary of the article's information in a semi-structured form. Such structured form of data is Infobox. An infobox is a table, consist of a set of attribute-value pairs and associated with an entity type. In a sense, it is a reflection of Wikipedia article in a potted format. It comes with a small box on the top right side of the Wikipedia article page. The availability of such information has opened up new possibilities for researchers. There are many applications (Freebase[2)], Google Knowledge Graph[3)], etc.), which used infobox information in the web. DBpedia [2] project made a rich knowledge base by extracting this (infobox) information from Wikipedia and are using for sophisticated queries.

In the web, accessing data is simpler and much faster from structured data than unstructured data. To assist in searching for a queries like "What is the name of that actor who acted in movie Troy and whose wife is Angelina Jolie", systems have been proposed to develop a model of Wikipedia article each. Every article is an entity associated with entity type and each entity contains a set of attribute-value pairs and every entity has relationship with each other [6, 7]. For the above query, the article *Troy* is associated with two entity types: *Film* and *Actor/Actress*, and the infobox of the *Troy* film has attribute

1 Dept. of Computer Science & Information Engineering
Inha University, Incheon, Republic of Korea
* Corresponding author (gsjo@inha.ac.kr)
[Received 10 February 2017, Reviewed 5 March 2017(R2 5 April 2017), Accepted 25 April 2017]
☆ This study was reconstructed using data from the master 's thesis written by Hanif Bhuiyan in February, 2016.

1) http://en.wikipedia.org/wiki/Infobox
2) http:www.Freebase.com/
3) http://www.google.com/insidesearch/features/search/knowledge.htm

value pairs about its actor *Brad Pitt*. The entity *Troy* is connected to the *Brad Pitt* Wikipedia article through the spouse relationship with *Angelina Jolie* and *Brad Pitt*. Similarly, infobox is playing important role in other applications such as review summarization, document categorization and question answering [8]. Inspired by this observation, many projects have made such structured form of data and many are still running to create them [1, 6, 9, 10]. Despite of those great achievements of Wikipedia knowledge, there exist some issues and untapped potential. However, infobox creation and enrichment still requires mammoth collaborative task from the author (expertise and volunteer of Wikipedia). This manual task can divide into two parts: 1) Identification of the infobox template and 2) Fill the attributes value of the template and create infobox.

For the template selection of articles, Wikipedia provide around 1809 infobox template[1] for the participator, while these templates are standard way to represent infobox. Therefore, contributors should choose the appropriate template from the template list. But it is quite hard to ensure whether the participant's selection are correct or not. Even there is no central authority checking for the template selection. Sometimes, the process becomes more complex for the general and new user who tend to create infobox.

In order to fill the attributes value users can fill the value based on their knowledge with authentic reference. But there is no justification that the value or attribute name is appropriate or not, which often leads to inaccurate infobox. Sometimes, different names are used to denote the same attribute (e.g. birthplace and placeofbirth) in different infobox [11], and hence the infoboxes possess several challenges in the scale of perfection and quality. Also, this kind of manual processing is complicated, time consuming and causing issues of inconsistency, heterogeneity, and the numbers of articles having infobox in Wikipedia [12]. Compare to the total articles in Wikipedia, the number of articles having infobox is half. As of 16 March 2015, almost 2.2 million Wikipedia English articles were being infoboxed whereas the total number articles was 4.72 million. Regarding the problems to generate Infobox, several research have been done and some of them reported significant progress [8, 11, 13, 14]. Therefore, an automatic process to generate infobox by reducing those error-prone issues seems conceivable.

To identify the infobox template and to generate infobox, prior works were mainly relied on corresponding article text. However, the accuracy of template identification and perfection of infobox generation depends on the availability of the information about its entity. But, due to the heterogeneity of the article's content in Wikipedia, a number of articles have very few important facts about its entity. On the other hand, due to the manual creation and updating information in the Wikipedia, all articles are not equally rich in information and therefore, dependent on the information from other sources. Generally, an article in Wikipedia consists of abstract (introduction), content, reference and categories. Interestingly, there are a lot of articles which does not have all these four features. To justify it, a survey was performed on 500 articles. Results showed that 28% article have no content and 18% article have no content and reference both. The scenario is much worse for the new articles but over time it becomes informative by the contribution of expert volunteers. Driven by this analysis and self-observation, it is speculated that when new article is created, the Wikipedia has only abstract (introduction) and categories. Therefore, considering all these facts, the methodology is designed to create infobox.

The aim of this work is to solve those aforementioned problems in order to generate infobox. We propose an effective and efficient method to automatically generate infobox for movie related Wikipedia article. Our approach works in two steps: first, an NLP based automatic method that semantically derives the infobox template of Wikipedia article using abstract and categories of the text and second fill up the attribute values of that template using semi-structured data. In order to identify the template, the proposed method works in an unsupervised fashion rather than supervised fashion [13, 15]. And to fill up the template attributes value, we focus on several web sources instead of extracting information from the unstructured text [8, 13, 14] of the article.

2. Related Works

Several research works had been undertaken on Wikipedia in terms of article classification and information extraction from article and web sources.

2.1. Article Classification to Identify Template

Classification of an article is a methodology of assigning documents into labeled groups, where labels come from a pool of pre-determined categories. The traditional approach is to represents documents by the words they contain, and use training data to mark out words that are indicative of each class label [18]. Wikipedia allows categorization techniques to draw on background knowledge about the concepts the word represent. Instead of classifying into infobox template types some approaches classified Wikipedia articles based on common Name Entity Recognition (NER: Person, Location and Organization) [19, 20]. To categorize Wikipedia article into NER, Dakka and Cucerzan [19] use NB and SVM classifier and found that SVM is better. Nothman et al. [20] transformed Wikipedia's links into named entity annotations by classifying the target articles into common entity types.

Sultana et al. [15] build a SVM classifier to classify Wikipedia articles into their respective infobox template from a large number of types. Their main idea was to train the classifier model through the labeled training example (having infobox article) and to apply the classifier over unlabeled entities (without infobox article).

Wu and Weld [13] also classified Wikipedia article by inbox type in a supervised way depending on the presence of infobox type keywords in the article (within Wikipedia list pages) and infobox class keywords in the article's categories. If articles are failed to fulfill any conditions among two, then this approach fails to work.

However, all the aforementioned approach for classifying Wikipedia article were based on supervised techniques, where large amount of training dataset was required. On the contrary, our approach works in an unsupervised fashion where system learns the features automatically from the corresponding article text to achieve the classification goal of article for identifying the infobox template.

2.2. Information Extraction to Generate Infobox

The explosive growth and popularity of the World Wide Web has caused many unsupervised and supervised learning of information extraction process to make infobox.

Zhang et al. [14] extract knowledge from Wikipedia using rich set of linked entities of the article. The main idea of this approach is to summarize the entity and its linked entities and then embedded a rank aggregation approach to remove noise and finally applied a clustering and labeling algorithm to extract knowledge and thus generate infobox. Although Wikipedia successfully generates infobox, but there are many article that does not have enough text content with insufficient number of linked entities.

Wu and Weld [13] devised an automatic process KYLIN, to extract information from Wikipedia article text to make infobox (structured information). The basic idea was to use existing infoboxes as a source of training data, then learns to CRF extractors and finally implied on each article page for creating new infobox. Even though they have shown impressive improvement by extracting knowledge from unstructured text and generating infobox but the whole process based on the supervised learning where large amount of training data was required.

Mousavi et al [8] proposed a Natural Language Processing (NLP) based text mining system (IBMINER) to derive structured information from the free text of Wikipedia article. In order to generate structured text, linguistic morphologies of the sentences is used by applying Semscape text mining framework, where texts are converted into TextGraphs. However, deducing infobox from the unstructured text by extracting triples from the existing text, IBMINER requires well written Wikipedia document.

On the contrary, for infobox information our approach extracts structured information from several sources of the web based on the template. We used API service technologies to obtain the structured information from the disparate web resources. Currently, API services are marked in the proliferation of web, for discovering or invoking the vast majority of data [21, 22]. Ly et al. [21] attempted to automatically extract relevant technical information such as operation names, operation descriptions etc. from the web pages by exploiting Web APIs.

3. Generating Infobox

This sections outlines in details about infobox generation through the proposed methodology. The proposed system

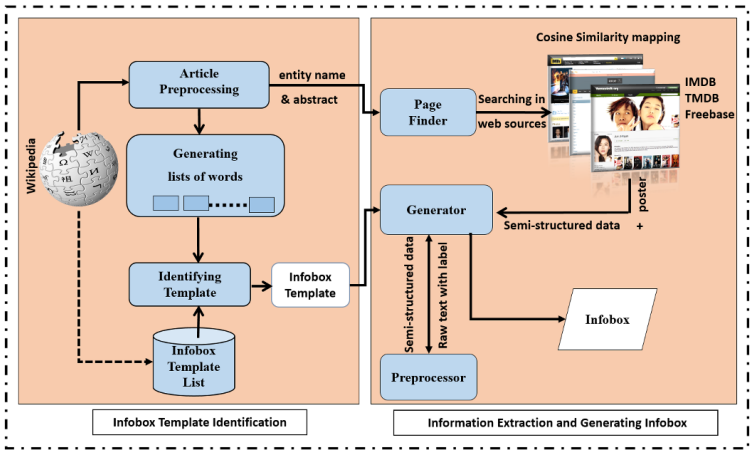

Figure 1. System Architecture for Generating Infobox

architecture is formulated in two step as shown in Figure 1. The first step identify the article infobox template and in the second step, fill that template attribute value and thus generate infobox. The brief description of these two major steps is given below.

3.1. Template Identification

A Natural Language processing (NLP) based unsupervised method is proposed to semantically identify the infobox template using abstract and categories of article instead of whole article text.

For an example, Figure. 2 (a) and (b) presents the abstract and categories of a Wikipedia article and Figure. 2 (c) presents the identification of the template of that article. Usually, every Wikipedia article has some important words which conveys implicit and effective meaning about the article and this type of words often appear in the text. Empirically, it is found that infobox type keywords were also used as an important word for the corresponding article and in most cases captured by the NN word of the relational sentences (containing relational patterns) of the article text. Therefore, to determine the infobox template keyword from the selected text, we focus on two features of the text: i)NN (Noun Singular Word) words in the semantic relational sentences and ii) important words of the text [3, 16] instead of using labeled training data [15]. The brief description of these two features are given below. Moreover, in the analysis of Wikipedia article (mentioned in Introduction) it is observed that article having infobox is well constructed and quiet rich in text than article without infobox. These differences in the text contents might create a gap in the accuracy of correctly identifying approach in both types of articles. Therefore, we evaluated the efficiency of the proposed approach in both types of articles. The process is completed in three segment as shown in Figure 1. Initially, selected article text (abstract and categories both or abstract only) is the primary input of the process. Secondly, the text is converted into raw format through preprocessing and generated some classified word lists. Finally, the Template_ Selection algorithm is applied on the word lists to mark out the template keyword and to determine the template from template list

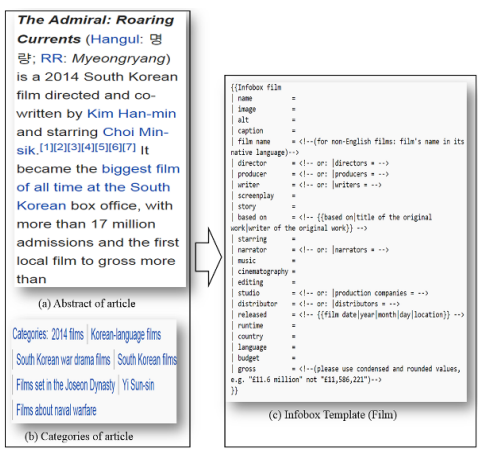

Figure 2. Identifying Infobox template of a Wikipedia article

3.1.1. Article Preprocessing:

Before generating the lists of words from the extracted article text we perform some preprocessing. All abbreviations are replaced to their original form (e.g. Ltd. to Ltd, U.S.A to USA, Dr. to Dr, Mr. to Mr, etc.). Except period ".", all punctuations are removed from the text. Subsequently, numbers, dates and unknown characters are also removed from the text. In essence, we only consider the raw text of the article where we can identify sentences and words based on period "." and space " " respectively.

3.1.2. Generating Lists of Words:

To obtain important words of the article text, Bag of Words (BOW) is made based on Term Frequency (TF). Before constructing BOW, all the words of the text are converted into their singular form. Stop words are removed from the text by applying MySQL full text stop words list. To make BOW, we consider the semantic meaning of each word and for that WS4J[4] (WordNet Similarity for Java) is used. We consider only Path Length and Wu & Palmer similarity measures [3, 17] for the semantic meaning of the word. For this two similarity measurements we set the threshold θ =0.85, where v (similarity value) >= θ, system determines similar word. Pragmatically and intuitively we consider that in the BOW, first seven words are the most important. Therefore, first word list W_1 is made from the first seven words of BOW. Occasionally, name entity word can become higher order in the BOW, which might create problem to determine the required keyword. Therefore, we determine NEs (Name Entity) by parsing first sentence of the article through Stanford Name Entity Recognizer[5] and remove those words from the first word list and make second word list W_2. By keeping first three word of W_2, new word list W_3 is made. In the extracted article text, we select the sentences those contain hyponym and holonym relation patterns (is-a and has-a). Then, in the sentences (Noun Singular) NN word after the patterns are marked by Stanford Part-of- Speech tagger[6] (POS Tagger) and making BOW (W_4). Such an example of semantic relational sentence is shown at Figure 3 and Table 1 shows the relational patterns. From W_4 select first three word and make the word list W_5. In the descending order from W_4 make another word list W_6 by keeping highest frequent word. Finally, we make the word list W_7 by extracting the first sentence's NN word.

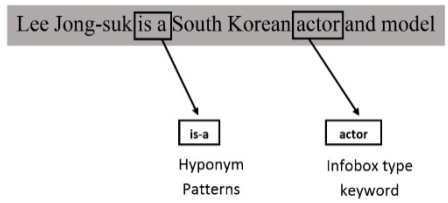

Figure 3. Sample example of Hyponym patterns

Table 1. Typical patterns for is-a (hyponyms) and has a (holonyms) relations

Relation	Patterns
Hyponym	is a, is an, is the, are a, are an, are the, was a, was an, was the, were a, were an, were the
Holonym	have a, have an, has a, has an

Algorithm: Template_Selection
Input: Article Text, Classified seven word list, Semantic relations (hyponym and holonym patterns)
Output: Infobox Template
1. Make the set of words from the word lists to search infobox type keyword.
{(W1 ,W4), (W3 ,W5), (W2 ,W4), (W6 ,W1), (W1 (first word) ,W4)}
2. If article text contains any semantic relation pattern
a. Search infobox type keyword in the set of words. If keyword found in any set then go to step 4. ELSE
b. Search the Noun Singular Word in W7 and define it as a keyword then go to step 4. ELSE
c. Define the first word of W4 as a keyword and go to step 4.
3. If article Text does not contains any relation patterns
a. Search the Noun Singular Word in W7 and define it as a keyword then go to step 4. ELSE
b. Define the first word of W4 as a keyword and go to step 4.
4. Semantically identify the template from Template List.
Based on the infobox type Keyword search in the template list and return the required template.

Figure 4. Summary of Infobox Template_Selection Algorithm

3.1.3. Identifying Template:

When word list is created, Template_Selection algorithm

4) http://code.google.com/p/ws4j/
5) http://nlp.stanford.edu:8080/ner
6) http://nlp.stanford.edu/software/tagger.shtml

(Figure 4) is applied. The whole process of the algorithm works in mainly four steps. First, it groups the word lists into different sets. Second, it searches for keyword in some sets of words, if article text contains the semantic relation patterns. But if article does not contain relation patterns then the third step works to mark the template keyword. Whenever in the second or third step infobox keyword is found, the process triggers the fourth step to get the infobox template for the article. To justify the template name with the derived infobox keyword, we also use the semantic similarity technique (WS4J[4] API mentioned above).

3.2. Information Extraction and Generating Infobox

Now a days, obtaining information about any topic is becoming spontaneous in the rapidly growing web. The World Wide Web (WWW) offers enormous number of facilities for the web users, collaboratively information sharing process, free access to extract information etc., which are mainly causing the significant growth of semi-structured information resources. Specifically, for movie domain, in the web, there are lots of such web resources like, Internet movie database (IMDB), Freebase, The movie database (TMDB), Rotten Tomatoes, Allmovie, etc., which are widely exploiting in the field of automatic information extraction for various aspects. Moreover, day by day the importance of those websites are increasing in the web information system due to the availability of such structured information for the specific areas. Therefore, to scale up the proposed approach to generate infobox from the template, system extracts information from three real data web sources of semi-structured information. Few sub- domains (film, actor / actress) are specified of movie domain of Wikipedia to create infobox. In this work, system exploits three websites, (http://www.imdb.com, http://www.tmdb.com, http://www.freebase.com).

However, Wikipedia does not provide specific infobox template for actor or actress article type, even though the template identification process can categorize those article into their respective infobox type. Therefore, for those articles, self-defined template is used which was made through the observation (mentioned in introduction) on articles about this domain. For film article type, Wikipedia recommended template is used. However, in infobox, often attribute name differs from article to article [11], to handle this, the attribute name is primarily set, like *Born* can be *Birth Date, Birthday*. Based on the template attribute, the number of achievable information can be different from source to source. So, a survey is made that what kind of information is provided by which source for which topic. According to the composition of obtainable and accessible information, the system made the query for each source. In terms of extracting information, sometimes one attribute can have several values (information), for such cases some heuristics are applied, i.e. attribute *production company* in the film template, the system will extract top three information from all available information. Also, the same information can be available in several sources, so the priority is given for the sources. If information of one attribute is found in the first source, then the system will not check that attribute information in other sources, it will check the remaining attribute information. The priority order of sources is *TMDB→IMDB→FREEBASE*. The same priority order is applied for extracting poster (image). The automatic process for extracting information is introduced briefly as below:

Figure 1 represents the methodology of the proposed approach for generating infobox from the template. The proposed approach consists of two major components: Page Finder and Generator. Through the template identification process, after getting the entity name, infobox template and article's abstract, the page finder process triggers on. Page finder module then search for the data on those selected websites, finally locate and extract the incidental information and send to the Generator. Before setting up the value, data are processed through the preprocessor for converting into the template format. In the end, generator provides the complete infobox. The brief explanation of these two components are given below.

3.2.1. Page Finder:

The goal of this section is acquisition of appropriate semi-structured data for the infobox template. A focused crawler is implemented to address this task. It uses the entity name and article type (entity name and article type obtain by identification technique) to look for the data in the web and formulates such a query using web API through HTTP GET method. Each query returns JSON data. But sometimes there are several sections

(example in Figure 5) of data can appear, among them which one is appropriate for the article, the system justifies it through cosine similarity mapping between two documents (article abstract and located overview text). The action of searching is different for source to source. To access in the web, use the Application Program Interface (API) but each website requires its own API style to allow the access. First, introduce how to search and extract the structured data in The Movie Database (TMDB), second Internet Movie Database (IMDB) and in the end Freebase.

A. The Movie Database (TMDB):

TMDB is a crowd sourced movie information data repository, conducted in many film related web sites, applications such as Wikipedia, Freebase, MythTV, Plex, etc. Due to the availability and accessibility of structured information through HTTP API, now a days it is used for many research applications. To fetch the information from TMDB first, system formulates a query through API using the entity name of the article. For answer of the query TMDB returns all the possible information including their unique ID that match with the entity name, where only one section of information is exactly related to the entity. Therefore, to pick the right information first, identify the ID from the answers by measuring the cosine similarity (equation 1) between article abstract texts with every section's (Figure 5 searching for movie "The Monkey King" information) overview text.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

In the calculation, which section will acquire the highest similarity value, the system fetches that ID of that section, formulates a new query using that ID and fetches the exact information (JSON data) and then send to the generator. From that information, poster path is used through new API and fetch the required image for the infobox template.

B. Internet Movie Database (IMDB):

IMDB is one of the biggest data repository for up-to-date movie related information on the web and freely available for

Figure 5. Querying for information in TMDB and the primary result

personal use and research purpose. Availability of such structured data has inspired the researchers about it. To build movie related application, websites, etc. IMDB provides the opportunity to use its data. Now a days many websites use IMDB information such as Freebase, Wikipedia etc.

For fetching IMDB information, system uses two HTTP API (OMDB and My Api Films). According to the entity name of the article text (obtain from type identification technique) search for the data and receive the JSON format data and then send to the generator. The extraction system extracts film information using OMDB (http://omdbapi.com/) and person entity information through My Api Films (http://www.myapifilms.com/). For an example, in Figure 6. fetching information for a film (Troy) and an actor (Brad Pitt)

C. Freebase:

Freebase is a practical, scalable, graph shaped repository of structure large collaborative knowledge, inspired by broadly used collaborative data communities such as the Wikipedia and semantic web. Freebase allows public read/write access through an HTTP based graph-query API using the Metaweb Query Language (MQL) as a data query and manipulation language for

the research purposes, the creation and compliance of structured knowledge, and building and developing applications [9].

The initial approach to access data storage of freebase is through its public HTTP-based API, where posed the MQL query (according to the entity name which obtain from template identification technique) and the answer will be formulated in JavaScript Object Notation (JSON) Notation syntax. For an example in Figure 7, fetching an actor information.


```
Title: "Troy",
Year: "2004",
Rated: "R",
Released: "14 May 2004",
Runtime: "163 min",
Genre: "Adventure",
Director: "Wolfgang Petersen",
Writer: "Homer (poem), David Benioff (screenplay)",
Actors: "Julian Glover, Brian Cox, Nathan Jones, Adoni Maropis",
Plot: "An adaptation of Homer's great epic, the film follows the assault on Troy by
the united Greek forces and chronicles the fates of the men involved.",
Language: "English",
Country: "USA, Malta, UK",
Awards: "Nominated for 1 Oscar. Another 3 wins & 17 nominations.",
Poster: "http://ia.media-
imdb.com/images/M/MV5BMTk5MzU1MDMwMF5BMl5BanBnXkFtZTcwNjczODMzMw@@._V1_SX300.jpg",
Metascore: "56",
imdbRating: "7.2",
imdbVotes: "346,143",
imdbID: "tt0332452",
Type: "movie",
Response: "True"
```


(a) Film (Troy) information using OMDB API


```
actorActress: "Actor",
birthName: "William Bradley Pitt",
born: "William Bradley Pitt December 18, 1963 in Shawnee, Oklahoma, USA",
dateOfBirth: "18 December 1963",
height: "5' 11" (1.8 m)",
idIMDB: "nm0000093",
name: "Brad Pitt",
placeOfBirth: " Shawnee, Oklahoma, USA",
spouse: [
  - {
      childrens: "6 children",
      dateFrom: "23 August 2014",
      dateTo: "present",
    - nameId: {
        id: "nm0001401",
        name: "Angelina Jolie"
      }
    },
  - {
      dateFrom: "29 July 2000",
      dateTo: "2 October 2005",
    - nameId: {
        id: "nm0000098",
        name: "Jennifer Aniston"
      },
      status: "divorced"
    }
],
urlPhoto: "http://ia.media-
imdb.com/images/M/MV5BMjA1MjE2MTQ2MV5BMl5BanBnXkFtZTcwMjE5MDY0Nw@@._V1_UX;
```


(b) Actor (Brad Pitt) information using My Api Films Api

Figure 6. Extracting information from IMDB through OMDB and My Api Films

Figure 7. Information extraction in Freebase

3.2.2. Generator:

The generator module devises the infobox by integrating all the information from the page finder module and template detection methodology (section 3.1). It receives the entity name and the infobox template from the template identification. Then prepare the template to attach the respective extracted information to the attribute. Alongside after obtaining the semi-structured data and image from the page finder module, it sends the data for preprocessing. Preprocessor applies the boilerplating to get the information respect to each attribute and then provide that processing information to generator. Generator then removes the ambiguity of data. Finally, attach the image on the template and information to the attribute and thus generate infobox. Figure 8. shows the example of devising new infobox through the proposed approach.

Figure 8. Appearance of new infobox respect to Wikipedia

4. Evaluation

Experimentation is conducted in order to prove the effectiveness of the proposed approach to generate infobox automatically. Initially, the accuracy of the template identification approach is evaluated first, for determining the infobox template. Afterward, the information extraction approach is evaluated with respect to manual processing to fetch information for the template attribute. Subsequently, the new

Table 2: Accuracy of identifying infobox template

Domain	Dataset	Number of articles	Experiment text of articles	Accuracy of Identification	Accuracy (WordNet similarity used)
Movie Domain	Dataset 1 With Infobox	500	Abstract	69.4%	72.4%
			Abstract & Categories	87.2%	88.6%
	Dataset 2 Without Infobox	250	Abstract	48.4%	51.2%
			Abstract & Categories	77.2%	80%
Other Domains	Dataset 3 With Infobox	500	Abstract	72.2%	74.8%
			Abstract & Categories	79.6%	83.8%
	Dataset 4 Without Infobox	250	Abstract	62%	63.6%
			Abstract & Categories	77.6%	79.6%

infobox (generated through the proposed approach) is evaluated with respect to the current infobox (Wikipedia infobox) based on user judgment. Finally, draw the end of this chapter with some observations on problem cases that could not yet be handled through our approach.

To run the experiment, data dump of 2015-03-16

English Wikipedia was downloaded, whereas 4.72 million articles was available, 2.2 million articles were being infoboxed and the rest 2.52 million articles have no infobox.

4.1. Template Identification

In this section, it is examined that how many articles of infobox template can be identified through the proposed approach (described in chapter 3.1). The effectiveness of the proposed approach was evaluated in different types (river, canal, bank, etc.) of Wikipedia articles alongside three types of movie related articles (actor, actress and film).

To evaluate the approach, four dataset was created from those 4.72 million articles (data dump of 2015-03-16 English Wikipedia). Dataset 1 and Dataset 2 are movie related article. Dataset 3 and Dataset 4 are other types of article. Dataset 1 was made by selecting 165 articles for each template type among three type of movie related article and made the total articles number 500. For dataset 2, 250 articles of those three types were randomly extracted from non-infoboxed articles. Dataset 3 was made by selecting 50 templates (except movie related templates) out of more than 1800 infobox template [1]. For each type of template, 10 articles were randomly extracted from infoboxed articles. Finally, Dataset 4 was made by randomly selecting 250 articles from non-infoboxed articles except movie related articles. In the experiment, existing infobox template was used as a ground truth for the article having infobox (Dataset 1 and Dataset 3) to justify the result given by the system. But, for article without infobox (Dataset 2 and Dataset 4) prior to the experiment, ground truth was prepared manually by labeling the articles to their infobox template. For both dataset, two different status was used for each article. First abstract part was considered and second, the abstract and categories part were considered together. For each article, accuracy was reported by determining whether the infobox template is correctly identified or not. Experiment result is shown in Table 2.

Compared to other domains of article, our system performs better for having infobox of movie domain. But for without infobox article, the accuracy of our system is much worse for movie domain than other domains. For the movie related article in Dataset 1 and Dataset 2 the accuracy of the identification methodology is 87.2% and 77.2% when abstract and categories were considered together. But when only abstract was used then the accuracy was decreased to 69.4% and 48.4% respectively. In all cases, the accuracy was increased marginally when WordNet semantic similarity (WS4J API) was applied. Similar trend of accuracy was observed in case of Dataset 3 and Dataset 4. However, better accuracy was seemed for Dataset 1 and Dataset 3 in terms of determining infobox template of articles.

From the result, it can be concluded that, the accuracy of the template identification is more accurate for having infobox article compared to article without infobox. As an experiment text, using abstract and categories is much effective to determine the infobox template rather than using only abstract. Although the accuracy for determining the infobox template is high but a good portion of article's template could not be identified through our methodology as it considers only the single infobox

template keyword. Therefore, this approach can cover around 400 infobox template out of 1809 which may provide scope of research for our future task

4.2. Information Extraction

In this subsection, the efficiency of the proposed approach is examined in order to extract information from several web resources. Three types (Film and Actor / Actress) of movie related Wikipedia articles were evaluated through this approach.

To evaluate the approach, 100 articles were randomly extracted from that 2.2 million infoboxed articles. The information was extracted for those articles through our information extraction process based on the corresponding template which is identified through our template identification method. Among 100 articles, 50 articles were selected from film infobox type and another 50 articles from actor and actress infobox type. In order to compare the system efficiency, 10 volunteers were recruited to extract the infobox information for those selected 100 articles using the infobox template. For the manual creation, author can use information from Wikipedia article text or any other web resources.

To justify the efficiency of both processes it was assumed that, for a particular article the current Wikipedia infobox served as a ground truth. To compare the proposed approach with manual processing (author's work) in terms of extracting relevant information, precision and recall was measured. For monitoring the recall of the manual process and proposed approach, threshold value 0.7 and 0.6 had been set respectively. This indicates that to make the infobox for the specific article author have to collect 70% information and proposed system have to collect 60% information regarding the ground truth. Slightly lower threshold 0.6 recall value was set for the proposed system than manual process. And for precision both process have to exactly identify the article type. The infobox template attribute denote A as the number of relevant information retrieved, B as the number of relevant information not retrieved and C as the number of irrelevant information retrieved.

$$\text{Precision: } \frac{A}{A+C} \times 100\% \; ; \quad \text{Recall: } \frac{A}{A+B} \times 100\%$$

Table 3. Performance measurement for extracting relevant information for infobox template.

Infobox Type	Volunteers		The Proposed Approach	
	Pre (%)	Rec (%)	Pre (%)	Rec (%)
Film	100%	96%	91.67%	88%
Actor / Actress	95.83%	92%	90.90%	80%

After retrieving the information from both processes, manual checking was performed for the precision and recall performance whether it fulfils the condition (threshold scale) or not for extracting relevant information. The result of the experiment was recorded in Table 3, which reflects the comparison scenario between manual process and the proposed automatic process in terms of extracting infobox template attributes value. For the *film* type of articles, compared to manual process, our proposed approach works slightly worse. In the *actor / actress* domain, our approach works badly because of inadequate information of the article text and unavailability of information in the web sources. So, far inaccuracy in the manual process that might occur because of hurries from the volunteers or data duplicity of the article text or unable to access the appropriate source or lacking of appropriate information in the web source.

It can be concluded that, extracting information for the popular and viewable articles type are much harmonic than obscure article type. The performance can vary depending on the data set. If the selected article is well organized and the entity information is easily achievable in the web, then the performance can be better in both processes. Moreover, in the manual process, the performance can vary from volunteers to volunteers. Even though, extraction rate of information for the template is good but the proposed approach works on movie domain due to the availability of API and accessible of free information. If other domains also provide API and free information like IMDB, TMDB then it is also possible to generate infobox for them through the proposed approach.

4.3. Effectiveness

The effectiveness of the proposed system had been justified by comparing with the current infobox which served as a ground

truth for the particular article. For the experiment, two popular classes (film and actor / actress) were selected, where each was among the top classes in terms of infobox usage. 10 Wikipedia articles having infobox (5 film, 3 actors and 2 actresses) has been taken randomly to formulate new infobox through the proposed process. After formulating, 50 volunteers were recruited to justify the perfection, relevancy and representation of those infoboxes. Prior, some parameters were defined to rate the new infobox. The parameters are shown in below.

According to the parameters, volunteers rated the infobox (devised from the proposed system) based on their justification from 1 (No relevant information) to 5 (Complete information with perfect relevancy). Figure 9 represents the user satisfaction level (every 5 volunteers rating average) for infobox and Table 4 shows the overall satisfaction (3.692 out of 5) on the generated infobox with respect to the Wikipedia infobox (it is assumed that all the regarding article's infobox of Wikipedia score is 5).

o Complete information with perfect relevancy	5
o Not Enough information but relevant	4
o Few information but relevant	3
o Information partially relevant	2
o No relevant information	1

From the satisfaction result in Figure 9 and Table 4, it comes out that the proposed methodology can efficiently generate infobox for the movie related articles of Wikipedia. The user satisfactory level for the generated infobox is almost 73.84% (3.692 out of 5). In addition, in Figure 9, from the user judgment, it can be concluded that infobox creation is much enriched and perfect for the popular entity rather than unpopular entity. However, this satisfaction level can vary from person to person.

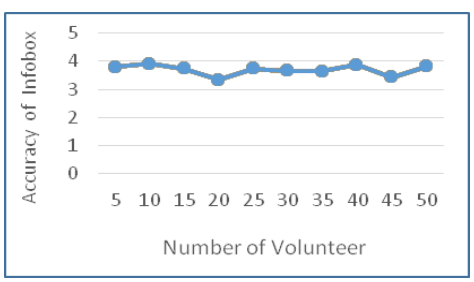

Figure 9. User satisfaction on the generated Infoboxes

Table 4. User satisfaction for the generated Infoboxes through the proposed approach

Matric	(Infobox) through the proposed Method	Wikipedia Infobox
Overall User Satisfaction	3.692	5

5. Conclusion

Infoboxes in Wikipedia pages are basically designed for instant look of the whole article to the readers. However, devising and enriching infobox is a challenging task. The generation of infobox is still dependent on manual processes, which is labor intensive and sometimes might create error prone infobox. Such problems mainly arise because there is no standard framework and integrity checking for this work.

This work proposes an effective and alternative approach to generate infobox automatically for movie domain of Wikipedia. Firstly, we used Wikipedia recommended template to select the appropriate template for a particular article. For that, we applied a Natural Language Processing (NLP) based approach on *abstract* and *categories* text instead of whole article text in an unsupervised fashion. Secondly, to fill the template attribute value, semi-structured information is extracted from several web sources through HTTP GET method using web API. Moreover, our approach determined almost all single type infobox template keyword, which indicates the robustness of the template identification method. Therefore, the proposed approach might be very effective and efficient for the volunteers or users of Wikipedia for generating infobox not only for movie related article but also for other articles following the system process. In future, we would like to integrate more methodology to the existing proposed methods to obtain the largest coverage of Wikipedia articles effectively and efficiently. In addition, it is also planned to generate infobox from template where information will be extracted from the web without specifying any web sources.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2015-R1A2A2A03006190)

Reference

[1] Wikipedia. htttp://www.wikipedia.org

[2] DBpedia. http://wiki.dbpedia.org

[3] Wordnet. http//wordnet.princeton.edu

[4] D. Milne, and I. H Witten, "An open-source toolkit for mining Wikipedia," Artificial Intelligence, vol. 194, pp. 222-239, 2013. https://doi.org/10.1016/j.artint.2012.06.007

[5] D. Milne, O. Medelyan, and I. H Witten, "Mining domain-specific thesauri from wikipedia: A case study," In Proceedings of the 2006 IEEE/WIC/ACM international conference on web intelligence, IEEE Computer Society, pp. 442-448, Dec. 2006. https://doi.org/10.1109/WI.2006.119

[6] C. Elkan and R. Greiner, "Building large Knowledge- based systems representation and inference in the Cyc project," Artificial Intelligence, vol. 61, no. 1, pp. 41-52, 2006.

[7] H. Nguyen, T. Nguyen, H. Nguyen, and J. Freire, "Querying Wikipedia documents and relationships," In Proceedings of the 13th International Workshop on the Web and Databases, ACM, p. 4, June 2010. https://doi.org/10.1145/1859127.1859133

[8] H. Mousavi, D. Kerr, M. Iseli, and C. Zaniolo, "Deducing infoboxes from unstructured text in wikipedia pages," CSD Technical Report# 130001, UCLA, pp. 1-13, 2013.

[9] K. Bollacker, C. Evans, P. Paritosh, T. Sturge and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, ACM, pp. 1247-1250, June. 2008. https://doi.org/10.1145/1376616.1376746

[10] Morsey, M. Lehmann, J. Auer, S. Stadler, C. Hellmann, and S. Hellmann, "Dbpedia and the live extraction of structured data from Wikipedia," program electronic library and information systems, vol. 46, no. 2, pp. 157-181, 2012. http://doi.org/10.1108/00330331211221828

[11] R. Yus, V, Mulwad, T. Finin and E. Mena, "Infoboxer: Using Statistical and Semantic Knowledge to Help Create Wikipedia Infoboxes," In Proceeding of ISWC-PD'14 Proceedings of the 2014 International Conference, vol. 1272, pp. 405-408, 2014.

[12] F. Wu and D. S. Weld, "Automatically refining the wikipedia infobox ontology," InProceedings of the 17th international conference on World Wide Web, ACM, pp. 635-644, April 2008. https://doi.org/10.1145/1367497.1367583

[13] F. Wu and D.S. Weld, "Autonomously semantifying Wikipedia," In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, pp. 41-50, Nov. 2007. https://doi.org/10.1145/1321440.1321449

[14] K. Zhang, Y. Xiao, H. Tong, H. Wang, and W. Wang, "The links have it: Infobox generation by summarization over linked entities," arXiv preprint arXiv: 1406.6449, 2014.

[15] A. Sultana, Q. M. Hasan, A. K. Biswas, S. Das, H. Rahman, C. Ding, and C. Li, "Infobox suggestion for Wikipedia entities," In Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 2307-2310, Oct. 2012. https://doi.org/10.1145/2396761.2398627

[16] A.M. Azmi and S. Al-Thanyyan, "A text summarizer for Arabic," Computer Speech & Language, vol. 26, no. 4, pp. 260-273, 2012. https://doi.org/10.1016/j.csl.2012.01.002

[17] T. Pederson, S. Patwardhan and J. Michelizzi, "WordNet:: Similarity: measuring the relatedness of concepts," In Demonstration papers at HLT-NAACL, Association for Computational Linguistics, pp. 38-41, May 2004.

[18] O. Medelyan, D. Milne, C. Legg and I. H. Witten, "Mining meaning from Wikipedia," International Journal of Human-Computer Studies, vol. 67, no. 9, pp. 716-754, 2009. https://doi.org/10.1016/j.ijhcs.2009.05.004

[19] W. Dakka and S. Cuecerzen, "Augmenting Wikipedia with Named Entity Tags," In IJCNLP, pp. 545-552, Jan. 2008.

[20] J. Nothman, J. R. Curran and T. Murphy, "Transforming Wikipedia into named entity training data," In Proceedings of the Australian Language Technology Workshop, pp. 124-132, Dec. 2008.

[21] P. A. Ly, C. Pedrinaci and J. Domingue. "Automated information extraction from Web APIs documentation," In Web Information Systems Engineering-WISE, Springer Berlin Heidelberg, pp. 497-511, 2012. http:// doi.org/10.1007/978-3-642-35063-4_36

[22] L. Faria, A. Akbik, B. Sierman, M. Ras, M. Ferreira, and J. C. Ramalho, "Automatic preservation watch using

information extraction on the Web," In Proceedings of the 10th International Conference on Preservation of Digital Objects (iPRES). Lisbon, 2013.

◐ 저 자 소 개 ◑

Hanif Bhuiyan

2009 B.Sc. in CSE, Ahsanullah University of Science & Technology, Bangladesh.
2014 ~ 2016: M.E. in Computer and Information Engineering, Dept. of CSIE, Inha Univ., Korea.
2016 ~ present: Lecturer, Dept. of CSE, University of Asia Pacific, Dhaka, Bangladesh.
Research Interest: Augmented Reality, Semantic Web, Natural Language Processing and Data Mining
E-mail: hanifbhuiyan.c@gmail.com

Kyeong-Jin Oh

2006 B.S. in CSIE, Inha Univ., Korea.
2008 M.S. in Information Engineering, Inha University, Korea.
2016 Ph.D. in Information Engineering, Inha University, Korea.
Research Interest: Data Mining, Recommendation System, Semantic Web, Ontology and Reasoning
E-mail: okjkillo@gmail.com

Myung-Duk Hong

2011 M.S. in Information Engineering, Inha Univ., Korea
2012~present: Ph.D. Candidate, Dept. of CSIE, Inha Univ., Korea.
Research Interest: Data Mining, Vehicle Routing Problem, and Recommendation System.
E-mail: hmdgo@eslab.inha.ac.kr

Geun-Sik Jo

1982 B.S. in Computer Science, Inha Univ., Korea.
1985 M.S. in Computer Science, City Univ., New York, U.S.A.
1991 Ph.D. in Computer Science, City Univ., New York, U.S.A.
1997~present: Professor, Dept. of Computer Science and Information Engineering, Inha Univ., Korea
Research Interest: knowledge-based scheduling, semantic Web, intelligent E-Commerce, constraint-directed scheduling, knowledge-based systems, decision support systems, and intelligent agents.
E-mail: gsjo@inha.ac.kr