Chinese Event Identification and Tracking Using Two Phase Clustering Algorithm

Jun Tang

Department of Information Engineering, Hunan Urban Construction College, Hunan, China, 411101

E-mail: xttangjun@163.com
doi:10.4156/jcit.vol6. issue3.32

Abstract

In this paper, two phase clustering algorithm is adopted to identify and track event. The first phase clustering is incremental clustering algorithm, and the new event will be identified. The second phase clustering is refined clustering algorithm, and the new event will be group and tracking. Experimental result shows that event identification and tracking using two phase clustering algorithm is effective.

Keywords: Clustering Algorithm, Event Identification, Two Phase

1. Introduction

The first step to track hot events is to identify all new events occur in news reports. New event identify is typically approached by reducing a news report to a set of features, either as a vector or a probability distribution. When a news report arrives, its feature set is compared to those of all past news reports. If there is a sufficient difference, the news report is marked as a new event. Another approach to do new event identify is using cluster algorithm. The cluster tracking system is not just to identify one set of a new topic in the news, but also to cluster new reports on the same topic into bins.

Yuxin Peng et al. [1] proposed a new approach for hot event detection and summarization of news videos. The approach is mainly based on two graph algorithms: optimal matching (OM) and normalized cut (NC). Initially, OM is employed to measure the visual similarity between all pairs of events under the one-to-one mapping constraint among video shots. Then, news events are represented as a complete weighted graph and NC is carried out to globally and optimally partition the graph into event clusters. Finally, based on the cluster size and globality of events, hot events can be automatically detected and selected as the summaries of news videos across TV stations of various channels and languages.

Tingting He et al. [2] proposed a method to detect hot event automatically. They used all the web pages from Jan 1st 2005 to Dec 31st 2005, and detected new events by using incremental TF-IDF model and incremental cluster algorithm. Based on analysis of the attributes of events, they proposed a method to measure the activity of events, then filter and sort the event according to the activity of events; finally a hot event list can be derived.

Canhui Wang et al. [3] proposed an automatic online news topic ranking algorithm based on inconsistency analysis between media focus and user attention. News stories are organized into topics, which are ranked in terms of both media focus and user attention. Experiments performed on practical Web datasets show that the topic ranking result reflects the influence of time, the media and users. They presented the quantitative measure of the inconsistency between media focus and user attention, which provides a basis for topic ranking and an experimental evidence to show that a gap between what the media provide and what users view.

Tingting HE et al. [4] proposed a method to detect hot event automatically. They download everyday documents pages from web sites, and detect temporary events by incremental clustering to get the events list of every month. Second, at the end of the year, they choose the events of which the number of related reports is greater than a certain threshold, and cluster them by single link method to get the events list of the year. Last, they proposed a formula to measure the hot level of events, filter and rank the events according to their scores, and then get the hot events of the year.

Hong Li et al. [5] proposed a method of bursty hot topic detection based on bursty feature. Firstly, determine the bursty features during a given time period based on term frequency, and cluster similar...
news stories together according to the content similarity and time decaying function to get the potential topic list, and then determine the final bursty hot topics based on bursty features. This method greatly reduces the complexity of the algorithm, improves efficiency, but also ensures the accuracy of bursty hot topic detection.

Junhua Li et al. [6] proposed a method of extracting “Chinese hot topic” from a set of text document downloaded from the Internet according to the web log. There are three major steps. Firstly, they get all corrective user information and the textual materials from web according to the web log. Secondly, they extract the hot terms of each web page, computing hotness of theme based on click-through rate and the forgetting factor. Finally, they form hot topics by merging correlative themes on the basis of common hot terms. It can deal with massive textual data with high efficiency and brings a new angle from the users in determining whether a topic is hot or not.

Masoomeh Zameni et al. [7] proposed a new algorithm for detection of expected goal events in soccer video. The proposed algorithm is composed of two main steps. Firstly, video is segmented into its constituent shots and these shots are categorized into two groups, namely long shots and non-long shots. Secondly, long shots are examined to detect expected goal events. Their scheme uses the playfield boundary feature for shot boundary detection, shot classification, and expected goal events detection. Therefore, this feature is extracted only one time and the algorithm is speedy. It is noticeable that the proposed algorithm is robust to blurring and spatial down sampling.

Qi He et al. [8] proposed a simple and effective topic detection model called the temporal Discriminative Probabilistic Model (DPM), which is shown to be theoretically equivalent to the classic vector space model with feature selection and temporally discriminative weights. They compared DPM to its various probabilistic cousins, ranging from mixture models like von-Mises Fisher (vMF) to mixed membership models like Latent Dirichlet Allocation (LDA). Benchmark results on the TDT3 data set show that sophisticated models, such as vMF and LDA, do not necessarily lead to better results; in the case of LDA, notably worst performance was obtained under variational inference, which is likely due to the significantly large number of LDA model parameters involved for document-level topic detection. On the contrary, using a relatively simple time-aware probabilistic model such as DPM suffices for both offline and online topic detection tasks, making DPM a theoretically elegant and effective model for practical topic detection.

Jianjiang Li et al. [9] proposed a cooperative alert topic detection model in distributed environment (named CATDM). The model abstracts the alert topic and represents it as the local alert case by analyzing the alert of campus network culture in depth. The model not only discovers new alert topic of local network domain, but also cooperatively schedules the information of alert case knowledge base (ACKB) between different alert monitor nodes. CATDM discovers new alert topic of each monitor node and optimizes the local ACKB periodically. Through cooperatively scheduling the information of ACKB between different alert monitor nodes, the model enables some alert monitor nodes to obtain the ability of detecting new alert topic and strengthens the detection ability of burst alert topic.

Jun Tang [10] proposed improved K-means clustering algorithm based on user tag. It first used social annotation data to expand the vector space model of K-means. Then, it applied the links involved in social tagging network to enhance the clustering performance. Experimental result shows that the proposed improved K-means clustering algorithm based on user tag is effective.

Ding Shifei et al. [11] studied a clustering algorithm based on information visualization. In this algorithm, through a nonlinear mapping (NLP), some high-dimensional and complicated feature data is transformed into low-dimensional feature data, such as one, two and three dimensionality. Its main aim is that the geometry image in high-dimensional space is mapped into one, two and three dimensional image in low-dimensional space, and the inherent data "structure" is approximately preserved after mapping.

Jiadong Ren et al. [12] proposed a novel approach PKS-Stream for clustering data streams, which is based on grid density and index tree Pks-tree. The new index structure Pks-tree is introduced to store the non-empty grid cells, which aims to improve the efficiency of storage and indexing. Simultaneously, they defined a novel time-based density threshold function to remove the noise points in real time. Based on Pks-tree, the data stream is clustered by grid density in the initial stage. With new data records arriving, the novel pruning strategy is adopted to periodically detect and remove noise points. Also, the generated clusters are dynamically adjusted.
Event identification and tracking, which are constructed from news stories using the techniques of Topic Detection and Tracking, bring convenience to users who intend to see what is going on through the Internet. However, it is almost impossible to view all the generated topics, because of the large amount. So it will be helpful if all topics are ranked and the top ones, which are both timely and important, can be viewed with high priority. Generally, topic ranking is determined by two primary factors.

In this paper, two phase clustering algorithm is adopted to identify and track event. The first phase clustering is incremental clustering algorithm, and the new event will be identified. The second phase clustering is refined clustering algorithm, and the new event will be group and tracking.

The rest of the paper is organized as follows. Section 2 is the description of event identification and tracking using two phase clustering algorithm. Section 3 focuses on experiments and evaluations. Finally, we end this paper with a conclusion and the future work.

2. Event identification and tracking using two phase clustering algorithm

This section is the kernel of this paper. Here, it illustrates two phase clustering algorithm. Traditional clustering algorithms are usually based on the bag-of-words (BOW) approach. A notorious disadvantage of the BOW model is that it ignores the semantic relationship among words. As a result, if two documents use different collections of core words to represent the same topic, they can be assigned to different clusters, even though the core words they use are probably synonyms or semantically associated in other forms.

K-means is a simple but efficient and highly scalable clustering method. It iteratively calculates the cluster centroids and reassigns each document to the closest cluster until no document can be reassigned. K-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data as well as in the iterative refinement approach employed by both algorithms.

Two key features of k-means which make it efficient are often regarded as its biggest drawbacks. Euclidean distance is used as a metric and variance is used as a measure of cluster scatter. The number of clusters k is an input parameter: an inappropriate choice of k may yield poor results. That is why two phase clustering algorithm are adopted.

The vector expression for documents is the foundation for documents clustering and document similarity calculation. Each dimension of the vector represents the weight of a certain term in the document. The inverse query weight scheme was used in this paper.

The $tf$ component in the weight scheme represents the frequency of a term $w$ in document $d$. The $idf$ component is the inverse of the number of documents in which the term occurs. The formula below shows the $tf \cdot idf$ scheme employed in paper as in [13]:

$$
\text{tf} = \frac{tf_{raw}}{tf_{raw} + 1 + \frac{\text{len}_d}{\text{len}_{avg}}}
$$

$$
\text{idf} = \log(N / df)
$$

Where, $\text{len}_d$ denotes the length of document $d$, and $\text{len}_{avg}$ is the average length of the documents.

Document similarity is calculated by cosine similarity.

The event identification and tracking would perform necessary actions to the newly downloaded pages dynamically. Therefore, an incremental scheme was used for calculation of $df$ as below. At time
point \( t \), a new set of document \( C_t \) was added to the model by \( df \).

\[
df_t(w) = df_{t-1}(w) + df_{C_t}(w)
\]

Where, \( df_{C_t}(w) \) denotes the document frequency of term \( w \).

Clustering used in event identification needs to perform a comparison between the recently added documents and previous stored documents. But because of the huge number of downloaded web pages in a group, the time needed to perform a comparison between all the stored documents and newly added documents is undesirable. Due to the requirement of dynamic handling to event identification, the incremental clustering algorithm was used.

We know that reports about same event tend to be dense, it is only necessary to compare the newly added document to documents happened within a short time period to it. The documents in group were stored according to their downloaded date. The newly added document only needed to be compared with a certain number of documents before it.

The following is a description of the clustering algorithm:

Create the vector expression of document \( D_1 \) and initialize a cluster with \( D_1 \). When a new document \( D_k \) \((k > 1)\) is downloaded, update the \( df \) value for each term in \( D_k \) and computed the vector expression of \( D_k \). If \( Sim(D_k, D^*) > \lambda \), then add \( D_k \) to cluster \( D^* \); otherwise, treat \( D_k \) as a new cluster, \( \lambda \) is threshold parameter.

After that, each group \( G_i \) of every month was processed by the following:

The incremental clustering was used to the group. After one month, a temporary event bin was generated, arranged in descending order by the number of documents in the group. After one year, in order to make it converge faster and result in better clusters in accordance with the original distribution, the improved the traditional k-medoids clustering algorithm was executed for those events in the monthly temporary event list whose number of the documents exceeded a certain threshold value. Then the event bin of the year was obtained.

The k-medoids algorithm is a clustering algorithm related to the k-means algorithm and the medoid shift algorithm. Both the k-means and k-medoids algorithms are partitional (breaking the dataset up into groups) and both attempt to minimize squared error, the distance between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the k-means algorithm, k-medoids chooses data points as centers (medoids or exemplars).

k-medoid is a classical partitioning technique of clustering that clusters the data set of \( n \) objects into \( k \) clusters known a priori. A useful tool for determining \( k \) is the silhouette.

It is more robust to noise and outliers as compared to k-means because it minimizes a sum of dissimilarities instead of a sum of squared Euclidean distances.

A medoid can be defined as the object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal i.e. it is a most centrally located point in the given data set.

The most common realisation of k-medoid clustering is the Partitioning Around Medoids (PAM) algorithm and is as follows:

Initialize: randomly select \( k \) of the \( n \) data points as the medoids
Associate each data point to the closest medoid. ("closest" here is defined using any valid distance metric, most commonly Euclidean distance, Manhattan distance or Minkowski distance)
For each medoid \( m \)
   For each non-medoid data point \( o \)
     Swap \( m \) and \( o \) and compute the total cost of the configuration
Select the configuration with the lowest cost.
Repeat above steps until there is no change in the medoid.

The improved the traditional k-medoids clustering algorithm is described as following:

When deciding the new center of a cluster, we first choose top-p documents closest to the old center in the cluster and take the document closest to the mean of the p documents as the new center. The
improved algorithm is described as follows:
1. Choose $m$ documents as the centers of clusters randomly: $C_1, C_2, \ldots, C_i, \ldots, C_m$;
2. Compute the similarity for each document to each cluster center. Insert the document to the closest cluster.
3. Determine the new center of each cluster as follows:
   Compute the average similarity of documents in cluster $i$ using the formula:
   \[
   \text{avgsim}[i] = \frac{1}{n} \sum_{k=1}^{n} \text{sim}(D, C_i)
   \]
   Where $n$ is the number of documents in cluster $i$.
   Compute the parameter: $p = m \times \frac{\text{avgsim}[i]}{\max(\text{avgsim})}$, then select $p$ documents most closest to the cluster $i$ center.
   Compute the mean of the above $p$ documents and choose the document closest to it as the new center of cluster $i$.
4. Repeat above steps until there is no change in the clusters.

Tracking hot events is very hard, we added two parameters to evaluate hot event: the exposure frequency of the event within a time unit, the consecutive reported days of the event within a time unit.

3. Experiments and evaluations

In the experiment, About 1376,546 news web pages from Jan 1st 2008 to Dec 31st 2008 were downloaded from five popular portal web sites. According to the subject, the web pages were divided into 5 groups: Sports, Science, Technology, Finance and Entertainment, denoted as $G_i$ respectively. A temporary event list was generated for each month by incremental clustering. At the end of the year, we chose the events of which the number of related reports was greater than a certain threshold, and clustered them by single link clustering to get the events list of the year. We scored and ranked the events, and then the events of the year were ultimately achieved by filtering them.

We got an event list arranged in descending order by their score.

<table>
<thead>
<tr>
<th>No</th>
<th>Events</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>北京 29届奥运会</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>汶川大地震</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>三鹿毒奶粉事件</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>中国股市狂跌</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>神七飞天</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>纪念中国改革开放 30周年</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>海峡两岸实现三通</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>新中国喜迎 60 华诞</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>中国受美国金融危机影响经济下滑明显</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>十七届三中全会在京召开</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 1. Events generation using our method
Most of the events published by other media are in time order so that our comparison is focused on the coverage of the events. From the above tables, it can be seen that the nine of the top ten events obtained by our method are included in the table 2. The percentage of coverage will be even higher if the number of our events selected is increased to 20.

Experimental results show that it is feasible to use our method to identify and track events. The ranking of the event lists got by using statistical information show the most important events of the year objectively.

4. Conclusions and future work

In this paper, two phase clustering algorithm is adopted to identify and track event. The first phase clustering is incremental clustering algorithm, and the new event will be identified. The second phase clustering is refined clustering algorithm, and the new event will be group and tracking.

Also, there is a lot of room for improvement. For example, clustering algorithm is time consuming, which also produces increase of computational complexity of the method. We need to speed up clustering algorithm in future. We will try to find better event tracking algorithm.

5. References


