International Journal of Electrical and Computer Engineering (IJECE)
Vol. 5, No. 6, December 2015, pp. 1536~1544
ISSN: 2088-8708

Graph-Based Concept Clustering for Web Search Results

Supakpong Jinarat*, Choochart Haruechaiyasak, Arnon Rungsawang***
* Department of Computer Engineering, Kasetsart University, Chatuchak, Bangkok, Thailand
** National Electronics and Computer Technology Center (NECTEC), Thailand

Article Info

Article history:

Received Jul 20, 2015
Revised Aug 27, 2015
Accepted Sep 14, 2015

Keyword:

Concept extraction
Graph-based clustering
Search results clustering
Wikipedia concept

ABSTRACT

A search engine usually returns a long list of web search results corresponding to a query from the user. Users must spend a lot of time for browsing and navigating the search results for the relevant results. Many research works applied the text clustering techniques, called web search results clustering, to handle the problem. Unfortunately, search result document returned from search engine is a very short text. It is difficult to cluster related documents into the same group because a short document has low informative content. In this paper, we proposed a method to cluster the web search results with high clustering quality using graph-based clustering with concept which extract from the external knowledge source. The main idea is to expand the original search results with some related concept terms. We applied the Wikipedia as the external knowledge source for concept extraction. We compared the clustering results of our proposed method with two well-known search results clustering techniques, Suffix Tree Clustering and Lingo. The experimental results showed that our proposed method significantly outperforms over the well-known clustering techniques.

Corresponding Author:

Arnon Rungsawang,
Department of Computer Engineering,
Kasetsart University,
50 Ngam Wong Wan Road, Ladyaow Chatuchak, Bangkok, 10900, Thailand.
Email: arnon@mikelab.net

1. INTRODUCTION

A search engine usually returns a large number of web search results corresponding to a query from the user. Consequently, users must spend a lot of time to browse and navigating the search results for the most relevant information. Many researchers used text clustering techniques, called web search results clustering, to handle the problem [1]-[6]. Text clustering is the method for grouping the related documents into the same cluster. Unfortunately, search result document returned from search engine is a very short text. It is difficult to cluster related documents into the same group because a short document has low informative content. We assume that the short document can be extended with some suitable terms to join the related document into the same cluster.

Many web search result clustering techniques used the "Bag of Words" model to convert a document into a term-vector and compute the similarity between 2 documents [2]. Some work used singular value decomposition model to generated cluster label first and then cluster the documents [4]. Some search result clustering techniques applied the suffix tree structure to maintain words or phrases [5]-[6]. The limitation of short document affect to these works encounters unsatisfactory clustering result.

Many researchers applied the external knowledge source to expand the information of short document. Some work applied web taxonomy of Open Directory Project (ODP) to generate the concepts for text enrichment [7]. The work of Di Marco and Naigli [8] applied the ODP to cluster the web search results by using techniques of Word Sense Induction (WSI). Limitation of using ODP as an external information source is too small content of each category, low informative content and lack of links to related categories.

Recently, many research works use Wikipedia as an external knowledge for many aspects in data mining techniques [9]-[12]. Due to Wikipedia is the largest online free encyclopedia which covers every topics and contains numerous categories including a people stories, important events and etc. the articles in Wikipedia are provided by a lot of volunteers around the world. Some research works applied Wikipedia to generate features for text categorization [11], annotate or identified document topics [12]-[13].

In this paper, we try to improve the quality of search results clustering by using the concept terms generated from Wikipedia (the largest online encyclopedia) and also introduce a short text clustering method using graph-based construction to connect the related documents. We compared the clustering results of our proposed method with well-known search result clustering techniques, Suffix Tree Clustering [5] and Lingo [4]. Experimental results show that our proposed method significantly outperforms the well-known clustering techniques.

2. RESEARCH METHOD

In this section, the proposed method has 2 primary steps. The First is Concept Extraction step, the process of extracting the concepts from external knowledge source to produce the input for next step. The second is the step of search results clustering based on graph model.

2.1. Concept Extraction

In this section, we explain what the Wikipedia is, what is the information which used to as a concept term and how to extract the concepts from Wikipedia.

2.1.1. Wikipedia

Wikipedia is the largest online encyclopedia of the world. In English version, it contains over 300 million words in nearly 5 million articles, contributed by over 160,000 volunteer editors. Wikipedia has several advantages. First, its articles are much cleaner than typical web pages. In the XML Wikipedia dump file, we can easily extract the parts of Wikipedia article by tracking with Wikipedia Tag contained in the article file.

The Wikipedia article is a page that contains knowledge of each article in the encyclopedia. There are three main sections for each article. First, the Title section is the name or topic of an article. Second, the Content section, this section use to explain what the article is about, and also contain the relative terms that link to the related articles. Last section is Category, it contains the category terms which this article belong to i.e. the article “Puma” belong to the categories “Cat stubs” and “Felines”.

From our observation, we found that the category terms from the Category section seem to properly represent as the concepts of its article. Therefore, we apply these category terms be a representative as the concepts of the document which contains the Title of the articles within.

2.1.2. Building the Wikipedia Corpus

We download the Wikipedia article pages dumped as XML file that contains all of Wikipedia articles in English. For each the article consist of Title, Content and Category which are the three mainly components. We use only the useful Wikipedia pages by removing the useless pages with criteria as following:

1) A title that contains only digits character such as "1944", "821"
2) A single little title such as “A”, “B”, “X” or “Y”
3) An article of years such as "1000s BC" or "500s BC (decade)"
4) Administrative Pages, a page that has the title start with "Wikipedia:", "File:", "Portal:" ,"Category:", "Special:" and "Template:"

The Wikipedia corpus has been created by indexing all Wikipedia pages which can download from Wikipedia website by using Apache Lucene API. We index the Title section for searching with an input text and use the *category terms* in the Category section as the concepts in the concept extraction module.

2.1.3. Extracting the Concepts

After Wikipedia corpus has been created. We perform the concept extraction by searching an input text into the Wikipedia corpus to match the titles of Wikipedia article and determine the category terms and out-link terms of the matched articles as candidate concepts as following steps in Figure 1.

1. Search input document into the Wikipedia corpus by matching with the Title of Wikipedia article.
2. Collect the returned Wikipedia articles search results
3. Get all category terms from returned article
4. Score each term by counting the number of occurrence
5. Return the high score concept term
   If the score is more than 1
   Else, return all terms

Figure 1. Concept extraction steps

2.2. Graph-Based Concept Clustering

In this paper, we proposed the clustering algorithm for short text, Graph-based concept clustering, which consists of 4 primary steps as following: 1) Feature extraction 2) Candidate Cluster Generation and 3) Sub graph detection and 4) Cluster scoring and labeling, that shown in Figure 2.

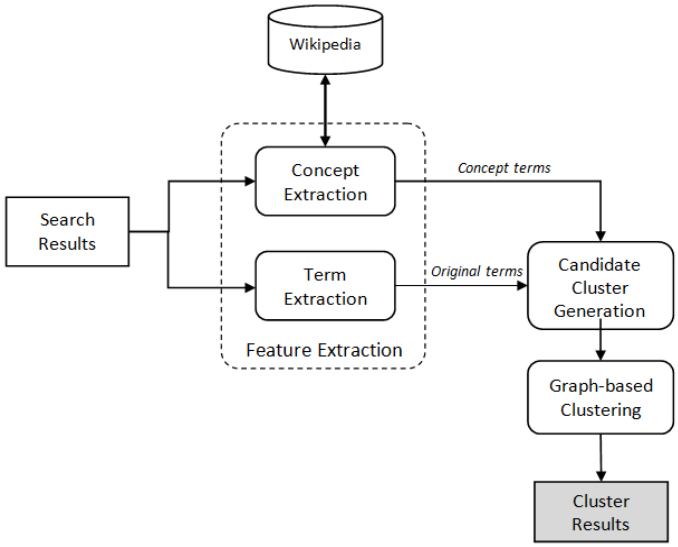

Figure 2. The workflow of Graph-based Concept Clustering algorithm

2.2.1. Feature Extraction

We separate this step into 2 parts. The first part is “Term Extraction”. We define the “Term” as the origin word or phrase that appears in the search results. We remove all stop-word and extract the terms with following rules that shown in Table 1.

Table 1. The rules of term extraction

Rules	Description
Punctuation Mark Term	The terms that appear in punctuation marks such as double or single quotes, parenthesis, bracket, etc.
Noun Phrase Term	We detect the Noun phrase term by using Part-Of-Speech Tagger tools [14]
Name Entity Term	The term that start with capital letter such as “Batman Begins”

The second step, “Concept Extraction”, we explained this step in above section (2. CONCEPT EXTRACTION)

2.2.2. Candidate Cluster Generation

Let $F = \{f_1, f_2, \ldots, f_i\}$ denote as a set of feature that generated from previous step, $D = \{d_1, d_2, .., d_j\}$denote as a set of input document for clustering and $CC = \{cc_1, cc_2, \ldots, cc_i\}$ denote as a set of candidate cluster. We can explain the algorithm to generate the candidate cluster as shown in Figure 3.

```
CC ← ∅ // initiate the candidate cluster set as an empty set
For each f_i ∈ F
    create a new cc_i and labeled it with f_i
    For each d_j ∈ D
      if f_i occur in d_j then
            assign d_j as a member of cc_i
      End if
    End for
End for
```

Figure 3. Algorithm of candidate cluster generation

We apply all features that returned by feature extraction step as candidate clusters (*cc*) and assign the document d_j which contains the features to each candidate cluster

2.2.3. Sub Graph Detection

Let *G* is a weighted undirected acyclic graph, denote as $G = (V, E)$. We denote $V = \{v_1, v_2, \ldots, v_i\}$ is a set of vertices and $E = \{e_{12}, e_{13}, \ldots, e_{ij}\}$ is a set of weighted edges in *G*. First, we define candidate cluster cc_i as v_i in *V*. We link v_i and v_j with the condition of cluster similarity which defined as *simCluster*(v_i, v_j). If the *simCluster*(v_i, v_j) is equal or greater than similarity threshold denote as *sim_th*, v_i and v_j will be linked with e_{ij}, with weighted by value of *simCluster*(v_i, v_j) that is defined as:

$$simCluster(c_i, c_j) = \left(\frac{|c_i \cap c_j|}{|c_i|} + \frac{|c_i \cap c_j|}{|c_j|}\right)/2 \quad (1)$$

The goal of this step is to find the coherent groups of clusters that connected by high weighted. We observe that the coherent sub graph is connected by the candidate cluster which has related documents. We apply the well-known graph algorithm, breadth-first search to detect the sub graphs from the graph G.

2.2.4. Cluster Scoring and Labeling

We score all cluster (vertices in the graph *G*) with scoring function *scoreCluster*(c_i) shown in (2). Then we sort by high score and select top *N* clusters as cluster results.

$$scoreCluster(c_i) = avgWeight(c_i) \times avgDocSize(c_i) \quad (2)$$

$$avgWeight(c_i) = \frac{\sum_{e \in E_i} w(e)}{|E_i|} \quad (3)$$

$$avgDocSize(c_i) = \frac{\sum |c_{ij}|}{|D|} \quad (4)$$

When E_i is a set of edge of c_i, $|E_i|$ is number of edges in E_i , *w(e)* is a weight of edge *e*, c_{ij} is a cluster that linked to c_i, $|c|$ is number of document in cluster *c* and $|D|$ is number of all documents

To display the cluster label for user's exploring. We generate the label of each cluster results by selection the highest number of document of a cluster in sub graph then use the feature term as a cluster label. In case of the same number of document for many clusters in the same sub graph, we choose the concept term as a label first.

3. RESULTS AND ANALYSIS

In our experiments, we compare the results of our proposed clustering framework with well-known traditional web search results clustering algorithms. First, the suffix tree clustering (STC) by [5]-[6] and second, Lingo (singular value decomposition based algorithm). We setup the experiments for analyzing in three aspects as following:

1) The experiment in order to proof that using knowledge from external source, concept term which generated in concept extraction step, can improve the quality of clustering.

2) Experiment to compare our proposed clustering algorithm with traditional ones.
3) Experiment to tune up the quality of proposed algorithm by varying the key parameters, the similarity threshold (sim_th) in the graph construction step.

3.1. Web Search Result Test Set

We use the web search result data set called AMBIENT to evaluate the quality of clustering. The AMBIENT data set consists of 44 ambiguous queries (topics). For each query used to search into Yahoo search engine. The top 100 search results of each query were labeled by subtopics of each query. For instance, the query "jaguar" consists of many subtopics such as "Atari jaguar Video Game", "Mammal", "Jacksonville Jaguar American Football (NFL)" and etc. This data set contains about 4,400 search results and widely used in many researches [1], [8] that related to document or short text clustering.

3.2. Evaluation Method

In this paper, we use the Precision, Recall and average F-score, in many research works [3], [7], [15-17] called F-measure, to measure the quality of clustering. *Precision* is fraction of documents in cluster *c* that belong to class *t* and all documents in cluster *c*. *Recall* is fraction of documents in cluster *c* that belong to class t and all document in class *t* that are defined as:

$$Precision(c,t) = \frac{|D_{c,t}|}{|D_c|} \quad (5)$$

$$Recall(c,t) = \frac{|D_{c,t}|}{|D_t|} \quad (6)$$

F-score measurement is harmonic mean of precision and recall that defined as:

$$Fscore(c,t) = \frac{2 \times Precision(c,t) \times Recall(c,t)}{Precision(c,t) + Recall(c,t)} \quad (7)$$

$$averageFscore = \frac{\sum_{t \in T} \max_{c \in C} Fscore(c,t) \times |D_t|}{\sum_{t \in T} |D_t|} \quad (8)$$

3.3. Experimental Results

For each our experiment, we denote STC as suffix tree clustering, denote Lingo as Lingo clustering and our proposed algorithm, we denote as GC (graph-based clustering). We used the AMBIENT dataset for testing the quality of each clustering algorithms and evaluated them with the average F-score measurement.

The results of evaluation in figure 4 show higher value of F-score for GC algorithm compare with STC and Lingo. For running the clustering algorithm without concept extraction, GC is 32.85% improved compare with STC and 50.25% improved compare with Lingo. For running with concept extraction, GC is 27.63% improved compare with STC and 44.62% improved compare with Lingo and GC with concept is 6.61% improved compare with GC without concept.

For analyzing the effect of similarity threshold on the GC algorithm, we modified the similarity threshold for several values. The result of the similarity threshold modification has shown in Figure 5 and 6

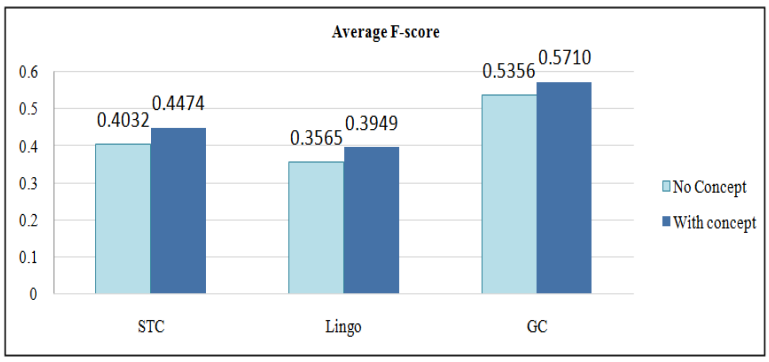

Figure 4. Clustering results comparison for each clustering algorithm

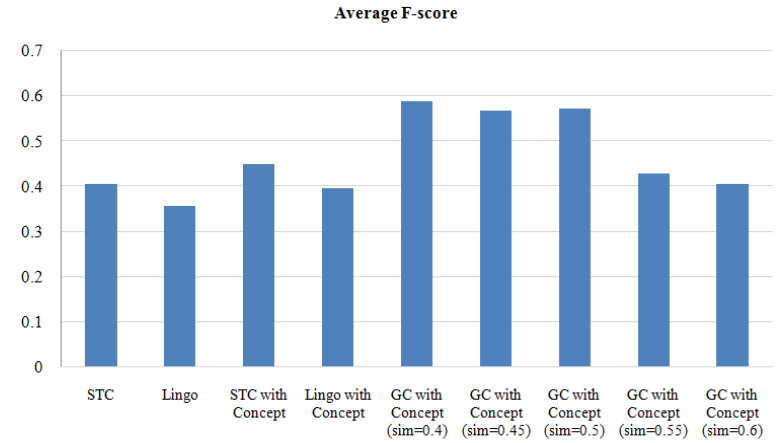

Figure 5. Clustering result of varying of Graph-based clustering's similarity threshold with other algorithms

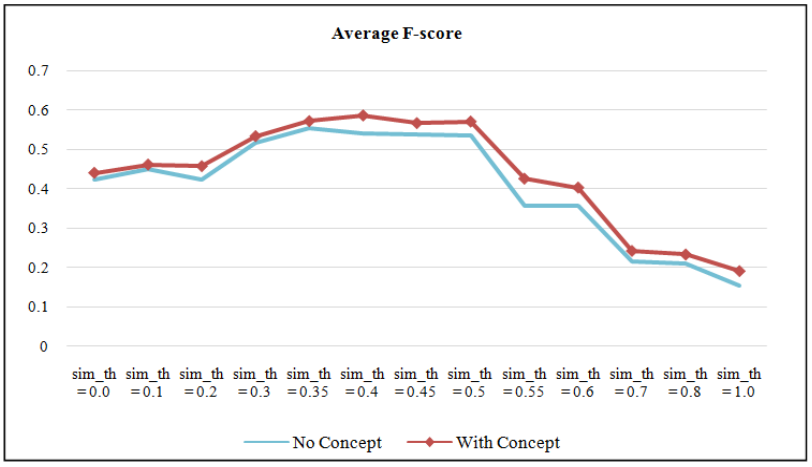

Figure 6. Clustering result of varying similarity threshold of Graph-based clustering

In Figure 7, we show the average number of sub graph that is generated from the sub graph detection step in the GC algorithm by changing the similarity threshold value.

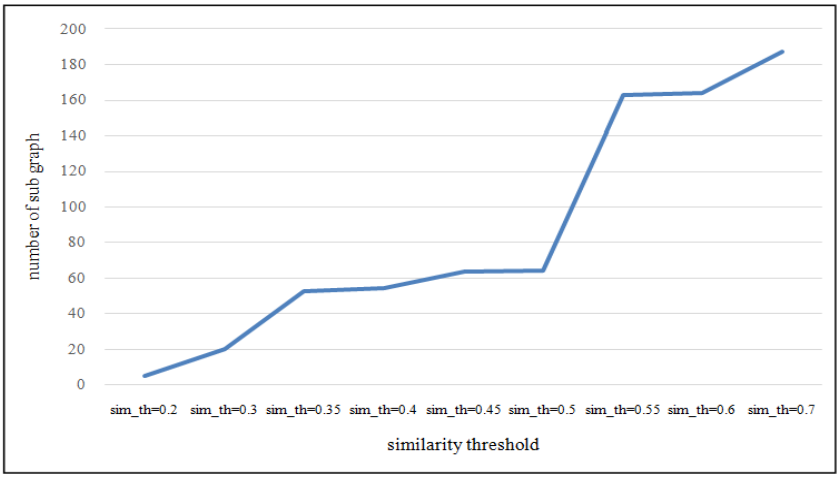

Figure 7. Average number of sub graphs by varying the similarity threshold value

For the search results exploring, users browse to each cluster results that generated by clustering algorithms. In figure 8, we show that the cluster result's labels generated by STC, Lingo and our proposed method GCC which perform the clustering on input document from topic "Jaguar" in the AMBIENT dataset.

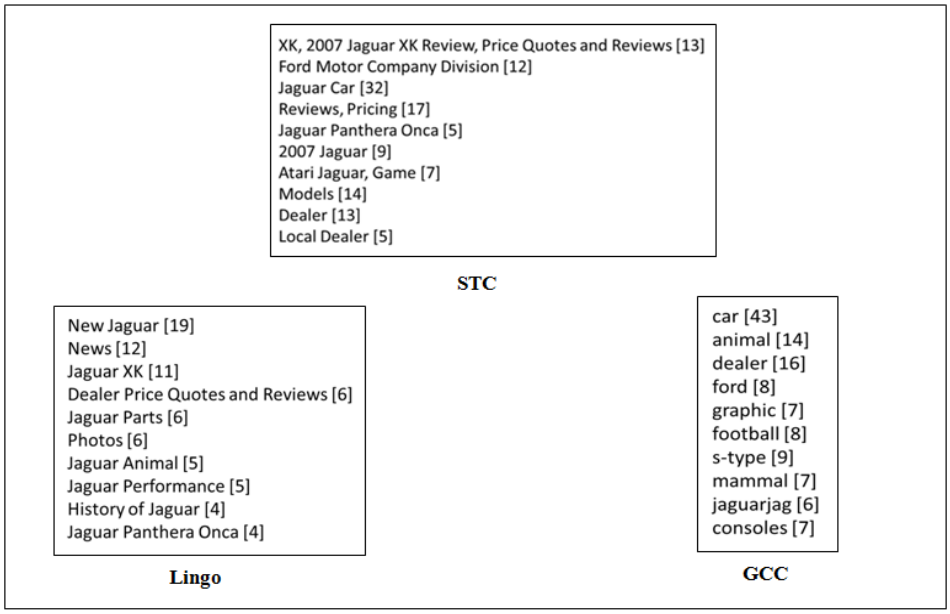

Figure 8. Cluster Labels generated by clustering algorithms

To improve the quality of cluster, we used phrase or term co-occurrence based with graph clustering algorithm by using the original terms from the input document and the extended concept terms which were generated from concept extraction step. The concept terms can help to group in the case of some related documents that do not share the same words but share the same concept. On the other hand, the concepts from the external knowledge source relieve the limitation of short text by increasing the informative data for effective clustering process. On the contrary, the concept extraction step consume a lot of processing time because its searching step need to access many times to the hard drive (I/O) into the Wikipedia corpus indexed files.

Another key variable that can affect the performance of our clustering algorithm is the similarity threshold. This variable used to determine the relation of two candidate cluster on the sub graph detection step. When the similarity threshold is changed to higher value, two candidate clusters will be connected if and only if they have almost the same document members which affect to the number of sub graph will be increased but the performance of clustering will be decreased (see Figure 5 and 6).

For the clustering result exploration, search results clustering algorithms need to provide the readable clustering result for users. Lingo algorithm worked on the description-come-first paradigm. It focused on the meaningful cluster label generation first and grouping the input document which similar to the label after. Likewise the STC algorithm, the phrase for the clustering result's label, will be maintained on the suffix tree construction step and the number of word in that phrase will be calculated for clustering scoring step as well.

For cluster label generation in our proposed algorithm, the label of each candidate cluster was generated in the candidate cluster generation step by using the terms that extract from original document or the concepts from concept extraction step. After the sub graph detection step, we select the cluster label by using the concept term first.

4. CONCLUSION

In this paper we have shown the achievement of web search result clustering by linking the related candidate cluster in graph-based model and reducing the variance of document within one cluster by finding the common words (concepts) for join all documents in the same cluster. We tested the document clustering

in the case of varying the value of similarity threshold. We found that the value between 0.4 – 0.5 that produced the high performance of clustering.

The main reason for using the Wikipedia as the external knowledge source instance for other sources to extract the concept of document. Because, Wikipedia is a largest online encyclopedia that contains a lot of articles which cover most of topics in the world including technical terms or name entities and newly word these can not found in English lexicon database.

In the future, we can improve the efficiency of extracting concept from Wikipedia corpus by finding the good feature selection techniques or combining the many external knowledge sources as a single corpus to extend information for the short document.

ACKNOWLEDGEMENTS

This paper has been supported by the National Electronics and Computer Technology Center (NECTEC) under grants TGIST-01-50-068.

REFERENCES

[1] X. Han and J. Zhao, "Topic-Driven Web Search Result Organization by Leveraging Wikipedia Semantic Knowledge," *in International Conference on Information and Knowledge Management, CIKM10*, pp. 1749-1752, 2010.
[2] C. L. Ngo and H. S. Nguyen, "A method of web search result clustering based on rough sets," *in Proceeding of the 2005 IEEE/WIC/ACM, International Conference on Web Intelligence*, pp. 673-679, 2005.
[3] D. Crabtree, X. Gao, and P. Andreae., "Improving Web Clustering by Cluster Selection," *in The IEEE/WIC/ACM, International Conference on WebIntelligence*, pp. 172-178, 2005.
[4] S. Osinski, J. Stefanowski, and D. Weiss, "Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition," *in The Proceedings of Intelligent Information Processing and Web Mining (IIPWM'04)*, pp. 359-368, 2004.
[5] O. Zamir and O. Etzioni, "Web Document Clustering: A Feasibility Demonstration," *in Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 1998.
[6] O. Zamir, *et al.*, "Fast and intuitive clustering of Web documents". *in Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pp. 287-290, 1997.
[7] S. Jinarat, C. Haruechaiyasak, and A. Rungsawang, "Web Snippet Clustering Based on Text Enrichment with Concept Hierarchy," *in Proceeding of the 16th International Conference of Neural Information Processing*, pp. 309-317, 2009.
[8] A. Di Marco, and R. Navigli, "Clustering and Diversifying Web Search Results with Graph-Based Word Sense Induction", *Computational Linguistics*. Vol. 39, No. 3, pp. 709-754, 2013.
[9] J. Hu, *et al.*, "Enhancing Text Clustering by Leveraging Wikipedia Semantics," *in International Conference on Research and Development in Information Retrieval, SIGIR 2008. Thirty-First Annual ACM SIGIR*, pp. 179-186, 2008.
[10] 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*, pp. 442-448, 2006.
[11] E. Gabrilovich and S. Markovitch, "Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge," *In AAAI Proceedings of the 21st National Conference on Artificial intelligence*, pp.1301-1306, 2006.
[12] P. Schönhofen, "Identifying document topics using the Wikipedia category network," *In the Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI'06)*, pp. 456-462, 2006.
[13] D. Milne and I. H. Witten, "Learning to Link with Wikipedia," *In International Conference on Information and Knowledge Management, CIKM08*. pp. 509-516, 2008.
[14] K. Toutanova and C. D. Manning. "Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger," *In The Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, pp. 63-70, 2000.
[15] P. Vahdani Amoli and O. Sojoodi Sh., "Scientific Documents Clustering Based on Text Summarization", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 5, No. 4, pp.782-787, 2015.
[16] Ravi kumar V., and K. Raghuveer, "Legal Documents Clustering and Summarization using Hierarchical Latent Dirichlet Allocation", *IAES International Journal of Artificial Intelligence (IJ-AI)*, Vol. 2, No. 1, pp. 27-35, 2013.
[17] O. Tsur, A. Littman, and A. Rappoport, "Efficient Clustering of Short Messages into General Domains," *In Proceeding of the 7th International AAAI Conference on Webblogs and Social Media*, 2013.

BIOGRAPHIES OF AUTHORS

Supakpong Jinarat received his B. Sc. of Computer Science from Khon Kaen University, Thailand, in 1999. He received his M. Sc. of Computer Science from Kasetsart University, Thailand, in 2004. He is currently a Ph.D. candidate in Computer Engineering at Kasetsart University. His research interests are Machine learning, Data and Web Mining and Big Data Analysis

Choochart Haruechaiyasak received his Ph.D. degree from the Department of Electrical and Computer Engineering, University of Miami, in 2003. After receiving his degree, he has worked as a researcher at the National Electronics and Computer Technology Center (NECTEC), National Science and Technology Development Agency (NSTDA). His current research interests include data and text mining, natural language processing, information retrieval, and data management. One of his objectives is to promote R&D works through IT applications for government offices and business sectors in Thailand. He is also a visiting lecturer at several universities in Thailand.

Arnon Rungsawang received his B.Eng. of Electrical Engineering from the King Mongkut Institute of Technology (Ladkrabang campus), Thailand, in 1986. He received his PhD in Computer Engineering from the ENST-Paris, France in 1997. Currently, he is an associate professor in the Department of Computer Engineering at Kasetsart University, Bangkok. His research interests are Information and Knowledge Retrieval, Internet Computing, and Social Network Mining.