Finding Topics in News Web Pages by Parameter-free Clustering

Ji Xiang[1] Neng Gao[1] Jiwu Jing[1]

[1]State key laboratory of information security, Graduate university of Chinese academy of sciences, 19A Yuquan road, Beijing, P. R. China
Email: {jixiang, gaoneng, jing}@lois.cn

Abstract

Topic detection is a novel technology which structures news stories into several topics. Present topic detection approaches are mainly based on clustering algorithms such as single pass or agglomerative clustering, and all these algorithms need at least one input parameter. We proposed a novel clustering algorithm which automatically determines the parameters for each corpus. Experimental results show that the parameters derived are close to optimal, and our algorithm has similar accuracy as the UPGMA algorithm which is manually set with optimal parameters. Another advantage of our algorithm is that it runs much faster than the UPGMA algorithm.

Keywords: Topic Detection, Clustering Algorithm, Parameter Free, Similarity Distribution

1. Introduction

Nowadays web pages have become the fastest ways for us to achieve news and publish individual opinions. It is hard to identify what pages exactly we want. By giving a set of keywords to search engines such as Google and Yahoo!, we can obtain a very long list of URLs referring to Web pages. However, it is still a difficult work to grasp and summarize the contents quickly from the search results. We need a convenient and efficient way to learn "what's happened" or "what's hot" from large number of news web pages. Topic detection is such a method which automatically finds topics in a group of news corpuses. At present, topic detection is widely applied to web information organization, such as automatic construction of online news issues [1] or organization of RSS news from different sources on topics [2].

It is easy to think of the use of data mining algorithms to find topics. Actually, clustering is the core algorithm of present topic detection approaches. Common clustering algorithms used for topic detection include single pass, kNN and agglomerative clustering [1][3][4][5]. A drawback of these algorithms is that they can not run automatically without manual intervention.

A common solution for the above problem is to use fixed parameters for all corpuses when they are similar in size and structure. A representative work on this approach is [1] which uses the UPGMA agglomerative clustering algorithm and specifies a fixed threshold to control the termination of the algorithm. But according to our experiments, when size or structure varies among corpuses, the optimal thresholds vary largely.

We suppose that different corpuses have different optimal parameters regardless of which algorithm and parameters are used, and it is impossible for users to determine the optimal

parameter for each corpus in advance. To solve the problem, we proposed a novel clustering algorithm which automatically derives parameters from and for the news corpuses. It does not need users to provide any kind of parameters, and it uses different parameters for different corpuses. We can not guarantee that the automatically determined parameters are optimized, but they are close enough to optimal according to our experiments.

Details of our algorithm are described as follows. (1) At first, pair-wise similarities of all the news stories in the corpus are computed, and they are used for automatically parameter determination. No user provided information is used. (2) A similarity threshold for the corpus is automatically selected according to the distribution of the pair-wise similarities of the corpus. (3)A novel algorithm is used to organize stories into topic groups based on the similarity threshold.

In the remainder of this paper, we proceed as follows: In the next section, we give a brief review of related work on topic algorithm in detail step by step. Section 4 gives the news corpuses to be evaluated and the evaluation metrics. After presenting the experimental results in section 5, we conclude in section 6.

2. Related Work

Topic detection in news corpuses [1][3][4][5][6][7][8] is an active research for years, but there still lacks accurate, efficient and practical solutions. Yang et al from CMU firstly bring forward the task of topic detection and propose two methods to achieve this goal [3]. One method is based on single pass clustering which needs users to provide a clustering threshold. The other is based on agglomerative clustering which needs users to specify desired cluster count or a threshold to control termination of the algorithm. Shah and ElBahesh [4] propose two other topic detection methods based on kNN and single linkage clustering, the former algorithm needs users to specify a parameter k while the latter takes a maximum distance value as input.

Wang et al [1] use UPGMA agglomerative clustering for topic detection because UPGMA is the most accurate of four algorithms they evaluated. And, they use fixed termination threshold for all corpuses to avoid user provided parameters. But according to our experiments, the optimal termination thresholds vary largely for different corpuses, even though they come from the same sources.

Fung et al [5][6] propose a well-developed parameter free approach for topic detection. Their approach focuses on detecting a few largest topics (called burst clusters), while our work aims to detect all topics in the news corpuses.

There also some efforts on general parameter free clustering [9][10][11], and the one mostly related to us is [9]. They automatically derive a threshold from the pair-wise similarities of elements using curve fitting, and group the elements into clusters using connected components algorithm. Their approach works in an iteration way and it can not automatically determine when to terminate.

3. Parameter free clustering for topic detection

In this section, we describe the topic detection method based on our designed parameter-free clustering algorithm. The method works as follows: At first, each story from a web page is vectored and pair-wise similarities of the news corpus are computed. Next, a similarity threshold is automatically selected based on the distribution of the pair-wise similarities. The similarity threshold is chosen so that ma-

Symbol	Definition
Corpus	The news web pages to be analyzed
PAIRS	$PAIRS = \{(s_i, s_j) \mid s_i, s_j \in Corpus, i < j\}$ Contain all distinct pairs of stories of the Corpus
SAME	$SAME = \{(s_i, s_j) \mid topic(s_i) = topic(s_j)$ $(s_i, s_j) \in PAIRS\}$ Contain all pairs of stories from same topics $topic(s_i)$ is the actual topic story s_i belonging to
DIFF	$DIFF = PAIRS \setminus SAME$ Contain all pairs of stories from different topics
ABOVE	$ABOVE = \{(s_i, s_j) \mid Similarity(s_i, s_j) > threshold, (s_i, s_j) \in PAIRS\}$ Contains pairs of stories whose similarities are above the threshold
P_V	$P_V = \frac{\mid ABOVE \cap SAME \mid}{\mid SAME \mid}$ Percent of story pairs of set *SAME* belonging to set *ABOVE*
P_N	$P_N = \frac{\mid ABOVE \cap DIFF \mid}{\mid ABOVE \mid}$ Percent of story pairs of set *DIFF* belonging to set *ABOVE*
SD(PAIRS)	The similarity distributions of set *PAIRS*
SD(SAME)	The similarity distributions of set *SAME*
SD(DIFF)	The similarity distributions of set *DIFF*

Table 1: Notations.

jority story pairs from the same topics have similarities above it while only few story pairs from different topics have similarities above it. Finally, a novel algorithm is used to organize stories into topic groups according to the similarity threshold. The whole topic detection process is performed automatically.

3.1. Pre-processing

The purpose of pre-processing is to generate the news corpus's pair-wise similarities, which are used for further threshold selection and topic detection.

At first, each story is segmented into a set of words based on a Chinese dictionary of about 100,000 words. Next, stopwords and those words appear in too few stories are removed. After then term vectors are generated for all stories using the LTC weighting method [3] (a standard version of TF-IDF weighting):

$$Weight(w,s) = \frac{(1 + \log Tf(w,s)) \times \log(N / n_w)}{\sqrt{\sum_w ((1 + \log Tf(w,s)) \times \log(N / n_w))^2}} \quad (1)$$

Where $Weight(w,s)$ is the weight of word w in story s; $Tf(w,s)$ is the number of occurrence of word w in story s; N is the number of stories in corpus; n_w is the number of stories that w occurs in.

Finally, similarities of all story pairs are calculated through Cosine similarity and so that the similarity value between any two stories ranges from 0 to 1.

$$Similarity(s_i, s_j) = \sum_{\substack{w \in s_i \\ w \in s_j}} Weight(w, s_i) * Weight(w, s_j) \quad (2)$$

3.2. Automatic similarity threshold selection

Table 1 shows the notation we use in our problem formulation and analysis.

Through analyzing the distribution of the pair-wise similarities, we expect to find a similarity threshold so that set *ABOVE* contains majority story pairs from *SAME* while contains only a small fraction of pairs from *DIFF*. To achieve this goal, it is preferred that P_V is maximized while P_N is minimized. In the perfect case where $P_V=1$ and $P_N=0$, the topics can be detected with 100% accuracy. But for real world news corpus, it is impossible to achieve the perfect result because:

$$\exists (s_i, s_j), (s_m, s_n) \mid Similarity(s_i, s_j) < Similarity(s_m, s_n), \quad (3)$$
$$(s_i, s_j) \in SAME, (s_m, s_n) \in DIFF$$

It is also generally impossible to find an optimal threshold to achieve maximum P_V and minimal P_N at the same time, for they are both inversely proportional to the threshold. From the similarity distribution, we can see $SD(PAIRS)$ is a mixed distribution of $SD(SAME)$ and $SD(DIFF)$. $SD(SAME)$ and $SD(DIFF)$ are generally overlapped. Because similarities of story pairs from different topics are generally less than those from same topics, $SD(DIFF)$ dominates the leftmost region (with least similarities) of $SD(PAIRS)$ and $SD(SAME)$ dominates remaining region. We choose the location of the demarcation point between $SD(DIFF)$ and $SD(SAME)$ as the appropriate similarity threshold.

Figure 1 shows an idea model of $SD(PAIRS)$ where $SD(SAME)$ and $SD(DIFF)$ both follow the normal distribution. It also shows the location of the demarcation point in the idea model.

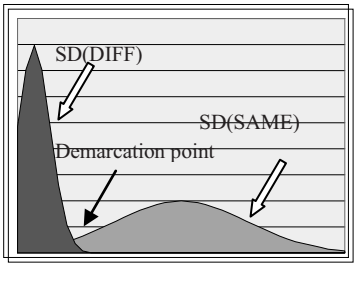


Fig. 1: Idea model of $SD(PAIRS)$ and the location of demarcation point.

In reality, the distributions of $SD(SAME)$ are unpredictable, and so that it is generally very difficult to locate the exact position of the demarcation point, but it is acceptable if a threshold around the location of demarcation point is found. After detailed study of various news corpuses, we found some characteristics of $SD(DIFF)$ and $SD(SAME)$ which may help us to locate the demarcation point.

--Similarity value between any two stories ranges from 0-1. $SD(DIFF)$ spreads only a small region around 0.0 because stories from different topics have very little similarities, while $SD(SAME)$ spreads a much larger region.

--Although the shapes of $SD(SAME)$ is random and unpredictable, the shape of $SD(DIFF)$ closely resembles the normal distribution in most cases.

--If we draw a distribution curve along the $SD(DIFF)$, the curve monotonously decreases with the increase of similarity in most cases. In the case of $SD(SAME)$, the curve generally contains many noticeable undulations.

Based on above findings, we assume that the position of the first evident undulation in the distribution curve of $SD(PAIRS)$ to be the location of demarcation point. We discrete the consecutive similarity distribution curve of $SD(PAIRS)$ with a parameter $CURVEPOINTS$, which is the number of points on x-axis of the Cartesian coordinate plane. The x-axis value of $point_i$ is $\frac{1.0}{CURVEPOINTS} \times i, i = 0,...CURVEPOINTS$, and the y-axis value is the number of similarities of $SD(PAIRS)$ falling into the region $[\frac{1.0}{CURVEPOINTS} \times i, \frac{1.0}{CURVEPOINTS} \times (i+1))$.

Finding undulations in distribution curves directly is relatively difficult, so we transform the distribution curves into slope angle curves by calculating the slope angles of each point on distribution

curve and replacing the y-axis value of each point with its slope angle. The slope angle of $point_i$ is calculated by fitting point set $(point_{i-K}, point_{i-K+1}, ..., point_i, ..., point_{i+K})$ with a one degree curve $Y(x) = ax + b$. The linear least squares method is used for curve fitting. After the curve is derived, the slope angle is equal to $arctag(a)$ which is a value in the range $[-1.0, 1.0]$. After transformation, undulations in distribution curve turns to peaks and troughs in the slope angle curve. Figure 2 show the slope angle curve of the SINA-1 corpus which is described in section 4 in detail.

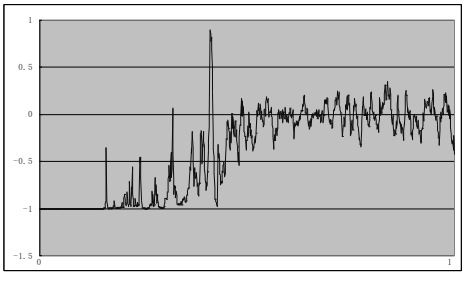


Fig. 2: Slope Angle curve of SINA-1 corpuses, undulations in distribution curve turns to peaks and troughs in the slope angle curve.

We search the slope angle curve to find the first noticeable peak and set the position of peak as the value of threshold. We use two thresholds T_H and T_W to specify the minimal height and width of the peak. The thresholds are carefully selected to exclude small or accidental peaks. We test each point against the two thresholds from $point_0$ to $point_{CURVEPOINTS}$ and select the first point which has y-axis value above T_H and is the largest point among point set $(point_{i-T_W}, point_{i-T_W+1}, ..., point_i, ..., point_{i+T_W})$.

$CURVEPOINTS, K, T_H$ and T_W are all parameters used by our algorithm, and we will describe their settings in section 5.

3.3. Topic detection

After the similarity threshold is determined, next step of our algorithm is to organize the stories into topic groups according to story pairs belonging to set $ABOVE$.

We propose a novel story grouping method which is resistant to little fraction of noises. Instead of generating the topics directly, we firstly select a set of stories as the principal story of each topic. With these principal stories, we can obtain overlapped or un-overlapped topic detection result as we want.

We call two stories are neighbors if their similarity above the similarity threshold. The principal stories are selected based on the following process.

```
Determine the neighbors of each
story based on the similarity thre-
shold.

while the corpus is not empty, do
   Select the story with the larg-
   est neighborhood count from the
   corpus.
   Create a new topic group, and
   set the story as the principle
   story of the topic group.
   Remove the story and all its
   neighbors from the corpus and
   continue.
   Output all topic groups created
```

In some cases, if principal stories belonging to different topic groups have lots of neighbors in common then they should be merged into a same topic group. For this reason, we propose a further combination method. Suppose two principal stories each has m and n neighbors, and they have l neighbors in common. If $\frac{l}{\min(m,n)}$ is above a certain threshold T_C, the two principal stories are combined into a single topic group. After that, we can use connected components algorithm to combine the principal stories, because

it is very small possibility that two principal stories from different topics have many common neighbors. So after combination, a topic may contain more than one principal story.

Next, we put stories into different topic groups based on these principal points. Two types of topic groups can be generated by using the principal stories: overlapped or un-overlapped.

4. Experimental Setup

4.1. News corpuses

Eight Chinese news corpuses from two different sources are used to evaluate our topic detection approach. The two news sources are:

--SINA hot topics: It is collected from news.sina.com.cn/zt/.

--TDT3 [13]: It were collected and annotated by LDC for the purpose of topic detection and tracking research.

News corpus	# of topics / # of stories	Description
SINA-1	78/3130	All topics from SINA
SINA-2	50/1879	Randomly selected 50 topics from SINA
SINA-3	30/1157	Randomly selected 30 topics from SINA
SINA-4	10/344	Randomly selected 10 topics from SINA
TDT-1	109/2695	All topics from TDT3
TDT-2	12/1422	Top 12 largest topics from TDT3
TDT-3	97/1273	All other topics not in TDT-2
TDT-4	12/458	Randomly selected 12 topics from TDT3

Table 2: Information of the news corpuses.

Eight topics with different sizes and internal structure are extracted from the two sources as follows:

4.2. Evaluation criteria

Two metrics are used to measure the quality of our topic detection approach: F-measure and Entropy [14][15]. F-measure measures the overall degree of how classes and clusters are matched, as well as Entropy measures the overall purity of the clusters.

For the given news corpus, we divided them into u classes manually and divided them into v un-overlapped topic clusters by our algorithm. Suppose n_{ij} indicates the number of documents from cluster j that belong to class i, $n_i = \sum_{j=1}^{v} n_{ij}$ is the number of stories belonging to class i, $n_j = \sum_{j=1}^{u} n_{ij}$ is the number of stories belonging to cluster j, and $n = \sum_i \sum_j n_{ij}$ is the total number of stories.

The F-measure of class i and cluster j can be computed with the following equation.

$$F(i,j) = \frac{2 \frac{n_{ij}}{n_i} \times \frac{n_{ij}}{n_j}}{\frac{n_{ij}}{n_i} + \frac{n_{ij}}{n_j}} \tag{4}$$

$\frac{n_{ij}}{n_i}$ and $\frac{n_{ij}}{n_j}$ are precision and recall of class i and cluster j respectively. To calculate overall F-measure value, we take the maximum $F(i,j)$ over all clusters and then sum across classes [15].

$$F - measure = \sum_i \frac{n_i}{n} \max_j F(i,j) \tag{5}$$

The entropy of cluster j is calculated through the following equation:

$$H_j = -\sum_i \frac{n_{ij}}{n_j} \log(\frac{n_{ij}}{n_j}) \tag{6}$$

The overall entropy can be computed by sum up the entropies of individual clus-

ters weighted with the proportion of stories in each [15].

$$Entropy = \sum_{j} \frac{n_j}{n} H_j \qquad (7)$$

5. Experiments and Discussions

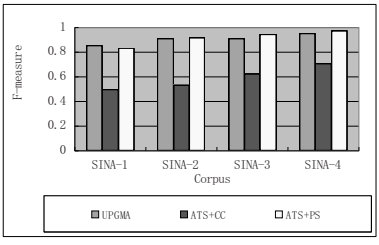
5.1. Topic detection on SINA corpuses

We detected topics in SINA corpuses using the following three algorithms:
--UPGMA: The UPGMA agglomerative algorithm
--ATS+CC: Automatic threshold selection with connected components algorithm
--ATS+PS: Automatic threshold selection with our principal story algorithm

The UPGMA algorithm is chosen for comparison because it is one of the most accurate clustering algorithms [12], and it is extensively used by other researchers in topic detection [1][3][5].

The parameters and thresholds used for automatic threshold selection are set as follows:
$CURVEPOINTS = 1000, K = 10, T_H = -0.8, T_W = 17$.
The combination threshold T_C described in section 3.3 is set to $1/3$.
The F-measure and Entropy values of the topic detection results are depicted in Figure 3.

From the results it can be figured out that for all four corpuses, ATS+PS algorithm has similar F-measure and Entropy values as UPGMA algorithm, and the results of ATS+CC are much worse. As described in section 3.3, CC algorithm performs worse because the P_N value is not negligible, so that the topics generated tend to be impure.
The similarity thresholds determined through ATS algorithm and the topics counts generated by PS algorithm are listed in the following Table 3.



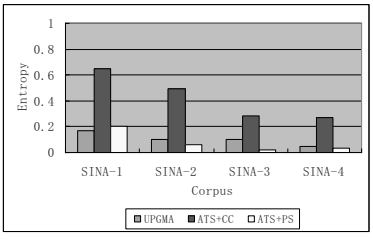


Fig. 3: F-measure and Entropy values of the UPGMA algorithm, ATS+CC algorithm and our ATS+PS algorithm.

	SINA-1	SINA-2	SINA-3	SINA-4
Threshold	0.161	0.145	0.166	0.128
Topic count	100	68	54	10

Table 3: Automatically selected threshold and generated topic count of SINA corpues.

The ATS algorithm selected different thresholds for different corpuses. We can not guarantee that the thresholds are optimized, but considered the situation that our algorithm has similar accuracy compare to the UPGMA algorithm which is set with the optimized parameters (the actual topic counts), the thresholds selected should be close to optimal.

The computation times of UPGMA algorithm and ATS+PS algorithm are listed in Table 4. The times used for preprocessing is not considered.

According to the results, our algorithm runs much faster than the UPGMA algorithm. The main reason is that UPGMA

algorithm usually runs hundreds to thousands of rounds. But in our algorithm, there is only one round so that a lot of time is saved for calculating similarities between clusters.

	SINA-1	SINA-2	SINA-3	SINA-4
ATS+PS	0.363s	0.143s	0.067s	0.029s
UPGMA	204.823s	49.93s	12.58s	2.263s

Table 4: Computation time compare of the two algorithms.

5.2. Topic detection on TDT corpuses

As mentioned in section 1, a method to avoid user provided parameters is setting fixed termination threshold in the UPGMA algorithm. We evaluated the possibility of use this method on TDT corpuses.

We use the cluster count as the parameter and specified the actual topic count manually. At the same time, we recorded the similarity threshold for to stop the merging. The results are showed in Table 5.

	TDT-1	TDT-2	TDT-3	TDT-4
similarity	0.071	0.041	0.106	0.066

Table 5: UPGMA algorithm's termination similarity of the TDT corpuses.

If we use termination threshold as the parameter and want to get desired topic counts, we must set the thresholds according to the value in above table. It is apparent that different corpuses should be set with quite different thresholds. For example, TDT-3 corpus's threshold is above 2.5 times larger than that of TDT-2. It is not a good idea to set fixed termination threshold for all corpuses, even they come from the same sources.

The TDT corpuses results of UPGMA and ATS+PS algorithms are depicted in figure 4, The ATS+PS algorithm use the same set of parameters described in section 5.1.

According to the results, for TDT-3 and TDT-4, our ATS+PS algorithm generated similar F-measure and Entropy values as UPGMA algorithm. But for TDT-1 and TDT-2 our ATS+PS algorithm generated relatively lower F-measure and Entropy values than the UPGMA algorithm.

The automatically selected threshold, the detected topic counts and P_V , P_N values of the four corpuses are list in Table 6.

	TDT-1	TDT-2	TDT-3	TDT-4
Threshold	0.265	0.198	0.18	0.112
Topic count	599	126	85	10
P_V	7.2%	8.3%	56.7%	55.8%
P_N	4.5%	1.9%	14.1%	14.2%

Table 6: Some results of the TDT corpuses.

It can be figured out that for TDT-1 and TDT-2 ATS+PS generated low F-measure and Entropy values because relatively large thresholds are selected so that the P_V values are quite low (less than 10%). The consequence is that much more topics than the actual ones are detected. Although these topics are relatively pure, users may still get discontent if too much topics as they expect are generated.

6. Conclusions and future work

In this paper, we proposed a novel and practical clustering algorithm for finding topics in news corpuses. It does not depend on the users to provide the parameters but automatically derives different parameters from and for different corpus-

es. It is proved that the selected parameters are close to optimal, so that the accuracy of the topic detection results is high and stable.

In the future, we will extend our work to topic trend tracking which automatically associate topics detected at different time into several topic trends.

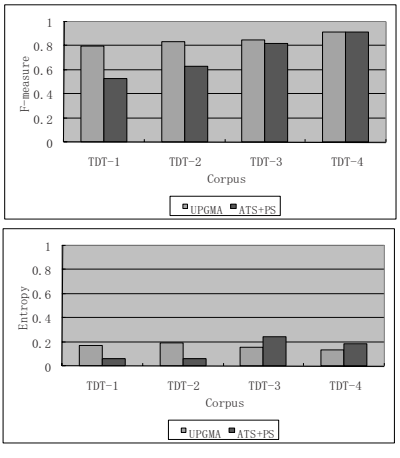


Fig. 4: F-measure and Entropy values of the UPGMA algorithm and our ATS+PS algorithm.

References

[1] Canhui Wang, Min Zhang, Shaoping ma, Liyun Ru: Automatic Online News Issue Construction in Web Environment: the 17th International World Wide Web Conference (WWW2008), Beijing, April, 2008, pp457-466.

[2] Yahoo Topic Clustered RSS Reader: http://research.yahoo.com/node/93

[3] Yiming Yang, Jaime Carbonell, Ralf Brown, et al: Learning Approaches for Detecting and Tracking News Events: IEEE Intelligent Systems, Volume 14,Issue 4,pp 32 - 43 , 1999.

[4] Najaf Ali Shah, Ehab M. ElBahesh: Topic-Based Clustering of News Articles: Proceedings of the 42nd annual Southeast regional conference, Huntsville, Alabama, 2004, pp 412 - 413.

[5] Qi He, Kuiyu Chang, Ee-Peng Lim: Using Burstiness to Improve Clustering of Topics in News Streams, ICDM 2007, Omaha, NE, pp 493-498.

[6] Gabriel Pui, Cheong Fung, Jeffrey Xu, Yu Philip, S. Yu, Hongjun Lu: Parameter Free Bursty Events Detection in Text Streams: Proceedings of the 31st international conference on Very large data bases, Trondheim, Norway, 2005, pp 181-192.

[7] Y Yang, T Pierce, J Carbonell: A study on Retrospective and On-Line Event detection: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. 1998, pp 28-36.

[8] M. Connell, A. Feng, G. Kumaran, and et al. UMass at TDT 2004. Topic Detection and Tracking Workshop Report, 2004.

[9] Javed Aslam, Alain Leblanc, and Clifford Stein: Clustering Data without Prior Knowledge: WAE 2000

[10] Christian Böhm, Christos Faloutsos, Jia-Yu Pan, Claudia Plant: RIC: Parameter-free noise-robust clustering: ACM Transactions on Knowledge Discovery from Data, Volume 1 , Issue 3, December 2007,

[11] Eugenio Cesario, Giuseppe Manco, Riccardo Ortale: Top-Down Parameter-Free Clustering of High-Dimensional Categorical Data: IEEE Transactions on Knowledge and Data Engineering, Volume 19 , Issue 12, December 2007, pp1607-1624.

[12] Ying Zhao and George Karypis: Hierarchical Clustering Algorithms for Document Datasets: Data Mining and Knowledge Discovery, Vol. 10, No. 2, pp. 141 - 168, 2005

[13] TDT-3 corpus: http://projects.ldc.upenn.edu/TDT3/TDT3_Overview.html

[14] Shen Huang, Zheng Chen, Yong Yu, and Wei-Ying Ma: Multitype Features Coselection for Web Document Clustering: IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 18, NO. 4, APRIL 2006

[15] ZDRAVKO MARKOV AND DANIEL T. LAROSE: DATA MINING THE WEB, Uncovering Patterns in Web Content, Structure and Usage: John Wiley & Sons, Inc, 2007