HKU CS Tech Report TR-2011-03

Detection of Sockpuppets in Online Discussion Forums

Xueling Zheng, Yiu Ming Lai, K.P. Chow, Lucas C.K. Hui, S.M. Yiu
Department of Computer Science
The University of Hong Kong
{xlzheng,ymlai,chow,hui,smyiu}@cs.hku.hk

Abstract

One common trick to cheat people to believe fake products or a high-return low-risk investment scheme in the online discussion groups is to make use of sockpuppets (i.e., use different fake identities pretending to be different persons to praise or create the illusion of support for the product). A fundamental problem is how to identify these sockpuppets. In this paper, we propose two approaches to identify these puppet pairs that occur in the same forum as well as cross forum. The evaluation based on millions of real posts in two popular discussion forums in Hong Kong shows that the methods are effective.

1 Introduction

Online discussion forum provides an excellent platform for users to communicate and share knowledge. On the other hand, it also facilitates criminals to swindle and scam. One example is the sexual grooming of children that occurs commonly in the Internet [1]. To cheat people in the Internet, for instance, to believe that a product is a good buy or a particular investment plan has extremely high return and low risk, a common trick is to use different fake online identities pretending to be different persons to praise or create the illusion of support for the product. Recently, a senior editor at The New Republic got in trouble for some particularly colorful sock puppetry [2]. It was found that he used a sockpuppet named "sprezzatura" to offer extravagant praise to himself on the blog, while receiving a good number of other posters' harshly critical [2]. Recently, a British historian Orlando Figes admitted that he used the alias "Orlando-birkbeck" and "Historian" to praise his own work as "fascinating" and "uplifting" while rubbishing that of his rivals [3].

Two different online entities (accounts), but belong to the same person, are referred as sockpuppet pairs. In this paper, we focus on identifying these sockpuppet pairs in the same online discussion forum or even in different discussion forums. In particular, we propose two methods for detecting sockpuppets. The first one is designed for detecting those sockpuppet pairs in the same discussion forum while the second one is for detecting sockpuppet pairs that appear in two different forums. Both approaches are simple and efficient. We try to avoid interpreting the contents of the posts. The first method only relies on the relative number of replies between the suspected sockpuppets while the second method makes use of an automatic approach to retrieve some keywords from the posts and base on these keywords, try to deduce if the candidates are sockpuppets. We evaluate our methods using more than 10 million real posts involving one million authors from two popular discussion forums of Hong Kong. The results show that our approaches are promising.

Related work: To the best of our knowledge, detection of sockpuppets has not been addressed in the current literature. Other related work includes the followings. Due to the rapid development of online social communities and discussion forums, some algorithms have been proposed for detecting the social roles of users in an online conversation space automatically. Welser et al. used structural signature methods to identify key roles, such as technical editors, substantive experts, vandal fighters, and social networkers in Wikipedia [4]. Chaoji et al. presented a method called Recursive Data Mining to mine the organizational roles and identify authors [5]. In [6], the authors applied social network analysis into their approach to characterize different authors in Usenet Newsgroups.

Some other researchers focus on finding the communities in online social networks. A community is a subset of the users that are more tightly interconnected compared to the whole network. Mining the communities can be useful for guiding information dissemination and acquisition and expressing access control policies. [7, 8, 9] defined and identified communities in networks. Du et al. also proposed ComTector to detect the communities efficiently in large-scale social networks [10].

The rest of this paper is structured as follows. We first describe the framework of the system that we implemented to automatically collect posts from different forums in Section 2. The method for detecting sockpuppet pair in the same forum is given in Section 3. In section 4, we present two approaches to detect sockpuppet pairs in two different forums. Section 5 concludes the paper.

2 Framework of the system

We designed a system to collect posts from more than 10 popular forums in Hong Kong. The system consists of 4 parts: Crawlers, Database, Data-mining engine and Statistical engine. Figure 1 shows the overview of the system. The crawler will automatically retrieve posts from the target forums and store these in the database. Both the data-mining and statistical engines are responsible for the analysis of the posts. The methods introduced in this paper is one of the components inside the data-mining engine.

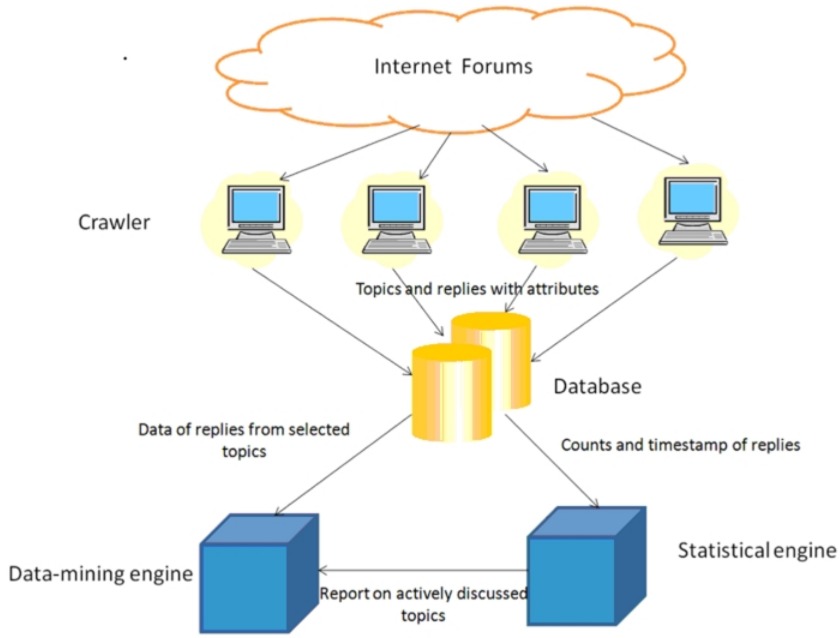

Figure 1: Overview of system

3 Mining sockpuppets in one discussion forum

Sockpuppet exists in blogs, online social networks, and discussion forums. In this paper, we focus on detecting sockpuppets in online discussion forums. According to our investigation, the followings are the purposes that one person registers two accounts in online discussion forums.

1. Two accounts are used for different purposes. For example, one account is used in category "sport", which the user's is interested in. The other one is used in category "buy & sell", where the user buys & sells things under the second account. In this situation, these two accounts can be treated as they belong to two different persons. We do not discuss a lot about this circumstance.

2. Two accounts are used to support each other in selling something or cheating someone else (e.g. for advertisement). In this case, the user usually posts topics for a product by one account, and then uses another account to reply the topics to support the product. The co-occurrence of the two accounts will be large.

Since the two accounts of (1) can be treated as they are owned by two different persons and usually do not involve in crime or cheating cases, we only focus on the second case. The following shows some of the observed characteristics of the second case (we use advertisement as a typical example).

(i) Since not too many other people actually use the products, it is unlikely that the topic will have a lot of replies from other users. Thus, the number of repliers for these topics are usually small if not only the sockpuppet pair.

(ii) The register dates of two accounts are usually very close. These two accounts are active for a short period after the register date since they do not want to use these accounts for a long period to be traced or tracked by other users.

(iii) The two accounts have equal status. Any one of the two can post a topic, and the other one behaves as a supporter. Moreover, both of them rarely reply other users' topics.

3.1 The algorithm of detecting sockpuppets in one forum

We first try to identify sockpuppets that exist in the same forum. It is possible that a person owns more than two sockpuppets. In this paper, we will focus on identifying sockpuppet pairs. Based on these pairs, further investigation should be carried out to relate these pairs with one another.

Based on the observation given in Section 3.1, we derive a simple approach to identify sockpuppet pairs without analyzing the contents of the posts. The core of our approach is based on the total (weighted) number of topics posted by one account and the relative number of replies from the other account with respect to the all his replies. Let $A_1, A_2, ..., A_n$ be n accounts and $t_1, t_2, ..., t_m$ be m topics in online discussion forum F during a period P. Denote $num(A_j \to A_i)$ as the number of topics that A_i posted and A_j has replied. The weight of a topic t, denoted as $weight(t)$, is the number of distinct users that have replied to the topic. Let $weight(A_i)$ be the sum of the weights of all topics posted by A_i and $num(A_j)$ be the number of all the topics that A_j has replied. We define the score of two accounts, A_i, A_j being the sockpuppet pair as follows.

$$DetectionScore(A_j \to A_i) = \frac{num(A_j \to A_i)}{num(A_j)} + \frac{num(A_j \to A_i)}{weight(A_i)} \tag{1}$$

If $DetectionScore(A_j \to A_i)$ is larger than a threshold α, we say $PuppetPair(A_j \to A_i)$ is true and A_i and A_j form a sockpuppet pair. The maximum value of $DetectionScore(A_j \to A_i)$ is 2 which means that A_j is the only person who has replied every topic posted by A_i. The larger the score, the more likely the two accounts is a sockpuppet pair. This approach has the advantage of not considering the contents of the posts and can be used in real forums with a huge number of posts.

To further enhance the accuracy of our prediction, based on the observation that these sockpuppet accounts may not be active for a long period of time to avoid being traced by other users, we consider the overall account active time. Here, we define the overall active time of an account A_k, denoted by $ActiveT(A_k)$, as the time difference between its first and the last post. If the active time of an account is longer than a threshold τ, say one month, we assume that this account is unlikely to be a sockpuppet.

3.2 Experimental results

We evaluate our approach based on real posts in Uwants. More than 8 million posts from Uwants during the period of Jan 2009 - Dec 2009 are retrieved. There are almost one million users in Uwants.

3.2.1 Overall performance

Recall that the detection score of $A_j \rightarrow A_i$ is at most 2, the larger the score, the more likely A_j and A_i belong to the same person. We use 2, 1.9, 1.8, 1.7, 1.6, 1.5 as the thresholds of α for evalution. Since there is no real answers, for each reported sockpuppet pairs, we examine all their posts and replies and based on our experience to determine if they are sockpuppet pairs or not. We define the accuracy of our approach as follows. $accuracy = \frac{\sum C_i}{(\sum C_i + \sum U_j)}$, where $\sum C_i$ is the total number of correctly detected pairs and $\sum U_j$ is the total number of false positives. In fact, although there are many users involved in the posts under evaluation, the number of returned sockpuppet pairs are not that many. For the threshold of 1.5, the total number of reported pairs is 111. It is true that there must be some sockpuppet pairs that are not located by our approach. However, from Figure 2, the accuracy of our approach (the green line) is quite high. ¿From the graph, it also shows that the use of the active time of an account is effective in improving the accuracy of the approach. In particular, for the threshold of 1.5, the accuracy is smaller than 0.34 without using the active account time, however, the accuracy increases to more than 0.9.

3.2.2 An example

To show one particular example, we select a sockpuppet pair whose detection score is 1.8. We look up the contents that were posted by the pair from the database. On December 7th 2009, one user posted a topic about a new formulation which contains lutein and also pointed out that lutein is good for eye health by the account Tom[1]. After several minutes, a user named Jerry pushed the topic and said that he had taken the formulation for three weeks and it was very good for him [11]. However, under another topic about the formulation by Tom on the same day, Jerry posted "how to get lutein" and it seems that Jerry knows nothing about the new formulation [12]. Moreover, on December 24th 2009, Tom asked a question about how to protect eyes and what was good for the eyes. After six minutes, Jerry recommended the formulation for Tom [13]. In fact, a user pointed out that this was an advertisement. Other users did not participate into the topics. In addition, Tom

[1]To respect privacy, the ID names are replaced by Tom and Jerry in the paper. You can find the real postings from the corresponding references.

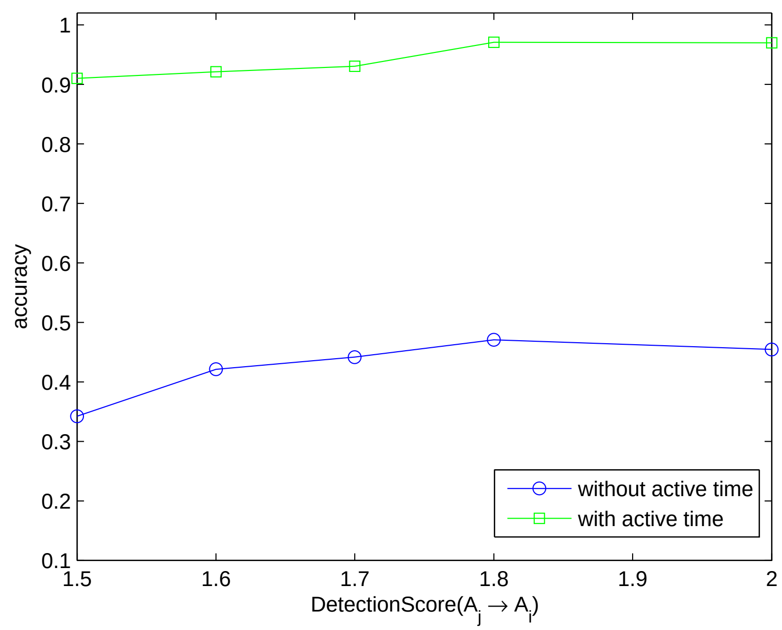

Figure 2: Compare the results under the same detection score

and Jerry were registered on the same day "December 7th 2009". So it is quite certain that Tom was in collusion with Jerry to advertise the product. There are other similar examples when the pair has a high detection score.

4 Sockpuppets in two different forums

The technique used detecting sockpuppets in the same forum is not applicable for sockpuppets exist in two different forums since they will not have direct replies on each other. Again, to have a detailed analysis on the contents of the posts should be avoided due to the heavy computational cost. Our approach relies on the idea of creating a keyword-based profile of the accounts based on the corresponding posts. We consider the posts of A_1 and A_2 and extract keywords from the posts, then compare the similarity of the two keyword profiles in order to determine whether they are likely to be a sockpuppet pair.

4.1 Keyword comparison

4.1.1 Simulation

Extract Keyword: Our system is implemented in Chinese discussion forum. Therefore, we concern the word segmentation in this section. In English and languages using some form of the Latin alphabet, the white space is a good approximation of a word delimiter. However, other languages such as Chinese and Japanese are written without any whitespace. Word segmentation of this kind of language is a difficult problem.

In Chinese, it is easier to segment words with a comprehensive dictionary. However, it is difficult to get a dictionary that contains all the existing Chinese words. It is almost impossible for online discussion forums, where new words are created every day. Moreover, sometimes a comprehensive dictionary may not work well since different segmentations can cause different results [14]. We propose a straightforward method to find the keywords efficiently and effectively.

The postings are processed to calculate the occurrences of n-adjacent-character word ($n = 2, 3, 4, 5, 6$) as shown in Figure 3. Since the weight of title is much higher than the weight of replies. The occurrence of a word in a title weight more than the occurrence of the word in the postings. Therefore, when a word appears in the title, its occurrence increases m ($m1$), otherwise its occurrence increases 1.

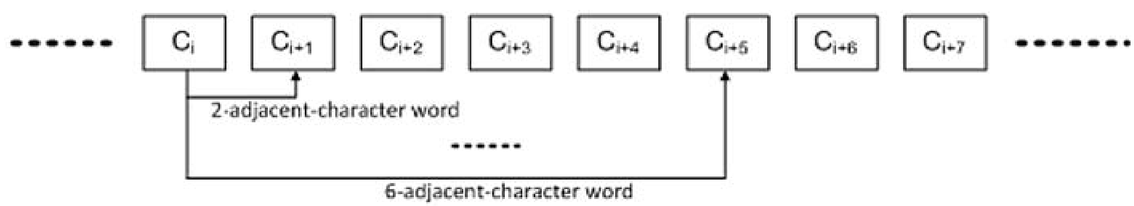

Figure 3: $C_i, ..., C_j (j > i \geqslant 0)$ are the characters in the postings. The occurrences of n-adjacent-character word (n=2, 3, 4, 5, 6) are calculated. The shift of the window is one character.

We can get the result of all the n-adjacent-character (n=2, 3, 4, 5, 6) words and their occurrences. If $(n-i)$-adjacent-character ($n > i+1$) word A is the substring of $(n-i+j)$-adjacent-character ($j > 0$) word B, and the occurrence of word B is larger than 70% of occurrence of word A, word A will be deleted; otherwise, word B will be deleted. With the steps of deletion, we can keep the good and longest keywords. The keywords are ranked by their occurrences to get an ordered keyword list. In Chinese, some words appear much more often than others. We build a common word list which contains the words that appear frequently but are not representative of authors. The common words in the ordered keyword list will be deleted.

Similarity computation: We retrieve A_1 and A_2' postings and topics respectively, extract the keywords and compare the keyword lists. The k-most frequent keywords of the two keyword lists are compared. If A_1 and A_2 have k_1 ($k_1 \leqslant k$) same keywords, their similarity will be calculated by

the function $similarity(A_1, A_2) = k_1/k$. Assume the threshold is θ, if $similarity(A_1, A_2) > \theta$, we say A_1 and A_2 belong to one person; if $similarity(A_1, A_2) \leqslant \theta$, A_1 and A_2 belong to different persons.
Parameter setting: It is difficult to get real data to set the parameters. We create our own testing data of sockpuppet pairs and non-sockpuppet pairs based on real posts to find the appropriate values of the thresholds. We believe that a more formal learning model can be employed later to get a better result. For the illustration of the effectiveness of our approach, we use a simple method to obtain a set of reasonable thresholds as follows.

500 authors who have posted more than 10 topics are randomly chosen from the database. (We do not choose the author who has posted topics less than 10 because the contents are not enough for test.)

Simulation of two accounts belong to one person

For each author, we retrieve its topics and postings. Assume Set S_1 contains all the topics and postings; and Set S_2 contains part of the topics and postings that are randomly chosen from S_1. Assume that account A_1 posts set S_1; and A_2 posts set S_2. We say the A_1 and A_2 belong to one person.

Simulation of two accounts belong to different persons

For each author, we retrieve its topics and posting and randomly divide the topics and postings into two mutually exclusive set: Set S_3 and Set S_4. Assume that account A_3 posts of set S_3; and A_4 posts set S_4. We say the owners of A_3 and A_4 are different persons.

We get 500 sockpuppet pairs (Set T) and 500 non-sockpuppet pairs (Set F). We try different combinations of m, k, and θ. The combination we found is $m = 15$, $k = 25$ and $\theta = 0.7$. And there are only 20 false classification in Set T and 22 false classification in Set F.

4.1.2 Experimental Results

We evaluate our approach based on real posts in two popular online discussion forums, Uwants and HK Discuss. Using our system, we retrieve more than 8 million postings from Uwants and HK Discuss during the period of Mar 2010 to May 2010[2]. There are more than one million users in both forums. We use our approach to detect sockpuppet pairs in the two forums.

We detected that 2257 sockpuppet pairs' out of 1000 billion account pairs. To show the experimental results, we discuss two detected sockpuppet pairs. For the first pair whose similarity is 1.0 (i.e. the highest similarity to be considered as a sockpuppet pair), we found that Tom posted 251 topics in Uwants and Jerry posted 251 topics in HK Discuss [15]. These 502 topics are all about online sex movie. Moreover, there are 248 topic pairs whose titles are the same. Each pairs whose topic titles are the same were posted almost at the same time. Therefore, we can confirm Tom and Jerry belong to one person who like to post postings about online sex movie and copy and past the same topic in another forum by another account. The second sockpuppet pair's similarity is 0.70 (i.e. the lowest similarity to be considered as a sockpuppet pair). This Tom in Uwants posted 8

[2]Due to some technical issues, we did not download the posts from Jan 2010 - Feb 2010

topics about asking help for his Apple iphone [16]. At the same time, Jerry in HK Discuss posted 7 topics about Apple iphone, and 5 topics are the same as the topics posted by Tom. Coincidentally, the ID names of Tom and Jerry are the same in two forums. So we can guess that Tom and Jerry belong to one person who bought a new iphone and encountered problems when using it. From the examples, we can see that our approach can detect the sockpuppet pairs in two forums.

4.2 Two-level method

After we have found our scheme can successfully detect sockpuppet pairs in two forums, we would like to improve the efficiency. From the results in section 4.1, we observed that user's different accounts in different forums trend to post the same topics. Moreover, the topics' posted time by the same user is usually near in two forums. That is said that the time interval of the topics' posted time is very small. We proposed a two-level method to detect the two forums sockpuppet pairs.

Assume that $A^1_1, A^1_2, ..., A^1_{n_1}$ are all the n_1 accounts and $T^1_1, T^1_2, ..., T^1_{m_1}$ are all the m_1 topics in online discussion forum F_1 during the period P; and $A^2_1, A^2_2, ..., A^2_{n_2}$ are all the n_2 accounts and $T^2_1, T^2_2, ..., T^2_{m_2}$ are all the m_2 topics in online discussion forum F_2 during the period P. The threshold of time interval is θ. A^1_i is an account in F_1 and A^2_j is an account in F_2. Two topics from two different forums form a topic pair if the time interval of two topics' posted time is smaller than θ. $num(A^1_i A^2_j)$ is the number of topic pairs that are posted by A^1_i and A^2_j and the titles of the two topics are the same. $num(A^1_i)$ is the number of topic pairs in which one topic is posted by A^1_i. Similarly, $num(A^2_j)$ is the number of topic pairs in which one topic is posted by A^2_j. $DetectionScore(A^1_i A^2_j) = \ln num(A^1_i A^2_j) + \{num(A^1_i A^2_j)/num(A^1_i) + num(A^1_i A^2_j)/num(A^2_j)\}$ is the detection score which shows whether A^1_i and A^2_j belong to one person. If $DetectionScore(A^1_i A^2_j)$ is larger than the threshold β, we say $PuppetPair(A^1_i A^2_j)$ is true and A^1_i and A^2_j are owned by one person.

If A^1_i and A^2_j are sockpuppet pair, it is more likely that their posted topics are the same. The larger $num(A^1_i A^2_j)$, the more likely $PuppetPair(A^1_i A^2_j)$ is true. In addition, the value of $num(A^1_i A^2_j)/num(A^1_i)$ and $num(A^1_i A^2_j)/num(A^2_j)$ will be close to 1, so the value of $num(A^1_i A^2_j)/num(A^1_i)+num(A^1_i A^2_j)/num(A^2_j)$ will be close to 2.

In first level of the method, we sort the pair by the value of $num(A^1_i A^2_j)$ and find the pair's ID names which are the same. In the second level, the left pairs are sorted by the $DetectionScore$.

Common Topic: We randomly picked up 10 million account pairs in Uwants and HK Discuss and computed the pairs' common topic number. Only 4 pairs' common topic number $num(A^1_i A^2_j) > 0$ and 3 pairs' common topic number is 1. Therefore, we say that common topic is quite rare in non-sockpuppet pairs. Common topic is a unique feature of sockpuppet pairs.

Experimental Result: We use the data in section 4.1.2 to test our two-level method. The second pair in the result list of level-one posted 234 topics in Uwants and 234 topics in HK Discuss and among them 218 topics are the same. These 468 topics are all about computer and software. Moreover, the ID names of the two accounts are the same. We can say that the two accounts belong to one person who was interested in computer and software. We also found that these two accounts' keyword similarity is 0.96 (i.e. to be considered as a sockpuppet pair in section 4.1). In the result list of second level, the first pair posted 2699 topics in Uwants and 795 topics in HK Discuss and among them 784 topics are the same. These 3494 (2699+795) topics are all about online sex movie. The common topic number of the pair is very large. We believed that the two accounts belong to one person who like to post postings about online sex movie and copy and past topics in another forum by another account. The two accounts' keyword similarity is 0.80 (i.e. to be considered as a sockpuppet pair) in section 4.1. More examples can be retrieved from [17][18].

We have shown that both the keyword comparison method (in section 4.1) and the two-level method (in section 4.2) are effective in finding sockpuppet pairs. From the efficiency point of view, the keyword comparison method consider all the postings and topics, yet the two-level method only consider account's topic titles. Therefore, the execution time will be much faster.

5 Conclusions

We proposed algorithms for detecting sockpuppet pair in one forum and two different forums. The algorithms are simple and efficient so that they can be applied to real online discussion forums. From our experiments, we found the results are promising. Our methods can be used for monitoring the online discussion forum for the police and the forum moderators. On the other hand, there are still a lot of sockpuppet pairs for which our methods cannot detect. A more effective solution is still desirable. Also, how the results of detecting sock puppet pairs can be used with other data mining results in order to provide effective clues for digital forensic investigation would be an interesting and challenging problem.

References

[1] "Child_grooming in wikipedia," *http://en.wikipedia.org/wiki/Child_grooming*.

[2] "Sock puppet bites man," *http://www.nytimes.com/2006/09/13/opinion/13wed4.htm*, 2006.

[3] R. Lea and M. Taylor, "Historian orlando figes admits posting amazon reviews that trashed rivals," *http://www.guardian.co.uk/books/2010/apr/23/historian-orlando-figes-amazon-reviews-rivals*, 2010.

[4] H. Welser, G. Kossinets, S. Marc, and D. Cosley, "Finding social roles in wikipedia," in *annual meeting of the American Sociological Association, Boston, MA, AllAcademic*, 2008.

[5] V. Chaoji, A. Hoonlor, and B. Szymanski, "Recursive data mining for author and role identification," in *3rd Annual Symposium on Information Assurance (ASIAˇ201908)*, p. 53, 2008.

[6] D. Fisher, M. Smith, and H. Welser, "You are who you talk to: Detecting roles in usenet newsgroups," in *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, vol. 3, p. 59b, IEEE, 2006.

[7] L. Donetti and M. Munoz, "Detecting network communities: a new systematic and efficient algorithm," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2004, p. P10012, 2004.

[8] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proceedings of the National Academy of Sciences*, vol. 101, no. 9, p. 2658, 2004.

[9] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Physical Review E*, vol. 74, no. 1, p. 16110, 2006.

[10] N. Du, B. Wu, X. Pei, B. Wang, and L. Xu, "Community detection in large-scale social networks," in *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pp. 16–25, ACM, 2007.

[11] nicyiu and tina_gg, "People who want their eyes more beautiful come in, please," *http://www.uwants.com/viewthread.php?tid=9113207*, 2009.

[12] nicyiu and tina_gg, "tell-tale signal," *http://www.uwants.com/viewthread.php?tid=9113233*, 2009.

[13] nicyiu and tina_gg, "Dark eye circles appear after party," *http://www.uwants.com/viewthread.php?tid=9181908*, 2009.

[14] X. Zheng, L. Huang, Z. Chen, Z. Yu, and W. Yang, "Hiding information by context-based synonym substitution," in *IWDW*, pp. 162–169, 2009.

[15] "Keyword comparsion result 1," `http://i.cs.hku.hk/~xlzheng/sockpuppet/similarity/1.htm`.

[16] "Keyword comparsion result 2," `http://i.cs.hku.hk/~xlzheng/sockpuppet/similarity/2.htm`.

[17] "Result of level one," `http://i.cs.hku.hk/~xlzheng/sockpuppet/levelOne.htm`.

[18] "Result of level two," `http://i.cs.hku.hk/~xlzheng/sockpuppet/levelTwo.htm`.