Search Result Clustering for Thai Twitter Based on Suffix Tree Clustering

Santipong Thaiprayoon, Alisa Kongthon, Pornpimon Palingoon and Choochart Haruechaiyasak

Speech and Audio Technology Laboratory (SPT)
National Electronics and Computer Technology Center (NECTEC)
Thailand Science Park, Klong Luang, Pathumthani 12120, Thailand
[santipong.tha, alisa.kon, pornpimon.pal, choochart.har]@nectec.or.th

***Abstract*—Today Twitter has become a popular online medium for posting and sharing news and events. Generally, many Twitter posts or "tweets" refer to the same topics or events. Searching on Twitter could return a long list of search results. To solve the problem, we propose an approach for clustering the Twitter search results based on the Suffix Tree Clustering (STC) algorithm. However, two main drawbacks of original STC are some of the returned cluster labels are unmeaningful and it is unable to create hierarchical structure. In this paper, we present a new approach called Suffix Tree Clustering with Label Merging (STC-LM). The key idea of the STC-LM is to merge partially overlapped cluster labels and then create two-level label structure. We performed experiments by using Thai Twitter posts from 12 topics such as flooding, traffic and entertainment. The performance based on the F1 measure is equal to 70%, an improvement of 9% from the baseline method.**

Keywords- Suffix tree clustering; search result clustering; Thai Twitter

I. INTRODUCTION

Twitter [16] is an online social networking and microblogging service that enables millions of people to share public information called "tweets". Tweets are dynamically generated in large volumes. Twitter users could "retweet", i.e., share interesting tweets in which someone else has posted. As a result, there are many Twitter posts containing similar messages on Twitter. Therefore, there is a need for tools to help users quickly find tweets that they are interested. At present, there are some existing Twitter search engines such as TwitterSearch [18], TweetScan [17] and Topsy [20]. Given a search query, these search engines typically return a long list of search results. Many tweets contain messages with the same topics or same events from several sources such as general users, news agencies and companies. Users have difficulties in finding relevant results from a long list of search results. Twitter does not solve the problem of information overload for users since it does not provide a mechanism to filter all duplicate tweets in search. Moreover, when hot issue occurs, such as flooding in Thailand, there is too much information to pay attention to. The key issue is to organize tweets into a list of cluster labels for facilitating user's browsing task.

One possible solution to the information overload problem is to perform a post retrieval organization of the search results from Twitter into coherent groups. If the Twitter search results are allocated in different groups, users will be able to have a look at the whole topics and just select interesting group to browse. It can also help in filtering out topics that the user is not interested in. Many previous works use Suffix Tree Clustering (STC) algorithm for clustering search results. STC is one of the popular text clustering because it is a fast incremental and linear time algorithm for search result clustering. STC will categorize huge volumes of data into coherent cluster labels. STC is based on the Suffix Tree Document (STD) model which was first proposed by Zamir and Etzioni [10, 11]. We apply the STC algorithm which is a part of Carrot2 framework. Carrot2 [13] is an open source search result clustering engine. It can automatically organize small collection of documents into thematic categories. However, the main drawbacks of Carrot2 are (1) it sometimes generates unmeaningful cluster labels, especially when applying for Thai language (2) it is unable to create a hierarchical structure of cluster labels.

In order to improve the original algorithm in Carrot2, we propose a new approach to merge and create two-level label structure called Suffix Tree Clustering with Label Merging (STC-LM). This algorithm could merge partially overlapped labels, which can be combined into one label. For example, the word "น้ำ" (water) will be merged into "น้ำท่วม" (flooding). To evaluate the performance of the proposed approach, we constructed a Twitter corpus from Twitter search API by using the Twitter4j [19] library. The Twitter corpus is created by collecting tweets from many Twitter profile pages in Thailand such as *Thaiflood*, *Manager* and *FM995radio*. The Twitter corpus, which contains approximately 160,000 tweets beginning on November 2011 to January 2012, is used in our experiments. Using this corpus, we select popular keywords under 12 topics such as *flooding*, *survival kits* and *oil prices*. We used this corpus to perform the evaluation of the cluster label results based on our proposed approach with the original Carrot2 by using the F1-measure. The advantage of clustering approach is it can generate the clusters of topics based on the terms extracted from tweets in real-time. The clustering results based on a user query can represent the current or hot topics related to the input keywords. Therefore, the proposed approach could reflect the real-time thoughts of users on social networks.

The remainder of this paper is organized as follows. In next section, we review some highlighted works related to the suffix tree clustering. Section 3 presents the proposed approach of search result clustering for Thai Twitter based on STC with label merging. Section 4 presents the experimental results and discussion. Section 5 gives conclusion and future works.

II. Related Works

There are numerous works related to search result clustering tasks. Yang et al. [9] implemented and proposed a novel algorithm called sentence-based suffix tree clustering algorithm (SSTC) for web documents. They implemented the algorithm based on the Carrot2 by extending it with structure weights of nodes. Wan et al. [8] proposed a new document clustering algorithm for the Chinese snippets in RSS. They extracted meaningful Chinese words from snippets based on STC and clustered RSS snippets using group-average link clustering algorithm with a new document similarity measure. The weighted suffix tree is built with sentences instead of documents. Each part of documents is assigned with a significance level as structure weight stored in the node of the suffix tree. Han and Zhao [3] proposed a novel topic-driven search result organization method, which can first detect the topic knowledge of a query by finding the coherent Wikipedia concept groups from its search results. They focus on improving the utility of organize search results.

Han et al. [2] proposed a method to automatically grouping Web snippets through an improved STC algorithm by combining the advantages of vector space model (VSM) and suffix tree clustering (STC) document models. Rangrej et al. [7] studied various clustering technique on short text documents and provided experimental results using corpus from Twitter. Zeng et al. Hu et al. [4] presented a method for merging the semantic duplicate clusters and hierarchicalizing the label-contained clusters by using cluster label results from STC algorithm.

Our approach is different from previous works in three aspects. Firstly, we apply Part-of-Speech (POS) tagging for each cluster labels to clean up uninformative cluster labels. Secondly, we merge and create two-label structure to improve the quality of cluster labels. Thirdly, we filter repetitious tweets to keep only legitimate cluster labels.

III. The Proposed Approach

In this section, we first provide a brief review of the STC and the Carrot2 framework. Then our proposed approach will be given with full details.

A. *Suffix Tree Clustering (STC)*

STC [1, 5] is a linear time (O(n)) clustering algorithm based on a suffix tree which efficiently identifies sets of documents that share common phrases or the suffix of a phrase into one cluster, and then create clusters according to these phrases. The original STC algorithm, however, cannot provide an effective evaluation method to assess the quality of clusters. STC algorithm has three logical steps: (1) document cleaning, (2) identifying base clusters, and (3) combining base clusters.

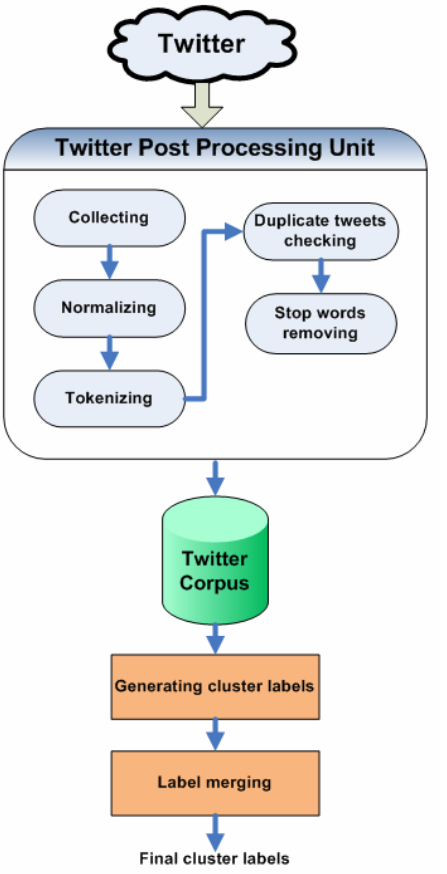

Figure 1. The process of STC-LM

B. *Carrot2 Framework*

Carrot2 is an open source search result clustering engine. It can automatically organize small collection of documents into thematic categories. Carrot2 is implemented in Java. It can fetch search results from various sources, e.g. Yahoo, Google, Wiki and more. Currently it offers two specialized search results clustering algorithms: Lingo and STC. The key data structure used in the STC algorithm is a Generalized Suffix Tree (GST) built for all input documents. The algorithm traverses the GST to identify words and phrases that occurred more than once in the input documents. Each such word or phrase gives rise to one base cluster. The last stage of the clustering process is merging base clusters to form the final clusters.

C. *Suffix Tree Clustering with Label Merging (STC-LM)*

To improve the quality of cluster labels, we propose an approach which consists of three components. Figure 1 illustrates the overall process, which can be explained in details as follows.

1) *Twitter Post Processing Unit*

For the first component, there are five modules which support the cleansing of tweets and prepare the tweets into Twitter corpus.

- Collecting: This module is for collecting tweets from many Twitter profile pages by using the twitter4j library for accessing the Twitter search API.
- Normalizing: Tweets are parsed and all non-word tokens such as HTML tags, punctuations, numbers, special symbols are removed. Each tweet including urls, retweet, hashtags and replies are stripped and converted to lowercase.
- Tokenizing: The main task of this module is to segment a given text into word tokens. We use the basic longest matching open-source word segmentation program called LexTo [14].
- Stopword removing: This module removes the stopwords such as title words, question words and particle words, including existing predefined English and Thai stopwords.
- Duplicate tweet checking: Twitter users often share interesting tweets or edit small parts of the tweet then retweet. As a result, many tweets obtained from the Twitter search API are duplicated. Similar tweets are not useful for constructing our Twitter corpus. In this module, we detect and filter out the near-duplicate tweets by using Jaccard coefficient for similarity calculation.

2) Generating Cluster Labels

This component will search tweets on Twitter corpus by using a query and then return the top-N tweets results from the Twitter corpus. The tweet results are cluster labels generated by using the Carrot2 framework.

3) Label Merging

The main task of the last component is to take the clustered labels generated from the Carrot2 framework and then merge and create two-level cluster label structure. Each cluster label is converted from abbreviated form to full word form, because some users abbreviate their messages due to the limitation of tweet characteristics input quantity. We tag the Part-of-Speech (POS) of each word in the cluster labels, and then remove the cluster labels, which do not contain any noun, to filtering out some unmeaningful words. The objective of this algorithm is to improve the cluster merging algorithm by reducing cluster label size. For each cluster label, we apply the label merging algorithm in Figure 2 to generate the informative cluster labels. If a cluster label is a substring or partially overlapped with any other cluster label, it will be merged into the same cluster. The final output from this algorithm is a merged two-level cluster label structure.

The process of the proposed algorithm begins by taking the resulting cluster labels from the Carrot2 framework. The *length* method returns the number of words of each cluster label. The *substringOf* method checks whether the cluster label is a substring of another cluster label. The *overlapWith* method checks whether the cluster label is partially overlapped with another cluster labels.

Algorithm: Merge and Create Two-Level Cluster Label Structure

Input: $L = \{l_1, l_2, l_i, \ldots, l_N\}$, a set N cluster labels
Output: $L' = \{l'_1, l'_2, l'_i, \ldots, l'_M\}$, a set of M merged cluster labels

```
For all l_i in L do
    For all l_j in L and l_j ≠ l_i do
        If l_j subStringOf l_i
            then merge l_j into l_i
        End if
        Else if l_j overlapWith l_i
            then add l_j as a sub-level cluster label of l_i
        End if
    End for l_j
End for l_i
return L'
```

Figure 2. The pseudocode of Merge and Create Cluster Label Structure

STC-LM is implemented in Java with JDK6 by using Eclipse tool. The key idea of the proposed algorithm is to eliminate any substring and partially overlapped cluster labels in order to generate meaningful cluster labels. Another difference is our proposed algorithm allows cluster labels which are substring or partially overlapped cluster labels to form a second-level hierarchical structure. The second-level cluster labels within the same parent level contains tweets with the same topics. The hierarchical structure could provide users a more convenient way to browse the topics of interest.

TABLE I. CLUSTER LABEL RESULTS

Original Carrot2 Framework *(Baseline)*	**STC-LM Approach**
สู้น้ำท่วม (fight against flooding) น้ำท่วมขัง (retained flood) ช่วย (help) จ่ายเงินช่วยน้ำท่วม (donated money for flooding) ผู้ว่าฯ กทม. (bangkok governor) น้ำ (water) ช่วยน้ำท่วม (help flooding) ยิ่งลักษณ์ (Yingluck) ฟื้นฟู (flood relief)	- สู้น้ำท่วม (fight against flooding) --- น้ำท่วมขัง (retained flood) --- จ่ายเงินช่วยน้ำท่วม (donated money for flooding) --- ช่วยน้ำท่วม (help flooding) - ผู้ว่า กทม. (bangkok governor) - ยิ่งลักษณ์ (Yingluck) - ฟื้นฟู (flood relief)

IV. EXPERIMENTS AND DISCUSSION

To evaluate the effectiveness and efficiency of our approach, we collected tweets from many Twitter profile pages in Thailand such as *Thaiflood*, *Manager* and *FM995radio*. The Twitter corpus contains approximately 160,000 tweets beginning on November 2011 to January 2012. Using this corpus, we selected popular keywords under 12 topics such as *flooding*, *survival kits* and *oil prices*. Each tweet is cleaned and tokenized by using the Twitter post processing unit. The evaluation corpus with answered cluster labels (gold standard set) was prepared by human judges.

We apply the Carrot2 framework to perform all the experiments. We used the default setting of Carrot2 for generating cluster labels and defined the maximum final cluster

labels equal to 20. We compare cluster label results between our approach and the original Carrot2 by using precision, recall and F1 measure. Precision is the number of returned cluster labels that are relevant over the number of returned cluster labels. Recall is the number of returned cluster labels that are relevant over the number of relevant number of cluster labels. F1 measure is the harmonic mean of precision and recall. The experimental results are summarized in TABLE II.

TABLE II. CLUSTER LABEL EVALUATION RESULTS

Topics	Baseline			STC-LM		
	P	**R**	**F1**	**P**	**R**	**F1**
พ่อปลาบู่ *(plaboo's father)*	0.900	0.642	0.750	1.000	0.785	0.880
น้ำท่วม *(flood)*	0.700	0.538	0.608	0.615	0.615	0.615
งูแมมบ้า *(mambas)*	0.900	0.642	0.750	0.900	0.642	0.750
ศปภ. *(flood center)*	0.700	0.583	0.636	0.666	0.666	0.666
กระสอบทราย *(sandbag)*	0.900	0.692	0.782	0.909	0.769	0.833
อุงยังชีพ *(survival kits)*	0.600	0.461	0.521	0.777	0.538	0.636
บิ๊กแบ๊ก *(big bag)*	0.800	0.571	0.666	1.000	0.714	0.833
คันกั้นน้ำ *(dike)*	0.700	0.437	0.538	0.857	0.750	0.800
ซี7 ตบบ้องหู *(C7 assault)*	0.600	0.352	0.444	0.687	0.647	0.666
ราคาน้ำมัน *(oil prices)*	0.900	0.642	0.750	1.000	0.642	0.782
ปรับ ครม. *(cabinet shuffle)*	0.600	0.461	0.521	0.777	0.538	0.636
บริหารจัดการน้ำ *(water management)*	0.500	0.294	0.370	0.428	0.352	0.387
Average	**0.733**	**0.527**	**0.612**	**0.802**	**0.639**	**0.707**

From the table, we can conclude that the STC-LM yields better performance than the original Carrot2 framework based on the STC algorithm. Our approach is better than the baseline because some unmeaningful words are pruned by using the POS tagger. Also many substring and overlapping cluster labels are merged into one cluster label. However, clustering short texts is a difficult task and challenging. Since tweets are also considered as short texts, the clustering task of tweets is also a complex problem to be solved [6]. Due to the nature of writing style of each one is different such as informal writing style with many out of vocabulary words, abbreviation and teenage slang words.

V. CONCLUSION AND FUTURE WORKS

In this paper, we present a new approach called Suffix Tree Clustering with Label Merging (STC-LM) for generating the informative cluster labels. The main task of the STC-LM is to merge and then create two-level label structure. We performed experiments by using Thai Twitter posts from 12 topics such as Flooding, Traffic and Entertainment. Based on the experiments, our approach yielded an improved performance with the F1 measure of 70% and improvement of 9% from traditional STC based on the Carrot2 framework. The experimental results showed that our approach can improve the quality of cluster labels and organize cluster label results because it can automatically filtering out some unmeaningful words. In the future work, we will include tweet's timestamp as another parameter to cluster labels. The search results could be ranked by time, therefore the users could view the tweets with the most up-to-date ordering. Another future work is to incorporate probabilistic and machine learning model into our label merging algorithm.

REFERENCES

[1] H. Chim and X. Deng, "A New Sufix Tree Similarity Measure for Document Clustering," *Proc. Of the 16th International World Wide Web Conference (WWW '07)*, pp. 121–129, 2007.

[2] W. Han, X. Nan-Feng and C. Qiong, "Snippets Clustering Based on an Improved Suffix Tree Algorithm," *Proc. Of the 6th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '09)*, pp. 542–547, 2009.

[3] X. Han and J. Zhao, "Topc-Driven Web Search Result Organization by Leveraging Wikipedia Semantic Knowledge," *Proc. Of the 19th ACM International Conference on Information and Knowledge Management (CIKM '10)*, pp. 1749–1752, 2010.

[4] G. Hu, Q. W, Zuo. Chen, F. He and Y. Wang, "Semantic-based Hierarchicalize th Result of Suffic Tree Clustering," *Proc. Of the 2nd International Symposium on Knowledge Acquisition and Modeling (KAM '09)*, pp. 221–224, 2009.

[5] S. Osinski and D. Weiss, "A Concept-Driven Algorithm for Clustering Search Results," *IEEE Intelligent Systems*, pp. 48–54, 2005.

[6] F. Perez-Tellez and D. Pinto, "Om the Difficult of Clustering Company Tweets," *Proc. Of the 2nd International Workshop on Search and Mining User-generated Contents (SMUC '10)*, pp. 95–102, 2010.

[7] A. Rangrej, S. Kulkarni and A. V. Tendulkar, "Comparative Study of Clustering Techniques for Short Text Documents," *Proc. Of the 20th International World Wide Web Conference (WWW '11)*, pp. 111–112, 2011.

[8] J. Wan, W. Yu and X. Xu, "Suffix Tree Based Chinese Document Feature Extraction and Clustering in RSS Aggregator," *Proc. Of the 2nd Symposium International Computer Science and Computational Technology (ISCSCT '09)*, pp. 462–466, 2009.

[9] R. Yang, H. Xie and Q. Zhu, "Suffix Tree Based Chinese Document Feature Extraction and Clustering in RSS Aggregator," *Proc. Of the International Journal of Digital Content Technology and its Applications (JDCTA '11)*, pp. 346–354, 2011.

[10] O. Zamir and O. Etzioni, "Web Document Clustering: A Feasibikity Demostration," *Proc. Of the 19th International ACM SIGIR Conferenceon Research and Development in Information Retrieval (SIGIR '98)*, pp. 46–54, 1998.

[11] O. Zamir and O. Etzioni, "Grouper: A Dynamic Clustering Interface to Web Search Results," *Proc. Of the Eighth International World Wide Web Conference (WWW 8)*, 1999.

[12] H. Zeng, Q. He, Z. Chen, W. Ma and J. Ma, "Learning to Cluster Web Search Results," *Proc. Of the 27th Annual International ACM SIGIR Conference (SIGIR '04)*, pp. 210–217, 2004.

[13] "Carrot2", Available at: *http://project.carrot2.org*

[14] "LexTo", Available at: *http://www.sansarn.com/lexto/*

[15] "Topsy", Available at*: http://topsy.com*

[16] "Twitter", Available at: *http://twitter.com*

[17] "TweetScan", Available at: *http://tweetscan.com*

[18] "TwitterSearch", Available at: *https://twitter.com/#!/search-home*

[19] "Twitter4J", Available at: *http://twitter4j.org/en/index.html*

[20] "Topsy", Availableat: *http://topsy.com*