

AN AUTOMATIC APPROACH FOR GENERATING TABLES IN SEMANTIC WIKIS

¹LULUAH AL-HUSAIN,² SAMIR EL-MASRI

¹Lecturer, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

²Assoc. Prof., College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

E-mail: ¹ lalusain@ksu.edu.sa, ² selmasri@ksu.edu.sa

ABSTRACT

Wiki is well-known content management systems. Semantic wikis extends the classical wikis with semantic annotations that made its contents more structured. Tabular representations of information have a considerable value, especially in wikis which are rich in content and contain large amount of information. For this reason, we propose an approach for automatically generating tables for representing the semantic data contained in wiki articles. The proposed approach composed of three steps (1) extract the semantic data of Typed Links and Attributes from the wiki articles and call them Article Properties (2) cluster the collection of wiki articles based on extracted properties from the first step, and (3) construct the table that aggregates the shared properties between articles and present them in two-dimensions. The proposed approach is based on a simple heuristic which is the number of properties that are shared between wiki articles.

Keywords: *Clustering, Semantic wikis, Resemblance Metric Wiki*

1 INTRODUCTION

Wiki is a well-known Web 2.0 content management platform[1]. A Wiki is a collection of web pages where anyone can create, edit and publish contents[2]. The recent advance of semantic wikis enriches the classical wikis by adding structured semantic annotations within plain wiki text[1]. Thus, it makes the wiki content organized and structured and supports better search and browsing facilities of wikis [3, 4]. Semantic Wiki combines classical wiki with semantic web technology by adding special semantic features like data schemes, categorization, and semantic annotations[4]. These new features improve the content organization in the wiki without affecting its main features of openness and flexibility[3].

In wiki, pages are edited using a simple markup language (the wiki-text). Wiki pages are also called articles. Each article has distinct name and mainly contains text and hypertexts (Links). Hypertext used to represent a link to other pages within or outside the wiki. In other hand, a wiki page in semantic wiki adds annotation to both text and hypertexts. In semantic wiki, a hypertext is annotated with semantic labels, which then called Typed Links. In addition, annotation can be used to

annotate specific attributes inside the text content of a wiki page such as date, number, entity...etc. In addition, semantic wiki supports queries using some simple query languages that can be used to access the structured data annotated on wiki pages[5].

Generally, we tried to find new ways to utilize the semantic data of the semantic wikis and provide structured and organized content for wiki users. Tables are considered a powerful and an efficient tool for organizing data. They are used to describe the structuring of data into two-dimensions. Moreover, tabular representations of information are useful and have a considerable value for reader and browsers. Since a wiki contains rich web content and large amount of information. Additionally, the semantic wiki enriches wiki data with semantic annotations that made this content more structured to support efficient retrieving and searching of the wiki contents. Based on these considerations, we propose to use semantic wikis' features to aggregate similar articles' structured data and present them in tabular format. As a result, this will improve the reading and browsing experiences of wiki users. To achieve this, we proposed an approach that automatically generates tables that summarize the content of wiki articles. Our approach is based on structured semantic data



of wiki articles. Semantic wiki article contains annotated semantic data including typed link and attribute. In our approach, we used typed link and attribute as features of wiki article and we called them Article Properties or Properties.

In this paper, we propose an approach based on semantic features of Wikis to automatically identify similar article and generate a table that aggregates the information contained in those articles. This approach is motivated by providing the wiki users structured information, which combines multiple pages contents, in friendly tabular representation. Especially, our contributions include:

- Method for extracting properties from wiki-text of wiki article.
- Algorithm for clustering a collection of wiki articles based on their semantic properties.
- Method for generating table that contains values of properties of articles that belong to one cluster.

This paper is organized as follows. First, in section 1, we provide a generic description about wiki, semantic wiki, and Semantic MediaWiki which is popular semantic wiki platform. Second, in section 2, we will detail steps of our approach and discuss how we extract semantic data, cluster wiki articles and generate wiki table. Then, in section 3, we illustrate the proposed approach with an example. Finally, we will conclude the paper and provide an overview of future work.

2 LITERATURE REVIEW

2.1 Wiki

A wiki is “web-based software that allows all viewers of a page to change the content by editing the page online in a browser”[2]. A wiki is “a Web application that manages a collection of Web pages, where anyone can create new pages and edit existing pages by using a simplified markup language”[6]. A wiki engine is “a software system that powers the wiki site and makes it work”[6]. A wiki enables collaborative editing of Web pages[7]. WikiWikiWeb is the first wiki which was developed in 1995 [2]. Wikis are utilized as primary tools for content management and knowledge sharing. Wikis have become popular to a large number of audiences due to the success of Wikipedia, the free online encyclopedia.

The main advantage of wikis is that they provide a simple and fast way to add and edit content. Basically, wiki provides a Web-based text editor with a simple markup language that used in creating content and links between pages, in addition to, versioning management that keeps track of the content changes and search capabilities for querying the wiki pages[7].

Wiki technology is based on a simple and easy-to-use platform. Any wiki page mainly contains two elements: texts and hypertexts. Hypertext can be used to link pages within a wiki. Text can be edited and formatted on the fly by using simple rules. Moreover, the user can navigate, read, or edit the content within wikis using any internet browsers; Wiki does not require any additional software or plugins.

Technically, wiki-Software is a script that is installed on a server and allows users to freely create and edit web page content using any web browser[2]. Hence, wiki is a collection of web pages called wiki pages or articles that are accessed via a browser. The content of the wiki page is written as text using a wiki-text and it is stored in a file or database[7]. When a wiki page is accessed, the wiki script translates the wiki data retrieved from database into HTML that is integrated in the layout template and embeds it in the web page. In a wiki, every wiki page must have distinct name that usually indicates the subject of the page.

2.1.1 Wiki Contents

Users can enter information into a wiki page using wiki-text, which is a simple markup language that provides the standard for formatting and structuring the content[8]. Once the page is saved, the wiki text is transformed to XHTML pages for reading. Wiki’s markup language provides many facilities for describing formatting, and for structuring content[9]. For example, the markup elements “ **. . .** ” is used when text should appear bold-faced.

The most important feature of wiki-text is defining hyperlinks to specify the interrelation of pages within a wiki. In wikis, there are three levels of links in a wiki: “interwiki” links within the wiki, internal links within a page, or external links[10]. Interwiki links is a simplest form that used to link to pages within a wiki. It is formed by putting the title of page in double brackets *[[page title]]*. Of



course, title pages are always distinct in a wiki and there can never be two pages with the same name.

2.2 Semantic Wiki

Semantic wikis are classical wiki extended with the features of semantic technologies. Wiki suffers from unstructured data which provides insufficient retrieving and searching of information [7]. Semantic wiki tries to address the structured data. Therefore, wikis are enriched with semantic data to structure the information of the wiki. Adding semantics into the wiki pages made the information on these pages structured enough for machine to process which will enhance retrieval, search, and navigation functionalities of wiki and enables knowledge exchange between applications. Moreover, knowledge discovery becomes easier for the wiki users. Basically all classical wikis provide simple annotation support in the form of tagging or categorization [11]. Semantic wikis extend these capabilities by adding semantic annotations or mark up to wiki pages and to the link relations between wiki pages. The semantic annotations are usually represented in RDF Schema and OWL (Web Ontology Language). Once the semantic annotations are created, they are then available for extended queries [7].

Different semantic-wiki systems are proposed that extend wikis with semantic web technologies. Platypus Wiki [12] uses RDF and OWL to describe the text through simple user interface. Semantic MediaWiki [9] extends MediaWiki with semantic links which encode relations between wiki pages. In [13], they discussed how to create a structured knowledge using semantic annotations, guided input and a predefined schema. WikSAR [14] and SemWiki [15] enrich the normal wiki text with semantic contents. In KiWi [16], content item is associated with meta-data in the form of RDF. In [4], they develop a social bookmarking extension for the wiki to support personalized access to wiki page by using both free tagging and structured tagging. In [17], a structured and context-aware Wiki system is developed based on semantic tag based categorization approach.

2.2.1 Semantic Mediwiki

Semantic MediaWiki (SMW) is an extension of the popular wiki software MediaWiki that is used by Wikipedia and many other sites[11]. Semantic MediaWiki enables wiki-users to semantically annotate wiki pages, based on which

the wiki contents can be browsed, searched, and reused in novel ways [18].

In SMW, users can add semantic annotation to the wiki-text via special wiki-markup rules. SMW annotates categories, relations, and attributes properties of the wiki page. Category annotation is shown at the bottom of a page, relations are displayed like normal links, and attributes just show the given value. All extracted annotations are viewed in a factbox at the bottom of each page outside the main text. Each category, relation, and attribute is represented as an article. Also, all available types of attributes must have dedicated articles.

2.2.2 Semantic Annotation

Semantic wiki uses the same format as wikis for annotation. It modifies the hyperlinks in the wiki-text to provide semantic data. In semantic wiki, the hyperlinks are annotated with properties and values. There are two new features: typed links and attributes[8] [9].

- **Typed link:** The typed link used to represent the relation between articles. For example, [[England]] links the London page to England page and London is the capital of England. With typed link, the link can be extended to by writing [[capital of::England]]. This means that “London” has a property called “capital of” with the value “England.”. This annotation first creates the ‘capital of’ property and applies it to the London wiki page. It also creates a separate wiki page for it titled ‘Property: capital of’ that may contain textual description of this property. It also lists all the wiki pages that use this particular property.
- **Attributes:** the attribute used to define a relationship between the article and a data value that represents semantic data about the subject of the article. For example, if London has population of 7,421,328, we can write [[Population::7,421,328]] within the wiki-text of London wiki page. Here, Population represents an attribute, and 7,421,328 is the value of the attribute and it is not a page. There are many types of values that can assigned to attributes such as numeric, string, date, and geographic coordinates. To assign a type to an attribute, built-in property ‘has type’ is stated in the attribute’s property page. To specify that the Population property can

take only numeric value, we add `[[has type::Number]]` in Population page. Figure 1 states an example of wiki-text with and without semantic annotations.

```

'''London''' is the capital city of [[England]] and of the [[United Kingdom]]. As of
[[2005]], the population of London was estimated 7,421,328. Greater London covers an
area of 609 square miles.
[[Category:City]]

'''London''' is the capital city of [[capital of::England]] and of the [[capital of::United
Kingdom]]. As of [[2005]], the population of London was estimated [[population::7,421,328]].
Greater London covers an area of [[area::609 square miles]].
[[Category:City]]

```

Figure 1 Sources code of London page with (top) and without (bottom) semantic annotations[9]

2.2.3 Semantic Data Browsing

In SMW, the semantic data can be browsed and queried in a simple way. Users can browse the semantic data for each page at its factBox.

Factbox, which is placed at the bottom of the wiki page, summarizes the semantic data in the wiki pages. The factbox displays the property and corresponding attribute, links to related functions and also an alert feedback in case of errors. This factbox can be used to browse the wiki based on semantic content. The page title in the factbox lists the annotations within a given page and also all annotations where the given page is used as a value. The magnifying glass icon next to a value returns all other pages with similar annotations. In addition, the Factbox shows links to property pages, which in turn list all annotations for a given property (See Figure 2).

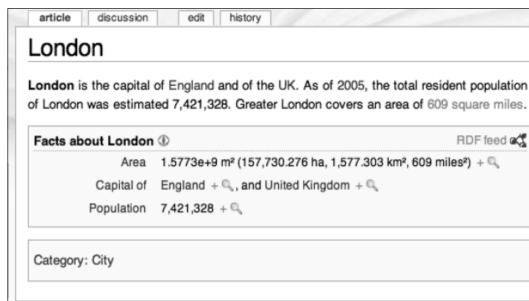


Figure 2 Wiki Page of London[9]

2.2.4 Semantic Data Querying

SMW has a simple query language which is based on the wiki's semantic data. The syntax of SMW's query language is closely related to wiki text. The query language can be used in two ways: directly query the wiki, or inline query. Inline

query can be used to embed dynamic content into pages[9].

The query language in semantic wiki is based in two constructs of semantic data contained in the wiki content: category and property[6]. Therefore, a query can be constructed by having constraints on either one or both of these constructs. For instance, this query `[[Category:Category_name]]` is a single constraint on category, and it returns every page that has Category_name as its Category. While this query `[[Category:Category_name]] [[Property::property_value]]` has two constraints connected by a logical AND relationship, one constraint on the category and one constraint on the property. Therefore, this query returns pages that have Category_name as its Category, and the annotation Property with property_value as its value. Moreover, a single query can have more property constraints.

As we note that if a given query involves more than one constraint on properties or categories, these constraints are treated as having a logical AND relationship. Therefore, it returns pages that satisfy all the constraints at the same time. In addition to, we can use OR operator between constraints to express a logical OR relationship in case we need to return pages that satisfy some of constraints. Also, the `||` operator is used to express a logical OR relationship among values, pages, and category names.

The query result is as a set of pages, and each page in the result set must satisfy the query conditions or constraints. The query result is displayed as titles of the result pages. Then, the user can click a page to see more information on the page, such as the page's property values or category values. However, we can use the printout statements to show the property and category values that we are interested in. The printout statements is written after the query constraints and start with a question mark `?` then the name of the property. As a result, the query result will be shown in a table column that is labeled by the name of the property specified in printout statement and filled out by the value of property for each page contained in the query result set. An empty field will be printed if printout statement has no value in some pages.

3 THE PROPOSED APPROACH

The main purpose of our approach is to automatically aggregate information from similar wiki articles and presents them in tabular format. For instance, the approach can automatically explore the wiki and extract the semantically annotated information from similar articles and provide them in tabular format without any efforts from users or technical persons. The proposed approach has three main steps. First, we extract the semantic data from wiki articles of Article

Properties that are composed of Typed Links and Attributes. Second, we find similar articles from collection of wiki articles based on extracted properties from the first step. Third, from each group, we can generate a table that aggregates the shared properties between articles. To accomplish this, the collections of wiki articles are clustered based on a simple heuristic that based on the number of properties (typed link and attributes) that are shared between wiki articles. Figure 3 provides an overview of the proposed approach.

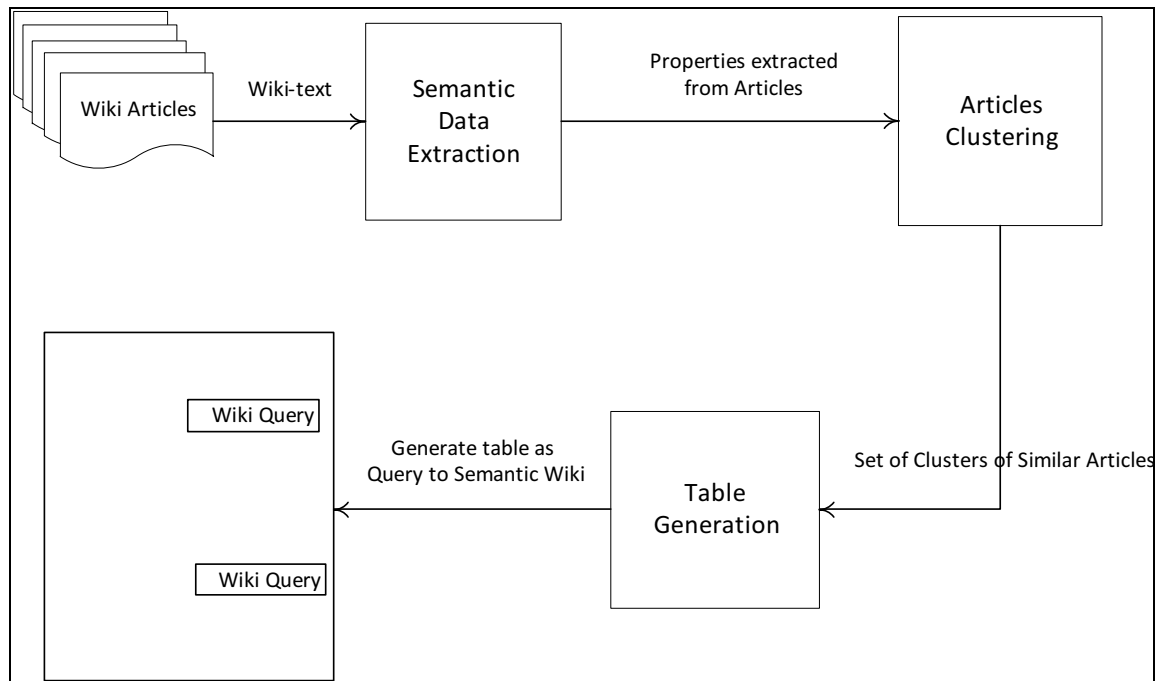


Figure 3 An overview of the Proposed Approach

3.1 Semantic Data Extraction

First step of our approach is semantic data extraction. We extract annotated semantic data from a wiki article. Two types of semantic data are extracted: typed links and attributes. They have the same syntax `[[property_name: value]]` which are inserted within wiki-text of articles. In our approach, we call typed links and attributes Properties of wiki articles. We needed an extraction method that parses the wiki-text in order to get the properties of wiki article. We cannot reuse an existing wiki-text parser because all existing parsers convert wiki-text directly to HTML[19]. So, we propose a simple extraction method that uses regular expressions for pattern matching of string to extract properties. This means the required

properties can be extracted from the wiki article independently of the rest of the article contents. This can be easily implemented using Java's Pattern Matching methods.

To extract semantic data from wiki-text, we need first to retrieve the wiki-text of the article rather than the HTML. For this, we can use "Special:Export" feature which can be used to get a list of all articles within a category. Articles are returned as wiki-text within an XML container. Another way that can be used to retrieve individual articles in its original wiki-text form by appending "?&action=raw" to the URL of any article.

3.2 Articles Clustering

The second step of our approach is articles clustering. Clustering is a natural way of semantically organizing information and abstracting important attributes of a collection of entities[20]. Clusters of wiki articles can provide more complete information about similar entities especially when this information exists on several pages. The intent here is to group wiki articles into meaningful clusters based on semantic annotation that can help to better organizing their contents in tabular format.

In this step, we aim to cluster collection of wiki articles into groups. The clustering is based on a simple heuristic that based on the number of properties (typed link and attributes) that are shared between different articles. For this purpose, we need similarity metrics which can measure the extent of relatedness between two or more wiki articles. Therefore, we use two generic similarity metrics [20, 21]: containment and resemblance. These two metrics can be used to measure the similarity of wiki articles based on articles' properties. In semantic wiki context, containment can be used to define article similarity as the amount of properties of the first article that are also contained in the second article and resemblance is defined to measure how many of all properties used in at least one of the two articles are used in both articles.

Given a pair of articles A and B and the set of properties of articles A and B (denoted $P(A)$, $P(B)$ respectively), the resemblance denoted ($r(A, B)$) and containment of A in B (denoted $c(A, B)$) are as follows:

$$r(A, B) = \frac{|P(A) \cap P(B)|}{|P(A) \cup P(B)|}$$

$$c(A, B) = \frac{|P(A) \cap P(B)|}{P(A)}$$

Conceptually, we can use resemblance to cluster articles of semantic wiki by first setting up a resemblance threshold. Given that extracted semantic properties for each article, we compute the resemblance for each pair of article and check if it exceed the threshold. If resemblance exceeds the threshold, the pair of articles is similar and would be combined in one cluster.

3.2.1 The clustering algorithm

We propose a clustering algorithm to cluster collections of wiki articles based on semantic annotated Properties. The algorithm is derived from technique proposed in[21] and uses resemblance metric. The algorithm is used to cluster web documents based on similar syntax. Broder et al. [21]claimed that their algorithm can be used to measure the similarity of any two objects and cluster the sets of similar objects from a large number of objects based on a set of features that are extracted from an object previously. We use their algorithm after making some changes to adapt to our domain requirements. Therefore, our clustering algorithm of wiki articles based on similar properties is performed in four steps:

Step 1: for every article, generate a sketch which composed of the properties contained in the article.

Step 2: produce a list of all the properties and the articles they appear in, sorted by property value. To do this, the sketch for each article is expanded into a list of <property_name, article_ID> pairs.

Step 3: generate a list of all pairs of articles that share any properties, along with the number of properties they have in common. To do this, we take the file of sorted <property_name, article_ID> pairs and expand it into a list of < articleA_ID, articleB_ID, count of common properties> triplets.

Step 4: produce the complete clustering by examining each < articleA_ID, articleB_ID, count of common properties> triplet and deciding if the article pair exceeds the threshold for resemblance. If it does, a link is added between the two articles in a union-find algorithm. The union-find algorithm forms the final clusters by finding connected components in a graph where edges denote the resemblance between article pairs.

The pseudo code of the clustering algorithm is shown in Figure 4.

Pseudo Code

- Set Resemblance Threshold (RT)
- For each article,
 - generate sketch that contain a list of <property_name>
 - Produce and Sort pairs of < property_name, article_ID>
- Compute list of article pairs with the number of shared properties as < articleA_ID, articleB_ID, count of common properties> triplets
- Generate clusters as follow:
 - For each triplet,
 - if $r(A,B) > RT$, then add link between A and B
 - Produce connected components using union-find algorithm

Figure 4 Pseudo Code of Clustering Algorithm

3.3 Table Generation

The third and final step of our approach is table generation. In this step, we generate a table, for each cluster, which composed of the properties and their values that are extracted from wiki articles and presents them in tabular format. To do this, first, we have to determine the most important properties that are shared by most of wiki articles in the same cluster. This can be done by collecting all properties of articles from one cluster and, then for each property; we calculate the occurrence frequency of property in the set of articles of that cluster. Therefore, the table will contain the properties that appear most frequently rather than others. However, we can set a threshold for occurrence frequency which can help in determining the most useful properties. Using threshold, occurrence frequency for a property can be defined by counting the number of occurrence of a property in the cluster divided by total number of articles of that cluster. This process is illustrated as pseudo code in Figure 5.

Pseudo Code

- Set the frequency threshold (FT)
- Get the articles of a given cluster
- For each article,
 - Get its properties
 - For each property, count the number of occurrence
 - Occurrence_frequency= number of occurrences/total number of articles
 - If Occurrence_frequency > FT then
 - Add the property to the set of table properties
- GenerateTable(table_properties)

Figure 5 Pseudo Code for Constructing Wiki Table

After determining the table properties, the table then can be generated using inline query. Query result is defined as a set of pages, and each page in the result set must satisfy the query conditions [6]. To show the properties and their values from pages contained in the query result, we use Printout Statements which must start with a question mark? followed by the property name. The result of query is shown in a table and each property is represented as column that is labeled by the name of the property. If some property has no value for any page contained in the query result, an empty field will be printed.

In our approach, we can generate the table using the query where the query condition is OR-ing of articles' titles in same cluster as [[Title::title_name]] ,followed by Printout Statements for table-properties that are determined before. The general query syntax is shown below (Figure 6).

Query

```
[[TITLE::Article_name1]] OR
[[TITLE::Article_name2]] OR...
?Property_name1
?Property_name2
?Property_name3
..
?Property_namen
```

Figure 6 A Query Syntax for Table Generation

4 DISCUSSION

In this section, we illustrate our approach with an example. Given a semantic wiki specialized in Germany country, it contains a lot of articles about its cities, people, places ... etc. As a prerequisite of our approach, those articles have texts and hypertexts that are semantically annotated. Applying our approach, these articles first are processed to extract their properties of Typed Link and Attributes (discussed in 0). Then, the clustering algorithm is executed to group together similar wiki articles based on extracted properties. Lastly, we can generate a query to generate a table of propertied of the articles for each cluster.

Figure7 represents the table generated for Cities cluster. It aggregates the semantic properties of 14 articles. The displayed properties are state, population, and Area with their values.

German cities			
State	Population	Area	
Berlin	3,391,407	891.69 km ²	
Celle	72,000	176.01 km ²	
Dresden	508,351	328.8 km ²	
Freiburg	217,547	153.07 km ²	
Halle (Saale)	240,000	135 km ²	
Hamburg	1,769,117	755 km ²	
Hannover	522,944	204.01 km ²	
Karlsruhe	285,812	173.46 km ²	
Koblenz	105,888	105.02 km ²	
Leipzig	510,274	297.6 km ²	
Marburg	79,139	124.5 km ²	
Munich	1,300,000	310.799 km ²	
Wiesbaden	300,427	204.1 km ²	
Worms	85,829	108.73 km ²	

Figure 7 Illustrated Example

5 CONCLUSION AND FUTURE WORK

The proposed approach aims to aggregate the semantic data of similar articles and presents them to the wiki users in tabular format. The approach first applies pattern matching expressions to extract Properties of articles from wiki-text. Then, clustering algorithm is applied to cluster a collection of wiki articles based on resemblance metric which measures the similarity of articles based on semantic properties. After that, the approach determines the table structure and content based on Occurrence Frequency measure and generates the table using semantic wiki's query.

As a possible future work, we intend to implement our approach and evaluate it to ensure that the approach is effective in aggregating the content of semantic wikis. That will depend on the accuracy of the properties extraction and clustering. In addition, we intend to examine its effect in terms of the time required to generate tables and the quality of those tables. Finally, the clustering algorithm would be extended to consider the value of property, in addition to, the property as a metric of similarity.

REFERENCES

- [1] J. Bao, et al., "A Semantic Wiki Based Light-Weight Web Application Model," in *The Semantic Web*. vol. 5926, A. Gómez-Pérez, et al., Eds., ed: Springer Berlin / Heidelberg, 2009, pp. 168-183.
- [2] A. Ebersbach, et al., "The Wiki Concept," in *Wiki*, ed: Springer Berlin Heidelberg, 2008, pp. 9-30.
- [3] J. Hoffart, et al., "An architecture to support intelligent user interfaces for Wikis by means of Natural Language Processing," presented at the Proceedings of the 5th International Symposium on Wikis and Open Collaboration, Orlando, Florida, 2009.
- [4] A. Singh, et al., "Personalized Information Access in a Wiki Using Structured Tagging," in *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops*. vol. 4805, R. Meersman, et al., Eds., ed: Springer Berlin / Heidelberg, 2007, pp. 427-436.
- [5] A. Bernier, et al., "Wiki, a new way to produce accessible documents," presented at the Proceedings of the 12th international conference on Computers helping people with special needs: Part I, Vienna, Austria, 2010.
- [6] L. Yu, "Semantic Wiki," in *A Developer's Guide to the Semantic Web*, ed: Springer Berlin Heidelberg, 2011, pp. 331-377.
- [7] S. Schaffert, et al., "Semantic Wikis," *Software, IEEE*, vol. 25, pp. 8-11, 2008.
- [8] J. Wesley, et al., "Semantic Wikis: Are They for You?," *Infosys Technologies*, vol. 7, 2009.
- [9] M. Krötzsch, et al., "Semantic MediaWiki," in *The Semantic Web - ISWC 2006*. vol. 4273, I. Cruz, et al., Eds., ed: Springer Berlin / Heidelberg, 2006, pp. 935-942.
- [10] A. Ebersbach, et al., "Multiple-page Structures," in *Wiki*, ed: Springer Berlin Heidelberg, 2008, pp. 81-105.
- [11] M. Krötzsch, et al., "Reasoning in Semantic Wikis," in *Reasoning Web*. vol. 4636, G. Antoniou, et al., Eds., ed: Springer Berlin / Heidelberg, 2007, pp. 310-329.
- [12] R. Tazzoli, et al., "Towards a Semantic Wiki Wiki Web," in *3rd International Semantic Web Conference (ISWC2004)*, Hiroshima, Japan 2004.
- [13] D. Herzig and B. Ell, "Semantic MediaWiki in Operation: Experiences with Building a Semantic Portal," in *The Semantic Web - ISWC 2010*. vol. 6497, P. Patel-Schneider, et al., Eds., ed: Springer Berlin / Heidelberg, 2010, pp. 114-128.
- [14] D. Aumueller and S. Auer, "Towards a Semantic Wiki Experience - Desktop Integration and Interactivity in WikSAR.," in *Semantic Desktop Workshop 2005 at ISWC'05*, Galway, Ireland 2005.



- [15] M. Volkel and E. Oren, "Semantic Wikis for Personal Knowledge Management," 2006.
- [16] S. Schaffert, et al., "KiWi – A Platform for Semantic Social Software (Demonstration)," in *The Semantic Web: Research and Applications*. vol. 5554, L. Aroyo, et al., Eds., ed: Springer Berlin / Heidelberg, 2009, pp. 888-892.
- [17] C. Trattner, et al., "The Austrian way of Wiki(pedia)!: development of a structured Wiki-based encyclopedia within a local Austrian context," presented at the Proceedings of the 6th International Symposium on Wikis and Open Collaboration, Gdansk, Poland, 2010.
- [18] M. V., et al., "Semantic Wikipedia," presented at the Proceedings of the 15th international conference on World Wide Web, Edinburgh, Scotland, 2006.
- [19] J. I. a. M. H. Butler., "Extracting and re-using structured data from wikis.," Digital Media Systems Laboratory Technical Report HPL-2007-182, November 2007.
- [20] D. Dhyani, et al., "A survey of Web metrics," *ACM Comput. Surv.*, vol. 34, pp. 469-503, 2002.
- [21] A. Z. Broder, et al., "Syntactic clustering of the Web," *Computer Networks and ISDN Systems*, vol. 29, pp. 1157-1166, 1997.