Expert Systems with Applications 38 (2011) 12807–12817

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Social trend tracking by time series based social tagging clustering

Shihn-Yuarn Chen [a], Tzu-Ting Tseng [b], Hao-Ren Ke [c,*], Chuen-Tsai Sun [a]

[a] Department of Computer Science, National Chiao Tung University, No. 1001 Ta Hsueh Road., Hsinchu 300, Taiwan
[b] Institute of Information Management, National Chiao Tung University, No. 1001 Ta Hsueh Road., Hsinchu 300, Taiwan
[c] Graduate Institute of Library & Information Studies, National Taiwan Normal University, No. 162, He-ping East Road, Section 1, Taipei 10610, Taiwan

ARTICLE INFO

Keywords:
Web 2.0
Social tagging
Time series clustering
Event tracking

ABSTRACT

Social tagging is widely practiced in the Web 2.0 era. Users can annotate useful or interesting Web resources with keywords for future reference. Social tagging also facilitates sharing of Web resources. This study reviews the chronological variation of social tagging data and tracks social trends by clustering tag time series. The data corpus in this study is collected from Hemidemi.com. A tag is represented in a time series form according to its annotating Web pages. Then time series clustering is applied to group tag time series with similar patterns and trends in the same time period. Finally, the similarities between clusters in different time periods are calculated to determine which clusters have similar themes, and the trend variation of a specific tag in different time periods is also analyzed. The evaluation shows the recommendation accuracy of the proposed approach is about 75%. Besides, the case discussion also proves the proposed approach can track the social trends.

1. Introduction

Social tagging has recently become a widely used application on the Internet. This process involves bookmarking part or all of a website for future reference. Social tagging can be used at a variety of websites, such as online shopping systems like Amazon.com, photo sharing communities like Flickr.com, and bookmarking services like Delicious.com. When someone finds something interesting online, he/she can tag it with some keywords. Tagging is very similar to bookmarking the entire page, and is similarly accessible.

Tagging also allows users to collaborate with other people online, including sharing collections and tag navigating. By sharing collections, a user can understand what other users bookmark and how others describe the same resource by various tags. Different resources tagged with the same word may refer to different subject matter, and this phenomenon can be found by navigating resources through one tag. For example, the tag "world-series" may highlight news reports regarding the 2009 World Series between the New York Yankees and the Philadelphia Phillies, but may also tag news reports about 2008 World Series between the Philadelphia Phillies and the Tampa Bay Rays. Tags can also be used to track news events. For example, news about Barack Obama's career as a senator to his presidential campaign and inauguration can be tagged simply "Obama."

This study analyzes social tagging information on time line, and each tag is represented by its tagging resources. Time series clustering is then applied to group tags with similar theme and find out the trends of events. In our example, there are five tags: 奧運 (Olympic Games), 中國 (China), 北京 (Beijing), 政治 (Politics) and 台灣 (Taiwan). Table 1 lists the usages of these tags in five sequential time points: p1, p2, p3, p4 and p5. Ignoring the chronological factor, traditional clustering algorithms group 奧運 (Olympic Games) and 中國 (China) in the same cluster, because of their similar usage count. However, according to Fig. 1, which depicts the usages of the tags at timeline, it is observably that 中國 (China), 政治 (Politics) and 台灣 (Taiwan) have similar polyline trends. Similar trends indicate these three tags have more similar theme than 奧運 (Olympic Games) and 北京 (Beijing), and these three tags should be grouped in the same cluster.

This study applies time series clustering to find out tags with similar trends. Based on clustering results, users can find related tags and documents in a particular time period. In addition, related documents from different time periods can be retrieved by calculating the similarities between clusters in different time periods.

The rest of this paper is organized as follows. Section 2 reviews previous studies on social tagging, time series analysis and clustering algorithms. Section 3 describes the proposed approach, covering data pre-processing, time series representation, time series clustering, and recommendation. Section 4 evaluates and compares the proposed approach and the counterpart approach that does not take into account the chronological factor. Section 5 concludes with future proposals.

* Corresponding author. Tel.: +886 2 77345203; fax: +886 3 5718925.
E-mail addresses: sychen@cs.nctu.edu.tw (S.-Y. Chen), baberainy@gmail.com (T.-T. Tseng), clavenke@ntnu.edu.tw, claven@lib.nctu.edu.tw (H.-R. Ke), ctsun@cs.nctu.edu.tw (C.-T. Sun).

0957-4174/$ - see front matter ∅ 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.073

Table 1
Tag usage example.

	p_1	p_2	p_3	p_4	p_5	Total
奧運	40	20	0	2	0	62
中國	8	15	22	12	10	67
北京	10	11	0	6	8	35
政治	5	10	20	10	8	53
台灣	6	9	19	8	10	52

2. Related works

2.1. Social tagging and folksonomy

"Folksonomy" is derived from the words "folks" and "taxonomy." It means a classification created by ordinary people. Vander Wal defined the term folksonomy as, "*... the result of personal free tagging of information and objects for one's own retrieval. Tagging is performed in a social environment (shared and open). Act of tagging is done by the person consuming the information.*" (Vander Wal, 2005) Folksonomy also includes collaborative classification, collaborative tagging, free tagging, tagsonomy, etc. Folksonomy emphasizes the spirits of social classification, collaboratively creation, and typically flat name-spaces.

Folksonomy consists of three aspects: user, resource, and classification (Fig. 2) (Pu, 2007). The user aspect involves social and collaborative concepts; the Resource aspect involves media information; the classification aspect defines the classification rules.

Social tagging is one type of folksonomy. Users can use tags, which are indicative keywords to annotate, describe or classify useful information. Flickr and Delicious.com are examples of websites which promote social tagging. Flickr is a photo sharing website where pictures can be tagged, and Delicious.com is a bookmark service provider which allows user to tag bookmarked URLs. In these instances, users are both consumers and contributors of tags, and these tags can be used for classification, indexing, searching and browsing content.

2.2. Clustering algorithm

There are various clustering algorithms which can be divided into five categories (Han & Kamber, 2001): partitioning methods (e.g.: *k*-means and fuzzy *c*-means), hierarchical methods (e.g.: agglomerative and divisive hierarchical clustering), density-based methods (e.g.: DBSCAN), grid-based methods (e.g.: STING) and model-based methods (e.g.: SOM). Clustering algorithms usually only process static data. Among the various clustering algorithms, the partitioning methods are most commonly used. A partitioning clustering method usually has to determine the number of clusters in advance, and then reduces the value of a goal function by iterative clustering computations. The halting condition of a partitioning clustering method is usually a threshold value of the goal function or a specific iteration count. For example, the *k*-means algorithm clusters data into *k* groups, and its goal function is the sum of square error between the centroid of a cluster and data items in the cluster.

2.2.1. Hierarchical clustering

This study uses hierarchical clustering to group time series data; this subsection introduces hierarchical clustering in greater detail. There are two types of hierarchical clustering: agglomerative (Voorhees, 1986) and divisive (Hastie, Tibshirani, & Friedman, 2009). Fig. 3 illustrates an example of hierarchical clustering. Agglomerative hierarchical clustering initially represents each data item as a cluster, and iteratively merges the two closest clusters till the halting constraint is satisfied. Divisive hierarchical clustering is different from agglomerative. Divisive method groups all data items in one group at beginning, and splits a cluster into two most distant clusters iteratively till the halting constraint is reached.

The criteria to decide cluster merging or splitting is the distance between clusters. The four ways to measure the distance between two clusters are single linkage, complete linkage, average linkage and Ward's distance (Ward, 1963).

I. Single linkage: Fig. 4(a) illustrates single linkage distance measurement, which only considers the shortest distance between two clusters. The distance is $D(C_i, C_j) = min\ d(a,b)$, where a belongs to cluster C_i, and b belongs to cluster C_j.
II. Complete linkage: Fig. 4(b) shows complete linkage distance, which considers the longest distance between two clusters. The distance is $D(C_i, C_j) = max\ d(a,b)$, where a belongs to cluster C_i, and b belongs to cluster C_j.
III. Average linkage: Fig. 4(c) displays average linkage, which considers the average distance between all data item pairs across two clusters. The distance is $D(C_i, C_j) = (\Sigma d(a,b))/(|C_i||C_j|)$, where a belongs to cluster C_i, and b belongs to cluster C_j.
IV. Ward's distance: Fig. 4(d) depicts Ward's distance; it finds out the centroid of two clusters first, and then calculates the square sum of distances between all data items and the centroid. The distance is $D(C_i, C_j) = (\Sigma|a - m|^2)$, where a belongs to $C_i \cup C_j$, and m is the centroid of C_i andC_j.

In addition to distance measurement of clusters, hierarchical clustering also has to consider the halting constraint before executing. The halting constraint is usually the cluster count or the average distance between clusters.

2.3. Time series analysis

A time series is a sequence of successive data measured at uniform time intervals (Box & Jenkins, 1976). Time series data is a set

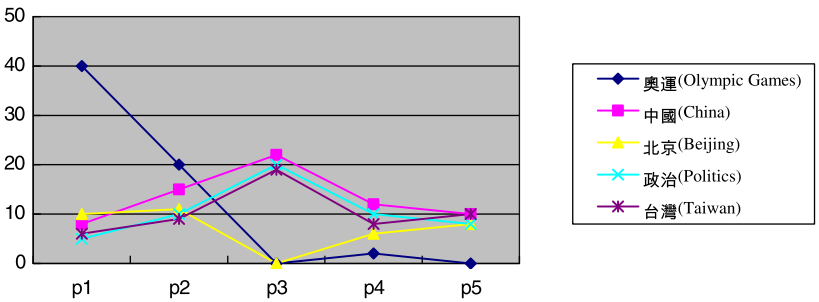

Fig. 1. Represent tags on time line.

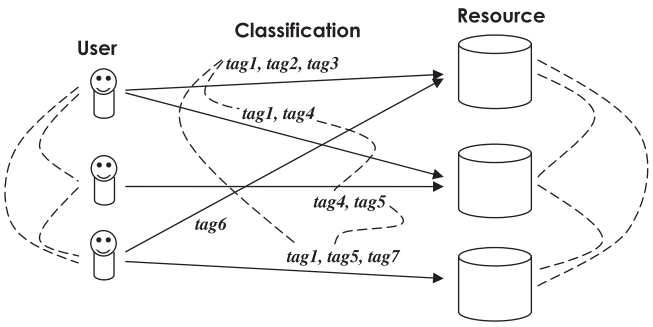

Fig. 2. Three respects of folksonomy (Pu, 2007).

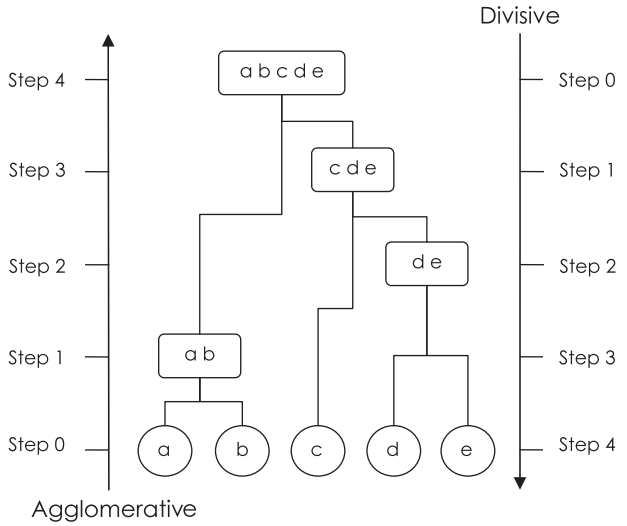

Fig. 3. Example of hierarchical clustering algorithm.

of values of an item's attribute in a particular time period. For example, the everyday market price of a company's stock in the first quarter 2009, and the weekly rainfall records of Taipei city in 2009. Time series analysis extracts statistics and other details from time series data. These statistics and details are helpful in forecasting the trends of future events.

This study is designed to cluster the time series data of social tags. However, time series data are chronological, and clustering algorithms are not proper to process non-static data. Before executing clustering, time series data should be transformed into a static form. The distance measurement between time series data is also essential, and some measurement methods are introduced as follows.

2.3.1. Euclidean distance

Euclidean distance is the simplest measurement between two time series data items. This method states a time series data of length N (i.e. N measured values on time line) as a data point in an N-dimension space. The similarity of two time series data items is the distance of each in the N-dimension space. However, Euclidean distance dose not afford for offset translation (Fig. 5(a)) or amplitude scaling (Fig. 5(b)) (Bollobás, Das, Gunopulos, & Mannila, 1997). Offset translation indicates that two time series are almost the same, except their amplitude offset. Amplitude scaling shows that two time series have similar trends, but one is the scaling of the other at certain time periods. For reducing the influence of offset translation and amplitude scaling, normalization is a solution. For example, Agrawal et al.'s approach (Agrawal, Lin, Sawhney, & Shim, 1995) normalizes every time series to a range $(-1,+1)$. After normalization, the Euclidean distance is calculated sequentially.

2.3.2. Dynamic time warping

Another issue presented is time series shifting (Fig. 5(c)), which indicates that two time series are similar but a delay time period exists between them. Euclidean distance and Agrawal et al. approach do not afford for measuring the similarity between time

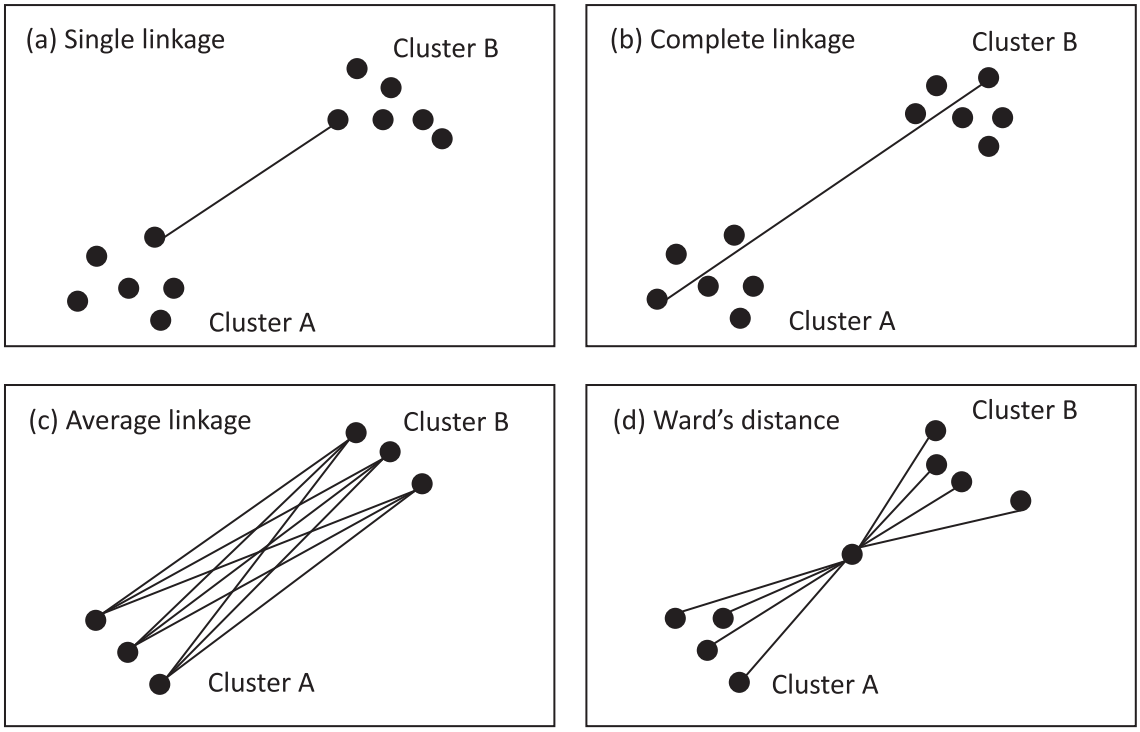

Fig. 4. Charts of four distance measure methods for hierarchical clustering.

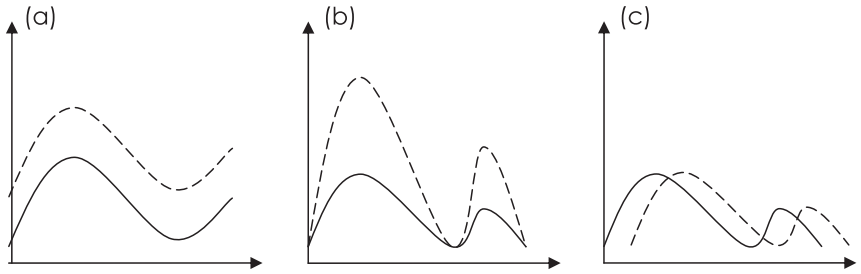

Fig. 5. Examples of offset translation, amplitude scaling and shifting of time series.

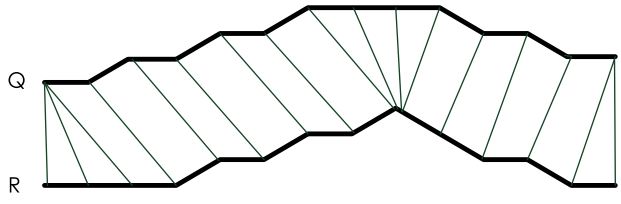

Fig. 6. Examples of dynamic time warping (DTW) (Oates et al., 1999; Salvador & Chan, 2007).

series with shifting. Dynamic time warping (DTW) is proposed to remedy this issue (Oates, Firoiu, & Cohen, 1999; Salvador & Chan, 2007). DTW allows referencing a time series data point consequently for various times while calculating the distance between two time series data, and Fig. 6 is an example.

For example, there are two time series, $Q = q_1, q_2, q_3, \ldots, q_n$ and $R = r_1, r_2, r_3, \ldots, r_m$. In order to minimize the distance between Q and R, DTW aligns Q and R by replicating certain data points. DTW generates a $n \times m$ matrix, M_{DTW}, to record the distances (e.g. Euclidean distance) between the data items q_i and r_j. Each warping path, W, is

$$\begin{aligned} &W = w1, w2, w3, \ldots, wk, \\ &\text{where} \quad \min(m,n) \leqslant K \leqslant (m+n-1), \\ &\quad w_k = M_{DTW}(i,j), w_1 = M_{DTW}(1,1), w_K = M_{DTW}(n,m). \end{aligned} \tag{1}$$

The minimum length of W is the minimum distance between Q and R, d_{DTW}, which can be calculated by dynamic programming (Liao, 2005).

$$d_{DTW} = \min \frac{\sum_{k-1}^{K} w_k}{K} = D(n,m), \tag{2}$$

$$D(i,j) = d(q_i, r_j) + \min \left\{ \begin{array}{c} D(i-1,j-1) \\ D(i-1,j) \\ D(i,j-1) \end{array} \right\}. \tag{3}$$

2.3.3. Longest common subsequence

Longest common subsequence (LCS) method finds the longest common subsequence in all sequences, and the similarity of two time series is the portion of the longest common subsequence and the original time series. However, LCS does not accommodate amplitude scaling and offset translation. Agrawal et al. (1995) proposed an approach to address these issues. Fig. 7 is an example to show their approach.

Agrawal et al.'s study details LCS time series analysis in three steps: **atomic matching**, **windows stitching** and **subsequence ordering**. The brief ideas of their approach are as follows. The first step is to define the gaps between the time series Q and R, and remove them. Second, align the time series to eliminate any shifting issues. The third step adjusts the time series to eliminate amplitude scaling and offset translation. Indicating the longest common subsequence of both time series is the final step.

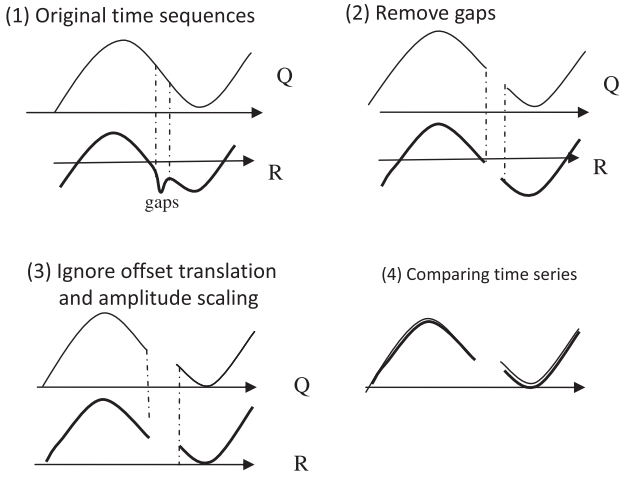

Fig. 7. Examples of longest common subsequence (LCS) (Agrawal et al., 1995).

3. Time series based social tagging clustering

3.1. Dataset and preprocess

This study focuses on social tagging in a traditional Chinese environment. The data is collected from Hemidemi.com, one of the largest traditional Chinese social bookmarking service providers. Hemidemi.com (Fig. 8) records the URL and title of a web page, when it was added (create date), and which tags were assigned by an individual user. The collected data includes 3842 distinct URLs, which were saved on Hemidemi.com from 2008/1/1 to 2008/12/31. Our information additionally contains the titles of these URLs, creation dates, and 2707 distinct tags which annotate these URLs. Besides, web page contents of these URLs are also crawled, and most of them are in traditional Chinese. The corpus covers various domains, such as sports, movie, cuisine, traveling, and politics.

This study uses CKIP (Chinese Knowledge Information Processing)[1] to preprocess the contents of the crawled web pages. CKIP tokenizes traditional Chinese documents into phrases and labels proper part-of-speech; Fig. 9 shows an example of CKIP results. After CKIP processing, this study only keeps nouns and verbs as feature candidates, as shown in Fig. 10.

However, some nouns and verbs do not efficiently represent information, and may hurt the clustering accuracy. These "stop words" can be removed by various ways, one of the simplest way is referring stop-word lists. This study uses Oracle Text Reference

[1] http://ckipsvr.iis.sinica.edu.tw/.

Fig. 8. Screenshot of Hemidemi.com.

《(PARENTHESISCATEGORY) 貧民(N) 百萬富翁(N) 》(PARENTHESISCATEGORY) 的(T) 拍攝(Nv) 地點(N) 和(C) 故事(N) 背景(N) 設(Vt) 於(P) 印度(N) ，(COMMACATEGORY)</sentence><sentence> ?(QUESTIONCATEGORY) 述(Vt) 一(DET) 名(M) 來自(Vt) 貧民窟(N) 的(T) 青年(N) ，(COMMACATEGORY)</sentence><sentence> 到(P) 孟買(N) 參與(Vt) 遊戲(N) 問答(N) 節目(N) 《(PARENTHESISCATEGORY) 百萬富翁(N) 》(PARENTHESISCATEGORY) ，(COMMACATEGORY)</sentence><sentence> 當中(N) 過程(N) 非常(ADV) 順利(Vi) ，(COMMACATEGORY)</sentence><sentence> 他(N) 答對(Vt) 了(Di) 一(DET) 條(M) 又(ADV) 一(DET) 條(M) 的(T) 難題(N) ，(COMMACATEGORY)</sentence><sentence> 卻(ADV) 被(P) 節目(N) 主持(Vt) 及(C) 警方(N) 懷疑(Vt) 其(DET) 作弊(Vi) 。(PERIODCATEGORY)</sentence><sentence> 之後(N) 在(P) 警方(N) 的(T) 拷問(Vt) 下(POST) ，(COMMACATEGORY)</sentence><sentence> 他(N) 道出(Vt) 一(DET) 段段(DET) 與(C) 題目(N) 相關(Vi) 的(T) 往事(N) 。(PERIODCATEGORY)

Fig. 9. Example of CKIP result.

貧民(N) 百萬富翁(N) 拍攝(Nv) 地點(N) 故事(N) 背景(N) 設(Vt) 印度(N) 述(Vt) 來自(Vt) 貧民窟(N) 青年(N) 孟買(N) 參與(Vt) 遊戲(N) 問答(N) 節目(N) 百萬富翁(N) 當中(N) 過程(N) 順利(Vi) 他(N) 答對(Vt) 難題(N) 節目(N) 主持(Vt) 警方(N) 懷疑(Vt) 作弊(Vi) 之後(N) 警方(N) 拷問(Vt) 他(N) 道出(Vt) 題目(N) 相關(Vi) 往事(N)

Fig. 10. Keep nouns and verbs of CKIP result.

Chinese stoplist[2] and Word List with Accumulated Word Frequency in Sinica Corpus 3.0[3] to remove these high frequent and low representative words and phrases.

3.2. Feature selection

This study uses the vector space model (VSM) to represent web pages and tags. Each term produced by data preprocessing is a dimension in the vector space. However, the enormous dimension size increases computation time and may deteriorate the clustering accuracy. In order to reduce the computation time and increase the clustering accuracy, this study applies three rules to remove insignificant and unrepresentative features.

1. Remove terms which do not appear in more than three web pages.
2. Remove terms which appear in more than 5% web pages.
3. In each web page, a term which appears only once is removed.

Once the three rules have been completed, Log Likelihood Ratio (LLR) (Lehmann, 1986; Neyman & Pearson, 1967) is applied to determine the features of a document. LLR is a statistical and

[2] http://download.oracle.com/docs/cd/B19306_01/text.102/b14218/astopsup.htm#sthref2545.
[3] http://www.aclclp.org.tw/doc/wlawf_abstract.pdf.

Table 2
Occurrence distribution of term ($term_i$) and document (d_x).

	d_x	$\overline{d_x}$
$term_i$	O_{11}	O_{12}
$\overline{term_i}$	O_{21}	O_{22}

probabilistic method, which tests the probabilities of two hypothesizes (null and alternative hypothesis), and determines which one is more possible to happen. In this study, the null hypothesis (H_1) states that the distribution of a term ($term_i$) occurring in a web page (d_x) is the same as other terms in d_x. The alternative hypothesis (H_2) presumes that the distribution of $term_i$ in d_x is different to other terms in d_x. The formulas for H_1 and H_2 are as follows, and the occurrence distribution of $term_i$ and d_x is shown in Table 2.

$$H_1 : P(term_i|d_x) = p = P(term_i|\overline{d_x}), \tag{4}$$

$$H_2 : P(term_i|d_x) = p_1 \neq p_2 = P(term_i|\overline{d_x}), \tag{5}$$

$$\begin{aligned} p &= P(term_i|d_x) = P(term_i|\overline{d_x}) = P(term_i),\\ p_1 &= \frac{P(term_i \cap d_x)}{P(d_x)},\\ p_2 &= \frac{P(term_i \cap \overline{d_x})}{P(\overline{d_x})}. \end{aligned} \tag{6}$$

O_{11} is the frequency of $term_i$ appearing in d_x, O_{12} is the frequency of $term_i$ appearing in web pages other than d_x, O_{21} is the frequency of terms other than $term_i$ appearing in d_x; O_{22} is the frequency of terms other than $term_i$ appearing in web pages except d_x. This study assumes the probability distribution is binomial distribution, as Eq. (7)

$$b(k;n,x) = (n)x^k(1-x)^{(n-k)}. \tag{7}$$

Then, *H1* and *H2* can be represented as Eq. (8).

$$\begin{aligned} L(H_1) &= b(O_{11};O_{11}+O_{12},p)b(O_{21;}O_{21}+O_{22},p),\\ L(H_2) &= b(O_{11};O_{11}+O_{12},p_1)b(O_{21;}O_{21}+O_{22},p_2). \end{aligned} \tag{8}$$

The Log Likelihood Ratio value, $-2\log\lambda$, can be calculated by using Eq. (9).

$$\begin{aligned} -2\log\lambda &= -2\log\frac{L(H_1)}{L(H_2)}\\ &= -2\log\frac{b(O_{11};O_{11}+O_{12},p)b(O_{21;}O_{21}+O_{22},p)}{b(O_{11};O_{11}+O_{12},p_1)b(O_{21;}O_{21}+O_{22},p_2)}\\ &= -2((O_{11}+O_{21})\log p + (O_{12}+O_{22})\log(1-p)\\ &\quad -(O_{11}\log p_1 + O_{12}\log(1-p_1)\\ &\quad + O_{21}\log p_2 + O_{22}\log(1-p_2)). \end{aligned} \tag{9}$$

Koller et al. believed that, in hierarchical clustering, the appropriate amount of features in a document ranges from 10 to 20 (Koller & Sahami, 1997). Furthermore, too many features may decrease the coherence between features and documents, and increase noises during clustering (Chang & Hsu, 2005). This study chooses at most 50 terms with the highest LLR in each document as features. After feature selection, the feature amount in the corpus is reduced from 1,760,840 (123,830 distinct) to 402,319 (20,371 distinct).

3.3. Tag representation

In the vector space model, a document d_x is represented as $d_x = \{w_{x1}, w_{x2}, w_{x3}, \ldots, w_{xn}\}$ where w_{xi} is the weight of $term_i$ in d_x. This study chooses TFIDF to calculate the weight of each term in a document. In social tagging, a tag is used to annotate one or more documents, so this study uses annotated documents to represent a tag, tag_j. Suppose tag_j annotates document d_x on date p, then tag_j can be represented as $tag_{j,p} = tag_{j,p,x} = d_x = \{w_{x1}, w_{x2}, w_{x3}, \ldots, w_{xn}\}$ on date p. If tag_j annotates two documents (d_x and d_y) on date p, then tag_j can be represented as $tag_{j,p} = tag_{j,p,x} + tag_{j,p,y} = d_x + d_y = \{w_{x1} + w_{y1}, w_{x2} + w_{y2}, w_{x3} + w_{y3}, \ldots, w_{xn} + w_{yn}\}$. The formal representation of tag_j on date p is shown in Eq. (10), where $W_{pk} = \sum_{x=1}^{q} w_{xk}$; q is the number of documents annotated by tag_j on date p.

$$tag_{j,p} = tag_{j,p,1} + tag_{j,p,2} + \cdots + tag_{j,p,q} = \{W_{p1}, W_{p2}, \ldots, W_{pk}\}. \tag{10}$$

3.4. Tag time series representation

This study normalizes each tag first in order to avoid offset translation and amplitude scaling. The normalization formula is shown as Eq. (11).

$$tag_{j,p} = \left\{ \frac{W_{p1}}{\sqrt{\sum_{k=1}^{n} W_{pk}^2}}, \ldots, \frac{W_{pm}}{\sqrt{\sum_{k=1}^{n} W_{pk}^2}} \right\} \tag{11}$$

$$v_{j,p} = tag_{j,p+1} - tag_{j,p} = \{W_{v_{jp},1}, W_{v_{jp},2}, \ldots, W_{v_{jp},k}\}. \tag{12}$$

The time series of a tag, tag_j, is the union of consecutive time segments of the tag. Each time segment $v_{j,p}$ is the difference between $tag_{j,p+1}$ and $tag_{j,p}$ (Eq. (12)). According to (Van Wijk & Van Selow, 1999), time series data is the sequence of N data pairs, $v_i = (y_i, t_i)$, where $i = 1, 2, 3, \ldots, N$, and y_i is the value of time t_i. The time line can be split into M time periods. $V_{j,m}$ represents the time series of tag_j in time period m, where $m = 1, 2, 3, \ldots, M$. Each $V_{j,m}$ contains N consecutive data pairs, $v_p = (v_{j,p}, t_p)$, where $p = 1, 2, 3, \ldots, N$. This study splits the whole time line (2008/1/1 ~2008/12/31) every two weeks, so that there are 26 time periods, and 14 consecutive data pairs in each time period.

3.5. Time series similarity

This study uses cosine similarity to compute the similarity between two tag time series in the same time period. Suppose tag_i and tag_j on the time line. The time series of tag_i and tag_j in period m are $V_{i,m} = \{v_{i,1}, v_{i,2}, v_{i,3}, \ldots, v_{i,N}\}$ and $V_{j,m} = \{v_{j,1}, v_{j,2}, v_{j,3}, \ldots, v_{j,N}\}$, respectively. The similarity of tag_i and tag_jin time period m, $sim(tag_i, tag_j)$, is calculated in Eq. (13)

$$\begin{aligned} sim(tag_i, tag_j) = (&similarity(v_{i,1}, v_{j,1}) + \cdots\\ &+ similarity(v_{i,N}, v_{j,N}))/N, \end{aligned} \tag{13}$$

$$similarity(v_{i,p}, v_{j,p}) = \frac{\sum_{k=1}^{n}(W_{v_{ip},k} \times W_{v_{jp},k})}{\sqrt{\sum_{k=1}^{n} W_{v_{ip},k}^2} \times \sqrt{\sum_{k=1}^{n} W_{v_{jp},k}^2}}. \tag{14}$$

Sometimes, the two time series shift. In Fig. 11, the solid and dashed time series are obviously similar, but they are shifted. Considering this issue, this study calculates the similarity of two tag time series by moving one backward and forward 1–4 days artificially. The highest similarity value is the shifting similarity, *sf*. The final similarity of two tag time series is the linear combination with a weighted parameter w (0.5 in this study) of *sf* and the similarity them without shifting (Eq. (15)).

$$sim''(tag_i, tag_j) = w \times sim'(tag_i, tag_j) + (1-w) \times sf(tag_i, tag_j). \tag{15}$$

The similarity between the two time series, $sim'(tag_i, tag_j)$, is between −1 and 1. The negative value means the two tag time series (tag_i and tag_j) have different trend in a period of time. For example,

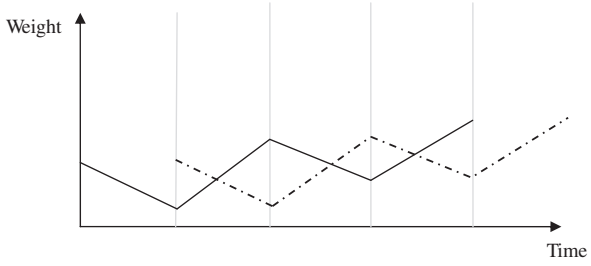

Fig. 11. Time series shifting example.

Table 3
Distribution of positive similarity pairs of time series in corpus.

Similarity interval	# of pairs
0.5 ~ 0.851	1,079
1.71E−02 ~ 0.5	17,888
3.41E−03 ~ 1.71E−02	19,713
2.33E−03 ~ 3.41E−03	9,221
6.81E−04 ~ 2.33E−03	43,557
1.36E−04 ~ 6.81E−04	49,478
2.72E−05 ~ 1.36E−04	15,461
0.0 ~ 2.72E−05	4,101

tag_i is seldomly used on date p, but tag_j is used more often. This is due to two reasons. First, the documents annotated by tag_j on date p are not relevant to tag_i. The other, although the documents annotated by tag_j on date p are relevant to tag_i, users seldom use tag_i to annotate these documents. Unfortunately, it takes time and efforts to judge the actual reason, so this study only considers the positive similarity and the negative values are set to 0. In the collected corpus, the amount of similarity of time series is 581,423, and 420,925 of them are negative. Out of the 160,498 positive similarity pairs of time series, the average is 0.00233, and the distribution is listed in Table 3.

3.6. Time series clustering

This study applies agglomerative hierarchical clustering algorithm to cluster time series and uses average linkage (Fig. 4(c)) for calculating the distance between clusters. The detailed steps are as follows:

1. For each time period m ($m = 1, 2, 3, \ldots, M$), every tag time series is treated as a cluster.
2. Calculate the average distance between cluster pairs (Eq. (16)), where $|C_i|$ is the size of cluster C_i, $d(a,b)$ is calculated by Eq. (15).

$$D_{avg}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{a \in C_i, b \in C_j} d(a,b). \quad (16)$$

3. Find the largest $D_{avg}(C_i, C_j)$, and merge C_i and C_j.
4. Iteratively execute steps 2 and 3, till reaching the halting constraint. The halting constraint is that the average distance of inter-clusters is less than the average distance between all tag time series in the period m.
5. Go back to step 1, and choose next m.

3.7. Recommendation

After time clustering, the time series in the same cluster have similar concept and similar trends. This study uses a mechanism to recommend relevant documents in the same time period and recommend relevant clusters across different time periods.

I. Recommend relevant documents in the same time period.
   (a) Recommending documents relevant to a cluster.
   The clustering result can be used to suggest similar documents to users for further reading. However, there are many documents annotated by tags in the same cluster, so the reasonable approach is to recommend the most relevant documents to users. Cosine similarity calculates the similarity between the cluster centroid and each document. Then suggest top n documents with the highest similarity to users. The cluster centroid $\overline{C_i}$ is calculated in Eq. (17), where $|C_i|$ is the cluster size.

$$\overline{C_i} = \sum_{j=1}^{|C_i|} \frac{V_{j,m}}{|C_i|} \quad (17)$$

   (b) Recommending documents relevant to multiple tags in a cluster.
   Sometimes, tag_i and tag_j are clustered together in time period m, but there is no overlap between documents annotated by tag_i and documents annotated by tag_j. This is due to users' tagging behavior patterns, not indicates that tag_i and tag_j are not relevant. In order to recommend

Table 4
Distribution of positive similarity pairs of time series in corpus.

The minimum tag count in a cluster	Hierarchical clustering (not consider the chronological factor)			Time series clustering		
	Cmp	Sep	Qcq	Cmp	Sep	Ocq
1	0.2866	0.0044	0.1455	0.2730	0.0029	0.1380
3	0.3704	0.0074	0.1889	0.3645	0.0058	0.1852
4	0.4023	0.0077	0.2050	0.4055	0.0061	0.2058

Table 5
Distribution of positive similarity pairs of time series in corpus.

		Expert A		
		No	Yes	Total
Expert B	No	53(21.2%)	22(8.8%)	75(30%)
	Yes	21(8.4%)	154(61.6%)	175(70%)
Total		74(29.6%)	176(70.4%)	250(100%)

Table 6
Distribution of positive similarity pairs of time series in corpus.

Kappa	Strength of agreement
0.00	Poor
0.01–0.20	Slight
0.21–0.40	Fair
0.41–0.60	Moderate
0.61–0.80	Substantial
0.81–1.00	Almost perfect

Table 7
Distribution of positive similarity pairs of time series in corpus.

		Experts' labeling	
		Y	N
Clustering	Y	130	37
Result	N	13	27
Total		143	64

documents which are most relevant to tag_i and tag_j, the time series of tag_i and tag_j in time period m are merged into $V_{ij,m}$ ($V_{ij,m} = V_{i,m} + V_{j,m}$). The similarity between $V_{ij,m}$ and each document annotated either by tag_i or tag_j is calculated by cosine similarity. The most similar documents are then suggested.

II. Recommend relevant clusters across different time periods.
This study groups tag time series in the same time period together. There may be relevant clusters in different time periods. This study retrieves relevant clusters from different time periods according to cosine similarity between two clusters. The relevant degree is $sim(\overline{C_i}, \overline{C_j})$, where $\overline{C_i}$ and $\overline{C_j}$ are the centroids of cluster i and j; and clusters i and j belong to different time periods. If the similarity is larger than a threshold (0.07 in this study), the two clusters are relevant.

4. Evaluation and case discussion

This section compares the proposed time series clustering approach, which produces 1225 clusters, and the hierarchical clustering without considering the chronological factor, which produces 1161 clusters. Besides, some clustering result cases are also discussed to show the proposed approach can find out the trends of events.

4.1. Quantification analysis

4.1.1. Clustering quality

Clustering is an unsupervised method to group similar data items together. The clustering quality depends on the in-cluster similarity and separation degree between clusters. The common ways to evaluate the quality of clustering are as follows: cluster compactness, cluster separation, and overall cluster quality (He et al., 2003).

I. Cluster compactness.
The cluster compactness, *Cmp*, is shown in Eq. (22), where $v(X)$ is the variance of all documents, and $v(c_i)$ is the variance of documents in a cluster. When the value of *Cmp* is smaller, the clusters are more compact.

$$v(X) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d^2(x_i, \bar{x})} \tag{18}$$

$$\bar{x} = \frac{1}{N}\sum_i x_i, \tag{19}$$

$$d(x_i, x_j) = 1 - \cos(x_i, x_j), \tag{20}$$

$$v(c_i) = \sqrt{\frac{1}{|c_i|}\sum_{j=1}^{|c_i|} d^2(c_{ij}, \overline{c_i})}, \tag{21}$$

$$Cmp = \frac{1}{C}\sum_i^C \frac{v(c_i)}{v(X)}. \tag{22}$$

II. Cluster separation.
The formula of cluster separation, *Sep*, is displayed in Eq. (23), where $\sigma$is the Gaussian Constant, C is the number of clusters, and $d(x_{ci}, x_{cj})$ is the distance between cluster c_i and c_j. *Sep* is valued between 0 and 1. When *Sep* has a smaller value, the clusters separate better.

$$Sep = \frac{1}{C(C-1)}\sum_{i=1}^{C}\sum_{j=1, j\neq i}^{C} \exp\left(-\frac{d^2(x_{c_i}, x_{c_j})}{2\sigma^2}\right). \tag{23}$$

III. Overall cluster quality.
Overall cluster quality, *Ocq*, is the linear combination of cluster compactness and cluster separation with a parameter β (0.5 in this study). The value of β is between 0 and 1. If the value of *Ocq* is smaller, the overall cluster quality is better.

$$Ocq(\beta) = \beta \cdot Cmp + (1-\beta)\cdot Sep. \tag{24}$$

The number of tags in each cluster may affect the comparison of *Cmp*, *Sep* and *Ocq*. Different settings of minimum tag count in a cluster are applied in this evaluation. Table 4 shows the values of *Cmp*, *Sep* and *Ocq* in different settings. The *Cmp* and *Ocq* values indicate that both approaches have similar cluster compactness and overall cluster quality. However, the *Sep* values of the proposed time series clustering are significantly better (>10%) than traditional hierarchical clustering.

4.1.2. Quality of relevant cluster recommendation

Before evaluating the quality of recommendation, this study removes 505 clusters, which contain less than three tags, and 720 clusters are left. 250 cluster pairs are then randomly chosen, and two computer science experts are asked to evaluate whether each cluster pair is similar or not. Table 5 lists the results of evaluation. "Yes" indicates that the expert determines the cluster pair is similar, and "No" indicates dissimilar. The Kappa[4] value of the evaluation is 0.589. According to Table 6, the strength of agreement is moderate.

$Kappa = (observed\ \ agreement - chance\ \ agreement)/(1 - chance\ agreement)$,
$observed\ agreement = (53 + 154)/250 = 0.828$,
$chance\ agreement = 0.296 \times 0.3 + 0.704 \times 0.7 = 0.5816$,
$Kappa = (0.828 - 0.5816)/(1 - 0.5816) = 0.589$.

According to Table 5, there are 207 (154 + 53) agreement cluster pairs. These 207 pairs are used to evaluate the clustering accuracy. Table 7 lists the result, and the sensitivity, specificity and accuracy of clustering (Han & Kamber, 2001) are as follows.

$$sensitivity = 130/143 = 0.909,$$
$$specificity = 27/64 = 0.422,$$
$$accuracy = 0.909 \times \frac{143}{207} + 0.422 \times \frac{64}{207} \cong 0.758.$$

4.2. Case discussion

4.2.1. Case of event trend on time line

This subsection uses the tag, 電影 (movie), to show that the proposed time series clustering approach can diagram the trend of events on the timeline. From 2008/5/6 to 2008/5/20, 鋼鐵人 (iron man) is the most relevant tag to電影 (movie), which coincides with the release of the movie in Taiwan (Fig. 12). From 2008/7/15 to 2008/7/29, the movie, 海角七號 (Cape 7), was released, and the tag 海角七號 (Cape 7) and 電影 (movie) are clustered in the same group (Fig. 13). However, the movie 海角七號 (Cape 7) does not lead an upsurge at the first few days after releasing, and the tag 海角七號 (Cape 7) does not increase in usage, too. The tags most relevant to 電影 (movie) are 瓦力 (Wall-E) and 動畫 (animation) between 2008/7/29 to 2008/8/12 (Fig. 14). After a few weeks, the tag海角七號 (Cape 7) increased in use and is clustered together with 電影 (movie) and 魏德盛 (the director of the movie), shown in Fig. 15. This movie also impulses the traveling fever in Taiwan, which causes 電影 (movie) and 海角七號 (Cape 7) are grouped to-

[4] http://www.dmi.columbia.edu/homepages/chuangj/kappa.

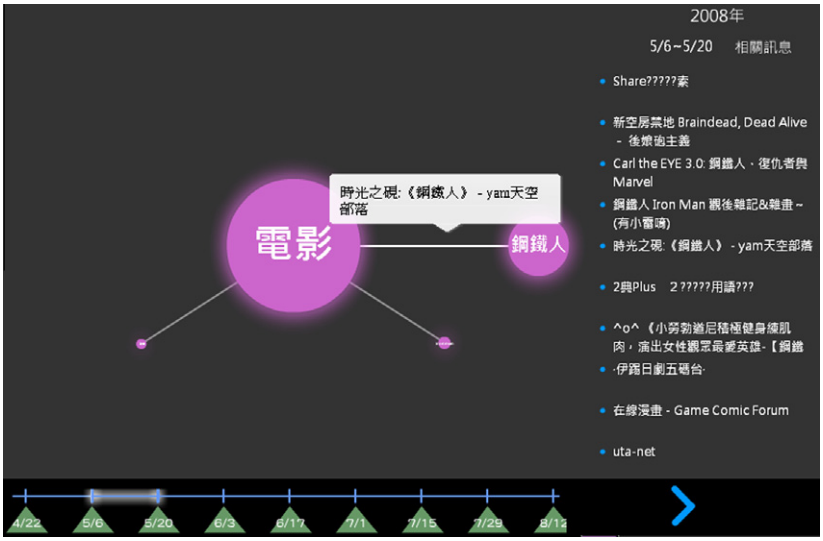

Fig. 12. Related tags of "電影" (movies) during 2008/5/6 ~ 2008/5/20.

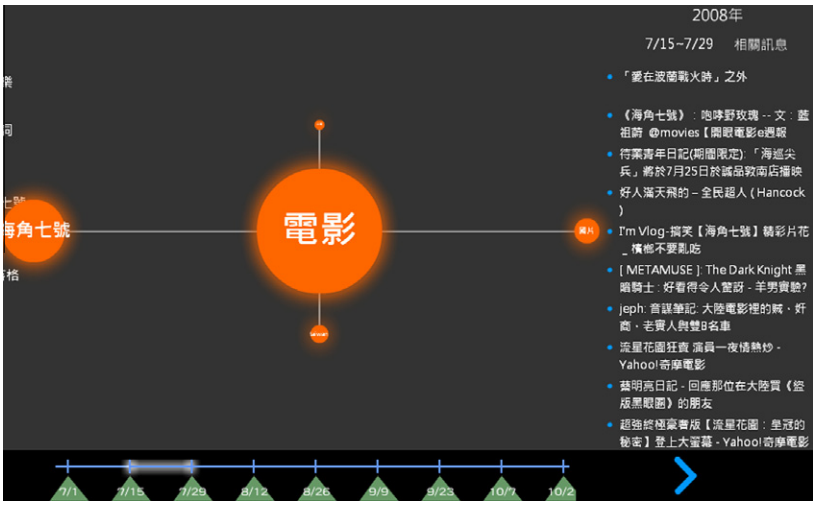

Fig. 13. Related tags of "電影" (movies) during 2008/7/15 ~ 2008/7/29.

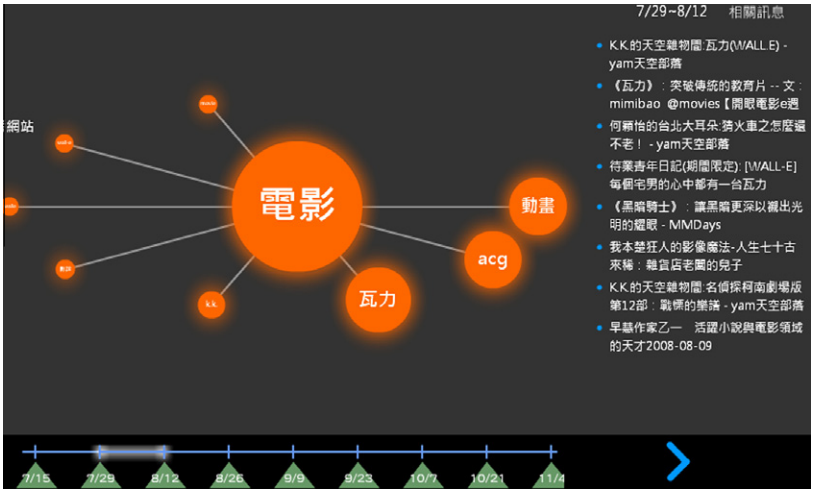

Fig. 14. Related tags of "電影" (movies) during 2008/7/29 ~ 2008/8/12.

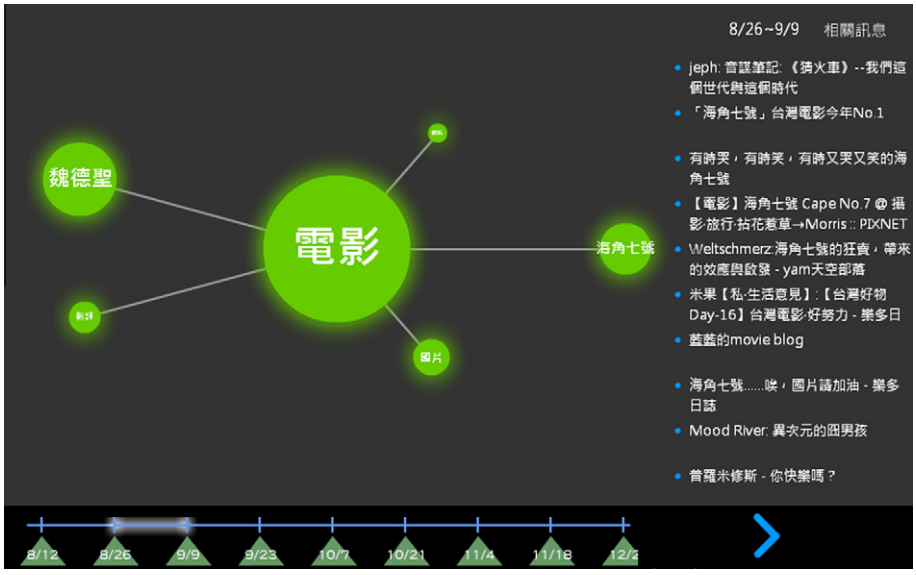

Fig. 15. Related tags of "電影" (movies) during 2008/8/26 ~ 2008/9/9.

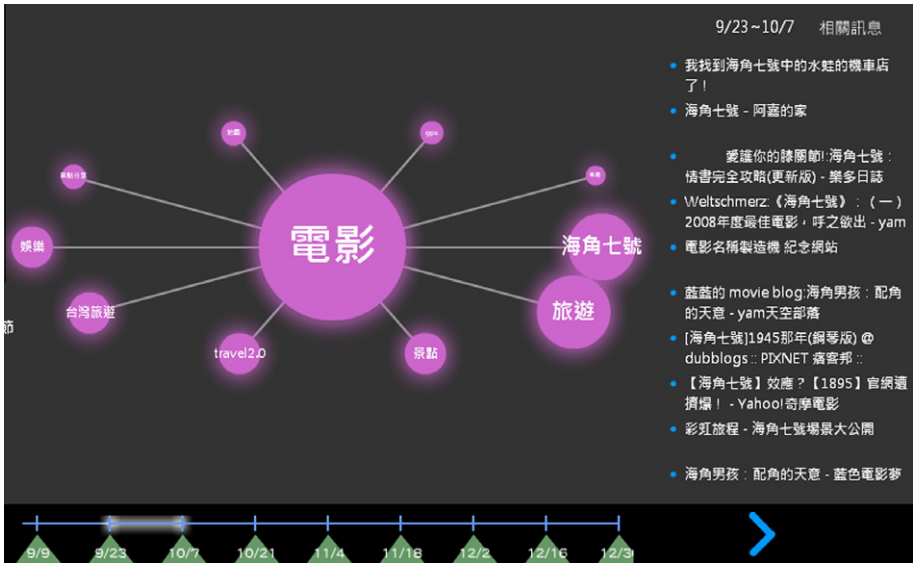

Fig. 16. Related tags of "電影" (movies) during 2008/9/23 ~ 2008/10/7.

gether with 旅遊 (traveling) and 墾丁 (Kenting, the main filming area in the movie), as shown in Fig. 16.

This proves that the proposed time series clustering approach can detect the societal trends by dividing the timeline and performing clustering in each period.

4.2.2. Clustering with and without the chronological factor

The previous Subsection 4.2.1 describes the advantage of clustering data by time period. This subsection shows the advantage of clustering with considering the chronological factor.

In the corpus, tags, including 中國 (China), 奧運 (Olympic Games), 北京奧運 (Beijing Olympic Games), BBC, 台灣 (Taiwan), 政治"(Politics), 新聞自由 (News Freedom), etc., are used between 2008/7/29 to 2008/8/12.

Table 8 lists the similarity values between 中國 (China) and other tags in this time period. If the chronological factor is not taken into account, 中國 (China), 奧運 (Olympic Games), BBC and 新聞自由 (News Freedom) are in the same cluster, 台灣 (Taiwan) and 政治 (Politics) are in another cluster, and 北京奧運 (Beijing Olympic Games) is in yet another (as illustrated in Fig. 17(a)). The proposed time series clustering approach clusters 中國 (China), 台灣 (Taiwan) and 政治 (Politics) in the same group, BBC, 新聞自由 (News Freedom) and 奧運 (Olympic Games) in another, and 北京奧運 (Beijing Olympic Games) in yet another (as show in Fig. 17(b)).

Table 8
Similarity values between 中國 (China) and other tags during 2008/7/29 to 2008/8/12.

Similarity (without the chronological factor)		Similarity (time series clustering)	
奧運	0.729	台灣	0.316
bbc	0.671	政治	0.270
新聞自由	0.563	奧運	0.265
台灣	0.547	Bbc	0.205
政治	0.394	北京奧運	0.118
北京奧運	0.356	新聞自由	0.052

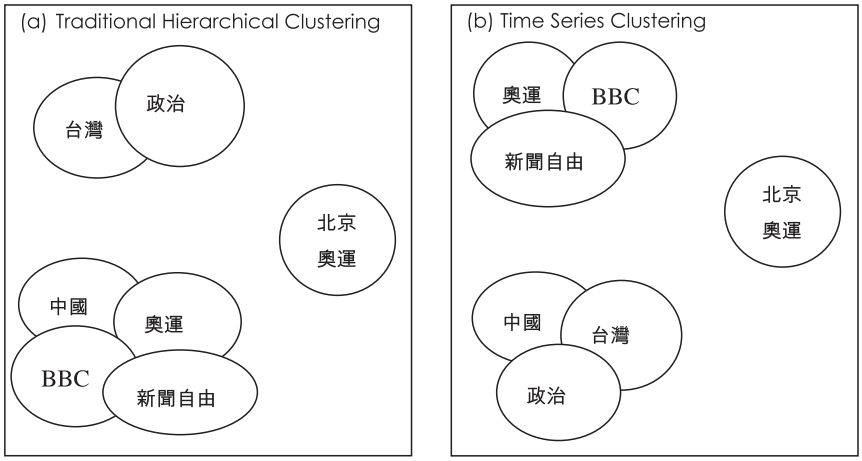

Fig. 17. Example of time series clustering result and traditional hierarchical clustering result.

If users only search under the cluster, 台灣 (Taiwan) and 政治 (Politics), in Fig. 17(a), they may misconceive that documents in this cluster are only related to political events in Taiwan. However, in this time period, there are also many documents related to political issues in China and across the strait. Users can easily miss these articles if they just browse the clustering result as shown in Fig. 17(a).

When the chronological factor is considered, like in Fig. 17(b), the proposed approach groups 中國 (China), 台灣 (Taiwan) and 政治 (Politics) together. This cluster would contain documents related to political events in Taiwan and in China.

5. Conclusion and future work

This study collects data from Hemidemi.com, and considers the chronological factor to represent each tag as time series by the vector space model. The corpus covers data created in 2008, including 3842 distinct web pages and 2707 distinct tags. This study divides the timeline into 26 periods, where each period is two weeks. The proposed approach produces 720 clusters by uses agglomerative hierarchical clustering with average linkage distance measurement. Cluster compactness, cluster separation and overall cluster quality are used to evaluate the proposed approach and traditional hierarchical clustering without considering the chronological factor. The evaluation results indicate that the proposed approach has similar qualities in cluster compactness and overall cluster quality measurements, and improves cluster separation significantly (>10%). The data is clustered periodically which allows for tracking societal trends. When considering the chronological factor, time series clustering is more precise than traditional hierarchical clustering in identifying the events in a time period. The proposed approach can also recommend relevant documents and clusters to users. The accuracy of these recommendations is around 0.758.

There are still some issues that need improvement. First, there can be irrelevant information in a web page. For example, a web page which introduces the movie "Iron Man" may contain information that is irrelevant to the movie, such as other movies released during same week. The second issue is the consistency of tags. Social bookmarking, a collection of folks' creation, is user designated, and is not under any straight set of rules or authority control. If an ontology can be created to identify different tags with similar concepts like "Web 2.0" and "Web2", clustering accuracy and quality would improve. The classification of tags is the last issue. Classifying tags can enable users to track the trends of a classification with a broader view, instead of simply tracking a tag.

References

Agrawal, R., Lin, K. I., Sawhney, H. S., & Shim, K. (1995). Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceedings of the 21st international conference on very large data bases, Zurich, Switzerland* (pp. 490–501).

Bollobás, B., Das, G., Gunopulos, D., & Mannila, H. (1997). Time-series similarity problems and well-separated geometric sets. In *Proceedings of the 13th annual symposium on computational geometry* (pp. 454–456).

Box, G., & Jenkins, G. (1976). *Time series analysis: Forecasting and control*. Oakland, California: Holden-Day.

Chang, H.-C., & Hsu, C.-C. (2005). Using topic keyword clusters for automatic document clustering. In *Third international conference on information technology and applications* (Vol. 1, pp. 419–424).

Han, J., & Kamber, M. (2001). *Data mining: Concepts and techniques*. San Francisco: Morgan Kaufman. pp. 346–389.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *14.3.12 Hierarchical clustering. The elements of statistical learning* (2nd ed.). New York: Springer. pp. 520–528.

He, J., Tan, A.-H., Tan, C.-L., & Sung, S.-Y. (2003). *On quantitative evaluation of clustering systems. Clustering and information retrieval*. Kluwer Academic Publishers, 105–133.

Koller, D., & Sahami, M. (1997). *Hierarchically classifying documents using very few words*. Stanford InfoLab.

Lehmann, L. E. (1986). *Testing statistical hypotheses*. Wiley.

Liao, T. W. (2005). Clustering of time series data – A survey. *Pattern Recognition, 38* (11), 1857–1874.

Neyman, J., & Pearson, E.S. (1967). *Joint statistical papers*. Hodder Arnold.

Oates, T., Firoiu, L., & Cohen, P. (1999). Clustering time series with hidden Markov models and dynamic time warping. In *Proceedings of the IJCAI-99 workshop on neural, symbolic and reinforcement learning methods for sequence learning* (pp. 17–21).

Pu, H.-T. (2007). *The development and applications of folksonomy*. <http://www.lib.ncku.edu.tw/journal/16/1.htm> Retrieved 21.06.08.

Salvador, S., & Chan, P. (2007). Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis, 11*(5), 561–580.

Vander Wal, T. (2005). Folksonomy coinage and definition. *Online information conference 2005*.

Voorhees, E. M. (1986). Implementing agglomerative hierarchical clustering algorithms for use in document retrieval. *Information Processing & Management, 22*(6), 465–476.

Van Wijk, J. J., & Van Selow, E. R. (1999). Cluster and calendar based visualization of time series data. In *Proceedings of 1999 IEEE symposium on information visualization* (pp. 4–9).

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association, 58*(301), 236–244.