May-June 2020
ISSN: 0193-4120 Page No. 3824-3830

Algorithmic Analysis on Optimization Oriented Hot Event Detection and Product Recommendation

[1]Manu G Thomas, [2]S. Senthil

[1]Research Scholar, School of C&IT, REVA University, Bangalore, India, manugthomas@gmail.com
[2]Professor & Director, School of Computer Science and Applications, REVA University, Bangalore, India, dir.csa@reva.edu.in

Article Info
Volume 83
Page Number: 3824-3830
Publication Issue:
May-June 2020

Article History
Article Received:*19 November 2019*
Revised: *27 January 2020*
Accepted: *24 February 2020*
Publication: *12 May 2020*

Abstract

"A hot topic is said to be a topic that people widely talk about during a specific time". However, there was no precise description of the hot topic and hotness evaluation standards. This paper presents a novel system that concerns on Hot Topic Detection (HTD) integrated with the recommendation o products as well. The proposed hot event detection model is performed via (i) Pre-processing (ii) Feature Selection and weighting (iii) Text Model Construction (iv) Cluster-based topic Identification. Initially, the keywords are extracted from every tweet. Subsequently, the feature vector space is evaluated that is then subjected for portraying text model construction. At last, clustering is carried out via optimization logic that meant for detecting the hot topic. Particularly, two diverse clustering processes are carried out: micro-clustering and macro-clustering, where the optimal centroid is selected by Monarch Butterfly Optimization (MBO). Following the hot event detection, the adopted model concerns the recommendation of the product associated with the discussed topic. Finally, algorithmic analysis is done by varying the weighting element of the MBO model from 0.2, 0.4, 0.6 and 0.8.

Keywords: *Hot Topic Detection; Pre-Processing: Feature selection; Optimization; Product Recommendation.*

Nomenclature

Abbreviation	Description
HTD	Hot Topic Detection
MBO	Monarch Butterfly Optimization
PLSA	Probabilistic Latent Semantic Analysis
TMHTD	Two-Phase Mic-Mac Hot Topic Detection
TD-HITS	Topic-Decision based Hypertext-Induced Topic Search
TS-LDA	Latent Dirichlet Allocation oriented Three-Step model
PPV	Positive Predictive Value
FPR	False Positive Rate
FNR	False Negative Rate
NE	Name Entities
TF-IDF	Term Frequency-Inverse Document Frequency
VSM	Vector Space Model
KWV	Key Word Vector
FSV	First Sentence Vector
OSV	Other Sentence Vector

1. Introduction

In recent years, social networking together with microblogging services has gained ever-increasing popularity in social media platform and it has attracted the users in a rapid manner [6] [7]. A huge count of data produced for the major events in microblogging needs the hot event determination along with the detection of key posts associated with those events. As a result, it is essential to detect the hot events in micro blogs [8] [9].

In recent days, event detection techniques [10] [11] depending on topic models are growing in popularity. For instance, LDA and PLSA were the two significant techniques for recognizing the hidden factors in microblogs [12]. These approaches

determine the occurrences of words depending on probabilistic theory and thus, the topical similarity is measured between the words. In addition, conventional event detection schemes necessitate human involvement for detecting the count of topics that greatly minimizes the accuracy and efficiency of event detection [13] [14].

Conventionally, the classification, identification, and annotation of hot events are manually carried out, which minimizes the efficacy of hot topic detection [15] [16]. In addition, several conventional topic models that are applied directly to microblogging will meet up with issues regarding data sparsity, which dynamically reduces the quality of detecting the user interest and hot events.

The major contribution of this paper is as follows:

- An automated system is introduced, which detects the hot events integrated with the product recommendation system.
- Clustering is carried out with the assistance of optimization, by which the micro-clustering and macro-clustering process takes place.
- The MBO algorithm is exploited for finding the optimal centroid.

The arrangement of the paper is specified as Section II portrays the review. Section III describes the framework of hot event detection: problem definition and section IV portrays the topic identification: optimization assisted clustering process. Section V defines the product recommendation system: integration of the detected hot topic. Section VI discusses the outcomes and section VII concludes the paper.

2. Literature Review

Related works

In 2018, Shi *et al.* [1] have established TD-HITS and TS-LDA schemes that aimed at identifying the hot topics. Accordingly, the suggested model has automatically recognized the count of topics and it also identified the associated key posts. Moreover, this scheme has detected the individuals who were spreading the hot event topics based on the user and post information. At last, the effectiveness of the presented model was presented with respect to the spreaders and events.

In 2016, Yang *et al.* [2] have demonstrated a novel HTD and extraction scheme for detecting hot topics depending on the topic and language models. Moreover, the significance of each microblogs was computed and consequently, the topic models were generated. Thus, the developed scheme assisted in managing and monitoring the hot topics.

In 2018, Wei *et al.* [3] have undergone research by means of TMHTD approach that was implemented in an "Apache Spark environment". The introduced TMHTD model has included two stages: "micro-clustering and the macro-clustering". For improving the precision, three optimization techniques were presented along with two-phase HTD. At last, the betterment of the adopted scheme was validated in terms of accuracy.

In 2019, Bok *et al.* [4] have developed an innovative approach for predicting the hot events in the near-future in social media. The adopted model has detected the hot topics by deriving the candidate keywords from the posts via the modified TF-IDF model. Moreover, the hot topic index for the entire candidate keywords was computed based on user interests and consequently, the predictions were carried out with regard to time. Finally, an assessment was done that established the supremacy of the designed technique.

In 2017, Shi *et al.* [5] have designed a "hot event evolution" scheme that considered the user interest distribution for finding the users' interestingness. In addition, this approach has resolved the issues associated with data sparsity and it has enhanced the accuracy on event prediction as well. In addition, an automated filtering scheme was exploited, which eliminated the occasions of general events. Thus, the quality and efficiency in extracting the hot events were improved by the proposed model.

3. Framework Of Hot Event Detection: Problem Definition

Assume Do as a document set over an interval of T, where

$w: Do = \{Do_1, Do_2, Do_3 .. Do_T\}$

$Do_i \cap Do_j = \phi;\ 1 \le i \ge j \le T$. The documents are simplified into the bag of keywords or feature terms. Further, the object $d_i \in Do_t$ is indicated by $d_i = \{f_{i,1}, f_{i,2}, ...f_{i,k}\}$, in which $d_{i\,f_{i,k}}$ denotes the feature item, which is extracted from d_i by deploying the pre-processing model. Fig. 1 shows the architecture of the presented model.

The adopted scheme includes the following phases

(i) Text pre-processing
(ii) Feature selection and weighting
(iii) Text model construction
(iv) Optimization assisted Topic clustering

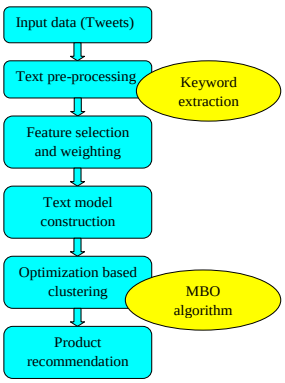

Figure 1: The framework of the presented model

TEST Engineering & Management

May-June 2020
ISSN: 0193-4120 Page No. 3824-3830

Text Preprocessing

Pre-processing is the initial step in detecting the hot topics of the micro blog text. Text pre-processing comprises of "word segmentation and filtering of words", which has no meaning. Depending on the language to be processed, it is essential to pick up the related word segmentation tool, like "ICTCLAS for a Chinese corpus". After segmenting the words, the senseless words will be neglected. Further, the feature items (i.e. group of words) are attained after pre-processing the micro blog documents.

Assume an example that includes 4 documents doc_1, doc_2, doc_3 and doc_4 . The extracted feature items from every document are indicated by Table II.

Table 1: Feature Items Extracted From Each Document

Features 1					No. of keywords
doc_1	f_{11}	f_{12}	f_{13}		K=3
doc_2	f_{21}	f_{22}			K=2
doc_3	f_{31}	f_{32}	f_{33}	f_{34}	K=4
doc_4	f_{41}				K=1

Number of texts= 10

adjectives are elected as the features for addressing every document in common micro blog HTD. Once the feature selection process is completed, the feature weight's values should be computed. There are two major techniques for evaluating the feature weights. They are "Boolean weights and TF-IDF weights" [18].

Generally, the VSM [19] is deployed for expressing micro blog text. Here, a feature vector space of every text is extracted that is portrayed as $U(doc_1)=(f_{11}, w_{11}, f_{12}, w_{12}, \ldots f_{1k}, w_{1k})$.

Subsequently, the weight of every feature item is computed by means of the normalized TF-IDF function as shown in Eq. (1).

$$w_{i,k}=\frac{ff_{ik}\times\log\left(\frac{N}{n_{ik}}+0.01\right)}{\sqrt{\sum_{j=1}^{k} ff_{jk}\times\left(\log\left(\frac{N}{n_{jk}}+0.01\right)\right)^2}} \quad (1)$$

Accordingly, $i=1,2\ldots No.of\ docs=4$, $k=1,2\ldots K$, in which K denotes the count of keywords in the document, ff_{ik} specifies the count of occurrence of ff_{ik} in doc_i , ff_{jk} specifies the count of occurrence of ff_{jk} in doc_j , N denotes the count of texts, here, 10 (as per example), n_{ik} denotes the count of texts that have f_{ik} . E.g. If $f_{ik}=f_{11}, f_{22}, f_{31}, f_{41}$, then $n_{ik}=4$.

For determining the contribution of feature items, the frequency function (f_{ik}), burst (f_{ik}), contribution function $CF(f_{ik})$ and score of the document $Score(doc_i)$ has to be considered.

Frequency function: It is utilized for evaluating how many texts include the feature f_{ik} as given in Eq. (2).

$$fr(f_{ik})=\frac{N(f_{ik})}{num} \quad (2)$$

$$N(f_{ik})=\frac{no.of\ documents\ containing\ f_{ik}}{no.of\ documents} \quad (3)$$

As per the above example,

$$N(f_{ik})=\frac{4}{4} \quad (\because f_{ik}\ is\ in\ all\ 4\ documents) \quad (4)$$

Burst (f_{ik}): Assume the given inputs for the time stamp "t". Therefore, $fr(f_{ik})_r$ is computed for all "t", in which $r=1,2..t$. The burst is computed as per Eq. (5).

$$Burst(f_{ik})=\begin{bmatrix} fr(f_{ik})_2\times\frac{1}{\sum_{r=1}^{1} fr(f_{ik})_r}; \\ fr(f_{ik})_3\times\frac{2}{\sum_{r=1}^{2} fr(f_{ik})_r} \\ \vdots \\ fr(f_{ik})_t\times\frac{(t-1)}{\sum_{r=1}^{1} fr(f_{ik})_r} \end{bmatrix} \quad (5)$$

Accordingly, the contribution function $CF(f_{ik})$ is portrayed for determining the contributor f_{ik} to the topic, which is done by computing its frequency and burst. Eq. (6) shows the computation of $CF(f_{ik})$.

$$CF(f_{ik})=fr(f_{ik})\times Burst(f_{ik}) \quad (6)$$

Then, the topic related likelihood for every text in dataset is computed as given in Eq. (7), where, score (d_i) specifies the topic related likelihood of d_i , $w_{i,k}$ denotes the weight of the feature f_{ik} and $CF(f_{ik})$ returns the contribution value associated with f_{ik} .

$$Score(doc_i)=\sum_{k=1}^{K} w_{i,k}\times CF(f_{ik}) \quad (7)$$

Thereby, the score formula formulates the topic related likelihood of doc_i .

May-June 2020
ISSN: 0193-4120 Page No. 3824-3830

4. Topic Identification: Optimization assisted Clustering Process

Micro Clustering

In the traditional model, micro-clustering was carried out depending on the single pass algorithm [1]. In this work, the micro clustering process is carried out with the involvement of optimization logic by exploits MBO algorithm that determines the optimal centroid.

Depending on the implemented logic, the selective documents like doc_1, doc_2, weight matrix of selective documents and keywords are provided as input. Accordingly, the KWV, FSV, and OPV are evaluated as given in Eq. (8), Eq. (9) and Eq. (10), correspondingly. For e.g. if $m = 2$, the KWV, FSV, and OPV can be determined as given in Table III, Table IV, and Table V respectively.

$$w_{11}^{KWV} = \frac{frequency\ of\ f_{11}}{No.of\ selected\ documents \times M} \quad (8)$$

$$w_{11}^{FSV} = \frac{frequency\ of\ f_{11}\ in the 1^{st}\ sentence\ of\ doc_1}{No.of\ words in\ doc_1} \quad (9)$$

$$w_{11}^{OPV} = \frac{frequency\ of\ f_{11}\ in the other\ sentence\ of\ doc_1}{No.of\ words in\ doc_1} \quad (10)$$

Table 2: Demonstration On Kwv Computation

doc_1	f_{11}	f_{13}	→	w_{11}^{KWV}	w_{13}^{KWV}
doc_2	f_{31}	f_{32}		w_{31}^{KWV}	w_{32}^{KWV}

Table 3: Demonstration On Fsv Computation

doc_1	f_{11}	f_{13}	→	w_{11}^{FSV}	w_{13}^{FSV}
doc_2	f_{31}	f_{32}		w_{31}^{FSV}	w_{32}^{FSV}

Table 4: Demonstration On Opv Computation

doc_1	f_{11}	f_{13}	→	w_{11}^{OPV}	w_{13}^{OPV}
doc_2	f_{31}	f_{32}		w_{31}^{OPV}	w_{33}^{OPV}

Thus, the centroid of every cluster is portrayed by means of the implemented scheme. For e.g.; for 2 clusters, the centroid is evaluated as specified in Table V. Accordingly, the centroid of every document is provided as input to the suggested MBO. Based on the example revealed in Table V, the solution is given in Fig. 2. Finally, the documents are clustered with the chosen centroids.

Table 5: Example Centroid Determination On Considering 3 Documents

	f_1	f_2
doc_1	w_{11}^{KWV}, w_{11}^{FSV}, w_{11}^{OPV}	w_{13}^{KWV}, w_{13}^{FSV}, w_{13}^{OPV}
doc_3	w_{31}^{KWV}, w_{31}^{FSV}, w_{31}^{OPV}	w_{32}^{KWV}, w_{32}^{FSV}, w_{33}^{OPV}
doc_4	w_{41}^{KWV}, w_{41}^{FSV}, w_{41}^{OPV}	w_{43}^{KWV}, w_{43}^{FSV}, w_{43}^{OPV}

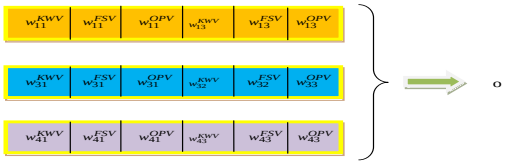

Figure 2: Solution encoding: Micro clustering

Macro Clustering

After the micro clustering process, macro clustering takes place. The unique documents containing each keyword are chosen from the micro-clusters. Generally, for evaluating the topic frequency, heat value of topic is deployed during the clustering process [20]. Moreover, for enhancing the accuracy of clustering resultants, the lower frequency topics are filtered and the counts of micro-topics are minimized before clustering the micro-topics into macro-topics. The election of topics depending on the topic heat is specified below:

Initially, consider $T = \{top_1, top_2, ...top_U\}$ as a micro-topic set, where V indicates the topic count. Consider $Heat(top_i)$ as the heat of top_i, which is modeled as per Eq. (11). Here, $PN(top_i)$ indicates the count of participators of top_i, $RN(top_i)$ and $CN(top_i)$ denotes the count of re-tweets and comments of top_i, and $MN(top_i)$ points out the micro blog count in top_i.

$$Heat(top_i) = PN(top_i) + CN(top_i) + RN(top_i) + MN(top_i) \quad (11)$$

Subsequently, the topics are arranged based on the heat values in downward order. "The jump point of the topic heat is defined as the sudden change point of the topic heat compared to the total topic heat, and the heat change rate is used to find the jump point". Consider $Rate(top_i)$ the heat change rate of top_i, which is computed as per Eq. (12).

$$Rate(top_i) = \left| \frac{Heat(top_{i+1}) - Heat(top_i)}{Heat(top_i)} - \frac{Heat(top_i) - Heat(top_{i-1})}{Heat(top_i)} \right| \quad (12)$$

Thus, further filtering is carried out based on $Rate(top_i)$. For threshold β, if $Rate(top_i) < \beta$, the heat value topics that are lesser or equal than the value of jump point are filtered. Or else, top_i can be considered as a micro-topic variable with a higher frequency for clustering. Therefore the count of

May-June 2020
ISSN: 0193-4120 Page No. 3824-3830

dimensions of the topic set is diminished from V to $v\ (v<V)$.

Especially, the cluster size cs and the frequency vector space of keyword in every document denoted by ff_{ik} are provided as solutions as specified in Fig. 3. Thus, the hot topic events are optimally detected in this proposed work.

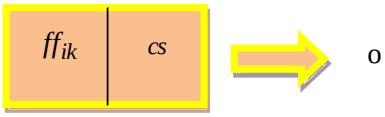

Figure 3: Solution encoding: Macro clustering

5. MBO Algorithm

The migration process of monarch butterflies include certain aspects like (i) "The overall population of monarch butterflies is limited to Land 1 and Land 2. (ii) The migration operator produces every single child monarch butterfly in both the regions such as Land 1 and Land 2. (iii) In the MBO approach [17], the parent butterfly will pass away when a new one is created so as to maintain a constant population (iv) The quality and performance of the monarch butterfly is assured even with the increased generations".

Migration Operator: Assume MP as the total count of the population, $Maxgen$ points out the utmost generation and c denotes the proportion of butterflies in Land 1 and 2. Here, Land 1 and 2 termed as subpopulation1 and 2 are denoted by $(S1)$ and $(S2)$.

$$o_{i,y}^{t+1} = o_{s1,y}^{t} \quad (13)$$

Eq. (13) reveals the migration process, in which, $o_{i,y}^{t+1}$ points out y^{th} component of o_i of monarch butterfly i at generation $t+1$. Also, $o_{s_1,y}^{m}$ specifies the y^{th} component of o_{s_1}, o_{s_1} denotes the generated location of monarch butterfly s_1, which is arbitrarily selected from $S1$. The parameter t symbolizes the current generation. By means of Eq. (13), the component y for novel monarch butterfly is determined when $s \le p$. The value for s is formulated as per Eq. (14), where $time$ denotes the migration time and ran indicates an arbitrary value taken from the uniform distribution.

$$s = ran * time \quad (14)$$

The component y for newly generated monarch butterfly is initiated when $s > c$ as specified in Eq. (15).

$$o_{i,y}^{t+1} = o_{s_2,y}^{t} \quad (15)$$

In Eq. (15), $o_{s_2,y}^{t}$ specifies the y^{th} component of o_{s_2}. o_{s_2} denotes the new location for the butterfly s_2, which is arbitrarily chosen from $S2$. Moreover, by regulating the ratio p, the direction of the migration operator could be sustained. Here, p is 5/12 for the current calculation and it determines if $S1$ and $S2$ can be chosen.

Butterfly Balancing Operator: The balancing operator is exploited for updating the location of the monarch butterflies. When a random value ran is equivalent or less than c for all components, then the memory of the monarch butterfly x is updated as revealed in Eq. (16).

$$o_{x,y}^{t+1} = o_{best,y}^{t} \quad (16)$$

In Eq. (16), $o_{x,y}^{t+1}$ indicates the y^{th} component of o_x for the generation $t+1$. Correspondingly, $o_{best,y}^{t}$ symbolizes the y^{th} component of o_{best} that indicates the best monarch butterfly in Land 1 and Land 2. On the other hand, when c is lesser than ran, the memory is updated as given in Eq. (17).

$$o_{x,y}^{t+1} = o_{s_3,y}^{t} \quad (17)$$

In Eq. (17), $o_{s_3,y}^{t}$ portrays y^{th} component of o_{s_3} which is randomly chosen in Land 2. At this point, $s_3 \in \{1,2,\ldots,MP_2\}$.

$$o_{x,y}^{t+1} = o_{x,y}^{t+1} + \lambda \times (fn_y - 0.5) \quad (18)$$

In case if $rn > BAR$, the memory is updated as given in Eq. (18), in which BAR represents the balancing value of butterfly. The variable fn indicates the walk steps of the monarch butterfly x. It can be defined via Levy flight as defined in Eq. (19) and (20).

$$fn = Levy(o_x^t) \quad (19)$$

$$\lambda = H_{\max}/t^2 \quad (20)$$

The weighting element λ is determined based on Eq. (20), in which $H_{\max}$ signifies the value that a single monarch butterfly can move for maximum walk steps.

II. Product Recommendation System: Integration of Detected Hot Topic

Product recommendation (PR) can be defined "as the semantic similarity between recommended products and grouped products" as given in Eq. (21). In Eq. (21), SS indicates the semantic similarity, RP specifies the recommended products and GP denotes the grouped products.

$$PR = SS(RP\ and\ GP) \quad (21)$$

May-June 2020
ISSN: 0193-4120 Page No. 3824-3830

6. Results and Discussion

Simulation Procedure

The presented HTD system integrated with product recommendation was executed in **JAVA** and the corresponding outcomes were achieved. The dataset was downloaded from the link "http://u.cs.biu.ac.il/~nlp/resources/downloads/twitter-events/." Here, the algorithmic analysis was carried out by varying the weighting element λ of the MBO model from 0.2, 0.4, 0.6 and 0.8 with respect to positive measures such as "accuracy, sensitivity, PPV" and negative measures such as "FDR and FNR". In addition, convergence analysis was carried out that demonstrated the macro-clustering and micro-clustering analysis of the proposed work.

Performance Analysis

The performance analysis of the MBO model for varying values of λ is summarized in Table VI. On observing the attained outcomes, the accuracy attained at $\lambda = 0.8$ is higher, which is 0.13%, 0.71%, and 0.14% better than the accuracy attained at λ =0.2, 0.4 and 0.6. Similarly, the value of sensitivity attained at $\lambda = 0.2$ is 0.4%, 0.16% and 0.09% superior to the values of λ at 0.4, 0.6 and 0.8. The PPV is found to be higher at $\lambda = 0.8$, whose value is 0.24%, 0.46%, and 0.09% better than the values of λ at 0.2, 0.4 and 0.6. On analyzing the negative measures, the value of FDR attained at $\lambda = 0.8$ is found to be minimal, which is 8.55%, 15.11% and 3.35% superior to the values of λ at 0.2, 0.4 and 0.6. Thus, the betterment of the proposed model is validated under all these given variations.

Table 6: Performance Of The Proposed Model By Varying λ

Measures	$\lambda = 0.2$	$\lambda = 0.4$	$\lambda = 0.6$	$\lambda = 0.8$
Accuracy	0.958819151	0.953290226	0.958728604	0.960055
Sensitivity	0.984420534	0.980472517	0.982869832	0.983449
PPV	0.972369035	0.970233683	0.973854462	0.974731
FDR	0.027630965	0.029766317	0.026145538	0.025269
FNR	0.015579466	0.019527483	0.017130168	0.016551

Convergence Analysis

Fig. 4 demonstrates the convergence analysis by considering the macro clustering and macro clustering of the proposed work by varying the value of $\lambda = 0.2$ for 100 iterations.

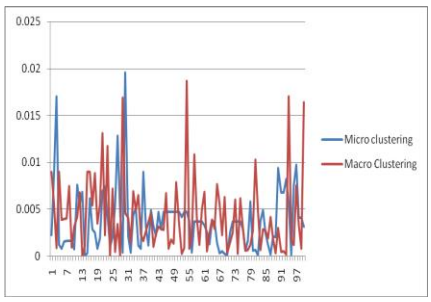

Figure 4: Convergence analysis of proposed MBO model by varying the value of $\lambda = 0.2$

7. Conclusion

This paper has established an automated system that detected the hot topics together with the product recommendation system, where the products associated with the events was suggested. Accordingly, clustering was carried out with the aid of optimization that enhanced the functionality of micro-clustering and macro-clustering. Here, MBO algorithm was deployed for optimization purpose that enhanced the accuracy of the HTD. Finally, from the outcomes, the value of accuracy attained at $\lambda = 0.8$ is higher, which is 0.13%, 0.71% and 0.14% better than the values of λ at 0.2, 0.4 and 0.6. Similarly, the value of sensitivity attained at $\lambda = 0.2$ is 0.4%, 0.16% and 0.09% superior to the values of λ at 0.4, 0.6 and 0.8. Likewise, the PPV is found to be higher at $\lambda = 0.8$, whose value is 0.24%, 0.46% and 0.09% better than the values of λ at 0.2, 0.4 and 0.6. Thus, the superiority of the presented approach was proved in an effective manner.

References

[1] L. Shi, Y. Wu, L. Liu, X. Sun and L. Jiang, "Event detection and identification of influential spreaders in social media data streams," Big Data Mining and Analytics, vol. 1, no. 1, pp. 34-46, March 2018.

[2] Liang Yang, Hongfei Lin, Yuan Lin, Shengbo Liu, "Detection and Extraction of Hot Topics on Chinese Microblogs",Cogn Comput, 17 February 2016.

[3] Wei Ai, Kenli Li, Keqin Li, "An effective hot topic detection method for microblog on spark", Applied Soft Computing, vol. 70, pp. 1010-1023, September 2018.

[4] Kyoungsoo Bok, Yeonwoo Noh, Jongtae Lim, Jaesoo Yoo, "Hot topic prediction considering infuence and expertise in social media", Electronic Commerce Research, 01 January 2019.

[5] Lei-Lei Shi ; Lu Liu ; Yan Wu ; Liang Jiang ; James Hardy, "Event Detection and User Interest Discovering in Social Media Data Streams", IEEE Access, vol. 5, pp. 20953 - 20964, 27 February 2017.

May-June 2020
ISSN: 0193-4120 Page No. 3824-3830

[6] Anuja Arora, Shivam Bansal, Chandrashekhar Kandpal, Reema Aswani, Yogesh Dwivedi, "Measuring social media influencer index- insights from facebook, Twitter and Instagram", Journal of Retailing and Consumer Services, vol. 49, pp. 86-101, July 2019.

[7] Xi Chen, Xiangmin Zhou, Timos Sellis, Xue Li, "Social event detection with retweeting behavior correlation", Expert Systems with Applications, vol. 114, pp. 516-523, 30 December 2018.

[8] Vinicius Monteiro de Lira, Craig Macdonald, Iadh Ounis, Raffaele Perego, Valeria Cesario Times, "Event attendance classification in social media", Information Processing & Management, vol. 56, no. 3, pp. 687-703, May 2019.

[9] Flora Amato, Vincenzo Moscato, Antonio Picariello, Giancarlo Sperlì, "Extreme events management using multimedia social networks", Future Generation Computer Systems, vol. 94, pp. 444-452, May 2019.

[10] Zhicong Tan, Peng Zhang, Jianlong Tan, Li Guo, "A Multi-layer Event Detection Algorithm for Detecting Global and Local Hot Events in Social Networks", Procedia Computer Science, vol. 29, pp. 2080-2089, 2014.

[11] Pengpeng Zhou, Zhen Cao, Bin Wu, Chunzi Wu, Shuqi Yu, "EDM-JBW: A novel event detection model based on JS-ID′Forder and Bikmeans with word embedding for news streams", Journal of Computational Science, vol. 28, pp. 336-342, September 2018.

[12] S. Zoletnik, T. Szabolics, G. Kocsis, T. Szepesi, D. Dunai, "EDICAM (Event Detection Intelligent Camera)", Fusion Engineering and Design, vol. 88, no. 6–8, pp. 1405-1408, October 2013.

[13] Xueming Qian, Mingdi Li, Yayun Ren, Shuhui Jiang, "Social media based event summarization by user–text–image co-clustering", Knowledge-Based Systems, vol. 164, pp. 107-121, 15 January 2019.

[14] Tiance Dong, Chenxi Liang, Xu He, "Social media and internet public events", Telematics and Informatics, vol. 34, no. 3, pp. 726-739, June 2017.

[15] Manar Alkhatib, May El Barachi, Khaled Shaalan, "An Arabic social media based framework for incidents and events monitoring in smart cities", Journal of Cleaner Production, vol. 220, pp. 771-785, 20 May 2019.

[16] Nicholas Roxburgh, Dabo Guan, Kong Joo Shin, William Rand, Jing Meng, "Characterising climate change discourse on social media during extreme weather events", Global Environmental Change, vol. 54, pp. 50-60, January 2019.

[17] Gai-Ge Wang and Suash Deb, "Monarch Butterfly Optimization", Neural Computing and Applications, 2015.

[18] G. Salton, "Automatic text processing: The transformation, analysis, and retrieval of," Reading: Addison-Wesley, 1989.

[19] Y. Zhu and Y. Hu, "Enhancing search performance on gnutella-like p2p systems," IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 12, pp. 1482–1495, 2006.

[20] C. Yang, J. Yang, H. Ding, and H. Xue, A Hot Topic Detection Approach on Chinese Microblogging. 2013.