HSPKNN: An Effective and Practical Framework for Hot Topic Detection of Internet News

Ping Lu[1], Shengyu Liu[2], Zhenjiang Dong[1], Shengmei Luo[1], Lixia Liu[1], Haodi Li[2], Qingcai Chen[2]
[1]ZTE Corporation, Nanjing, China
[2]Intelligent Computing Research Center, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China
lu.ping@zte.com.cn, liushengyu.hit@gmail.com, dong.zhenjiang@zte.com.cn, luo.shengmei@zte.com.cn,
liu.lixia@zte.com.cn, lhd911107@gmail.com, qingcai.chen@gmail.com

Abstract*—With the rapid growth of information on the Internet, many Single-Pass based clustering methods are used in topic detection and tracking (TDT) because of Single-Pass's characteristics of incremental processing. In Single-Pass based methods, similarities between the feature vectors of news reports and the cluster centers of historical topics are calculated. The accuracy of TDT will be affected if the cluster centers can not precisely represent the topics. To overcome the shortcoming of Single-Pass based methods. This paper proposes an effective and practical framework for hot topic detection of Internet news. Firstly, news report streams are partitioned into segments by a time window, and then an agglomerative hierarchical clustering algorithm is used to acquire candidate topics. Finally, an algorithm fusing Single-Pass and KNN is proposed to detect topics from the candidate topics. Furthermore, in order to make it easier for the users to understand what the topics discuss, an algorithm generating descriptive labels for detected topics is proposed. Experimental results show that the proposed framework can outperform Single-Pass based methods and agglomerative hierarchical clustering based methods for TDT. In addition, the proposed framework has been used in the TDT module of an application system. Both the experimental results and application system demonstrate the effectiveness and practicality of the proposed framework.

***Index Terms*—TDT, Single-Pass, Agglomerative Hierarchical Clustering.**

I. INTRODUCTION

The advent of the Internet makes information increase explosively. It is impossible for people to organize and manage such a large amount of information manually. Therefore, an effective tool that can automatically organize and manage massive information is urgently needed. TDT[1] studies how to detect hot topics and track the development of events related to the topics. It can make people know the hot topics and details of events related to the topics. As an important task of TDT, topic detection studies how to cluster together the new reports that discuss the same topic in the news stream and text clustering algorithm is the key technology of topic detection. However, news reports are dynamic information flow in chronological order, general clustering algorithm can't timely cluster the news reports one by one. The clustering algorithms used in topic detection are based on incremental ways. As an efficient and widely used incremental clustering algorithm, Single-Pass[2] deals with one news report each time in the order of input. According to the similarities between an input news report and the centers of existing clusters, it assigns the news report to an existing cluster or creates a new cluster. The similarities between the feature vector of news reports and centers of historical topics have an important impact on the performance of topic detection. Because each topic may contain different aspects of the narrative, cluster center can't represent a topic precisely. Furthermore, most of the news hot topic detection methods just list the news reports about the topic and don't extract descriptive words or abstract information of the topics. Therefore, users can't easily understand what the topics discuss.

In this paper, an effective and practical framework for hot topic detection is introduced, which combines an agglomerative hierarchical clustering algorithm and an algorithm fusing Single-Pass and KNN. The proposed framework can overcome the shortcomings of Single-Pass based methods and thus achieve superior performance over them. Furthermore, an effective label generation algorithm is proposed in the framework.

The paper is organized as follows. Section II reviews the related works. Section III describes the proposed framework in details. Experimental results are demonstrated in Section IV. Section V concludes this paper.

II. RELATED WORKS

Most of early researches mainly concentrate on the selection of clustering algorithms. Yang et al. [3] used hierarchical clustering for offline topic detection. Allan et al. [4] applied Single-Pass clustering to topic detection and achieved good performance. There exists extensive works extending or integrating hierarchical clustering and Single-Pass clustering. Most of above mentioned methods based on the traditional vector space model and ignored some topic-indicative terms. Lam[5]、Yiming Yang[6]、James Allan[7] and Kumaran[8] used named entity recognition technology in hot topic detection and proved that named entity can improve the performance of hot topic detection. Gupta[9] proposed a Single-Pass based clustering algorithm GenIc for incremental processing of news reports. Genic partitions data stream into blocks, and decide whether to retain or delete the cluster centers according to the evaluation of cluster centers. For the inaccuracy of incremental cluster at the initial time，Shui Y.[10] proposed a TDT method of periodic classification and Single-Pass cluster.

In Single-Pass based incremental clustering algorithm, the similarities between the news reports and cluster centers of historical topics greatly affect the performance of the topic detection. In this paper, the framework HSPKNN selects the top K historical topics similar to the candidate topics rather than the most similar one. Each news report of the candidate topic is assigned to one of the top K historical topics through KNN classifier and decisions of all news reports are fused to get the final decision of the candidate topic. That can eliminate the negative impact of using cluster center to represent a topic. Most previous methods don't generate descriptive words for the detected topics. A label generation algorithm is proposed in HSPKNN to generate descriptive labels for the detected topics.

III. Flow of the Framework

Flow of the framework for hot topic detection is shown in Fig.1.

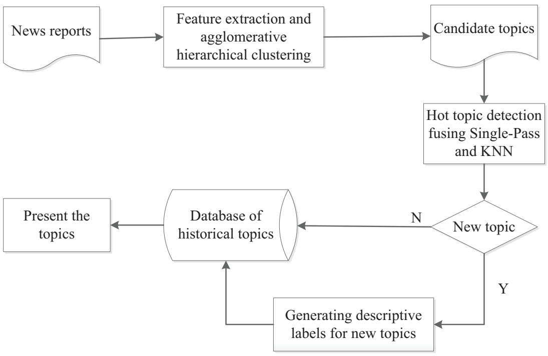
News reports
Feature extraction and agglomerative hierarchical clustering
Candidate topics
Hot topic detection fusing Single-Pass and KNN
New topic
N
Y
Database of historical topics
Present the topics
Generating descriptive labels for new topics

Figure 1 The flow of the framework for hot topic detection

A. News document presentation and similarity calculation

The news report streams are firstly partitioned into segments by a time window. We borrow the idea of [11] to present the news documents. Instead of using a single term vector as document representation, the terms are split into three semantic classes, including space, character and content and the classes are weighed separately. Each news report in a period of time is presented by three feature vectors:

1. A vector of place names presenting the locale of the news document.
2. A vector of person names and organization names presenting the character of the news document.
3. A vector of other words presenting the content of the news document.

The similarity between vectors of place names or between vectors of person names and organization names is defined as,

$$sim(v_1, v_2) = \begin{cases} 1 & v_1 \; and \; v_2 \; have \; common \; components \\ 0 & else \end{cases} \quad (1)$$

where v_1 and v_2 are vectors of place names or vectors of person names and organization names.

The content vector of news document is got by calculating TF-IDF value of each term. We take the position of the terms into account when calculating TF-IDF value. Weights of terms in the document title are larger than that in the text.

The similarity between two news document are defined as,

$$sim(d_1, d_2) = \alpha sim(v_{con1}, v_{con2}) + \beta sim(v_{pl1}, v_{pl2}) + \lambda sim(v_{ch1}, v_{ch2}) \quad (2)$$

where d_1 and d_2 are two news documents, $sim(v_{con1},v_{con2})$ is similarity between content vectors, $sim(v_{pl1}, v_{pl2})$ is similarity between place vectors and $sim(v_{ch1}, v_{ch2})$ is similarity between character vectors, α , β , λ are weighting coefficients and $\alpha + \beta + \lambda = 1$.

B. Acquiring candidate topics

Since news reports on the same topic often appear in the same time period, we introduce time window referring to a period of time. All news reports in a time window are divided into several candidate topics by agglomerative hierarchical clustering. Each candidate topic may be a new topic or a continuation of a historical topic.

C. Topic detection and tracking

Similarities between a candidate topic and historical topics should be calculated to determine whether the candidate topics is s new topic or a continuation of a historical topic. Since topic has a life cycle, relative to news reports that have a long time interval, time adjacent news reports are more likely to discuss the same topic. Therefore, it is not necessary to calculate the similarities between a candidate topic and all historical topics when determining whether the candidate topic is a new topic. Otherwise, it will cause a great deal of useless computation and affect the performance of topic detection. From the above, we introduce a time window to limit the number of historical topics to be compared with candidate topics.

Although the number of historical topics to be compared is limited by a time window, it is still very large. Single-Pass is an efficient cluster algorithm; it has an obvious advantage of processing massive data. In Single-Pass, the accuracy of similarities between candidate topics and historical topics greatly affect the performance of the topic detection. An algorithm fusing Single-Pass and KNN is proposed to eliminate the negative impact of using cluster center to represent topic. Figure 2 shows the flow of hot news topic detection in the framework HSPKNN. The dark geometries present topics and the white geometries present news reports. The algorithm fusing Single-Pass and KNN is given in Algorithm 1.

The four parameters θ_1 , θ_2 , K_1 and K_2 used in Algorithm 1 are determined experimentally from the training corpus.

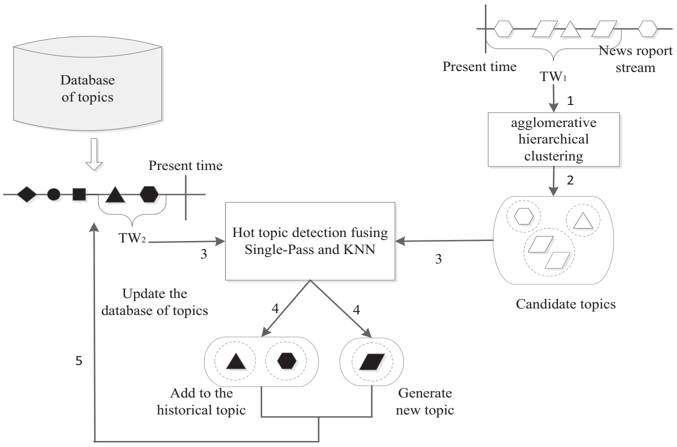
Database of topics
Present time
TW2
3
Hot topic detection fusing Single-Pass and KNN
Present time
News roport stream
TW1
1
agglomerative hierarchical clustering
2
3
Candidate topics
Update the database of topics
4
4
5
Add to the historical topic
Generate new topic

Figure 2 Flow of hot news topic detection

Algorithm 1 Hot topic detection algorithm fusing Single-Pass and KNN

```
Input: CT--set of candidate topics in time window $TW_1$.
       HT--set of historical topics in time window $TW_2$.
Output: T--set of topics after update.
  i = 0
  while i < CT.size()
    if ( HT.size() == 0 )
      Conclude that the candidate topic CT[i] is a new topic and then add it to the set HT
    else
      Calculate the similarities between the cluster center of CT[i] and that of each historical topic in HT. The
      historical topic most similar to CT[i] is HT[max], and the similarity between CT[i] and HT[max] is
      $sim_{max}$.
      if ($sim_{max} < \theta_1$)
        Conclude that the candidate topic CT[i] is a new topic and then add it to the set HT.
      else
        Select the top $K_1$ historical topics similar to the candidate topics, and D is the set of all
        news documents from the top $K_1$ historical topics.
        while( j < CT[i].size() )
          Add top $K_2$ news documents most similar to CT[i][j] from D to a empty set P.
          For each historical topic in HT, a similarity is calculated; take HT[k] for example,
          the similarity is defined as, Sim(CT[i][j], HT[k]) = $\sum_{d \in HT[k],\, d \in P} Sim(CT[i][j]), d)$
          $Score_{\arg\max_{HT[k]} sim(CT[i][j], HT[k])}++$
      $l = \arg\max_{k} Score_{HT[k]}$
      if ( $\frac{Score_{HT[l]}}{\sum_k Score_{HT[k]}} > \theta_2$ )
        Conclude that the candidate topic is a continuation of the historical topic HT[l], and then add it into
          the historical topic HT[l]. Update the cluster center of HT[l] using algorithm 2 in the next
          section and remove the terms whose weight less than 0.01 from the cluster center vector.
      else
        Conclude that the candidate topic CT[i] is a new topic and then add it to the set HT.
```

Algorithm 2 Update-Cluster-Center

```
Input: hisCluster--cluster of a historical topic to be updated.
       canCluser--cluster of a candidate to be added into hisCluster.
Output: hisCenter--cluster center of the updated hisCluster
  1: hisTS is the term set of hisCluster, canTS is the term set of canCluster.
  2: If t ∈ hisCenter ∩canCenter, the weight of term t is,
       $w_t = (w_{ht} \cdot hisCluster.size() + w_{ct} \cdot canCluster.size()) / hisCluster.size() + canCluster.size()$
     where $w_{ht}$ is the weight of t in hisClusetr, and $w_{ct}$ is the weigth of t in canClusetr.
  3: If t ∈ hisCenter －canCenter, the weight of term t is,
       $w_t = w_{ht} \cdot hisCluster.size() / hisCluster.size() + canCluster.size()$
  4: If t ∈canCenter －hisCenter, the weight of term t is,
       $w_t = w_{ct} \cdot hisCluster.size() / hisCluster.size() + canCluster.size()$
```

Algorithm 3 DCF-FPGrowth label generation algorithm

```
Input: T ={$d_1$, $d_2$, …,$d_n$} -- document set of a topic.
       $\alpha$ , $\beta$ -- weighting coefficients of DCF and FP-Growth.
       KW = {$kw_1$, $kw_2$, …, $kw_{|KW|}$}--top K high frequency keywords in T.
Output: Descriptive labels of topic T.
  1: Calculate the similarities between keywords in KW and the cluster center of topic T.
  2: Add top p keywords similar to the cluster center of topic T to an empty set KW-DCF.
     KW-DCF = {<$kw_1$, score1($kw_1$)>, …, <$kw_i$, score1($kw_i$)}>,… , <$kw_p$, score1($kw_p$)>}, score1($kw_i$) is the
     normalized similarity between $kw_i$ and the cluster of topic T.
  3: Take every document in topic T as a set of keywords, and mine the maximal frequent item set X in T.
     X = { $item_1$, …, $item_i$, …, $item_{|X|}$ }
  4: Grade every keyword in set X using the following equation.
       $score2(item_i) = (\sum_{k=1\cdots m} f(I_k, item_i) * |I_k|) * (1 + posScore(item_i) * pageScore(T, item_i))$
       $f(I_k, item_i) = \begin{cases} 1 & item_i \in I_k \\ 0 & else \end{cases}$
       $pageScore(T, item_i) = \frac{1}{n} \sum_{k=1}^{n} td(item_i, d_k)$
     where posScore($item_i$) is the weight of Part-of-speech of $item_i$, td($item_i$, $d_k$) is the TF-IDF value of $item_i$ in
     document $d_k$, $I_k$ is a frequent item set mined in T. The set KW-FP = {<$item_1$, score2($item_1$)>, …, <$item_j$,
     score2 ($item_j$) >, …, <$item_{|X|}$, score2 ($item_{|X|}$)>}.
  5: Calculate the weight of every candidate label in KW-DCF∪KW-FP, and the weight of $label_i$∈KW-DCF
     ∪KW-FP is defined as weight($label_i$) = $\alpha$ ×score1($label_i$)+ $\beta$ ×score2($label_i$).
  6: A certain number of candidate labels with the highest weights are selected as the descriptive labels of topic
     T.
```

D. Updating cluster center of the topics

The document set of a topic is dynamic, and focus of a topic is constantly changing. Therefore, when merging the cluster of a historical topic and the cluster of a candidate topic, cluster center should be updated. The updated cluster center can represent the topic more accurately. The Update-Cluster-Center algorithm is given in algorithm 2.

E. Generating descriptive labels for topics

In order to make it easier for the users to understand what the topics discuss, a method based on combination strategy, i.e. combination of the DCF(Description Comes First)[12]and FPGrowth(frequent pattern growth)[13] is proposed to generate descriptive labels for the detected topics. The proposed algorithm is given in Algorithm 3.

IV. EXPERIMENTS

In this paper, we implement Single-Pass(SP) and agglomerative hierarchical clustering(AHC). The proposed method is evaluated and compared with SP and AHC on three datasets.

A. Datasets

Table 1 gives the statistics of datasets.

Table 1 Statistics of datasets

Dataset No.	Number of report news	Number of Topics
1	120	7580
2	24872	2548
3	723	38

We collected dataset 1 from news topics module of sina (http://news.sina.com.cn/zt/). Dataset 1 is reliable because the news reports are edited manually by the website editors. Dataset 2 is SogouTDTE an evaluation corpus for TDT provided by Sogou Labs (http://www.sogou.com/labs/resources.html) . The web pages of SogouTDTE were collected from May 2008 to June 2008. Dataset 3 is extracted from the standard evaluation dataset TDT4. There are 27142 Chinese documents in TDT4 and 723 Chinese documents are annotated. The annotated documents are used to evaluate the proposed method.

B. Evaluation criterion

False alarm rate, missed alarm rate and the normalized detection error cost are adopted to evaluate the performance of the hot topic detection methods. The definition of false alarm rate and missed alarm rate of a single topic are shown in Eq. 2 and Eq. 3.

$$FA_i = \frac{the\,number\,of\,detected\,news\,reports\,irrelevant\,to\,topic\,i}{the\,number\,of\,all\,news\,reports\,irrelevant\,to\,topic\,i} \tag{2}$$

$$miss_i = \frac{the\,number\,of\,undetected\,news\,reports\,relevant\,to\,topic\,i}{the\,number\,of\,all\,news\,reports\,relevant\,to\,topic\,i} \tag{3}$$

The average false alarm rate and average missed alarm rate are defined as,

$$P_{FA} = \frac{\sum_{i=1}^{n} FA_i}{n} \tag{4}$$

$$P_{miss} = \frac{\sum_{i=1}^{n} miss_i}{n} \tag{5}$$

The detection error cost is defined as,

$$C_{Det} = C_{Miss} \cdot P_{Miss} \cdot P_{t\arg et} + C_{FA} \cdot P_{FA} \cdot P_{non-t\arg et} \tag{6}$$

where C_{FA} and C_{Miss} are cost coefficients of false alarm rate and missed alarm rate. P_{target} is the prior probability of target topic and $P_{non\text{-}target} = 1 - P_{target}$. C_{FA}, C_{Miss} and P_{target} are predefined value. In this paper ,they are separately set to 1.0, 0.1 and 0.02.

The detection error cost is usually normalized to a value between 0 and 1. The normalized detection error cost is defined as,

$$(C_{Det})_{norm} = \frac{C_{Det}}{\min\{C_{Miss} \cdot P_{t\arg et}, C_{FA} \cdot P_{non-t\arg et}\}} \tag{7}$$

$(C_{Det})_{Norm}$ is often used to evaluate the performance of the system. The smaller the value of $(C_{Det})_{Norm}$, the better the performance of the system.

C. Experimental results

We compared SP, AHC and HSPKNN on the aforementioned datasets. The overall performance is shown in Table 2.

Table 2 Performance results of SP, AHC and HSPKNN

Evaluation criterion	**Sina**		
	SP	**AHC**	**HSPKNN**
average false alarm rate	0.014	0.012	0.010
average missed alarm rate	0.386	0.235	0.198
$(C_{Det})_{Norm}$	0.385	0.288	0.247
	SogouTDTE		
	SP	**AHC**	**HSPKNN**
average false alarm rate	0.013	0.011	0.009
average missed alarm rate	0.326	0.220	0.194
$(C_{Det})_{Norm}$	0.350	0.240	0.230
	TDT4		
	SP	**AHC**	**HSPKNN**
average false alarm rate	0.014	0.011	0.009
average missed alarm rate	0.253	0.192	0.196
$(C_{Det})_{Norm}$	0.301	0.233	0.228

From the performance results in table 2, we can see that HSPKNN perform better than SP and AHC on all datasets. AHC performs better than SP, but AHC is not suitable for the actual application system due to its high time complexity. The threshold of topic detection in SP, AHC and HSPKNN is 0.3. The threshold is determined experimentally. 0.10, 0.11, 0.12, …, 1.00 are used as threshold to calculate the corresponding $(C_{Det})_{Norm}$. On the dataset Sina, the minimum value of $(C_{Det})_{Norm}$ is 0.274 when the threshold is 0.29. On SogouTDTE, the minimum value of $(C_{Det})_{Norm}$ is 0.251 when the threshold is 0.31. On TDT4, the minimum value of $(C_{Det})_{Norm}$ is 0.236 when the threshold is 0.32. Given all of that, we select 0.3 as the threshold in this paper.

Figure 3 shows the missed alarm rate corresponding to the false alarm rate on three datasets.

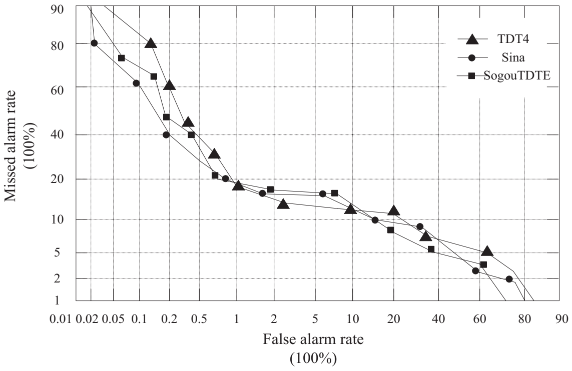

Figure 3 Missed alarm rate vs. False alarm rate

Table 3 gives some examples of descriptive labels generated by DCF-FPGrowth. We can see that the generated labels can reflect the content of the topic to some extent.

Table 3 Some examples of generated descriptive labels

Topic	Descriptive labels
Click of the Korean video "Gangnam Style" exceed 300 million and break the Guinness record.	JiangnanStyle, Youtube, Hot
A small plane crashed and caused two deaths in Air Show of Bandung, Indonesia.	Bandung, Indonesia, plane, crash
2012 U.S. presidential campaign.	Obama Romney, campaign
Avalanche of Manaslu in Nepal.	Nepal, Avalanche, accident

D. Applications

The proposed framework has been used into the topic detection module(http://www.haitianyuan.com/hotevent/) of the web knowledge service platform Haitianyuan (http://www.haitianyuan.com/) and provides users with hot topic presentation service. The proposed framework is shown to be an effective and practical way for hot topic detection.

V. Conclusions and Future Works

This paper proposes a framework for hot topic detection. It can overcome SP's shortcoming of using cluster center to represent the topic and achieves competitive performance over SP and AHC. Moreover, a label generation algorithm is proposed in the framework to generate descriptive labels for topics. For future work, the threshold used in HSPKNN is a fixed number determined experimentally, a dynamic threshold generation algorithm should be developed. Label generation in this paper is based on the weights of keywords in the topics, the semantic relationship should be considered in label generation.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China (No. 61173075 and 60973076), ZTE Foundation and Science and Technology Program of Shenzhen.

References

[1] Hong Yu, Zhang Yu, Liu Ting and Li Sheng. "Topic Detection and Tracking Review", Journal of Chinese Information Processing, vol. 21, pp. 71-86, 2007.

[2] J Allan, R Papka, and V Lavrenko. "On-line New Event Detection and Tracking", Proceedings of the 21st Annual International ACM SIGIR Conference, pp. 37-45, 1998.

[3] Y. Yang, T. Pierce, and J. Carbonell, "A Study of Retrospective and On-Line Event Detection," Proceedings of ACM SIGIR , 1998.

[4] J. Allan, S. Harding, D. Fisher, A. Bolivar, S. Guzman-Lara, and P. Amstutz, "Taking Topic Detection from Evaluation to Practice", 2005

[5] W Lam, H Meng, K Wong, and J Yen. "Using Contextual Analysis for News Event Detection". International Journal on Intelligent Systems, vol. 16, pp. 525-546, 2001.

[6] Y Yang, J Carbonell, C Jin. "Topic-Conditioned Novelty Detection". Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 688-693, 2002.

[7] J Allan, H Jin, M Rajman, and C Wayne. "Topic-based Novelty Detection". Proceedings of the Johns Hopkins Summer Workshop, 1999.

[8] G Kumaran and J Allan. "Text Classification and Named Entities for New Event Detection". Proceedings of the 27th Annual International ACM SIGIR Conference, pp 297-304, 2004.

[9] Gupta C, Grossman R L. "GenIc: A Single-Pass Generalized Incremental Algorithm for Cloustering", Proceedings of the 2004 SIAM International Conference on Data Mining, pp.137-153, 2004.

[10] Shui Yidong, Qu Youli and HUANG Houkuan. "A New Topic Detection and Tracking Approach Combining Periodic Classification and Single·Pass Clustering", Journal of Beijing JiaoTong University. Vol. 33, pp. 85-99, 2009.

[11] Dong Dan, Wang Weidong, Chen Ying. "Topic Detection and Tracking with a develop Vector Space Model", Computer Technology and Development, vol. 16, pp. 62-65, 2006.

[12] Dawid W. "Descriptive Clustering as a Method for Exploring Text Collections", PhD Thesis: Poznan University of Technology, pp. 7-56, 2006.

[13] J Han, J Pei, Y Yin and R Mao. "Mining Frequent Patterns without Candidate Generation". Proceedings of Special Interest Group on Management of Data. pp. 1-12, 2000.