Mining an enriched social graph to model cross-thread community interactions and interests

Conference Paper · June 2012

DOI: 10.1145/2310057.2310066

Tarique Anwar
Macquarie University

Muhammad Abulaish
Jamia Millia Islamia

Mining an Enriched Social Graph to Model Cross-Thread Community Interactions and Interests

Tarique Anwar
Center of Excellence in Information Assurance
King Saud University, Riyadh, Saudi Arabia
tAnwar.c@ksu.edu.sa

Muhammad Abulaish, *SMIEEE*
Center of Excellence in Information Assurance
King Saud University, Riyadh, Saudi Arabia
mAbulaish@ksu.edu.sa

ABSTRACT

In this paper, we present a text mining approach to generate an enriched social graph to model cross-thread community interactions and interests of Web forum users. In addition to modeling *reply-to* relationships between users, the proposed approach models message-similarity relationship to keep track of all similar posts resulting out of deviated discussions in different threads. The generated social graph resembles a network of clusters, where the clusters are the group of similar posts and the binding links are the *reply-to* relationships between them. The graph can be presented at the granule of users who authored the posts to generate a social network, and at the same time it keeps information for all other users with similar interests.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentations**]: Hypertext/Hypermedia—*Navigation*; I.5.3 [**Pattern Recognition**]: Clustering—*Similarity measures*

General Terms

Algorithms, Design

Keywords

Web forum analysis, Social graph generation, User interactions modeling

1. INTRODUCTION

Unlike other Online Social Media (OSM), *Web forums* or *discussion boards* provide a platform for formal, vivid and dynamic discussions among an unrestricted number of participants. In this folksonomy, discussions are started by its members in the form of a discussion thread with a title and an entry message post. The viewers annotate their own opinions or replies to this thread, and thus the system keeps on evolving as the number of posts grows in it.

A common phenomenon observed in online threaded discussions is that they usually start from a specific topic, but as they grow with more posts, their context goes on deviating from its actual title [6]. Very often a deviated discussion is found to be overlapping with a different thread in the forum. A person replying to a deviated post in one thread is very much likely to reply similar posts in other threads if he comes to know about this kind of thread overlaps. The state-of-the-art research makes it very clear that the *reply-to* relationships play a prime role in interaction graph generation [4]. But in case of a deviated discussion, a simple reply-to relationship fails to capture the relation between a reply-post in a thread, and the posts in other threads which are similar to the post to which the former replied. In this paper, we propose a novel enriched social graph generation method, which, in addition to identifying reply-to relationships, identifies message-similarity relationship to keep track of all similar posts resulting out of deviated discussions and thus models cross-thread community interactions and interests. The novelty of the proposed method lies in establishing cross-thread linkages using the post-similarity relationship, and generating a condensed social graph of the entire forum community.

Starting with a review of related works in section 2, we discuss the proposed social graph generation method in section 3. Section 4 presents experimental results. Finally, section 5 concludes the paper.

2. RELATED WORK

The inherent complexities and lack of support from the online platforms powering forums bring about various challenges in capturing user interaction structures, roles and behaviors. Chan and Hayes [3] established user communication roles in discussion forums by analyzing several categories of features including structural, reciprocity, persistence, popularity, and initialization. Gomez *et al.* [5] created a social network from discussion threads in *Slashdot* using user interactions and their main objective remained statistical analysis of the generated network. The Hybrid Interactional Coherence (HIC) algorithm [4] generates an interaction graph of users that is basically composed of reply-to interactions. As reply-to relations are not always explicit in Web forums, Fu *et al.* adopted three key feature-matches including system feature match (consisting of header information match and quotation match), linguistic feature match (consisting of direct address match and a lexical match algorithm), and residual match. Rather than using reply-to relationship between posts, Liu *et al.* [8] exploited similarity measure to

MSM'12, June 25, 2012, Milwaukee, Wisconsin, USA.

generate social network structure of a forum. In [7], Kang and Kim generated an information flow network from discussion threads, in which a node represents either a user or a message, and an edge represents the reply-to or authorship relationship. Messages posted by same user are connected globally across the forum in different threads using an authorship relationship. Unlike others, Aumayr *et al.* [2] applied a machine learning approach to capture the reply-to relationships using a set of five fundamental features – *reply distance*, *time difference*, *quotes*, *cosine similarity*, and *thread length.* They used SVM and C4.5 classifiers and comparatively analyzed them by varying feature combinations.

3. PROPOSED METHOD

The proposed social graph mining method primarily consists of four major tasks – *forum crawling and pre-processing*, *reply-to relationship identification*, *similarity-based clustering*, and *social graph generation.* The crawling and pre-processing tasks form the base of the system to get an organized data set as a collection of threads having a title and a unique id, each thread consisting of one or more posts that in turn comprise a post id, time-stamp, body, author and quotations, if they exist. Details about each author comprising user id, joining date, location, and total posts are collected separately. The remaining core tasks performed on these data are described below in further detail.

3.1 *Reply-to* Relationship Identification

When a thread is initiated, it is assigned a title, and an initial post is attached with it, often called *entry post.* The entry post simply elaborates its title and waits for other's comments on it. Viewers, who find interest on the newly initiated thread, comment on it either by quoting an existing post to respond specifically or by a quote-less post.

Most of the time quotations accompanying a post occur as a simple single quote to another post. Multiple quotes (a post quoting multiple other posts at a time) and nested quotes (a post quoting a quoted post), are also encountered occasionally to focus on specific points in the discussion. All of them are neutralized by breaking down the multiple quotes into multiple single quotes, and processing the nested quotes to drop all the nested inner quotes except the outermost. An author may sometimes find a lengthy quote message to be cumbersome, and to focus on a specific point may edit the message to delete rest of its body. In this kind of behavior, it becomes difficult to trace the post to which is it responding by the quote. To overcome these issues, if a simple complete match fails to identify a reply-to relation, we follow a sliding window technique [4]. In this technique, the text of earlier posts as well as the quote is broken down into substrings (windows) and the quote-post pair with highest number of substring matches are linked.

For comments that are posted without quoting any of the existing posts, because of having no sound clue it becomes very difficult to establish the reply-to ($\Rightarrow$) relationship. Although some prior research works use the notion of similarity of posts to establish a reply-to relationship [4], contradictory to this, we found that simply a similarity of textual contents doesn't provide much evidence for a reply-to relationship. Rather a higher similarity shows an imitation of the same words, which very often is not true in posts connected with a reply-to linkage.

While commenting in a thread, very often people use author name of an earlier post in text to reply to that specific user, instead of quoting [4]. To capture this information, a search for a match of usernames of earlier posts in the body text may lead to establish an obscured reply-to link. At the same time, as we know that an online conversation is hardly given a serious attention, the writing style remains far from a formal way of writing. Unintentional misspellings and grammatical errors are commonly found in them, and many times usernames which do not look like real names are intentionally trimmed to make it like a real name. To overcome this hurdle, we apply the approximate string matching (ASM) metric of Jaro-Winkler [9], which is primarily intended for short strings, to check if there exist a misspelled author name in the body text of a comment.

Even after applying username string matching algorithm in the body text, there remains considerable number of reply-to relationships undiscovered, and to identify which we follow a rule based classification. In this matching, we make use of communication patterns as used in HIC [4].

3.2 Similarity-Based Clustering

To capture inter-thread similarity of posts, we apply a similarity-based clustering approach to group posts irrespective of threads to which they belong. Prior research show that a similarity comparison of Web forum posts is not as trivial as usual content similarity [8]. Liu *et al.* [8] defined this measure as a function of body text appended by thread title and author of the post. However, we noticed an additional factor to count for the similarity measure. Generally, time plays a substantial role in deciding the topics of discussion and its deviation, with respect to the daily happenings in one's personal life. For example, immediately after the tsunami outbreak in Japan in March 2011, all social media got flooded with this hot discussion all over the world. Hence, we observed that the discussions going in close proximity are likely to be more similar than those with a considerable time gap, and we have incorporated timestamp of a post along with other factors to measure similarity as described here. In our earlier work [1], we have applied a similarity-based clustering approach to identify cliques in dark Web forums. Here we follow the same approach to get cluster of posts.

The overall similarity between each pair of posts is identified using four different similarity measures – *content similarity*, *title similarity*, *author similarity* and *time similarity.* After transforming the posts into vector space model (VSM), content similarity between a pair of posts is calculated as the cosine between their vectors. Title similarity measures the cosine similarity between the thread titles in the same way as content similarity. Author similarity value is set to 1 if a pair of post is written by the same author, otherwise 0. Time similarity is determined based on the difference between the time-stamps of the posts.

Finally, the overall similarity, $Sim(p_j^i, p_l^k) \in [0, 1]$, is determined by aggregating all four measures using equation 1, where α, β, γ and δ are constants such that $\alpha+\beta+\gamma+\delta = 1$.

$$\begin{aligned} Sim(p_j^i, p_l^k) = \alpha \times CSim(p_j^i, p_l^k) + \beta \times TSim(p_j^i, p_l^k) \\ + \gamma \times ASim(p_j^i, p_l^k) + \delta \times LSim(p_j^i, p_l^k) \end{aligned} \quad (1)$$

The calculated similarity measures are used in our agglomerative clustering algorithm, shown in Figure 1. Considering n_0 number of total posts in a forum at time $t = 0$ the algo-

```
Algorithm postClustering(P)
Input: A set of posts $P = \{p_1, p_2, \cdots, p_n\}$ in a forum
Output: A set of cluster of posts $C = \{c_1, c_2, \cdots, c_m\}$
$C^0 = \{c_1^0 \leftarrow p_1, c_2^0 \leftarrow p_2, \cdots, c_{n_0}^0 \leftarrow p_n\}$      // assign all posts to a separate cluster
$t \leftarrow 0$
do
    if $t = 0$ then
        $\Phi_{n_t \times n_t}^t \leftarrow createSimilarityMatrix(C^0)$
    else
        $\Phi_{n_t \times n_t}^t \leftarrow createMatrix(n_t \times n_t)$
        for each $i$ and $j$ in $\Phi_{ij}^t$ do
            $\Phi_{ij}^t \leftarrow \frac{\sum_{c_k^{(t-1)} \in c_i^t, c_l^{(t-1)} \in c_j^t} \Phi_{kl}^{(t-1)}}{|c_i^t| \cdot |c_j^t|}$
            if $\Phi_{ij}^t \geq \varepsilon$ then
                $\Lambda^t \leftarrow \Lambda^t \cup \{(c_i^t, c_j^t, \Phi_{ij}^t)\}$
            end if
        end for
    end if
    $rank(\Lambda^t)$ on decreasing value of $\Phi_{ij}^t$
    do
        $\{(c_i^t, c_j^t, \Phi_{ij}^t)\} \leftarrow top(\Lambda^t)$
        $merge(c_i^t, c_j^t)$
        for each element in $\Lambda^t$ do
            $\{(c_k^t, c_l^t, \Phi_{kl}^t)\} \leftarrow \Lambda^t$
            if $c_i^t = c_k^t, or\, c_i^t = c_l^t, or\, c_j^t = c_k^t, or\, c_j^t = c_l^t$ then
                remove $\{(c_k^t, c_l^t, \Phi_{kl}^t)\}$ from $\Lambda^t$
            end if
        end for
    until $\Lambda^t$ becomes empty
    $t \leftarrow t + 1$
    $C^t \leftarrow getClusters()$
until $|C^t| = |C^{(t-1)}|$      // until clusters remain unchanged
$C = C^t$      // set of clusters returned
```

Figure 1: Proposed agglomerative clustering algorithm

rithm starts with a set $C^0 = \{c_1^0, c_2^0, \cdots, c_{n_0}^0\}$ of n_0 clusters assuming every post dissimilar from others. At each iteration, t, a similarity matrix $\Phi_{n_t \times n_t}^t$ is maintained to contain the similarity information between each pair of clusters.
At time, t, each value in the matrix, $\Phi_{n_t \times n_t}^t$, is compared with the similarity threshold value, ϵ. The pair of clusters for whom this value is found to be greater are added to the set of pairs, Λ^t, that need to be merged. After collecting all such cluster pairs, we rank them by their corresponding matrix values. Starting with the top ranking pair, two clusters are merged to form a unified cluster and all those pairs in Λ^t containing either of the two sub-clusters are removed from the set. The merging process is continued until Λ^t becomes empty. After merging, it proceeds to next iteration, t+1, the new set of clusters becomes $C^{(t+1)}$ with number of clusters as $n_{(t+1)} < n_t$, and the new matrix becomes $\Phi_{n_{(t+1)} \times n_{(t+1)}}^{(t+1)}$.

Each cluster, c_i^t, at time, t, keeps information about all its posts grouped into two sub-clusters, $c_k^{(t-1)}$ and $c_l^{(t-1)}$, if c_i^t is a result of merging $c_k^{(t-1)}$ and $c_l^{(t-1)}$, else c_i^t contains a single cluster of posts, $c_k^{(t-1)}$, the same as it was in last iteration. Each value, Φ_{ij}^t, in the new matrix is calculated using equation 2, where $|c_i^t|$ and $|c_j^t|$ denote the number of sub-clusters in c_i^t and c_j^t, respectively.

$$\Phi_{ij}^t = \frac{\sum_{c_k^{(t-1)} \in c_i^t,\ c_l^{(t-1)} \in c_j^t} \Phi^{(t-1)}(k, l)}{|c_i^t| \cdot |c_j^t|} \tag{2}$$

3.3 Social Graph Generation

We have differentiated a reply-to relationship from the property of posts being similar. Our enriched social graph considers a cluster of similar posts in the forum as a node, and the reply-to relationships between posts from different clusters as directed links to connect the nodes. Let $P = \{p_1, p_2, \cdots, p_m\}$ be the set of total posts in all the threads and $R = \{r_{ij}\}$ be the set of relationships $p_i \Rightarrow p_j$ between posts, p_i and p_j. Let us suppose that the set of clusters generated using the clustering algorithm is $C = \{c_1, c_2, \cdots, c_n\}$. Now, the enriched social graph consists of n cluster nodes with the set of relationships, $R^e = \{r_{kl}^e\}$, where r_{kl}^e is defined in equation 3.

$$r_{kl}^e = \bigcup_{p_i \in c_k, p_j \in c_l} r_{ij} \tag{3}$$

Each post associates with it the thread title, author name or user id and timestamp, and this enriched social graph can be presented in various forms for its analysis.

4. EXPERIMENTAL RESULTS

The experiments are conducted on a real dataset of "*eActivism and Stormfront Webmasters*" forum under the *Activism* category in the popular Stormfront social Web forum. The reply-to relationship identification task is evaluated by using the metrics, Precision (π), Recall (ρ) and F-score (F_1). A gold standard set is created by manually selecting some age-long threads relevant to the forum category. Only 29 threads are found having more than 40 comments on them. Discarding the irrelevant ones, we stuck to 10 threads. Two independent users are assigned to manually identify all actual reply-to relationships based on their context, and finally conflicts were resolved on a mutual consent. Another set of relationships identified by the proposed method are also collected. Values of the evaluation metrics are calculated using these two sets, shown in Table 1 along with their statistical summary. As another part of this experiment, the established relationships between posts are transformed to establish them between users. As there exist some common users in different threads, the relationship established for a user in one thread is continued over and integrated with the relationships in other threads, which connected users in different threads to form a network. The generated network consisted of 3 distinct components, each of whom represents a closed-group of inter-related users, shown in Figure 2. There are a total of 310 nodes (overall participants) and 545 directed edges (reply-to relationships) generated from a total of 934 posts distributed in 10 discussion threads.

The similarity-based clustering algorithm is evaluated in terms of $F_{\alpha=0.5}$ (or F_P[1] at $\alpha = 0.5$) and $F_{B\text{-}cubed}$ (or F_B[2])

[1] If C is the set of clusters generated by the automated system and L is the gold standard set, then $purity = \sum_i \frac{|C_i|}{n} maxPrecision(C_i, L_j)$ and $inversepurity = \sum_i \frac{|L_i|}{n} maxPrecision(L_i, C_j)$. F_P is calculated as their harmonic mean.

[2] For each element (or post), i, precision and recall val-

Table 1: Result summary of *reply-to* relationship identification process

Thread No.	Posts	Participants	π	ρ	F_1
1	68	22	0.784	0.816	0.800
2	105	13	0.727	0.715	0.721
3	122	37	0.831	0.847	0.839
4	82	54	0.802	0.790	0.796
5	185	48	0.878	0.845	0.861
6	58	41	0.691	0.733	0.711
7	55	11	0.856	0.862	0.859
8	169	52	0.773	0.816	0.794
9	44	11	0.758	0.809	0.783
10	46	37	0.887	0.914	0.900
Avg.	**93.4**	**32.6**	**0.799**	**0.815**	**0.806**

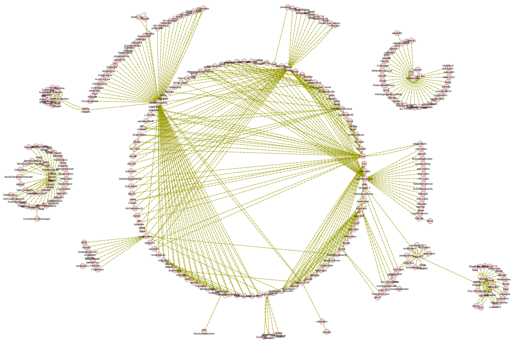

Figure 2: Network generated by system-identified *reply-to* relations between users

measures. Same strategies as earlier are followed to group the posts manually, which produced a set of 207 clusters as a gold standard from the same set of 934 posts. For automatic clustering, tuning the parameters α, β, γ and δ, is another challenge. The ideal way is to learn them from the manually annotated set, and we leave it as an application issue. In our case, we experimentally set them to 0.7, 0.1, 0.1 and 0.1, respectively, and generate the similarity matrix. Thereafter the clustering algorithm is executed by varying similarity threshold, ϵ, from 0.2 to 0.5 in intervals of 0.05.

Figure 3 presents the impact of varying ϵ on the evaluation metrics. Considering $\epsilon = 0.3$ as the ideal threshold, the F_p and F_B values in this experiment are found as 0.825 and 0.804, respectively.

Proceeding forth, all the 545 reply-to relationships among 934 posts are unified to construct the social graph at cluster-level. On unifying these post-to-post relations to map them to cluster-to-cluster relations, we got 332 relations in-between 173 clusters, identified above.

5. CONCLUSION

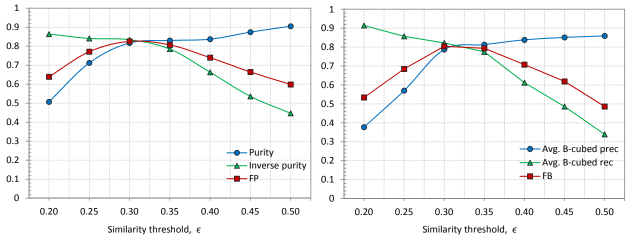

Figure 3: Impact of ϵ on *Purity* and *B-Cubed* measures

In this paper, we have presented a methodology to model Web forum discussions into an enriched social graph using user interactions and their overlapping interests, with a deliberate consideration of deviated discussions. The user interactions link posts through *reply-to* relationships, whereas the overlapping interests lead to merge similar posts into clusters, and thus collapse the generated network.

6. ACKNOWLEDGMENT

The authors would like to thank King Abdulaziz City for Science and Technology (KACST) and King Saud University for their support. This work has been funded by KACST under the NPST project number `11-INF1594-02`.

7. REFERENCES

[1] T. Anwar and M. Abulaish. Identifying cliques in dark web forums- an agglomerative clustering approach. In *Proc. of the IEEE ISI Conf.*, 2012.

[2] E. Aumayr, J. Chan, and C. Hayes. Reconstruction of threaded conversations in online discussion forums. In *Proc. of the AAAI ICWSM Conf.*, pages 26–33, 2011.

[3] J. Chan, C. Hayes, and E. Daly. Decomposing Discussion Forums using User Roles. In *Proc. of the WebSci'10*, 2010.

[4] T. Fu, A. Abbasi, and H. Chen. A hybrid approach to web forum interactional coherence analysis. *J. Am. Soc. Inf. Sci. Technol.*, 59(8):1195–1209, 2008.

[5] V. Gómez, A. Kaltenbrunner, and V. López. Statistical analysis of the social network and discussion threads in slashdot. In *Proc. of the Int'l WWW Conf.*, pages 645–654, 2008.

[6] Y.-H. Guan, C.-C. Tsai, and F.-K. Hwang. Content analysis of online discussion on a senior-high-school discussion forum of a virtual physics laboratory. *Instructional Science*, 34(4):279–311, 2006.

[7] J.-H. Kang and J. Kim. Analyzing answers in threaded discussions using a role-based information network. In *Proc. of the IEEE Int'l Conf. on Soc. Comp.*, 2011.

[8] D. Liu, D. Percival, and S. E. Fienberg. User interest and interaction structure in online forums. In *Proc. of AAAI ICWSM Conf.*, pages 283–286, 2010.

[9] W. E. Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proc. of Sec. on Survey Res.*, pages 354–359, 1990.

ues are computed individually as $precision_i = \frac{C_i \bigcap L_i}{C_i}$ and $recall_i = \frac{C_i \bigcap L_i}{L_i}$. The average B-cubed precision and recall are computed as the mean of individual values. F_B is calculated as their harmonic mean.