Reorganization of Search Results Based on Semantic Clustering

Chihli Hung, Zhen-Bang Wang, Pei-Fen Hu, Chen-Yu Yen, Tsung-His Lin, and Li-Hao Chiang

***Abstract*—This research proposes a novel framework for clustering search results, whereby efficiency and effectiveness are considered simultaneously. Search engines are an essential method for searching the Internet. Due to communication issues between information providers, information requesters and search engines, some relevant results may not be shown at the top of the list of search results. Based on the cluster hypothesis, whereby documents containing similar concepts will match the same search requests, clustering techniques are able to reorganize search results and improve performance. Traditionally, search results clustering works by mainly focusing on document snippets, due to the need for a quick response to the user's query. However, snippets contain poor quality semantics, which may cause the problem of poor effectiveness. On the other hand, using full-text clustering is impractical as it is very time consuming. This research integrates the real-time and batch processing phases. Batch processing achieves greater effectiveness, and real-time processing returns the clustering results to users quickly. From the experiments, the proposed method is able to achieve search efficiency and effectiveness at the same time.**

***Index Terms*—Search result clustering, document organization, information retrieval, semantics indexing, web search.**

I. Introduction

Vast volumes of information on the Internet have made it the world's largest data repository. Finding information that is relevant to users or information requesters from a huge data repository has always been the main goal of search engines [1]. Existing popular search engines, such as Google, Yahoo, Bing, etc., use their own specific searching and sorting algorithms to search for keywords, and present them on web pages ordered linearly, based on the degree of relevance to the user's query. However, there is a communication gap between information providers, information requesters, and search engines. The nature of the gap lies in the characteristics of human language, such as polysemy, synonyms, word sense vagueness and ambiguity, which reduce the accuracy of search results. On average a query contains between 1.6 and 3.3 terms [2]. A shorter query has the further difficulty of semantic ambiguity. On the other hand, according to the research of Höchstötter and Lewandowski [3], most users only read the first few results returned by a search engine. More than half of users only browse the first page returned by a search engine. In general, users rarely use the scroll-down function of a mouse wheel to view further results. Such browsing behavior results in a decrease in the quality of search results due to the fact that the relevant information may fall outside the focused area.

Existing search engines use their own specific mapping and ranking algorithms in response to a user's query and display the search results in a linear format. Theoretically, the higher the relevance of the search results to the purpose of the user's query, the higher the ranking will be. In order to help users to evaluate the relevance, search engines usually present a snippet for each search result. A snippet is a tiny piece of a document which contains the keywords in the user's query, which enables the user to evaluate the usefulness of the search result without reading the whole document. A search engine provides an online real-time service that seeks to achieve both search speed and accuracy of search results simultaneously. Due to the user's limited attention capacity, a search engine delivers the search results page by page. One page usually includes 10 pieces of linked information. However, word sense vagueness and ambiguity sometimes result in highly relevant information falling outside the focused area, i.e. outside the first few results of the first web page, which thus reduces search effectiveness.

In the literature of search results clustering, most researchers [4]-[8] focus on snippets instead of whole documents, and reorganize the search results based on various clustering techniques. However, the use of snippets, due to the lack of textual content, risks the predicament of lack of semantic content, which will reduce the aggregation of semantics of clusters [9], [10]. On the other hand, researchers such as Mecca *et al*. [9], Soliman *et al*. [10] and Schenker *et al*. [11] show that focusing on whole documents to reorganize the search results sacrifices the basic requirement of the search engine's rapid response to a user's query. Therefore, this research proposes a framework that combines batch and real-time processing modes. The whole documents linked by the search results are used in the batch mode, and the snippets are directly used in the real-time processing mode. After text preprocessing, the dynamic K-means clustering approach is proposed. In this research, our proposed framework combines the advantages of using snippets and whole documents. The user is able to ignore irrelevant information and narrow the scope of browsing based on our dynamic K-means clustering approach.

The rest of the paper is organized as follows. Section II discusses related works. Section III outlines the proposed

Manuscript received July 20, 2018; revised October 8, 2018. This work was supported in the Ministry of Science and Technology of Taiwan under Grant MOST 106-2410-H-033-014-MY2.

C. Hung and Z.-B. Wang are with Chung Yuan Christian University, Taoyuan, Taiwan (e-mail: chihli@cycu.edu.tw, bang@cycu.org.tw).

P.-F. Hu and C.-Y. Yen are with SYSCOM Computer Engineering Co., Taipei, Taiwan (e-mail: Pei-fen_Hu@SYSCOM.com.tw, Jan-Chang_Yan@SYSCOM.com.tw).

T.-H. Lin and L.-H. Chiang are with Institute for Information Industry, Taipei, Taiwan (e-mail: tsunghsilin@iii.org.tw, lihaochiang@iii.org.tw).

DOI: 10.7763/IJCTE.2018.V10.1217

framework. The experimental results are presented in Section IV. A conclusion and possible further work are presented in Section V.

II. Related Work

Search engines are a useful method for quickly searching for information that users are interested in from the Internet. However, due to a communication gap between information providers, information requesters and search engines, the usefulness is degraded, especially for very short and vague queries. On the other hand, both effectiveness and efficiency are the main goals for search engines. In order to quickly respond to a user's query, search engines present their results ranked by the relevance to the query. Thus, it is inevitable that some relevant results may not be presented at the top of the list of search results. Based on the technique of clustering, whereby the same or similar objects are grouped together, van Rijsbergen [12] proposed the cluster hypothesis in the information retrieval field, so that documents with similar concepts will match the same search requests. The technique of document clustering for this issue is called search results clustering, which groups the documents with the same or similar concepts together, and thus provides a shortcut for the user to explore search results with the same or similar concepts [7].

Traditional document clustering has two main categories, which are hierarchical clustering and flat clustering. Due to the $O(n^2)$ computational time complexity, traditional hierarchical clustering techniques may be infeasible when dealing with a large data set. The K-means technique proposed by MacQueen [13] is the most popular flat clustering method owing to its simple computational time complexity. For example, Giannotti *et al*. [14] developed WebCat to automatically cluster web search results using K-means. However, the traditional K-means technique suffers from the requirement to pre-assign the value of K before clustering. Thus, Alam and Sadaf [15] proposed a heuristic search approach to decide the value of K and clustered search results using the K-means technique. We propose a novel dynamic K-means technique to cluster search results based on the distribution of data.

The main purpose of clustering search results is to improve the effectiveness of the search engines. For an online real-time service, such as the search engine, efficiency should not be compromised too much. Schenker *et al*. [11] argued that the whole document processing task is very time-consuming, which makes this task infeasible. According to the experiments of Zamir and Etzioni [16], there was no significant degradation in the quality of clusters between using whole documents and document snippets. Generally speaking, most works in the field of search results clustering focus on document snippets rather than whole documents from the search results [4]-[8]. However, Osiński and Weiss [5] argued that the amount of data from document snippets is often extremely small and of low quality, so hierarchical clustering or K-means does not perform well. Mecca *et al*. [9] reached a similar conclusion, namely that the quality of document snippets is very poor and not sufficiently informative, so the classification performance severely degrades when snippets rather than whole documents are used.

Evidently, in the field of search results clustering, efficiency and effectiveness are a trade-off. In contrast to researchers such as Mecca *et al*. [9] and Soliman *et al*. [10], who focused on whole documents, Schenker *et al*. [11] designed a framework consisting of asynchronous and online modes. The user chooses the online mode for clustering snippets from the search results and chooses the asynchronous mode for clustering whole documents from the search results. Although their work has considered both efficiency and effectiveness, it retains both the advantages and disadvantages of the two modes. The requirement for user registration in order to receive the 'clustering job complete' email is unusual. In this research, we propose a novel framework, which integrates the real-time processing with the batch processing. In contrast to Schenker *et al*. [11], our proposed framework combines the advantages of clustering snippets and whole documents of search results.

III. Approach

This research proposes a framework that integrates real-time and batch processing to combine the advantages of clustering snippets and whole documents. In this framework, the keyword-link pair storage and the keyword queue (Fig. 1 and Fig. 2) are used to connect the real-time and batch processing. The keyword-link pair storage stores the batch processing results, and the keyword queue stores the keywords queried by the users. Batch processing is fired when the keyword queue is not empty.

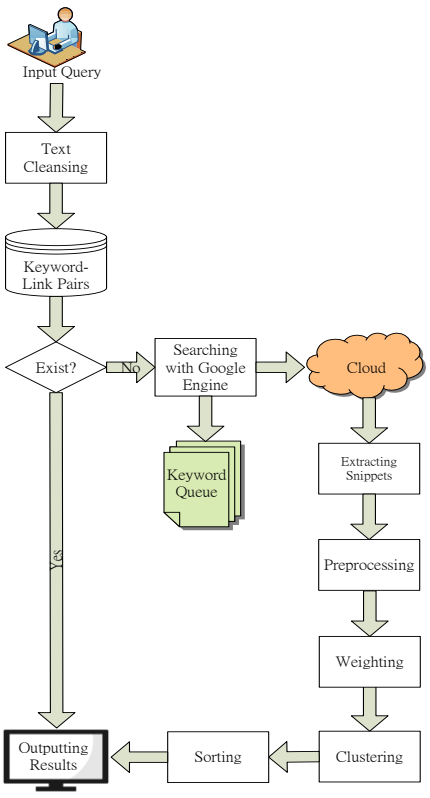

Fig. 1. The proposed real-time processing.

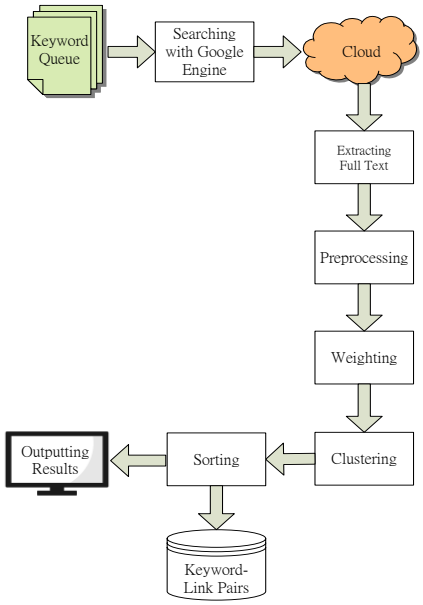

Fig. 2. The proposed batch processing.

A. Real-Time Processing Phase

Real-time processing is shown in Fig. 1. When the user enters a query, the first step in this phase is text cleansing, which converts each English word to its lower case form, and lemmatizes each word to its based lemma through the WordNet lemmatizer function. We then look up the keyword-link pair storage, and if the query keywords already exist, the batch processing results already stored are directly shown in the browser, as in Fig. 3. If the query keywords are not already stored in the keyword-link pair storage, they are saved in the keyword queue, which will be processed by the batch processing. The query keywords are then passed to the Google search engine and the following processes are divided into five steps, namely extracting snippets, preprocessing, weighting, clustering and sorting. Finally, we display the clustering results to users through our proposed user interface (Fig. 3).

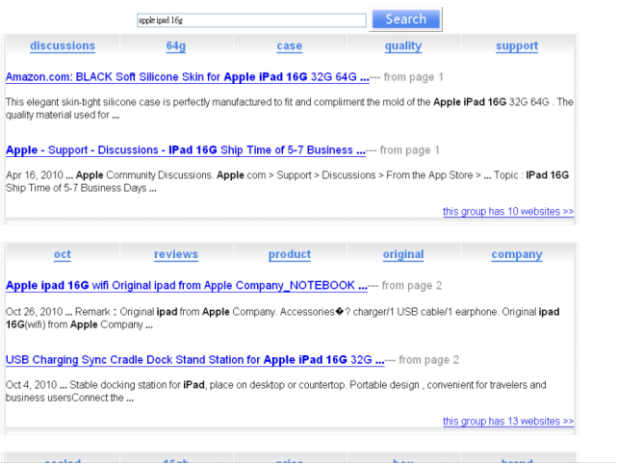

Fig. 3. An example of our proposed user interface for user search results.

B. Extracting Snippets

In comparison with the whole document, a snippet is extremely small. In order to respond quickly to the user's query, the real-time processing phase only collects snippets from the top 100 search results returned by the Google search engine.

C. Preprocessing

In this stage, only English articles are addressed, and text cleansing is applied. Firstly, all HTML tags and punctuation are removed from 100 documents. Each English word is then converted to its lower case form, and lemmatized to its base lemma through the WordNet lemmatizer function. All stop words are removed. In order to increase the processing speed, only words which occur 5 or more times are retained.

D. Weighting

Based on the traditional vector space model [17], the frequency of occurrence of a word is related to the concept of the document. This research uses the term frequency representation method to convert each document to a document vector as shown in (1).

$$D = [W_1, W_2, \ldots, W_m] \tag{1}$$

where D indicates a document vector, m indicates the total number of different words formed by 100 documents, and W is the weight of each word, which is shown in (2).

$$W_i = \mathrm{TF} \tag{2}$$

where TF indicates term frequency in a document.

E. Clustering

There is a relationship between the top100 documents, as all documents that are returned from a search engine contain the same query keywords. These documents are clustered based on their semantics. One cluster may consist of more than 50% of the documents, while at the same time the average similarity for this cluster is lower than that of the other clusters, as the number of documents is much greater than that of the others. K-means is a popular clustering technique due to its efficiency and simplicity. From our experiments, we find that K-means also suffers from this issue. Therefore, this research proposes the dynamic K-means clustering approach. The initial K value is set at 2 and the maximum is set at 5. That is, the number of clusters is between 2 and 5. If there is a cluster whose number of documents is greater than 50% of the entire dataset, this cluster is forced to separate into two clusters by K-means, where K is 2. This process continues until the number of clusters reaches five, or no clusters contain more than 50% of documents. The five most frequent words in each cluster represent the associated cluster.

F. Sorting

This step determines the order of document clusters. A cluster order value, C, is calculated by its average document order in the original search results (3). A cluster with a smaller cluster order value is arranged first on the results list.

$$C = \frac{\sum O}{n} \tag{3}$$

where C indicates the cluster order, O indicates the document order of the original search results and n indicates the number of documents in the cluster.

G. Output Results

The real-time processing phase and the batch processing phase show the same user interface as shown in Fig. 3.

H. Batch Processing Phase

The batch processing phase is shown in Fig. 2. This phase is fired when there are keywords in the keyword queue. We search the user's query keywords through the Google search engine and further retrieve the whole document from the first 100 search results. This batch processing phase follows the same preprocessing step as in the real-time processing phase. In contrast to the snippets in the real-time processing phase, the full-text web pages contain many HTML tags, which highlight more relevant texts. In this research, we apply greater emphasis to certain words highlighted by specific HTML tags, shown in Table I. At the weighting stage, Equation (2) has been modified into Equation (4) to give greater weight to more relevant words.

The *remaining* steps of the batch processing phase, i.e. clustering, sorting and outputting results, are the same as those in the real-time processing phase. Finally, the search results are saved in the keyword-link pair storage for use in the real-time processing phase, and the query keywords are removed from the keyword queue.

TABLE I: WEIGHTS FOR HTML TAGS

HTML Tag	<h1>	<h2>	<b> <a> <u> <i> <big>	<font>	Others
Weight	3	2.5	2	1.5	1

$$W_i = \mathrm{TF} \times \mathrm{Weight} \tag{4}$$

IV. EVALUATION

Due to a lack of open benchmarking datasets in the field of search results clustering, human evaluation is mainly used in the literature, such as [16] and [10]. Based on the concept of an experimental method, this research pre-selects five web pages as our search targets and invites 30 human testers to compare the performance between a traditional search engine, i.e. Google, and our proposed method, by searching for these target web pages. The maximum time allowed for each tester using a search engine for a topic search is three minutes. A search fails if they cannot find the chosen target web page in three minutes. For a target web page, we count the number of web pages the testers have visited for each search attempt. Thus we evaluate our proposed method based on two criteria: the average page visit number and the average search success rate.

Finally, we further analyze our proposed method using statistical analysis, which is also used by several researchers in various domains such as [18] and [19]. Based on user experiences of five search targets, the paired *t*-test is used to test whether or not our proposed method is significant different from the traditional Google search engine in provision of informative results and satisfaction of search results.

A. Demographic Variables

This research invited 30 volunteer testers. The age distribution of these testers is shown in Table II and the occupational distribution is shown in Table III.

TABLE II: THE AGE DISTRIBUTION OF TESTERS

Age	~18	19 ~22	23 ~25	26 ~29	30 ~34	35~
%	13%	44%	17%	13%	10%	3%

TABLE III: THE OCCUPATIONAL DISTRIBUTION OF TESTERS

Profession	Industry	Student	Service	IT	Others
%	7%	60%	17%	10%	6%

B. The Average Page Visit Number

We assign five target web pages in advance. All 30 testers search five target web pages via the Google search engine and the proposed method. The average page visit number, *A*, is calculated as (5).

$$A = \frac{\sum_{i=1}^{Q \times T} P}{Q \times T} \tag{5}$$

where *Q* indicates the number of target web pages, i.e. 5, *T* indicates the number of Testers, i.e. 30, and *P* indicates the number of visiting pages for a specific target web page searched by a specific tester. From the experiments, the average number of visiting web pages is 6.32 using the Google search engine and 3.83 using the proposed method. These results show that the proposed method has the potential to find the targeted web pages more efficiently than the Google search engine does.

C. The Average Search Success Rate

In this research, a successful search, as shown in (6), is defined as a search that is able to find a predefined target web page using a search engine within three minutes.

$$B = \frac{\sum_{i=1}^{Q \times T} S}{Q \times T} \tag{6}$$

where *Q* indicates the number of target web pages, i.e. 5, *T* indicates the number of testers, i.e. 30, and *S* equals 1 for a successful search, or else 0. According to the experimental results, using the traditional Google search engine, the success rate is 74%, and it is 83.3% while using the proposed method. Thus, based on the criterion of the average search success rate, the proposed method is able to effectively find pre-selected target web pages.

D. Statistical t-Test

We finally use a questionnaire, which is based on a Likert 5-point scale, where 5 indicates strongly agree and 1 indicates strongly disagree, to collect subjective feedbacks from 30

testers according to their search experiences. The paired sample *t*-test is used to further analyze whether or not our proposed method is significant different from the traditional Google search evaluated by the criteria of providing informative results and search satisfaction. The experimental results in Table IV show that these two methods achieve a significant level of difference based on the significance value of 0.05. Thus, our proposed method significantly provides results that are more informative and satisfactory than the traditional Google search method.

TABLE IV: THE PAIRED T-TEST RESULTS FROM TWO METHODS

	Google	Proposed Method
Informative Results	3	4.067 (p-value=0.00)
Satisfaction	3	3.8 (p-value=0.00)

V. CONCLUSION AND FURTHER WORK

In this research, we propose a search framework for clustering search results while taking into account the search efficiency and effectiveness. In the field of search results clustering, due to the importance of the efficiency of the search engine service, most researchers cluster search results based on snippets. However, as the length of a document snippet is very short, this may result in low quality semantics and poor search effectiveness. Thus, there is a trade-off between search efficiency and effectiveness. This research proposes a framework that integrates real-time and batch processing to combine the advantages of using both snippets and whole documents. We use the keyword-link pair storage and the keyword queue to connect real-time and batch processing. The keyword-link pair storage stores the batch processing results, and the keyword queue temporarily stores the keywords queried by the users. Batch processing is fired when there are keywords in the keyword queue, and there are clustering results stored in the keyword-link pair storage. Thus, the batch processing achieves greater effectiveness and the real-time processing quickly returns the clustering results to the users. On the other hand, a traditional K-means clustering technique suffers from the pre-selection of the cluster number, K, before clustering. In this research, we propose a dynamic K-means, which automatically increases the cluster number based on the distribution of data. In short, the proposed framework and the dynamic K-means are the two main original contributions of this research. Finally, we evaluate our proposed framework based on the average page visit number and the average search success rate. From the experiments, we demonstrate that the proposed framework is able to take into account both search efficiency and effectiveness at the same time.

In terms of further work, some possible directions are as follows:

1) Further work may try and compare different vector representation approaches, such as TFIDF, Binary, singular value decomposition (SVD), etc. Different vector representation approaches may produce different cluster results.
2) There are many clustering algorithms [20], which may be used in further work.
3) Further work may concentrate on evaluation of the performance of search results clustering, which can be a complex process.
4) Further work may integrate the technique of word sense disambiguation (WSD), as the ambiguity of word sense is a characteristic of language, and is of great importance when dealing with text [21].

ACKNOWLEDGMENT

This work was supported in the Ministry of Science and Technology of Taiwan under Grant MOST 106-2410-H-033-014-MY2.

REFERENCES

[1] B. J. Jansen and A. Spink, "How are we searching the world wide web? A comparison of nine search engine transaction logs," *Information Processing & Management*, vol. 42, pp. 248-263, 2006.
[2] N. Höchstötter and J. Koch, "Standard parameters for searching behaviour in search engines and their empirical evaluation," *Journal of Information Science*, vol. 34, 2008.
[3] N. Höchstötter and D. Lewandowski, "What users see – Structures in search engine results pages," *Information Sciences*, vol. 179, pp. 1796-1812, 2009.
[4] O. Zamir and O. Etzioni, "Grouper: A dynamic clustering interface to web search results," *Computer Networks*, vol. 31, no. 11-16, pp. 1361-1374, 1999.
[5] S. Osiński and D. Weiss, "Conceptual clustering using Lingo algorithm: Evaluation on open directory project data," *Intelligent Information Processing and Web Mining*, pp. 369-377, 2004.
[6] S. Osiński and D. Weiss, "A concept-driven algorithm for clustering search results," *IEEE Intelligent Systems*, vol. 1541-1672, pp. 48-54, 2005.
[7] C. Carpineto, S. Osinski, G. Romano, and D. Weiss, "A survey of web clustering engines," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1-38, 2009.
[8] R. Navigli and G. Crisafulli, "Inducing word senses to improve web search result clustering," in *Proc. the 2010 Conference on Empirical Methods in Natural Language Processing*, 2010, pp. 116-126.
[9] G. Mecca, S. Raunich, and A. Pappalardo, "A new algorithm for clustering search results," *Data & Knowledge Engineering*, vol. 62, pp. 504-522, 2007.
[10] S. S. Soliman, M. El-Sayed, and Y. F. Hassan, "Semantic clustering of search engine results," *The Scientific World Journal*, vol. 2015, pp. 1-9, 2015.
[11] A. Schenker, M. Last, and A. Kandel, "Design and implementation of a web mining system for organizing search engine results," *International Journal of Intelligent Systems*, vol. 20, no. 6, pp. 607-625, 2005.
[12] C. J. V. Rijsbergen, *Information Retrieval*, 2nd ed. London: Butterworths, 1979.
[13] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281-297.
[14] F. Giannotti, M. Nanni, D. Pedreschi, and F. Samaritani, "WebCat: automatic categorization of web search results," in *Proc. the 11th Italian Symposium on Advanced Database Systems*, 2003, pp. 507-518.
[15] M. Alam and K. Sadaf. (2015). Web search result clustering based on heuristic search and K-means. [Online]. Available: https://arxiv.org/abs/1503.06609
[16] O. Zamir and O. Etzioni, "Web document clustering: A feasibility demonstration," in *Proc. the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998, pp. 46-54.
[17] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol.18, no.11, pp. 613-620, 1975.
[18] C. Hung and H.-K. Lin, "Using objective words in SentiWordNet to improve word-of-mouth sentiment classification," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 47-54, 2013.
[19] C. Hung, "Word of mouth quality classification based on contextual sentiment lexicons," *Information Processing & Management*, vol. 53, no. 4, pp. 751-763, 2017.
[20] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, no. 2, pp. 165-193, 2015.

[21] C. Hung and S.-J. Chen, "Word sense disambiguation based sentiment lexicons for sentiment classification," *Knowledge-Based Systems*, vol. 110, pp. 224-232, 2016.

Chihli Hung is a professor at the Department of Information Management of Chung Yuan Christian University, Taiwan. He obtained a PhD at School of Computing and Technology from the University of Sunderland, UK in 2004. His current research interests are in information retrieval, text mining, machine learning, data mining and intelligent systems.

Zhen-Bang Wang earned a bachelor degree at the Department of Information Management from Chung Yuan Christian University, Taiwan. He worked for Pivsti Corporation since 2014 and currently holds a title of chief of technical officer for Pivsti.

Pei-Fen Hu works for Syscom Computer Engineering Co. She is a deputy chief engineer for highly distributed big-data management system with Intelligent Scalable SQL Index Servers Programs.

Chen-Yu Yen works for Syscom Computer Engineering Co. He is an assistant chief engineer for highly distributed big-data management system with Intelligent Scalable SQL Index Servers Programs.

Tsung-Hsi Lin is a section manager of Institute for Information Industry (III), Taiwan. He is a Taiwan PostgreSQL User Group promoter and his current research interests are in open source relational/NoSQL database and promoting industry 4.0 in Taiwan.

Li-Hao Chiang is a section manager of Institute for Information Industry (III), Taiwan. His current interests are in open source cloud fundamental infrastructure and cloud native applications.