CLUSTERING POSTS IN ONLINE DISCUSSION FORUM THREADS

Dina Said[1] and Nayer Wanas[2]

[1]Department of Computer Science, University of Calgary, Calgary, Canada
dasaid@ucalgary.ca
[2]Informatics Department, Electronics Research Institute, Giza, Egypt
nwanas@gmail.com

ABSTRACT

Online discussion forums are considered a challenging repository for data mining tasks. Forums usually contain hundreds of threads which in turn consist of hundreds, or even thousands, of posts. Clustering posts can be used to discover outlier and off-topic posts and would provide better visualization and exploration of online threads.In this paper, we propose the Leader-based Post Clustering (LPC), a modification to the Leader algorithm to be applied to the domain of clustering posts in threads of discussion boards. We also suggest using asymmetric pair-wise distances to measure the dissimilarity between posts. We further investigate the effect of indirect distance between posts, and how to calibrate it with the direct distance. In order to evaluate the proposed methods, we conduct experiments using artificial and real threads extracted from Slashdot and Ciao discussion forums. Experimental results demonstrate the effectiveness of the LPC algorithm when using the linear combination of direct and indirect distances, as well as using an averaging approach to evaluate a representative indirect distance. Furthermore, the results show the potential of the LPC algorithm for detecting off-topic or outlier posts compared with two state-of-the-art methods for off-topic post detection.

KEYWORDS

Distance metrics, Clustering, Outlier Detection, off-topic Detection, Online Forums Mining

1. INTRODUCTION

Online discussion boards, also known as newsgroups or online forums, are amongst the most popular forms of user generated content. Through these discussion boards, users share opinions, experiences, post questions and search for answers. Online discussion boards are unique compared to other web-based information resources in that they are organized in tree structures known as threads. The lead post (thread head) within this thread initiates the discussion. Subsequent posts contribute additional content that extends the discussion. This, in turn, implies that knowledge within threads is retained in a sequence of posts within them, rather than a specific post. Overall, forums remain to be a rich repository of user generated content that contain vast resources of knowledge.

However, there are several issues that render discussion boards difficult to use, more than other forms of user generated content. One major problem in using discussion boards is the limited ability to filter and search the content to meet a specific need. In addition, irrelevant posts that infiltrate the sequence could obscure the ability to isolate nuggets of knowledge. Moreover, users might deviate from the initial topic of the thread to discuss other issues and several trains

DOI : 10.5121/ijcsit.2011.3201

of thought might flow concurrently. These issues might obscure the usability of discussion forums.

In order to overcome these issues, posts need to be organized differently based on their relevance to each other. Clustering posts within a given thread based on their content could assist this organization. While document clustering has been a well addressed problem in the literature, online discussion boards are significantly different in their nature. Posts in forums are short and fragmented allowing limited detection of context. To the best of our knowledge, this is the first work to attempt to cluster online discussion posts within a thread. However, several researchers have addressed the issue of clustering short text. Clustering of web snippets for search organization has been suggested in [1,2,3] and primarily focused on hierarchal clustering. These approaches aim to identify clusters based on low level element matching between the different snippets and assign appropriate tags and structure to each cluster. However, they don't consider the interdependence between these elements which is more profound in online threads. Several researchers have clustered email messages, yet most have focused on spam-detection rather than topical clustering of emails [4,5,6]. Huang and Mitchell [7] suggested a hierarchal email clustering algorithm that is adaptable based on user feedback. However, these approaches have been trying to cluster emails threads at a coarse level. This is in contrast to the need to cluster posts within individual threads in online discussion forums.

In this work we present an iterative distance based approach to cluster posts within online discussion forums. This approach is rooted in the fact that the order is important in online discussion forums and that the relationships between posts can be both direct and indirect.

2. CLUSTERING POSTS IN DISCUSSION BOARDS

Discussion forums have several characteristics that should be considered when clustering posts within their threads. Among these characteristics are the following:

1- Online discussion boards usually include different topics which in turn have sub-topics. Each sub-topic usually involves many threads. Some of these threads may be very large and/or very diverse. This makes different threads in a single discussion board potentially demonstrating different characteristics. Hence, the clustering algorithm should not require any predefined parameters to make it as general as possible.

2- Posts are ordered in the thread, mostly according to the post date. Therefore, one might model this as a sequential clustering to capture the time dependency among posts. Therefore, a post that is not related to clusters formed for previous posts should be assigned to a new cluster.

3- The head post in a thread is of a special importance. Eventually, posts are determined to be off-topics, or outliers, based on how relevant they are to the head post in the thread [8]. Therefore, the head post should be considered a core node in the clustering algorithm.

4- Threads may span several discussions, and hence would include several clusters. Therefore, no individual post can be the only representative to its cluster, and in turn the distance between a new post and the cluster should consider all posts within the cluster.

5- The number of discussions addressed in a single thread is hard to estimate. Additionally, online threads may have a large number of off-topic and outlier posts that do not relate to any discussion in thread. Therefore, the clustering algorithm should allow the number of clusters to grow accordingly, and each outlier post should be assigned to a single-post cluster.

6- Posts may be subsets of each other by using the tagging facility available in most discussion boards. Therefore, pair-wise distance between any two posts should reflect this tagging, or referencing. Consequently, the probability of assigning the new post to the same cluster of the post it tags should increase.

7- Discussion boards usually involve a hierarchy of discussions, where a post P_i may refer to or comment on another post P_l. In turn, another post P_j may refer to or comment on P_l. Therefore, there is indirect relation between posts P_i and P_j that should be captured in the assessment of pair-wise distance. This also dictates that the pair-wise distance between posts should be asymmetric.

With these characteristics in mind, we suggest the Leader-based Posts Clustering (LPC) algorithm. This algorithm is a modification of the leader algorithm [9], which starts with selecting a pattern randomly to be the first leader. In turn, the distance between every other pattern with current selected leaders is evaluated. If the minimum distance between the new pattern and the current leaders is less than a predefined threshold, the corresponding pattern is assigned to the cluster of the closest leader. Otherwise, the pattern is identified as a new leader.

Using the leader algorithm enables the maintenance of the dependencies amongst posts in online threads. First, it captures the time dependency among posts. Second, it allows novel posts to form new clusters. Moreover, it does not require any prior knowledge about the number of clusters in the thread. Several modifications are suggested to the leader algorithm to adapt to clustering posts on online threads. First, the initial leader is predefined to be the head post, instead of selecting it randomly. In addition, the distance between a post P_i and a cluster C_m is considered to be the average distance between P_i and all posts $P_j \in C_m$. Eventually, this leads to a better assessment of distances between posts and the candidate cluster. Additionally, after assigning all posts to clusters, we iteratively repeat the whole process until no change in the assignment of posts to clusters occurs, or the number of iterations exceeds a maximum pre-set threshold.

While the Leader algorithm does not require the predetermination of the number of clusters, it however requires a threshold of distances which is a very critical parameter. A large threshold would produce a smaller number of clusters with low cohesion. Consequently, a small threshold would produce more clusters with higher cohesion and less separation. In order to adjust to the diversity that exists between different threads, the LPC algorithm uses the median of pair-wise distances between posts in the same thread as a robust threshold of distances.

3. DISTANCE METRICS

As previously mentioned, the nature of discussion posts suggests the potential of using asymmetric pair-wise distance between posts, while taking into consideration the indirect distance between them. We define asymmetric direct distance D_d between posts P_i and P_j as follows:

$$D_d(P_i,P_j) = 1 - \frac{\sum_{k=1}^{|b_i \cap b_j|} \min(w_i^k, w_j^k)}{\sum_{k=1}^{|b_i|} w_i^k} \qquad (1)$$

whereb_i and b_j are the bags of non-stop stemmed words of posts P_i and P_j respectively, and w_i^k is the term frequency of word k in post P_i. Hence, $D_d(i,j)=0$ if $b_i \subseteq b_j$. Consequently, $D_d(i,j)=1$ if $b_i \cap b_j = \emptyset$ which means that there is no direct distance between these posts. Besides its appropriateness to the domain of post clustering, asymmetric distance has been used in [10,11] to cluster text documents. Asymmetric distance has shown a potential to enhance the clustering performance compared to the symmetric distance, based on the cosine similarity.

In order to find the indirect pair-wise distance between posts P_i and P_j, the indirect links between them should be first determined. In this research, we consider only indirect links that span one level. Therefore, an indirect link exists between posts P_i and P_j through post P_l if there is direct links between $\{P_i,P_l\}$ and $\{P_l,P_j\}$. In turn, the indirect distance (D_i) between P_i and $P_j$through post P_l is defined as follows:

$$D_i = (P_i, P_l, P_j) = \begin{cases} \left. D_d(P_i,P_l) + D_d(P_l,P_j) \middle/ 2 \right. & \text{if} D_d(P_i,P_l) < 1 \text{ and} D_d(P_l,P_j) < 1 \\ 1 & \text{Otherwise} \end{cases}$$

(2)

The aggregated indirect distance (D_a) between posts P_i and P_j can be evaluated using one of the following functions:

- **Minimum Distance (Min):** representing the shortest indirect distance between posts P_i and P_j.
- **Average Distance of Indirect Links (Avg)**: suppress the bias to the shortest indirect path
- **Median Distance of Indirect Links (Med)**: eliminates the effects of very small and very large indirect distances.
- **Average Distance of the Smallest Five Indirect Links (AvgF)**: assumes that the smallest five indirect links are the most representative links to the indirect distance between the posts. It should be noted that if the number of indirect links is less than five, the AvgF is calculated based on only actual indirect links.

In order to cluster posts, direct and indirect distance between posts should be combined together to form the combined distance (D_c). D_c between posts P_i and P_j is based on the direct distance ($D_d(P_i,P_j)$) and the aggregated indirect distance ($D_a(P_i,P_j)$) can be defined as follows:

- **The Constant Function**: where combined distance between two posts equals the direct distance between them without considering the indirect distance.

$$D_c(P_i, P_j) = D_d(P_i, P_j)$$

- **The Power Function**: bounds the effect of the indirect distance on the direct distance.

$$D_c(P_i, P_j) = D_d(P_i, P_j)^{D_a(P_i, P_j)}$$

- **The Linear Function**: provides equal effect of the direct distance and indirect distance on the combined distance.

$$D_c(P_i, P_j) = D_d(P_i, P_j) \times D_a(P_i, P_j)$$

- **The Tanh Function**: increases the contribution of the indirect distance in the combined distance.

$$D_c(P_i, P_j) = D_d(P_i, P_j) \times \tanh(D_a(P_i, P_j))$$

In total, four different functions of aggregating indirect distances are suggested, along with four to combine indirect and direct distances. In the following, we present an experimental study to evaluate these different approaches.

4. EXPERIMENTS

In this section, the various experiments performed in order to evaluate our methods are introduced.

4.1. Datasets

In this work, we using two corpora provided through CAW 2.0[1], the first corpus is crawled from the Slashdot discussion board[2], while the second corpus is collected from the Ciao discussion board[3] for movies reviews. Table 1 provides a summary of the main highlights of these two corpora.

Table 1: A comparison between Slashdot and Ciao Corpora

	Slashdot Corpus	Ciao Corpus
No. of threads	495	183
Total No. of posts	140,788	20,879
Average No. of posts per thread	576.7	41.46
Max. No. of posts per thread	1062	195

[1] http://caw2.barcelonamedia.org

[2] http://slashdot.org

[3] http://www.ciao.com

Min. No. of posts per thread	20	22
Average No. of contributed users per thread	287.5	n/a
Average No. of posts sent by a single user per thread	1.64	n/a
Average No. of non-stop stemmed words per post	40.67	66.2
Average No. of unique non-stop stemmed words per post	32.74	58.79

The previous table outlines the differences between these two corpora. It is worth noting that the Slashdot corpus is significantly larger than the Ciao corpus in terms of both the number of threads and the number of posts. Threads in Slashdot average over 500 posts per thread, which is substantially larger than those of the Ciao, whose average is just over 40. Another distinction between the two corpora is in the number of words per post. Ciao is mainly a forum regarding movie reviews, and hence posts are generally lengthy, which renders clustering easier compared to Slashdot. Moreover, the potential of off-topic, outliers and deviated discussion posts is smaller in Ciao posts comparing with Slashdot. This is due to the nature of threads in Ciao, which are more independent posts about specific movies. This is in contrast to threads in Slashdot which cover a wide spectrum of topics and sub-topics, which in turn means that clustering is a more challenging task. It should be noted that each thread is considered as a dataset since we are performing clustering for posts within each thread separately.

Due to the absence of labelled data, we evaluate the performance on both corpora based on the clustering quality. Additionally, and to overcome the lack of labelled data, we have constructed two artificial corpora (a) Slashdot-Art and (b) Ciao-Art which are formed from the Slashdot and Ciao respectively. A number of artificial threads are created by concatenating several posts from different threads in the original corpus. The posts are labelled to belong to the same cluster if they are selected from the same original thread, which provides a pseudo-label for all posts. Each artificial corpus consists of 15 threads. The number of true clusters is maintained to be {2,5,10,15,20} while the number of posts per cluster varies between {1,5,10}. To ensure the quality of the constructed corpus, we exclude posts that are either tagged as off-topics and those labelled automatically according to [8] to be outliers. Additionally, the diversity among clusters is maintained in the Slashdot corpus by selecting posts in threads from different topics.

4.2. Performance Measure

Clustering performance of the artificial threads, where true clusters are known, is measured in terms of F_1measure [12] defined as:

$$F_1(C) = \frac{2 * Recall(C) * Presicion(C)}{Recall(C) + Precision(C)}$$

For real threads, we have adopted the silhouette factor measure [12] which combines the classical separation and cohesion measures in one measure. The silhouette factor (SF) for post *P* assigned to cluster *C* is defined as:

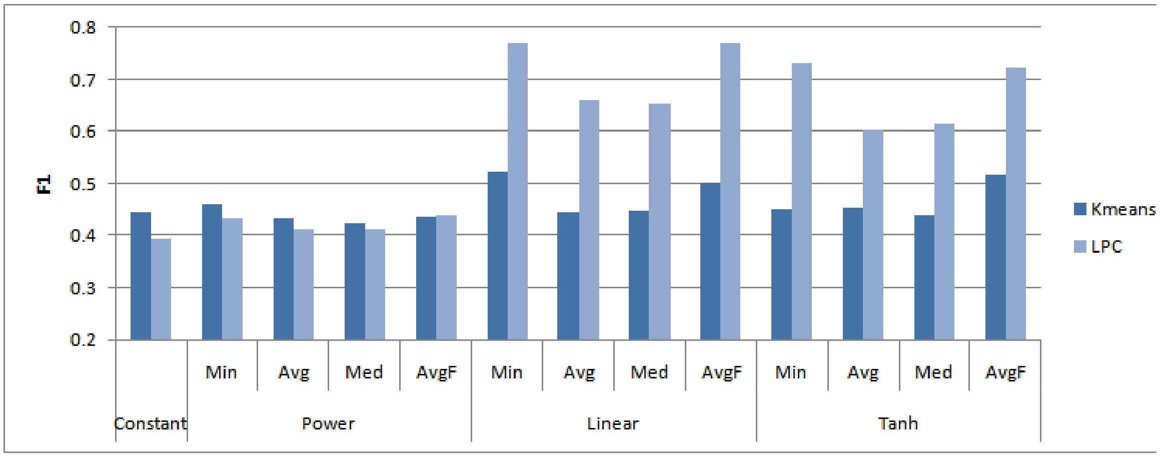
$$SF(P) = 1 - \frac{a(P)}{b(P)}$$

where*a(P)* is the average distance between Post *P* and all posts in *C* while *b(P)* is the minimum average distance between Post *P* and all clusters in the thread excluding *C*. The clustering performance for *SF(P)* is considered pretty good when $a(P) \ll b(P)$ and hence $SF(P) \approx 1$. It should be noted that the silhouette factor of Cluster *C* (*SF(C)*) is the average SF of all posts assigned to this cluster.

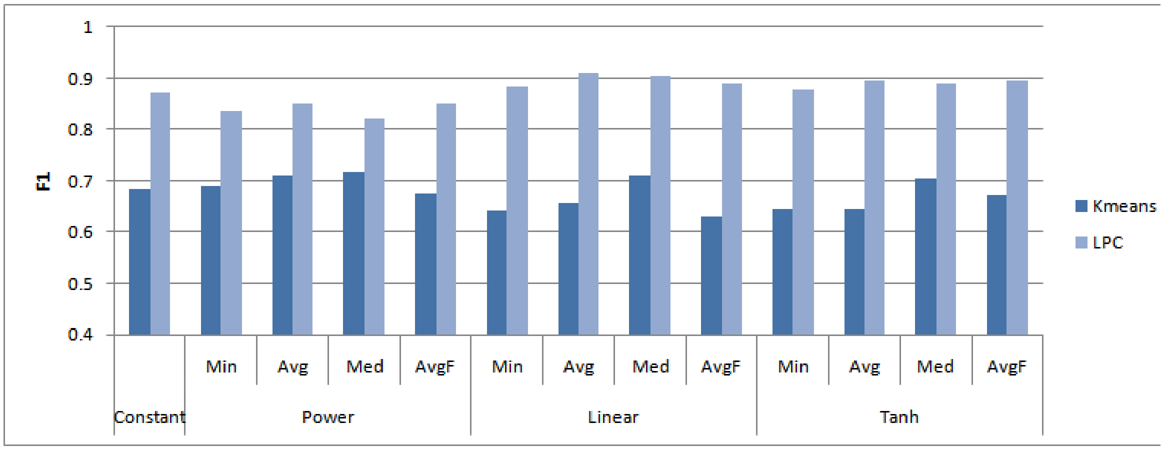
The overall F_1 and SF measures for the whole thread is calculated based on the weighted average of the F_1 and SF measures of all the clusters in the thread. Since it would be hard to provide results for each thread, we further calculate these measures for the whole corpus based on weighted averaging over all threads in the corpus. In turn, this implies that the threads with a large number of posts have more contribution in the performance evaluation compared with those with a small number of posts.

4.3. Results

Figure 1compares the clustering performance based on the F_1 measure of different combining and aggregate functions using Slashdot and Ciao artificial corpora. In this set of experiments, the *k*-means algorithm is used to benchmark the performance of the LPC algorithm suggested. It is worth noting that a cap on the number of iterations has been set to 100 for both algorithms. Moreover, we set *k* for the *k*-means algorithms to be the true number of clusters. Clearly, this is the best setting for *k*-means since its performance is expected to decline if *k* is over-estimated or under-estimated.



(a)



(b)

Figure 1: Weighted average F1 for the *k*-Means algorithm and the LPC algorithm using different combining and aggregate functions (a) Slashdot-Art Corpus and (b) Ciao-Art Corpus.

The results demonstrate the superiority of the LPC algorithm compared with the *k*-means for most distance functions. This is with the exception of using the power and constant distance functions for the Slashdot-Art corpus where *k*-means slightly outperforms LPC. It is worth noting that unlike the *k*-means, the LPC algorithm doesn't require any prior knowledge about the number of clusters in each thread.

Moreover, while the range of performance of the LPC algorithm is limited on the Ciao-Art dataset, the Linear and Tanh combining-aggregation functions demonstrate a better performance on Slashdot-Art compared to the corresponding Constant and Power functions. This is not as significant when using the *k*-means algorithm. As mentioned in section 4.1, the clustering task for the Slashdot-Art corpus is more challenging than that of the Ciao-Art corpus. This is reflected in the limited diversity in the performance of both LPC and *k*-means in the Ciao-Art corpus using different combining-aggregate functions compared to Slashdot-Art. For the LPC algorithm, the best F_1 achieved by the LPC was 0.911 while the worst $F_1$was 0.821. For the *k*-means algorithm, the best F_1 attained was 0.711 and the worst F_1 was 0.643. Since, the performance of the *k*-means is not significantly affected by the combining-aggregate function used, it was excluded from the performance evaluation of Slash and Ciao corpora.

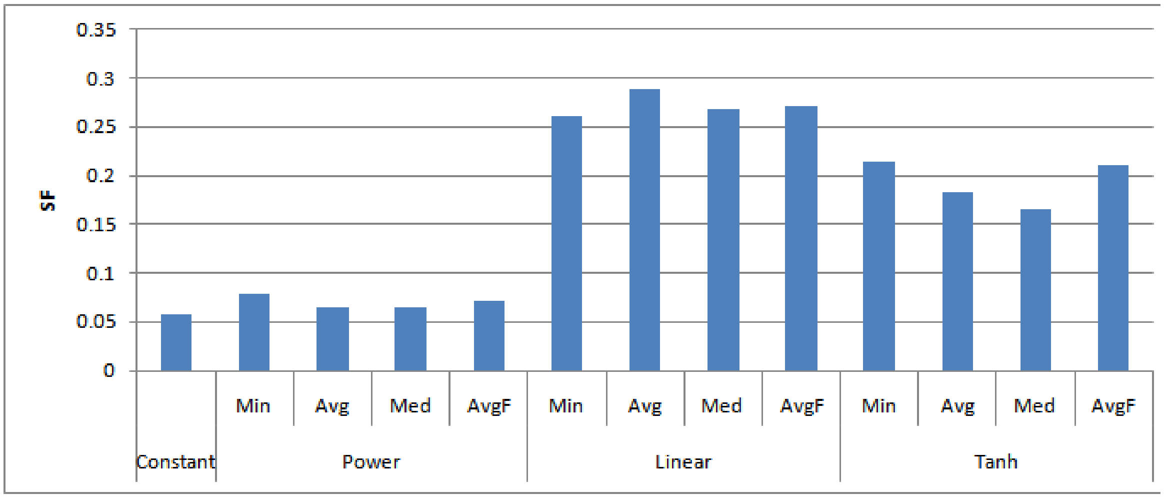
Figure 2 shows the performance of the LPC algorithm using the Slashdot and Ciao corpora. Since the true clusters for these corpora are unknown, the weighted average Silhouette Factor (SF) has been used to evaluate the performance (section 4.2).

The results demonstrate the superiority of the Linear and Tanh combining functions where indirect distance contributes intensively in the combined distance. This is more profound for the Slashdot corpora. In this case, the SF of the Linear and Tanh functions is at least three times better than that of the Constant and Power functions. Overall, the Linear function demonstrates a slightly better performance compared with the Tanh function. This is due to the fact that Linear function gives equal weights to the direct and indirect distances, while the Tanh function is more biased to the indirect distance. This may lead to a concealing of the effect of the direct distance which represents the direct dissimilarity between posts. Generally, incorporating the indirect distance using any of the three combining function (Power, Linear, Tanh) improves the performance on the Ciao-Art and Ciao corpora by at least 4%.

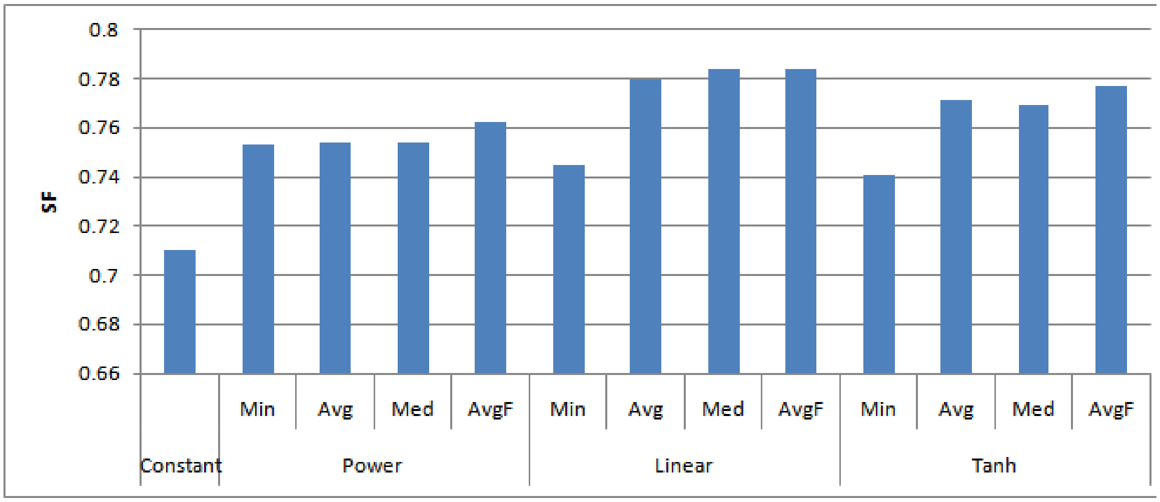
The diversity of performance of the LPC according to using different aggregate function is limited using the same combining function. For example, the performance of the Avg and Med is about 1% less that of the Min and AvgF for the Slashdot-Art corpus while the performance of the Min is about 0.4% less that of the AvgF for the Ciao corpus. In general, we recommend the using of the AvgF function. This is because it is not biased towards the minimum indirect distance like the Min. Additionally, it considers only the five indirect links which makes it a more reflecting to the indirect distance compared with the Avg and Med.

5. DETECTION OF OUTLIER POSTS

One of the most significant applications of clustering posts in threads is to provide a convenient way for users to navigate and explorer threads. Furthermore, clustering posts may help detect outlier posts. This is realized by the fact that outlier posts are those which are not related to the head post or more diverse compared to the cluster that contains the head post (the main cluster). The larger the distance is between a certain cluster and the main cluster, the more likely to be an outlier cluster. Therefore, posts can be declared to be outliers if their clusters are deviated from the main cluster with a distance larger than a specific threshold. In this work, this threshold is defined to be the median of distances between all clusters and the main cluster. We call this method Far Outlier Cluster (FOC). This is to distinguish it from the Small outlier Cluster (SOC) method in which we declare posts that belong to small cluster as outliers. Following the same concept of automatic thresholding, we define the median size of clusters to be the threshold that any cluster with size less than it will be declared as outlier.



(a)



(b)

Figure 2: Weighted average Silhouette Factor (SF) for the LPC algorithm using different combining and aggregate functions for (a) Slashdot Corpus, and (b) Ciao Corpus.

Figure 3illustrates the outlier detection recall of LPC with different combining function for the FOC and SOC methods using the Slashdot corpus where some posts have been tagged as being off-topic. Since the majority of posts have no tags, we will limit our focus on only recall. It should be noted that there is no tagging of posts in the Ciaodataset, hence, it will not be included in this assessment.We used AvgF as an aggregate function since the diversity in the performance of different aggregate functions is limited and AvgF was generally recommended. The results demonstrate the effectiveness of the SOC method to detect outlier posts compared with the FOC method for all combining methods. Additionally, the results show the superiority of the Linear and Tanh combining functions compared with the Power and Constant functions which is consistent with the clustering evaluation results. This indicates the importance of taking the indirect distance into consideration in the domain of the clustering posts and outlier discovery.

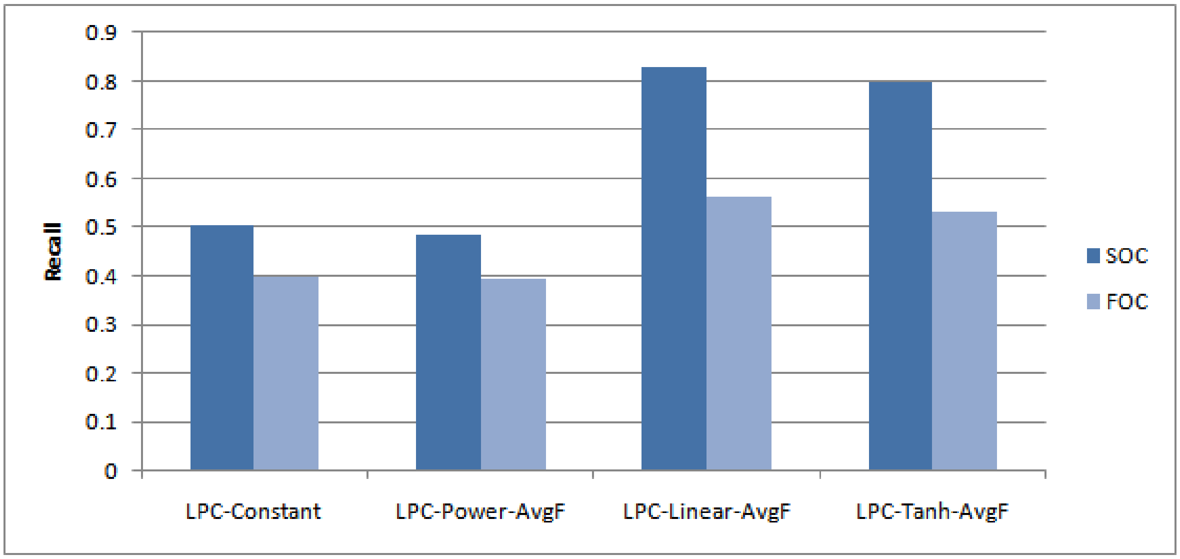


Figure 3: Outlier detection Recall of SOC and FOC methods for the Slashdot Corpus using the LPC algorithm with different combining methods

In order to benchmark our results, we have used the method proposed by [8] in order to discover outlier posts in Slashdot corpus. The recall achieved by this method is 0.537 which is about 25% less than the recall attained by the LPC-Linear-AvgF and LPC-Tanh-AvgF clustering with the SOC method.

Since using only recall in evaluating the performance of outlier detection is not sufficient, we have constructed another artificial corpus where outlier posts are known. This corpus (Slash-Dot-Art2) contains 72 artificial threads with different number of inlier and outlier posts. We attempted to model the nature of online threads by forming clusters with relatively large number of posts. Posts that belong to the same cluster have been selected from the same thread in Slashdot Corpus in a similar way of constructing artificial threads for clustering evaluation. Additionally, we formed a variety of threads in this corpus using different number of clusters and different cluster sizes. This is followed by planting outliers from threads of a completely different topic. The performance has been evaluated using weighted averaging F_1 to aggregate the performance of different threads for the whole corpus.

Table 2 compares the performance of the proposed methods (SOC and FOC) with a method for distance-based outlier detection (DBO) suggested in [13]. DBO defined outliers as those observations (posts) that have the largest *M*% summation of distances to the nearest neighbours. We use the threshold *M* for SOC and FOC methods to benchmark our results with the results attained by DBO. It should be noted that we used five neighbours in DBO. We further used the same distance Linear-AvgF for the three methods. The results indicate that both SOC and FOC attain a better performance compared with the DBO method especially for small *M*. Additionally, both SOC and FOC demonstrate similar performance.

Table 2: Weighted Average F1 for DBO, SOC, and FOC methods using Slash-Dot-Art2 and Linear-AvgF distance

Threshold	DBO	SOC	FOC
1	0.053	0.169	0.169
5	0.127	0.202	0.209
10	0.200	0.209	0.221
15	0.199	0.230	0.245
20	0.204	0.250	0.260
25	0.198	0.294	0.293

6. CONCLUSIONS

Online discussion boards represent a rich repository for data mining tasks in user generated texts. This research addresses the problem of clustering posts in different threads. The purpose of this clustering is mainly to provide improved visualization to threads in online discussion boards. This may also facilitate the discovery of off-topic and outlier posts in discussion threads. The Leader-based Posts Clustering (LPC) approach suggested captures the time dependency between posts. Starting from the head post, subsequent posts are assigned to either the most related cluster or to new clusters, based on an automatically-determined threshold of distances. An asymmetric distance is suggested for measuring the pair-wise distance between posts. This distance allows for modelling the inter-post tagging and commenting. Additionally, we suggest incorporating the indirect distance between posts. Four functions have been suggested for aggregating different indirect links, including the Minimum, Averaging, and Median aggregating functions. In addition, four methods for combining indirect and direct distances have been proposed; namely the Constant, Power, Linear, and Tanh functions.

Our experiments have been conducted using four corpora, two of them are artificially generated, where true clusters are known and the others are real corpora crawled from Slashdot and Ciao discussion boards. The results show the potential of the LPC, while using Linear combining function and averaging aggregate function. This is in comparison with the *k*-means algorithm using the artificial corpora, where the true number of clusters is known. In addition, the LPC algorithm, unlike the *k*-means, eliminates the requirement to estimate the number of actual clusters or predefined thresholds. For real corpora, the Linear combining function along with the averaging aggregate function has demonstrated the best performance among all the examined methods. This is supported by evaluating both the clustering quality and the ability to identify outlier posts.

ACKNOWLEDGEMENTS

This research has been conducted duringan internship granted to the first author at theCairo Microsoft Innovation Lab.

REFERENCES

[1] Moreno Carullo, ElisabettaBinaghi, &Ignazio Gallo, (2009) “An online document clustering technique for short web contents’, Pattern Recognition Letter, Vol. 30, No. 10, pp. 870–876

[2] Paolo Ferragina& Antonio Gulli, (2005) “A personalized search engine based on web-snippet hierarchical clustering”, In the proceedings of the 14th World Wide Web Conference (WWW’05), Chiba, Japan, May 10-14, 2005, pp. 801–810.

[3] Hua Jun Zeng, Qi Cai He, Zheng Chen, Wei Ying Ma, &Jinwen Ma, (2004) “Learning to cluster web search results’, in the proceedings of the 27th annual international ACM SIGIR conference on research and development in Information Retrieval (SIGIR’04), Sheffield, United Kingdom, July 24-29, 2004, pp. 210–217.

[4] Fulu Li & Mo-Han Hsieh, (2006) “An empirical study of clustering behaviour of spammers and group-based anti-spam strategies”, in the proceedings of the 3rd Conference on E-mail and Anti-Spam (CEAS’06), Mountain View, CA, USA, July 27-28, 2006.

[5] ArunSurendran, John Platt, & Erin Renshaw, (2005) “Automatic discovery of personal topics to organize email”, in the proceedings of the 2nd Conference on E-mail and Anti-spam (CEAS’05), Stanford University, CA, USA, July 21-22, 2005.

[6] Yang Xiang, (2009) “Managing email overload with an automatic nonparametric clustering system”, Journal of Supercomputing, Vol. 48, No. 3, pp. 227–242.

[7] Yifen Huang & Tom Mitchell, (2009) “Toward mixed-initiative email clustering”, In the proceedings of the AAAI Spring Symposia 2009: Agents that learn from human teachers, Stanford University, CA, USA, March 23-25, 2009.

[8] Nayer Wanas, Amr Magdy, &HebaAshour, (2009) “Using automatic keyword extraction to detect off-topic posts in online discussion boards”, in the proceedings of the Content Analysis in Web 2.0 Workshop (CAW2.0), Madrid, Spain, April 21, 2009.

[9] T.RavindraBabu&M.NarasimhaMurty, (2001) “Comparison of genetic algorithm based prototype selection schemes”, Pattern Recognition, Vol. 34, No. 2, pp. 523–525.

[10] Shaoxu Song &Chunping Li, (2005) “TCUAP: a novel approach of text clustering using asymmetric proximity”, in the proceedings of the 2nd Indian International Conference. on Artificial Intelligence (IICAI-05), Pune, India, December 20-22, 2005.

[11] Shaoxu Song &Chunping Li, “Improved rock for text clustering using asymmetric proximity”, in the proceeding of the 32nd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM’06), Merın, Czech Republic, January 21-27, 2006, pp. 501–510.

[12] Pang-Ning Tan, Michael Steinbach, &Vipin Kumar, (2005) “Introduction to data mining”, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

[13] FabrizioAngiulli& Clara Pizzuti, (2005) “Outlier mining in large high-dimensional data sets”, IEEETransactions on Knowledge and Data Engineering Vol. 17, No. 2, pp. 203–215.

Authors

Dina Saidis a PhD student at the Dept. of Computer Science, University of Calgary. Dina graduated with a B.Sc and M.E from the Computer Engineering Department, Cairo University, Egypt in July 2003 and Feb 2007 respectively. Her research interests are in text and data mining, machine learning, and soft computing methods.

Nayer Wanas is an Assistant Research Professor in the Department of Informatics, Electronics Research Institute, Cairo, Egypt. He conducts research in the area of data mining. His main research interests are in data and text mining, information fusion and machine learning. Nayer has published over 40 technical papers in international journals and conference. He is a member of the IEEE and ACM and acts as a reviewer in several top tier journals and conferences.

Currently, he is the development manager at Smartec-Group, a start-up in the area of intelligent content delivery. Previously, Nayer spent over 3 years in Microsoft Research conducting research in the area of mining social media and web2.0 platforms. Prior to that, he was also involved in establishing and leading the Data Mining and Computer Modelling Centre of Excellence in Egypt. He was a postdoctoral fellow in the Pattern Analysis and Machine Intelligence lab at the University of Waterloo, Ontario, Canada where he focused on research in Multiple Classifier Systems. Nayer Wanas received his Ph.D. in Systems Design Engineering from the University of Waterloo in 2003, and his Bachelors and ME from the Electronics and Communications department at Cairo University in 1992 and 1996 respectively.