Personalized Facets for Faceted Search Using Wikipedia Disambiguation and Social Network

Hong Son Nguyen, Hong Phuc Pham, Trong Hai Duong, Thi Phuong Trang Nguyen and Huynh Minh Triet Le

Abstract The main aim of this paper is to deal with semantic search based on personalized facets using Wikipedia disambiguation data which can help to solve lexical ambiguity. User profile is learned from his/her activities and preferences in Facebook social network. Faceted graph visualization for result collaborative filtering is proposed. The facets are vertices representing ontological concepts. Other vertices represent instances belonging to the concepts, which are known as facets values. The vertices are highlighted by matching with user profile using TF-IDF feature vector model in order to individually produce search interfaces. The ties between vertices are ontological relations or properties considering as variables/attributes of facets. An algorithm to construct the faceted graph visualization and collaboratively filter search result is also provided. The faceted search method presented here is implemented to demonstrate these ideas.

Keywords Faceted search · Personalization · Wikipedia disambiguation · Semantic search · Social network

H.S. Nguyen (✉)
Faculty of Information Technology, Le Quy Don University, Hanoi, Vietnam
e-mail: son_nguyenhong2002@yahoo.com

H.P. Pham
Nguyen Tat Thanh University, Ho Chi Minh City, Vietnam
e-mail: phphuc7989@gmail.com

T.H. Duong · H.M.T. Le
International University, Vietnam National University, Ho Chi Minh City, Vietnam
e-mail: haiduongtrang@gmail.com

H.M.T. Le
e-mail: trietle95@gmail.com

T.P.T. Nguyen
Banking University of Ho Chi Minh City, Ho Chi Minh City, Vietnam
e-mail: phuongtrangict@gmail.com

T.B. Nguyen et al. (eds.), *Advanced Computational Methods for Knowledge Engineering*, Advances in Intelligent Systems and Computing 453, DOI 10.1007/978-3-319-38884-7_17

1 Introduction

The brain of human being connects the knowledge into a huge network of ideas, memories, definitions, perceptions. We can fully understand the meaning of a word even though it has ambiguous meaning, we can understand the same thing based on various terms. It is easy for human being but it is a big problem for computer. The major problem is shortage of specification of semantic heterogeneousness and ambiguity. We aim to look the "apple fruit" by putting the word "apple" in today famous search engines including Google, Yahoo! and Bing. We could not find the page which mentions "apple fruit" easily; most of the top results are about the "Apple Inc.", an American multinational corporation. All Google, Yahoo! or Bing are search engine not knowledge engine. They are good at returning a small number of relevant documents from a tremendous source of webpages on the Internet; but they still experience the lexical ambiguity issue, the presence of two or more possible meanings within a single word. Another problem with today search engines is that their filtering for their search results is not enough. In the process of finding the word "apple", we have observed that three big search engine only provide one visible criterion which is time to refine the result. In summary, there are two problems in today search engines:

- Lexical ambiguity: the question is that can the search engine return the correct meaning of the word that we are looking for when limited information of search query is provided?
- Search results filtering: Can the search engine offer better search results filtering such as collaborative filtering or content-based filtering?

In recent years, we have witnessed the emergence of Human-Computer information retrieval program where user can interact with the program to bring more complex information-seeking tasks.[1] Facets play the major part of this program. It is a way of classifying information and it also helps to solve the weakness of earlier knowledge representations. The faceted classification has been developed by scientists to offer an approach of knowledge representation which are rich and practical. However, the Faceted classification is only a solution to knowledge representation. We also need a mean which help to utilize that information, that mean is called Faceted Search. Faceted Search is becoming more and more popular especially in online shopping sites and site search. However, the facet types (category, price, brand, etc.) and the possible values for each facet are usually manually defined for a specific e-commerce site. For general purpose retrieval, automatic facet and facet-value recommendations are needed [1].

Facebook is now the most famous social networking site. It contains an extensive data of each member. The availability and extent of the profile data depends on the user's attitude towards entering and making the information visible in his profile [2]. The following data from Facebook can be utilized as user preferences: age, gender, group, geography data, posts, comments and likes. In Facebook, when user clicks the

[1] http://www.alexa.com/siteinfo/wikipedia.org.

"Like" button on an object such as fan page. That object is stored as an item in user profile. And certainly, the first thing that comes to mind when thinking about user preferences is likes as they explicitly express affinity. The same is for groups where many people with same interest gather as a small community. The geography data can point out the location-related affinity. In general, with the help of social networks, the many online services have the opportunity to know the users and get close to them in order to provide more relevant results. The valuable insights from networking sites are useful in many areas such as searching, recommendation or advertising.

Wikipedia is a free access and free content internet encyclopedia. Wikipedia is ranked as among the most popular website and constitutes the Internet's largest and most popular general reference work. In Wikipedia, internet users can freely create or edit a Wikipedia page's content. This "freedom of contribution" has a positive impact on both the quantity (fast-growing number of articles) and the quality (potential mistakes are quickly corrected within the collaborative environment) of this online resource [1].

To address the mentioned problems, this article's approach is to build a smart search that will utilize wiki disambiguation data and the information of social networks from user, the search is supposed to return the most relevant results with collaborative filtering. This approach includes the following:

- Provide the search results related to user's intentions based on his social network data. In particular, the faceted search program will solve the lexical ambiguity problem in current search engines.
- Based on the search results, provide a filtering that assists user to choose the correct result.

2 Related Works

Dynamic queries defined as interactive user control of visual query parameters that generate a rapid animated visual display of database search results [2–14]. The authors emphasize the interface with outstanding speed and interactivities. Ahlberg and Shneiderman built Film Finder to explore the movie database [1]. The graphical design contains many interface elements; parametric search is also included in a faceted information space. However, the results of Film Finder returning to users are not proactive and users are still able to select unsatisfactory combination. Later on, Shneiderman and his colleagues addressed the above problem on query previews. Query Previews are prevent wasted steps by eliminating zero-hit queries. That is mean the parametric search is replace by faceted navigation. In general, it helps user to have an overview over the selected documents. The mSpace project [7] described as an interaction design to support user-determined adaptable content and describe three techniques, which supports the interaction: preview cues, dimensional sorting and spatial context. Parallax was developed by David Huynh, its interface provides a "set-based browsing", that extends faceted search to shift

views between related sets of entities. When user is browsing a set of results, Parallax provides the connections to related entities along with filter-base for current search result. Parallax is more like a semantic-web browser than a faceted search, because it supports more general ontology. Parallax made an important step that makes semantic web explore able, using many of the same techniques that have made faceted search successful [4]. In our previous work [15], we proposed an effective method to build a faceted search for unstructured documents that utilizes wiki disambiguation data to build the semantic search space; the search is to return the most relevant results with a collaborative filtering. The faceted search also can solve the lexical ambiguity problem in current search engines.

3 Personalized Facets for Faceted Search Using Wikipedia Disambiguation and Social Network

We present a proposed personalized facets methodology for faceted search using Wikipedia disambiguation and social network:

- Phase 1: Data Preparation: The data was not available for our experiment. In order to obtain the Wikipedia disambiguation data, we decided to download the Wikipedia dumps file which contain Wikipedia contents. And we will edit it to find a suitable data for our method.
- Phase 2: Prepare User Profile: We made user profile from user Facebook profile. It contains an extensive data of each member.
- Phase 3: Search visualization: we present a semantic search method using facets and user profile to automatically provide the most expected results in Wikipedia Disambiguation.

3.1 Phase 1: Data Preparation

The disambiguation pages [16] in Wikipedia are extracted from the main page file; each disambiguation entity contains list of all existing Wikipedia article of the give word; for example Java, an island, a programming language, an animal. Each meaning is categorized into facets/sub-facets.

In the Fig. 1, facets and sub-facets of Java disambiguation are:

- Facets: Places, Animal, Computing, Consumables, Fictional characters, Music, People, Transportation, Other uses.
- Sub-facets: Indonesia, United States, Other are sub-facets of facet Places.

Fig. 1 Sample facet structure of java

Contents [hide]

1 Places
  1.1 Indonesia
  1.2 United States
  1.3 Other
2 Animal
3 Computing
4 Consumables
5 Fictional characters
6 Music
7 People
8 Transportation
9 Other uses
10 See also

In order to obtain the Wikipedia disambiguation data, we decided to download the Wikipedia dumps file which contain Wikipedia contents and load them into a MySQL database server (see Fig. 2).

After importing Wikipedia, a Disambiguation extraction process is carried out. All the disambiguation pages are extracted out from the database based on template of Disambiguation; for every disambiguation entity, an algorithm is applied to get the feature vector of its documents. Each document also includes its facet information. The process is as below Fig. 3.

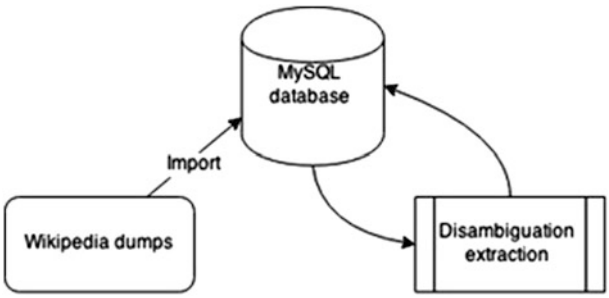

Fig. 2 Data preparation overview

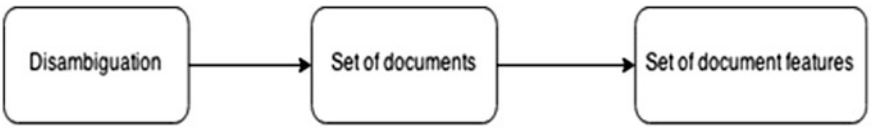

Fig. 3 Document features extraction process

The feature vectors of documents will also be used as search data.

```
Document features extraction algorithm
Input: Set of document, D
Output: Set of features, F
1. for each d ∈ D do
2.  Let f = GetFeaturealgorithm(d);
    /*converts document content into a feature vector */
3.  return f
4. return R_w

Get Feature algorithm
Input: A document, d
Output: A feature vector
1. Remove stop words from d
2. Convert d into set of terms, T
4. for each t ∈ T do
    Remove verbs, get noun phrase
5. return feature vector contains list of noun phrases
```

3.2 Phase 2: Prepare User Profile

We use the information from Facebook page include: Category, Name, Description. In our method, only the Likes of user are used as user preferences which includes all the pages which have been liked by user [17, 18]. For each page, the associated information Name, Category and Description are extracted to form user profile data, the same will be used for matching with search results. We use Facebook Spring Social tool to enables the connection between our program with Facebook's Graph API which helps to get data from Facebook.

3.3 Phase 3: Search Visualization

The sample of partial Apple disambiguation

Figure 4 shows an example of a surfing user browses web to get more information about Apple Inc. which produces the iPhone and iPad that he is using. As modern search engines are keyword-based, user may get the correct "apple" that he is looking for.

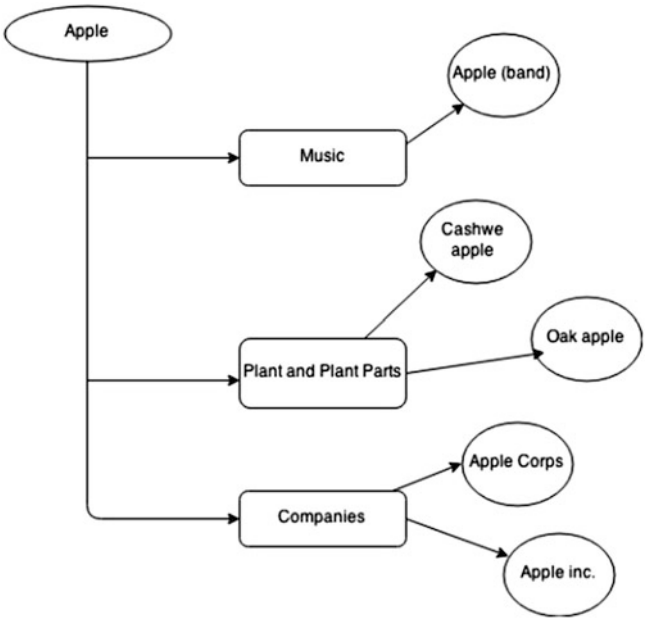

Fig. 4 Sample of partial Apple disambiguation

The search program is designed to solve the above user query. As query is entered, the program will return all the related result, the most relevant results are highlighted. Along with search result, the facet function is provided for result filtering, on selecting a facet, only its facet values are left.

Beside the search input, the user search graph interface includes two sections. The Facet graph which has a list of facets, the same is displayed in tree format gives user an overview about the result. Each facet has a list of available values associated with it. Every facet is displayed as a rectangle.

The second section Document Graph visualization is a graph to present results $\mathcal{R}(\mathrm{V_r}, \mathrm{E_r})$ on which each vertex $\mathrm{v} \in \mathrm{V_r}$ presented with different sizes. Here, we proposed the Algorithm 1 [10] to compute the size of vertices. Let $\mathcal{R}_w\left(V_r^{'}, E_r^{'}\right)$ be a new result after matching $\mathcal{R}$ with User Profile. Out of which, $V_r^{'} = \{(v_i, w_i, s_i) | v_i \in V_r\}$ with s_i is the size and w_i is the weight of vertex v_i. This algorithm uses two loops to compute the weight of each vertex w_i and determine the maximum weight w_{max}. Then, it computes the size of each vertex (s_i) by using following equation:

$$s_i = S_{min} + w_i \times \frac{S_{max} - S_{min}}{w_{max}} \tag{1}$$

where $S_{max} S_{min}$ stand for the constant of minimum and maximum size.

Computing the sizes algorithm
Input: $\mathcal{R}(V_r, E_r)$
Output: $\mathcal{R}_w(V'_r, E'_r)$
Initialize $\mathcal{R}_w(V'_r, E'_r)$ with $V'_r = \emptyset$ and $E'_r = E_r$
Initialize $w_{max} = 0$
1. for each $v \in V_r$ do
2. Let $v' = (\mathrm{v}, \mathrm{w} = 0, \mathrm{s} = 0)$
3. Let $weight = tfidf(v, P)$; // P is user profile feature vector
/*tfidf compute the $weight$ v on P */
4. if $weight > w_{max}$ then
5. $w_{max} = weight$
6. Adding v' into V'_r
7. $step = (S_{max} - S_{min})/w_{max}$
8. for each $v' \in V'_r$ do
9. $v'.s = S_{min} + v'.w * step$
10. return $\mathcal{R}_w$

4 Experiment

In this section we present experimental evaluations for our techniques. We first evaluate our disambiguation extraction process that we used to create data for our search program. Then, we evaluate the search results in term of efficiency, effectiveness and performance. Finally, we examine the algorithm which was used to compute the size of vertices. In order to implement the search program for the demonstration, we use Spring Tool Suite[2] as development tool.

4.1 Disambiguation Extraction Evaluation

There are 158613 disambiguation pages in total as of 11/2014. Each disambiguation page contains zero or more facets. The evaluation was carried out by comparing the raw data with extracted result. Two important aspects are considered during this duration: the first aspect is the number of extracted facets along with their structures in a one disambiguation entity so that our semantic search can benefit from this information, a list of facets is provided so that user can base on that to seek for correct documents; another thing is number of documents in one disambiguation

[2] https://spring.io/tools.

Table 1 Sample extracted data for places facet in Java disambiguation

Facet	Document
Java->Places->Indonesia	Java sea
Java->Places->Indonesia	Java trench
Java->Places->Other	Java (town)
Java->Places->Other	Java road
Java->Places->Other	Java, São Tomé and Príncipe
Java->Places->Other	Java district
Java->Places->Other	Java eiland
Java->Places->United States	Java, New York
Java->Places->United States	Java, Virginia
Java->Places->United States	Java, Ohio
Java->Places->United States	Java, South Dakota
Java->Places->United States	Coffee County, Alabama

entity, the information from these document are used for matching the query entered by user, the same are indexed to be used as the search data for the program. Below is a same extracted data for Places facet in Java (Table 1).

In Wikitext, raw content of Wikis page stored in database, the Facets are within double equal symbols "== ==", sub-facets are within triple equal symbols "=== ===", the documents start with asterisk symbols "*" and are within pairs of double square "[[]]" brackets. We based on those notations to pull out the data. Six sample disambiguation entities are taken for extraction "apple", "obama", "java", "joker", "iphone", "alien".

According to Table 2, the number of extracted facets are always the same as original one, however when extracting the documents from disambiguation page, the results are not completely correct, this is because the complexity of structure of page in raw format. In general, the result of extraction is quite accurate; it can retain the information from original disambiguation entity.

Table 2 Extracted facets/documents compared to original data

Disambiguation page	Extracted facets/original facets	Facets accuracy (%)	Extracted documents/original documents (%)	Documents accuracy (%)
Java	12/12	100	45/48	93.76
Apple	8/8	100	46/46	100
Obama	3/3	100	14/14	100
Joker	11/11	100	58/59	98.3
Iphone	0/0	100	14/14	100
Alien	6/6	100	39/39	100

4.2 Search Result Evaluation

In this section, we will evaluate our search program, we analyze the personalization aspect of the program, how it accommodates the differences between individuals. The accuracy is evaluated using precision and recall technique; finally we evaluate the satisfaction of user about our program.

4.2.1 Personalization Evaluation

To perform this evaluation, we created three test users; Facebook offers test user feature, a test user can experience the app as regular user but it is invisible to normal users, furthermore the app can be granted any permission from test user without the approval from Facebook. Our three test users have liked various pages in Facebook; Richard has interests in Java programming language, Apple Inc., and United States; for Tom, they are Indonesia, Java sea and Apple fruit. The last user, Patricia, has showed no interests on Facebook. In order to examize how the program reponses to different profiles, we use two queries: "java" and "apple" (Table 3).

Six tests were executed and the search program behaves differently for different profiles; in the search results, sizes of vertices for specific documents are varied for various users (Tom, Richard and Patricia).

4.2.2 Accuracy Evaluation

In the Fig. 5, the effectiveness of the returned results of our program is evaluated. In most of cases, all the relevant documents are retrieved but the accuracy is not very good, many times the irrelevant documents are shown to user.

4.2.3 Filtering Evaluation

Because of the problem when matching words together, there will be the cases that user is not able to get his interested results (Table 4); in these situation, user can refine search results by selecting facet or sub facets in the facet graph.

Table 3 Test users with different interests

User	Interests (Likes)
Richard	Java programming language, Apple Inc., United States
Tom	Indonesia, Java sea, Apple fruit
Patricia	None

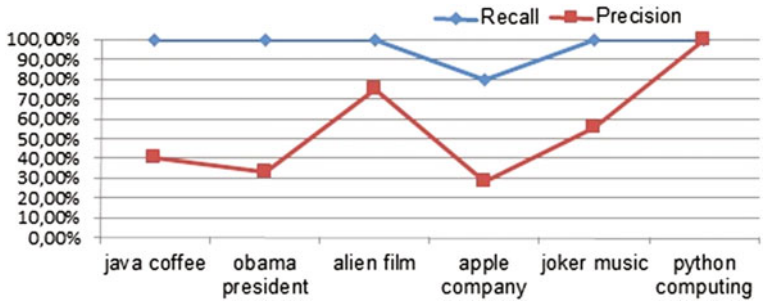

Fig. 5 Measure search result using precision and recall

Table 4 Search result summary

User	Query	Most highlighted vertices
Richard	Java	Java programming language, java software platform, java virtual machine
	Apple	Apple Inc., Apple II series, Apple store
Tom	Java	Coffee, java (an island), java (DC Comics)
	Apple	Apple (fruit), tomato, Apple Inc.
Patricia	Java	None (all vertices are at default size)
	Apple	None (all vertices are at default size)

In a common flow, user will be able to select the correct information after two steps in our search program; the first is to enter search query, next step is to refine the search result based on facet value on facet graph. This is considered very fast.

4.3 Vertices Size Effectiveness

To evaluate the algorithm that was used to compute the size of vertices, we compare the original graph (G1) and the matched graph (G2). In G1, let the size of each vertex be 75 px. And in G2, the size of each vertex is computed by using Eq. 1 with s_min = 50 px and s_max = 100 px.

The comparison was considered with tasks named apple. The detailed results of this comparison were presented in Fig. 1. To have more conviction, we expanded 10 other tasks. For each task, we compute the rate between the total size of G2 and G1:

$$r = \frac{\text{Total size of graph 2}}{\text{Total size of graph 1}} \times 100\,\%$$

A graph of rates is shown in Fig. 6.

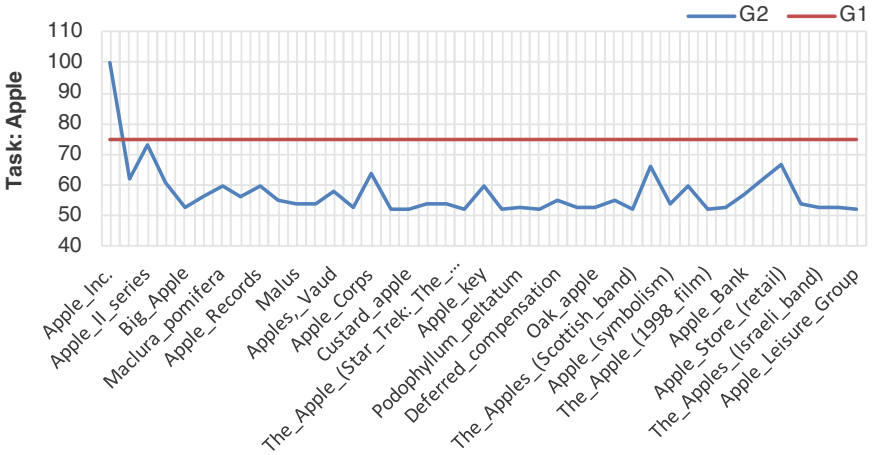

Fig. 6 Compare the size of G1 and G2

According to this results, we realize that G2 not only uses less resources (occupies less pixel on screen) than G1 does, but also facilitates exploring of the expected results.

5 Conclusions

We have presented a methodology regarding personalized facets for semantic search using two data sources, the first one is Wikipedia Disambiguation, a dataset that helps to solve lexical ambiguity; the other one is Facebook user profile. A complex process was carried out to make the data available for experiment from raw Wikis data. The graph with modern user interface was implemented to show individualized search results by matching facet values as documents with Facebook user profile.

In experiment, firstly we evaluate the effectiveness of our process to extract the disambiguation data from Wiki. The result shows that the process retains most of the information of disambiguation page from Wikipedia. Secondly the evaluations for effectiveness and efficiency our search program is conducted. The program not only helps to solve the ambiguity of words when searching but also help user to seek for the information quickly with few interactions.

References

1. Ahlberg, C., Shneiderman, B.: Visual information seeking: tight coupling of dynamic query filters with starfield displays. In: Adelson, B., Dumais, S., Olson, J. (eds.) Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'94), pp. 313–317. ACM, New York, NY, USA (1994)
2. Brunk, S., Heim, P.: tFacet: hierarchical faceted exploration of semantic data using well-known interaction concepts. In: Proceedings of DCI 2011. CEUR-WS.org, vol. 817, pp. 31–36 (2011)
3. Heim, P., Ertl, T., Ziegler, J.: Facet graphs: complex semantic querying made easy. In: Proceedings of the 7th Extended Semantic Web Conference (ESWC 2010). LNCS, vol. 6088, pp. 288–302. Springer, Berlin, Heidelberg (2010)
4. Heim, P., Ziegler, J., Lohmann, S.: gFacet: a browser for the web of data. In: Proceedings of the International Workshop on Interacting with Multimedia Content in the Social Semantic Web (IMC-SSW 2008). CEUR-WS, vol. 417, pp. 49–58 (2008)
5. Heim, P., Ziegler, J.: Faceted visual exploration of semantic data. In: Human Aspects of Visualization. Lecture Notes in Computer Science, vol. 6431, pp. 58–75. Springer, Berlin, Heidelberg (2011)
6. Koren, J., Zhang, Y., Liu, X.: Personalized interactive faceted search. In: Proceedings of the 17th International Conference on World Wide Web, pp. 477–486. ACM New York, NY, USA (2008)
7. Schraefel, M.C., Karam, M., Zhao, S.: mSpace: interaction design for user-determined, adaptable domain exploration in hypermedia. In: AH 2003: Workshop on Adaptive Hypermedia and Adaptive Web Based Systems, pp. 217–235. Nottingham, UK (2003)
8. Thi, A.D.H., Nguyen, T.B.: A semantic approach towards CWM-based ETL processes. In: Proceedings of I-SEMANTICS, vol. 8, pp. 58–66 (2008)
9. Tunkelang, D.: Faceted Search. Morgan & Claypool Publishers (2009)
10. Wagner, A., Ladwig, G., Tran, T.: Browsing-oriented semantic faceted search. In: DEXA'11 Proceedings of the 22nd International Conference on Database and Expert Systems Applications, vol. 1, pp. 303–319. Springer, Heidelberg (2011)
11. Hostetter, C.: Faceted searching with apache solar. In: ApacheCon, US (2006)
12. Nguyen, T.B., Schoepp, W., Wagner, F.: GAINS-BI: business intelligent approach for greenhouse gas and air pollution interactions and synergies information system. In: iiWAS 2008, pp. 332–338 (2008)
13. Nguyen, T.B., Wagner, F., Schoepp, W.: Cloud intelligent services for calculating emissions and costs of air pollutants and greenhouse gases. ACIIDS **2011**, 159–168 (2011)
14. Le, T.,Vo, B., Duong, T.H.: Personalized facets for semantic search using linked open data with social networks. In: IBICA 2012, pp. 312–317 (2012)
15. Dang, B.D., Nguyen, H.S., Nguyen, T.B., Duong, T.H.: A framework of faceted search for unstructured documents using wiki disambiguation. In: ICCCI, vol. 2, pp. 502–511 (2015)
16. Mihalcea, R.: Using wikipedia for automatic word sense disambiguation. In: Proceedings of NAACL HLT, pp. 196–203 (2007)
17. Duong, T.H., Uddin, M.N., Li, D., Jo, G.S.: A collaborative ontology-based user profiles system. In: Proceedings of ICCCI'09, Social Networks and Multi-agent Systems, pp. 540–552. Springer, Heidelberg (2009)
18. Duong, T.H., Mohammed, N.U., Nguyen, D.C.: Personalized semantic search using ODP: a study case in academic domain. ICCSA **2013**, 607–619 (2013)