International Journal of Hybrid Information Technology
Vol.9, No.2 (2016), pp. 247-258
http://dx.doi.org/10.14257/ijhit.2016.9.2.22

# Online Hot Topic Detection Based on Segmented Timeline and Aging Theory

Ruiguo Yu[1], Xiaodong Xie[2], Yongxing Li[3], Mankun Zhao[4], Xuyuan Dong[5*],
Muwen He[6], Peng Chang[7] and Zan Wang[8]

*[1 2 3 4]School of Computer Science and Technology, Tianjin University, Tianjin, P.R. China*
*[1 2 3 4]Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin, P.R. China*
*{[1]rgyu; [2]xxd_dream; [3]liyongxing; [4]zmk}@tju.edu.cn*
*[5 6 7]Information and Network Center, Tianjin University, Tianjin, P.R. China*
*{[5]dongxuyuan; [6]hemuwen; [7]snow}@tju.edu.cn*
*[8]School of Computer and Software, Tianjin University, Tianjin, P.R. China*
*wangzan@tju.edu.cn[8]*
*(Communicated by Xuyuan Dong)*

### Abstract

*With a great deal of digitized textual information now available on the internet, it is almost impossible for people to assimilate all the information timely. Therefore, the technologies of topic detection and tracking are used for constructing news topics from news stories in order to bring convenience to people. However, traditional topic detection methods are not always so effective in detecting emerging hot news topics in a short period of time, and most topic detection methods use single-pass clustering algorithm which is with low accuracy and very sensitive to the input sequence of news documents. In order to improve clustering accuracy, we utilize a temporal distance factor to segment timeline into equal parts and propose a novel two-times single-pass clustering algorithm to deal with news stories in each part of timeline separately. Moreover, the aging theory is combined with our approach to build life-span model of topics from which we can obtain variation trend of hotness value of topics. The results of experiments show that our approach is effective and the life circle model of topics established by our method can conform to reality well.*

**Keywords**: *hot topic detection, two-times clustering, short term, aging theory, single-pass*

## 1. Introduction

Along with the rapid development of network technology, the internet has become one of the most important information sources for people to see what happens every day. According to the thirty-fourth statistical report of Chinese internet development situation given by Internet Network Information Center of China, the number of Internet users in China has reached 633 million by the end of June 2014 including 503.16 million persons who read news online every day, and the scale of websites in China is 2.73 million [1]. When an event occurs, different news websites publish news stories with various viewpoints. As web news documents grows, finding required information from hundreds of thousands of news documents becomes increasingly time consuming and laborious. Such huge amount of news information has been beyond human’s ability of assimilating information, information overloaded has happened. Facing such an embarrassing situation,

Copyright ⓒ 2016 SERSC

people eagerly expect that news selection work would be done for them. Hot topic extraction is a solution for this problem. This subject has emerged as a part of topic detection and tracking (TDT) [2] which attempts to identity topics by searching and organizing content of digitized news documents.

In this paper, we are only interested in detecting hot topics from online news documents that are published in a short period of time. Our goal is to alleviate information overloaded problem by focusing on extracting important topics that appears with high frequency and a large number of reports in short time. In fact, people usually don't have urgent need to obtain news of events that last for a long time, they can get those information easily. But on the other hand, when an emerging topic with a great deal of news reports appears, much more attention is always paid to it. To detect these "burst" topics effectively, we propose a novel two-times single-pass clustering algorithm to extract and update hot topics from news archives online automatically. Meanwhile, through building life circle model of extracted topics with aging theory, we can get a clear understanding of variation trend of hotness value of topics.

The rest of this paper is organized as follows: In Section 2, we give a brief review of related work. A novel two-times single-pass clustering algorithm for extracting hot topics in short term is proposed at Section 3 and we combine the proposed algorithm with aging theory to build life circle model of topics. In Section 4, we describe our experiment data and results to demonstrate that our approach is suitable to solve to this problem, followed by the conclusion and a discussion of our future work in Section 5.

## 2. Related Work

### 2.1. Topic Detection and Tracking

Topic Detection and Tracking was proposed in 1996, since then many researchers were attracted by the subject and invested their time and energy in finding new TDT approaches. Topic Detection and Tracking is intended to structure news stories from news-wires and broadcasts into topics [3]. In the past years, approaches in TDT were mainly improvements of feature extraction methods and news stories clustering algorithms [4-8]. Meanwhile, researchers also made progress in using different text representation models in TDT such as the language model and the probabilistic model [9, 10].

Khoo, *et al.*, designed an information system that will extract main topics in the news archive in a weekly basis with a novel TF-PDF algorithm which can calculate weight of terms [5]. Sentence vector clustering is carried out after TF-PDF weight calculation for summa-rization of the main topic.

Chen, *et al.*, finished their research following Khoo [8]. Based on the TF-PDF algorithm, they proposed a new hot topic detection method combining timeline analysis and multidi-mensional sentence modeling. The results of their empirical experiments showed that their approach is more effective than other hot topic detection methods.

Most of previous work done in Topic Detection and Tracking were focused on detecting topics that distributed in a long period of time. Because of using a lot of background news archive in long term, it's so difficult for their topic detection method to extract emerging topics reported in a short time properly, and what's worse, those "burst" topics which are with valuable information would be miss. Different from previous work, our algorithm focus on detecting topics with vast amount of news stories appearing in a short time. We deal with dynamic-increasing online news data while the algorithms in [5] and [8] require news documents in a long timeline.

Single-pass is most relevant to our task because of its incremental, scalable and dynamic solution to process news data. However, single-pass algorithm is sensitive to input sequence of news documents, and the clustering accuracy is low. For the purpose of getting a better clustering accuracy, we come up with a novel two-times single-pass clustering algorithm.

### 2.2. The Application of Aging Theory

Aging theory was first used in event detection by Chen, *et al.,* in 2003 [11]. In their research, they consider a new event as a life circle of birth, growth, decay and death, and life-span model of a news event is simulated based on this idea. After that, an aging theory for event life-circle modeling was proposed by Chen et al in 2007 to build the life circle of sequential events [12].

The general process of aging theory in [12] is described as follows: Firstly, news documents that belong to a topic are transformed to “nutrition” value, a transformation factor is used to measure it. Secondly, an energy function is defined to gain “energy” value of topics. At the same time, the “energy” value of every topic decays with time, there is a decay factor to measure it. The experiment result in [11] and [12] shows that this proposed aging theory achieves a better overall performance for both long-running and short-term topic detection.

Chen, *et al.,* incorporated the traditional single-pass clustering algorithm with aging theory at [11] and [12]. After a single-pass clustering process, they can know which topic a news document belongs to, followed by the energy value updating process of that topic. Different from the approach proposed by Chen *et al*, we can just identify which topic a news document belongs to after two times single-pass clustering process in our algorithm, so a novel combination way is used to incorporate aging theory with our algorithm.

## 3. Hot Topic Detection and Tracking

In this section we will give a detail description of our work in online topic detection and tracking. It can be divided into three parts: the first part is collection and preprocessing of news documents, the second part is online topic detection from web news archives in short term, the last part is construction of life circle model of every topic.

### 3.1. Preprocessing and Text Representation

In order to obtain news documents from websites, a news page crawler is developed to finish this work. Web pages crawled by our crawler are analyzed to get useful information. Their titles and contents are extracted along with meta-data such as their source and their published time. We call a web page’s content part (including the title) a story, following the name used in TDT.

ICTCLAS [14] is an excellent Chinese lexical analysis system with high Chinese word segmentation accuracy developed by Chinese Academy of Sciences, we use it to solve word segmentation problem. After word segmentation, stop words are also removed from every document.

Incremental TF-IDF model is widely applied to term weight calculation in TDT [3, 13]. We choose the incremental TF-IDF as a base of our title-pivot incremental TF-IDT term weight calculation approach. The document frequency of term *w* in time slot *i* is calculated as:

$$df_i(w) = df_{i-1}(w) + df_{s_i}(w) \tag{1}$$

where $s_i$ means a set of stories coming during time slot *i*, and $df_{s_i}(w)$ means the number of stories that term *w* appears in. $df_{i-1}(w)$ represents the number of stories that contain term *w* before time slot *i* (not included). A training set comprised of a sufficient amount of news stories is used to calculate the initial DF. As shown in the formula above, DF is updated dynamically in each time slot.

Next each story *d* can be represented as an n-dimensional vector, where *n* is the number of distinct terms in story *d*. Among most news stories, news title is either a brief summarization of news content or an incisive comment, we consider that terms in news title are more import than terms in content and assign higher weight values to terms included in news title in our title-pivot incremental TF-IDF model. The weight of term *w* in story *d* can be counted as:

$$weight(w,d) = \frac{tf(w,d)}{\sqrt{\sum_{w'\in d}(tf(w',d))^2}} * \log(\frac{N_i+1}{df_i(w)+0.5}) * (1+\lambda) \quad (2)$$

where $N_i$ is the total number of stories before time slot *i* (included). $tf(w,d)$ means the times that term *w* appear in document *d*. $\lambda$ is a weighted factor for terms appearing in title:

$$\lambda = \begin{cases} 0.3 & \text{if } w \text{ appears in title} \\ 0 & \text{if } w \text{ doesn't appear in title} \end{cases} \quad (3)$$

Terms are sorted by their weight value and the first *K* ( $K = 100$ in this paper) terms are selected to represent a document in terms of time efficiency in online topic detection.

Cosine similarity is used to calculate the similarity between two vectors. For vector *e* and vector *f*, their similarity is calculated as:

$$similarity(e,f) = \frac{\sum_{w\in e\cap f} weight(w,e)*weight(w,f)}{\sqrt{\sum_{w\in e}(weight(w,e))^2} * \sqrt{\sum_{w\in f}(weight(w,f))^2}} \quad (4)$$

### 3.2. A Two-Times Single-Pass Clustering Algorithm

Most TDT approaches use single-pass clustering algorithm to contract news stories into topics. The general process of single-pass algorithm in online topic detection and tracking is described as Algorithm 1. As we all know, single-pass is easily affected by input order of data. If news documents that have a big similarity value between each other are adjacent in the input sequence, single-pass can get a better clustering effect. But the input sequence of news stories in online topic detection must follow time, thus two news documents with a big similarity between each other have a lower probability in adjacent input sequence, which is a main reason for single-pass's low clustering efficiency in online topic detection.

Based on the analysis above, we propose a novel two-times single-pass clustering algorithm which performs better in online topic detection. The basic idea of our method is: A time distance factor is utilized in our approach to segment timeline into equal parts and the length of each part is a time distance. News stories in each time distance are gathered separately as document sets. First we use single-pass to cluster news stories in every time distance into subtopics. Then we use single-pass the second time to construct subtopics of each time distance into topics. Compared to the traditional single-pass, for two news stories which are not adjacent in timeline but with a big similarity between each other, our approach improves the probability that those two news stories are clustered into a topic. A detailed description of our algorithm is shown in Algorithm 2.

**Algorithm 1: Single-Pass Algorithm in TDT**

```
Init the set of topics Topics=null
for each document d in online news stream:
    feature selection and text representation
    if Topics is null:
        add d to Topics as a new topic
    else:
        for each topic T in Topics:
            calculate the similarity between d and T then
            choose the biggest similarity Smax and the topic Tmax who has the
            biggest similarity between d
        end for
        if Smax>cluThre:
            add d to Tmax
        else:
            add d to Topics as a new topic
end for
```

Because of the temporal relation of news stories, the topic's weight of every dimension must be updated progressively to reflect topic's evolution. In our algorithm, there are two cases in which we need to update the topic center vector: the first case is when a new document is added to a subtopic, the second is when a subtopic is added to a topic. In the first case, we update the center vector of subtopic s as: if term *w* both appears in the vector of subtopic *s* and the added news story vector, we update the weight of term *w* as:

$$vector(w_{new}) = \frac{slen - 1}{slen} + weight(w_{sub}) + \frac{1}{slen} * weight(w_d) \qquad (5)$$

where $vector(w_{new})$ is the new weight value of term *w*, *slen* is the number of stories that subtopic *s* includes, $weight(w_{sub})$ is the weight of term *w* in subtopic vector and $weight(w_d)$ is the weight of term *w* in the adding story vector. If term *w* only appears in the subtopic vector or only appears in the added story vector, we use the old weight value. In the second case, we update the center vector of topic *t* as:

$$vector(w_{new}) = \frac{slen * weight(w_{sub}) + tlen * weight(w_t)}{slen + tlen} \qquad (6)$$

where *tlen* is the number of stories that topic *t* includes and $weight(w_t)$ is the weight of term *w* in the center vector of topic *t*. After updating weight value, we also sort terms by their weight value and select the top *K* ( $K = 100$ in this paper) terms as a new vector.

### 3.3. The Definition of Aging Theory

Aging theory is used to build life circle model of events. Three important functions of aging theory are described below:

$getNutrition()$ : calculate the nutrition value that news documents contribute to topics. We define the similarity between a story *d* and a subtopic *s* as $sim(d,s)$ , the similarity between a subtopic *s* and a topic *t* is defined as $sim(s,t)$ . In our algorithm, we can just decide which topic a document belongs to after two times single-pass process, so a dict is used during the first time single-pass process to record the biggest $sim(d,s)$ for every document. Also, we utilize another dict in the second time single-pass process to record the biggest $sim(s,t)$ of every subtopic.

**Algorithm 2: Two-Times Single-Pass Clustering Algorithm**

```
Init the set of topics: $Topics=null$
Init the set of subtopics: $subTopics=null$
Init the start point of time distance: $timeStart=start\_time$
Init the end point of time distance: $timeEnd=timeStart+time\_dis$
for each news document $d$ in online news stream:
    extract the publish time of $d$ as $pTime$
    if $pTime>=timeStart$ and $pTime<timeEnd$:
        feature selection and text representation using VSM
        if $subTopics$ is null:
            add $d$ to $subTopics$ as a new subtopic $S$
        else :
            for each subtopic $S$ in the set $subTopics$:
                calculate the similarity between $d$ and $S$ then choose the subtopic
                $subTmax$ which has the biggest similarity $Smax$ with $d$
            end for
            if $Smax>cluThre1$:
                add $d$ into $subTmax$ and update the center vector of $subTmax$
            else:
                add $d$ to the set $subTopics$ as a new subtopic $S$
    else if $pTime>=timeEnd$:
        if $Topics$ is null:
            copy the set $subTopics$ to the set $Topics$ and set $subTopics=null$
        else:
            for each subtopic $S$ in the set $subTopics$:
                for each topic $T$ in the set $Topics$:
                    calculate the similarity between $S$ and $T$ then choose the topic
                    $Tmax$ which has the biggest similarity $Smax$ between $S$
                end for
                if $Smax>cluThre2$:
                    add subtopic $S$ to topic $Tmax$ then update the center vector and
                    file list of $Tmax$
                else:
                    add subtopic $S$ to the set $Topics$ as a new topic
            end for
            set $subTopics=null$
        $timeEnd=timeStart$
        $timeStart=timeEnd+time\_dis$
end for
```

If a document is the first document of a subtopic or a subtopic is directly added to the topic set without combining with a topic, we record 0. Then two kinds of nutrition function are applied to deal with two different situations: (1) A subtopic is directly added to the topic set but not combined with a topic. The nutrition is calculated as:

$$getNutrition = \alpha * sim(d,s) \tag{7}$$

where $\alpha$ is the nutrition transformation factor. (2) A subtopic is added to a topic. Then the nutrition is calculated as:

$$getNutrition = \alpha * \frac{sim(d,s) + sim(s,t)}{2} \tag{8}$$

$energyFunction()$ : It is a monotonically increasing function for transforming the nutrition value of topics to energy value. The inverse function of $energyFunction()$ is used to calculate the previously accumulated nutrition value. It is defined as:

$$energyFunction(nutrition) = \frac{2.0 * nutrition}{1.0 + 2.0 * nutrition} \tag{9}$$

Copyright ⓒ 2016 SERSC

**Algorithm 3: Hot Topic Detection and Life-Span Modeling Algorithm**

```
Init the set of topics: Topics=null
Init the set of subtopics: subTopics=null
Init the start point of time distance: timeStart=start_time
Init the end point of time distance: timeEnd=timeStart+time_dis
for each news document d in online news stream:
    extract the publish time of d as pTime
    if pTime>=timeStart and pTime<timeEnd:
        feature selection and text representation using VSM
        if subTopics is null:
            add d to subTopics as a new subtopic S, record the sim_value as 0 in simDict1
        else:
            for each subtopic S in the set subTopics:
                calculate the similarity between d and S then choose the subtopic
                subTmax which has the biggest similarity Smax with d
            end for
            if Smax>cluThre1:
                add d into subTmax and update the center vector of subTmax
                record Smax in simDict1
            else:
                add d to the set subTopics as a new subtopic S, record the
                sim_value as 0 in simDict1
        record the document processing order in order_list
    else if pTime>=timeEnd:
        if Topics is null:
            copy the set subTopics to the set Topics
        else:
            for each subtopic S in the set subTopics:
                for each topic T in the set Topics:
                    calculate the similarity between S and T then choose the topic
                    Tmax which has the biggest similarity Smax between S
                end for
                if Smax>cluThre2:
                    add subtopic S to topic Tmax then update the center vector and
                    file list of  Tmax, record Smax in simDict2
                else:
                    add subtopic S to the set Topics as a new topic
                    record the sim_value as 0 in simDict2
            end for
        for each time slot i in the current time distance:
            for each news document d of order_list in time slot i:
                search the topic T that d belongs to
                if d is the first document of T:
                    update the energy value of T
                        $E(T)^{i} = energyFunction(\alpha)$
                else:
                    search the subtopic S that d belongs to, then get sim(d,s) and sim(s,t)
                    if sim(s,t)==0
                        $E(T)^{i} = energyFunction(energyFunction^{-1}(E(T)^{i-1}) + \alpha * sim(d,s))$
                    else:
                        $E(T)^{i} = energyFunction(energyFunction^{-1}(E(T)^{i-1}) + \alpha * \frac{sim(d,s) + sim(s,t)}{2})$
            end for
            for each topic T in the set Topics:
                if the energylist of T is not null:
                    $E(T) = E(T) - \beta$
                    if $E(T) < 0$ : $E(T) = 0$
            end for
        end for
        set $subTopics, order\_list, simDict1, simDict2\ null$
        $timeEnd=timeStart$
        $timeStart=timeEnd+time\_dis$
    end for
```

$energyDecay()$: In each time slot, the energy value of a topic would decay a constant $\beta$ which is called the decay factor. The nutrition transformation factor $\alpha$ and the decay factor $\beta$ can be obtained from training data. If the energy value of a topic is below a threshold (we set 0 in this paper), it will be considered as a "dead" topic.

### 3.4. Hot Topic Detection and Life-Span Modeling Algorithm

Different from what Chen, *et al.*, done in [11, 12] to incorporate the traditional single-pass algorithm with aging theory, our algorithm can only identify which topic a news story belongs to after two times single-pass process, so we separate the process of clustering and the process of energy value calculation. Our algorithm updates the energy value of every topic after the second single-pass process, thus an order-list is used to record the document processing order of every time distance in the first single-pass process. A search work is done firstly to determine which topic a document in order-list belongs to and then we update the energy value of the topic. The detail description of our hot topic detection and life-span modeling algorithm combining two-times single-pass with aging theory is given in Algorithm 3.

## 4. Experiments and Results Analysis

### 4.1. Dataset Description

We get more than 83000 news stories that are posted from March 1, 2014 to March 8, 2014 on five most popular news websites in China (163.com, Sina.com, People.cn, Chinanews.com and Ifeng.com). 8300 news stories are randomly selected as our dataset. 10000 news documents are randomly chosen from the rest data to be used to calculate the initial DF. 8 topics including total 991 news documents are manually labeled. Table 1 shows the number of news stories of each labeled topic.

**Table 1. The Manually Labeled Topics in the Dataset**

| ID | Topic | Number of stories |
|---|---|---|
| 1 | The tense situation in Ukraine | 299 |
| 2 | Disappearance of  Malaysia flight MH370 | 136 |
| 3 | Terrorist attack in Yunnan Kunming railway station | 194 |
| 4 | The internet financial in China | 77 |
| 5 | Hazy weather everywhere | 101 |
| 6 | North Korea fired cannonball | 32 |
| 7 | Housing price | 101 |
| 8 | The Oscar Awards | 51 |

### 4.2. Evaluation Metrics

We treat each topic as a cluster and choose traditional evaluation metrics widely used in Information Retrieval and clustering: Recall, Precision and F-measure. Topics generated by our approach are called clusters and actual topics labeled manually are called classes. Then Recall and Precision are calculated as:

$$Recall(i, j) = \frac{n_{ij}}{n_i} \tag{10}$$

$$Precision(i, j) = \frac{n_{ij}}{n_j} \tag{11}$$

where $n_{ij}$ is the number of news stories of cluster *j* in class *i*, and $n_i$ is the size of class *i* and $n_j$ is the size of cluster *j*. The F-measure of class *i* and cluster *j* is calculated as:

$$F(i, j) = \frac{2 * Recall(i, j) * Precision(i, j)}{Recall(i, j) + Precision(i, j)} \tag{12}$$

### 4.3. The Selection of Time Distance

Firstly, all parameters except time distance are settled optimized from the data set. The TDT results on labeled topics are best with *cluThre1*=0.17 and *cluThre2*=0.19 .The nutrition transformation factor $\alpha$ and the energy decay factor $\beta$ are trained by the methods proposed in [11], we get the final values: $\alpha = 0.14332$ and $\beta = 0.014671$ . Then we finish the selection work of time distance. In term of time efficiency, we don't consider time distance value less than 4 hours. Six time distance values are measured by our approach; the results are shown in table 2. As the result shows, our approach has an overall better performance when time-dis is 24-hours.

### 4.4. Topic Detection

We sort 8300 news documents by their publish time to simulate the online environment. Then we use our proposed approach and the traditional single-pass algorithm to detect topics on the data set respectively. The clustering threshold we use in traditional single-pass is 0.14. The results are shown in table 3 and table 4. The experiment results indicate that our proposed algorithm has obvious promotion effect in online topic detection.

**Table 2. Topic Detection Results of Different Time-Dis Values**

| time-dis(hour) | Topic ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | AVG |
|---|---|---|---|---|---|---|---|---|---|---|
| 4 | Precision | 0.911 | 0.977 | 0.901 | 0.762 | 0.851 | 0.549 | 0.893 | 0.933 | 0.847 |
| | Recall | 0.957 | 0.934 | 0.840 | 0.831 | 0.733 | 0.875 | 0.495 | 0.824 | 0.811 |
| | F-measure | 0.933 | 0.955 | 0.869 | 0.795 | 0.787 | 0.675 | 0.637 | 0.875 | 0.816 |
| 6 | Precision | 0.916 | 0.977 | 0.925 | 0.708 | 0.873 | 0.609 | 0.925 | 0.953 | 0.861 |
| | Recall | 0.950 | 0.949 | 0.763 | 0.883 | 0.683 | 0.875 | 0.733 | 0.804 | 0.830 |
| | F-measure | 0.933 | 0.963 | 0.836 | 0.786 | 0.767 | 0.718 | 0.818 | 0.872 | 0.837 |
| 12 | Precision | 0.923 | 0.985 | 0.903 | 0.709 | 0.824 | 0.718 | 0.899 | 0.938 | 0.862 |
| | Recall | 0.963 | 0.949 | 0.820 | 0.948 | 0.743 | 0.875 | 0.792 | 0.882 | 0.871 |
| | F-measure | 0.943 | 0.966 | 0.859 | 0.811 | 0.781 | 0.789 | 0.842 | 0.909 | 0.863 |
| 18 | Precision | 0.924 | 0.977 | 0.899 | 0.682 | 0.866 | 0.460 | 0.859 | 0.958 | 0.828 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Recall | 0.980 | 0.941 | 0.923 | 0.974 | 0.703 | 0.906 | 0.782 | 0.902 | 0.889 |
| | F-measure | 0.951 | 0.959 | 0.911 | 0.802 | 0.776 | 0.611 | 0.819 | 0.929 | 0.845 |
| **24** | Precision | 0.949 | 0.937 | 0.881 | 0.664 | 0.895 | 0.750 | 0.890 | 0.922 | **0.861** |
| | Recall | 0.987 | 0.985 | 0.954 | 1.000 | 0.842 | 0.844 | 0.802 | 0.922 | **0.917** |
| | F-measure | 0.967 | 0.961 | 0.916 | 0.798 | 0.867 | 0.794 | 0.844 | 0.922 | **0.884** |
| 30 | Precision | 0.936 | 0.915 | 0.843 | 0.667 | 0.804 | 0.614 | 0.887 | 0.942 | 0.826 |
| | Recall | 0.977 | 0.956 | 0.938 | 0.987 | 0.772 | 0.844 | 0.851 | 0.961 | 0.911 |
| | F-measure | 0.956 | 0.935 | 0.888 | 0.796 | 0.788 | 0.711 | 0.869 | 0.951 | 0.862 |

**Table 3. Results using the Traditional Single-Pass**

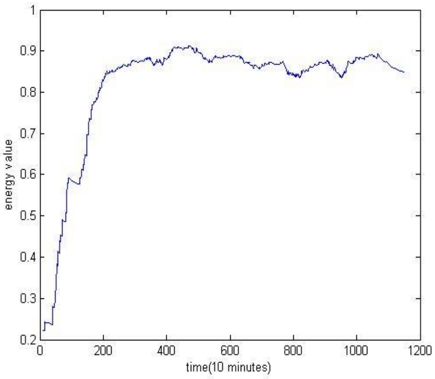
| Topic | Precision | Recall | F-measure |
|---|---|---|---|
| 1 | 0.897(210/234) | 0.702(210/299) | 0.788 |
| 2 | 1.0(96/96) | 0.706(96/136) | 0.828 |
| 3 | 0.964(163/169) | 0.840(163/194) | 0.898 |
| 4 | 0.846(66/78) | 0.857(66/77) | 0.852 |
| 5 | 0.852(52/61) | 0.515(52/101) | 0.642 |
| 6 | 0.547(29/53) | 0.906(29/32) | 0.682 |
| 7 | 0.820(41/50) | 0.406(41/101) | 0.543 |
| 8 | 0.940(47/50) | 0.922(47/51) | 0.931 |
| AVG | 0.858 | 0.732 | 0.770 |

**Table 4. Results with Two-Times Single-Pass Algorithm**

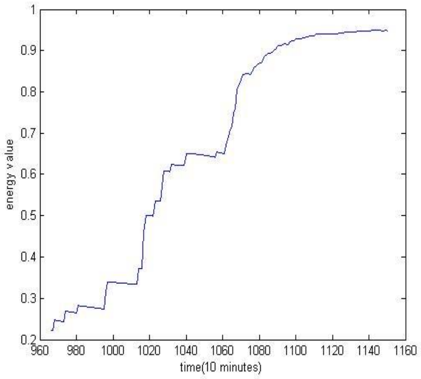
| Topic | Precision | Recall | F-measure |
|---|---|---|---|
| 1 | **0.949(295/311)** | **0.987(295/299)** | **0.967** |
| 2 | 0.937(134/143) | **0.985(134/136)** | **0.961** |
| 3 | 0.881(185/210) | **0.954(185/194)** | **0.916** |
| 4 | 0.664(77/116) | **1.0(77/77)** | 0.798 |
| 5 | **0.895(85/95)** | **0.842(85/101)** | **0.867** |
| 6 | **0.750(27/36)** | 0.844(27/32) | **0.794** |
| 7 | **0.890(81/91)** | **0.802(81/101)** | **0.844** |
| 8 | 0.922(47/51) | 0.922(47/51) | 0.922 |
| AVG | **0.861** | **0.917** | **0.884** |

## 4.5. Life-Span Modeling of Topics

We utilize our data set containing 8300 news stories to build life-span model of topics. The life circle curves of four topics are shown in Figure 1.

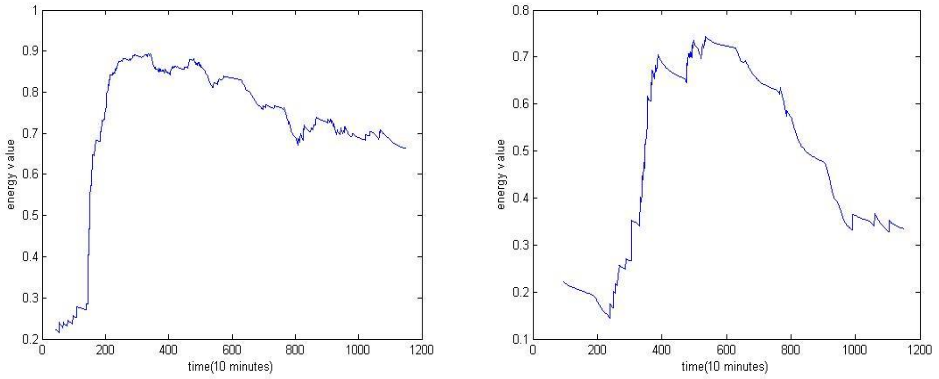


**Topic 1**



**Topic 2**

Copyright ⓒ 2016 SERSC



Topic 3 Topic 8

**Figure 1. The life-Span Model of Topics (The Horizontal Axis Represents the Time Point of Topics that Starts at 0:0:0 1/3/2014. We Set Time Slot as 10 Minutes so there is a Time Point Every Ten Minutes in the Horizontal Axis)**

The life span curves of topic 3 and topic 4 describe the general topic life-span model defined by aging theory. The energy value of topic 3 got a rapid growth soon after the Kunming event happened at March 1, 2014, and after the criminal case was solved at March 3, 2014, the energy value of topic 3 decayed as time gone. The energy value of topic 8 increased rapidly as The Oscar Award Ceremony was held at March 2, 2014 and then it decreased with time.

Topic 1 is a hot topic that emerged before March 1, 2014 and wasn't over at March 8, 2014, there were many news stories about the tense situation of Ukraine every day. So the energy value of topic 1 got a rapid growth and reached a high value, but the energy value didn't decrease obviously. Topic 2 became a hot topic immediately after disappearance of Malaysia flight MH370 at March 8, 2014. We just got news documents about it of March 8, so there is only the rapid increase process of topic 2 in the life-span curve. The analysis above shows that our life-span models clearly describe the energy value variation trend of topics and our life-span model of topics conform to reality.

## 5. Conclusions and Future Work

In this paper, we come up with a two-times single-pass clustering algorithm based on segmented timeline. The experiments above show that our approach has a better performance at online topic detection in a short period of time. What's more, we have utilized a new way to incorporate the aging theory with our algorithm to build the life circle model of topics. The next step of our research will focus on the analysis of life circle curves of topics to discover the conditions that a topic becomes a breaking event. And then we can make breaking event prediction based on the rapid-growth part in their life span. The work in this paper lay a foundation of our following research on breaking topic prediction.

## Acknowledgments

We'd like to thank the anonymous reviewers for their helpful and constructive comments. This work was partly funded by Project 61202030 supported by NSFC and Project x2013-005 supported by ECTC.

## References

[1] “The 34th statistical report of Chinese internet development situation”, http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201407/P020140721507223212132.pdf.

[2] J. Allan, “Introduction to topic detection and tracking”, Springer US, **(2002)**.

[3] J. Allan, R. Papka and V. Lavrenko, “On-line new event detection and tracking”, Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, Melbourne, Australia, **(1998)**.

[4] Y. Zhuang, “An improved TFIDF algorithm in electronic information feature extraction based on document position”, Advances in Mechanical and Electronic Engineering, **(2012)**, pp. 449-454.

[5] K. K. Bun and M. Ishizuka, “Topic extraction from news archive using TF*PDF algorithm”, Proceedings of the Third International Conference on Web Information Systems Engineering, Singapore, **(2002)**, pp. 73-73.

[6] D. Trieschnigg and W. Kraaij, “Hierarchical topic detection in large digital news archives”, Proceedings of the 5th Dutch Belgian Information Retrieval workshop, Utrecht, the Netherlands, **(2005)**, pp. 55-62.

[7] K. Zhang, J. Li and G. Wu, “New event detection based on indexing-tree and named entity”, Proceedings of the 30th Annual International ACM SIGIR Conference, Amsterdam, the Netherlands, **(2007)**, pp. 215-222.

[8] K. Y. Chen, L. Luesukprasert and S. Chou, “Hot topic extraction based on timeline analysis and multidimensional sentence modeling”, IEEE Transactions on Knowledge and Data Engineering, **(2007)**, pp. 1016-1025.

[9] X. Guo, Y. Xiang, Q. Chen, Z. Huang and Y. Hao, “LDA-based online topic detection using tensor factorization”, Journal of Information Science, **(2013)**.

[10] Y. Chen, H. Amiri, Z. Li and T. Chua, “Emerging topic detection for organizations from microblogs”, Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, Dublin, Ireland, **(2013)**, pp. 43-52.

[11] C. C. Chen, Y. T. Chen, Y. Sun and M. C. Chen, “Life cycle modeling of news events using aging theory”, Machine Learning, **(2003)**, pp. 47-59.

[12] C. C. Chen, Y. T. Chen and M. C. Chen, “An aging theory for event life-cycle modeling”, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, vol. 37, no 2, **(2007)**, pp. 237-248.

[13] C. Wang, M. Zhang, L. Ru and S. Ma, “Automatic online news topic ranking using media focus and user attention based on aging theory”, Proceedings of the 17th ACM conference on Information and knowledge management, Napa Valley, CA, USA, **(2008)**, pp. 1033-1042.

[14] http://www.ictclas.org.