

Detecting Collusive Spamming Activities in Community Question Answering

Yuli Liu[†], Yiqun Liu[†], Ke Zhou[‡], Min Zhang[†], Shaoping Ma[†]

[†]Department of Computer Science & Technology, Tsinghua University, Beijing, China

[‡]School of Computer Science, University of Nottingham, UK.

yiqunliu@tsinghua.edu.cn

ABSTRACT

Community Question Answering (CQA) portals provide rich sources of information on a variety of topics. However, the authenticity and quality of questions and answers (Q&As) has proven hard to control. In a troubling direction, the widespread growth of crowdsourcing websites has created a large-scale, potentially difficult-to-detect workforce to manipulate malicious contents in CQA. The crowd workers who join the same crowdsourcing task about promotion campaigns in CQA collusively manipulate deceptive Q&As for promoting a target (product or service). The collusive spamming group can fully control the sentiment of the target. How to utilize the structure and the attributes for detecting manipulated Q&As? How to detect the collusive group and leverage the group information for the detection task?

To shed light on these research questions, we propose a unified framework to tackle the challenge of detecting collusive spamming activities of CQA. First, we interpret the questions and answers in CQA as two independent networks. Second, we detect collusive question groups and answer groups from these two networks respectively by measuring the similarity of the contents posted within a short duration. Third, using attributes (individual-level and group-level) and correlations (user-based and content-based), we proposed a combined factor graph model to detect deceptive Q&As simultaneously by combining two independent factor graphs. With a large-scale practical data set, we find that the proposed framework can detect deceptive contents at early stage, and outperforms a number of competitive baselines.

Keywords

Community Question Answering; Crowdsourcing Manipulation; Spam Detection; Factor Graph.

1. INTRODUCTION

Community question answering (CQA) portals, such as Yahoo! Answers, have become a popular platform for people to share their knowledge and learn from each other [30]. These Web sites have attracted a great number of users, and have accumulated a large amount of user-generated contents (i.e., questions and answers or Q&As). To seek advice or enrich knowledge, Internet users can find answers provided for previously asked questions in response to new queries. Because CQA has great influence on users' cognitions and judgments [24], tremendous malicious users try to manipulate contents to mislead common users, which makes the CQA environment less credible. Several previous research works focus on evaluating the quality of answers or question-answer (QA) pairs [1, 2, 14, 24], or identifying and removing manipulated contents from the archived Q&A resources [5, 15].

© 2017 International World Wide Web Conference Committee (IW3C2), published under Creative Commons CC BY 4.0 License.
WWW 2017, April 3-7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
<http://dx.doi.org/10.1145/3038912.3052594>



Nowadays, with the wide usage of crowdsourcing systems, massive organized manipulated contents pollute the CQA platforms. As shown in Figure 1, to gain economic benefits, malicious commercial campaign owners release tasks on crowd-sourcing platforms (e.g., Amazon Mechanical Turk). As we can see in one of real-world CQA promotion campaigns, the CQA crowdsourcing promotion task includes detailed descriptions and guidelines that the crowd workers (*CWers*) need to follow. The task requester only approves those submissions that meet the task description. Almost all these tasks provide task templates for workers to refer to, which contain keywords (marked in red color) such as product name and domain-specific word.

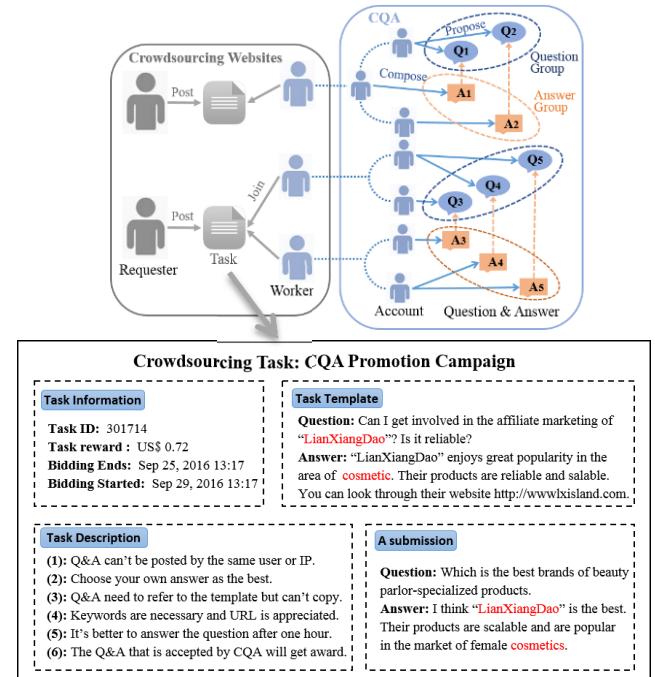


Figure 1: Collusive Spamming Activities on CQA Platforms.

After accepting those promotion campaign crowd-sourcing tasks, the crowd workers then accomplish those campaigns through posting malicious questions and using deceptive answers to respond the questions collaboratively. In this scenario, both questions and answers are polluted systematically. To avoid being identified, a *CWer* may create multiple accounts to make sure that one account will not answer his/her own question, and he/she reorganizes Q&As to make them superficially dissimilar. These collusive manipulated contents exert full control over the opinions of the commercial campaigns, which may be more detrimental than the common deceptive Q&As.

In this paper, we aim to detect the above-described collusive spamming activities in CQA in a very early stage. Rather than blocking certain spamming accounts, we focus on effectively detecting deceptive Q&As in a timely fashion. This is due to the fact that on average a crowd worker creates about 90 CQA accounts to manipulate Q&As (see the statistical analysis in Sec.3.2) and the spammers can be only detected after they post many spam contents [10].

Compared to prior works, many challenges arise regarding this problem: (1) (**Annotation Difficulty**) Considering the fact that CQA spamming is usually a collaborative activity, it is difficult to ascertain which contents are deceptive and which ones are legitimate; (2) (**Asymmetric Q&A Attributes**) In CQA, the questions and answers are asymmetric with different attributes and linguistic structures, which are different from deceptive product reviews [9, 10, 23] or promotional microblog posts [13] that can be analyzed uniformly; (3) (**Unclear Group Base**) Previous works group the spamming activities that review multiple common products in the review platforms [19, 28, 29] and post common URLs or contents in the microblog environments [4]. However, in CQA, there are not any clear existing connections that can group Q&As, because *CWers* can generate unlimited distinct questions, and the deceptive answers can respond to any of them. This makes deceptive contents in CQA more flexible; (4) (**Obscure Signals**) Compared to traditional spam bots that leave identifiable attributes [11, 12], these human-powered deceptive contents in CQA are inherently distinct and lack any easily identifiable signals [7]; (5) (**Early Detection**) Detecting the fast-growing crowd-sourcing deceptive contents at an early phase can reduce the damage of them, but it is challenging due to the very limited information in the timely detection tasks.

To tackle these issues, we propose to exploit the crowdsourcing tasks (promotion campaigns) to form the ground truth dataset. We regard questions and answers as two heterogeneous and independent networks while the asymmetric Q&A attributes therefore can be analyzed and utilized respectively. The submitted Q&As for a task are collusively posted to achieve a promotion campaign. Since they share the common theme (i.e., same keywords and domain information as shown at the bottom half of Figure 1), we consider the questions and answers extracted from the submissions of a task as the ground truth collusive (deceptive) question group and collusive answer group respectively (as top half of Figure 1 shows).

We then detect cooperative groups from two graphs (question and answer) that are built based on the common theme of their contents. Distinctive attributes (group-level) are extracted from the detected groups of the graphs. By integrating individual-level attributes and correlations (content-based and user-based), we construct question and answer factor graphs respectively. Finally, a combined factor graph model is proposed by combining the two factor graphs to detect collusive Q&As. Through extensive experimental comparisons with competitive baselines, we empirically show that our framework is robust, effective and capable of detecting collusive contents early.

This work is the first to analyze the group spamming activities in CQA, and apply group attributes to detect deceptive Q&As.

Our contributions are four-folds:

- Through locating CQA commercial tasks in crowdsourcing platforms, we create a CQA collusive spamming data set that contains deceptive Q&As and collusive group information, which is publicly available.¹

- We provide comprehensive analysis of deceptive and normal Q&As in CQA on both their individual and group attributes.
- We propose a group detection framework that can facilitate extracting identifiable collusive (group-level) attributes.
- We propose a novel detection framework that can effectively detect deceptive contents at the early stage.

2. RELATED WORK

Prior work on social spam detection can be categorized into two groups: individual spam detection and collusive spam detection.

Individual spam detection. The problem of opinion spam (i.e., deceptive review) detection has been extensively studied on the individual level. For example, Jindal and Liu [9] first study the opinion spam problem by analyzing Amazon data and detecting individual fake reviews. They identify three types of spam, and detect them by using supervised learning with manually labeled training examples. Feng et al. [8] regard the opinion spam as a distributional anomaly. They find a connection between distributional anomalies and the time windows when spam reviews are posted. In [22], the authors create a gold-standard fake review dataset through Amazon Mechanical Turk and use n-gram and POS tag features to train a classifier to detect them. Besides spam review detection, the problem of review spammer detection has also been widely studied in [28, 16, 23]. These research studies identify several features related to rating behaviors and model these features so as to detect the spam reviewers. However, those works can be only applied in the review systems.

With respect to spam detection approaches on CQA platforms, most of the previous works focus on estimating the quality of answers or QA pairs. The authors of [24] present a study to evaluate and predict the quality of an answer in a CQA setting based on logistic regression model using extracted features from questions, answers, and the users who posted them. In [14], the authors estimate question quality with a mutual reinforcement-based label propagation algorithm. Besides, Chen et al. [5] study the phenomenon of malicious commercial campaigns by analyzing more context information rather than textual similarities only. They develop a system that automatically analyzes the hidden patterns of commercial spam and raises alarms instantaneously to end users whenever a potential commercial campaign is detected. Li et al. [15] focus on promotion channels (URLs, telephone numbers and social media accounts) which are relied by spammers to connect users to achieve promotion goals. A propagation algorithm is proposed to detect possible spamming activities at individual level.

Collusive spam detection. Compared to individual detection problem, collusive detection receives less attention. Mukherjee et al. [19] are among the first to study group level spammers in review communities and propose a novel relation-based approach to detect spammer groups. Although many group behavior indicators are extracted and analyzed, they only aim to detect spam groups but not spammers or review contents. Xu et al. [28] focus on collusive review spammer detection by combining individual and collusive indicators to detect colluders. In [7], the authors use a Conditional Random Field model to cluster reviewers. They embed the results of this probabilistic model into a classification framework directly for detecting crowd-manipulated reviews rather than extracting attributes from detected groups. Besides, collusive activities are also common in the Microblogging environment. Cao et al. [4] find that embedding individual-based behavioral signals in URL posting activities can uncover groups whose members engage in similar behaviors while group-level behavioral signals can distinguish between organic and organized user groups.

¹ <http://www.thuir.cn/group/~YQLiu/>

Most of the collusive spam detection efforts focus on the spamming activities in product review sites based on the clear review-product relations. As mentioned, collusive spamming activities on CQA platforms might be more flexible and more challenging to be detected. In addition, although prior work exists on detecting individual or group spamming activities in product review sites and uncovering commercial campaigns (deceptive questions or answers) at individual level, little is known about how colluders disseminate deceptive Q&As and how to effectively detect the crowdsourcing manipulated contents on CQA platforms. Therefore, in this work, we focus on studying the more challenging CQA collusive spam detection problem and incorporate the group attributes to simultaneously detect deceptive questions and answers.

3. DATA COLLECTION AND ANALYSIS

In this section, we focus on collecting and analyzing deceptive contents in CQA. We aim to collect the data in order to generate a publicly available test set that can enable us to provide insights and evaluate our algorithms.

3.1 Data Collection

Our collected data consists of two parts: the collusive spamming dataset and the ordinary dataset.

3.1.1 Collusive Spamming Dataset

In several popular crowdsourcing platforms, such as Zhubajie.com and RapidWorkers.com, the crowd workers who participate in a CQA promotion task are required to submit the promotion CQA URLs to indicate that they have accomplished the task. This provides a chance for us to acquire ground truth of deceptive Q&As. To collect this data, we first locate the CQA promotion tasks in the crowdsourcing platforms using manual searching and filtering (using the key words such as CQA and promotion) of the search results. All the product or service names that the tasks aim to promote are manually extracted, which are clearly mentioned in the requirement descriptions (see Fig.1). Through this way, we obtain 2,625 tasks that contain about 40K promotion CQA URLs.

Based on these URLs, we crawl all the Q&As in the corresponding pages in CQA, and obtain 40K gold deceptive questions ($+Q_s$). Usually, to increase the visibility of the promotion answers, the tasks require asker to adopt the promotion answer as the best answer. Therefore we deem the best answer of a malicious question as a deceptive answer ($+A$) because this is adopted by a deceptive user for collusive spamming purpose. In addition to the best answers, about 24K other answers are also collected. Since normal users may also answer the deceptive questions, we consider these non-best answers as candidate deceptive answers ($+A^*$ s) rather than simply treating them as deceptive ones. A normal answer is unlikely to mention a promoted product name because most of these products are not popular and of low quality, therefore we label a candidate answer as deceptive, if it contains any promoted product names. By this means, we obtain another 21K deceptive answers. This result shows that only a few normal answers (3K, 12.5%) respond to deceptive questions.

As mentioned in related work, spam activities can be investigated at different levels. It is relatively easier to define individual abnormal activities than collusive activities. In the review platforms, the products can be used to connect reviewers or reviews to help cluster groups [19]. Such connection may not exist in CQA, because the Q&As are not organized according to specific targets (i.e., no product items in practical CQA). However, the submitted Q&As of the same crowdsourcing task share the common theme [7] and they are collusively posted to achieve a promotion cam-

paign (mentioned above). Therefore, since the promotion tasks are definite in our collected dataset, the deceptive questions ($+Q_s$) and deceptive answers ($+A_s$) can be clustered into the corresponding groups (two types) clearly according to the crowd-sourcing tasks (i.e. the promotion campaign) they belong to. So besides deceptive Q&As, the spamming dataset also contains the ground-truth deceptive question groups ($+QGs$) and deceptive answer groups ($+AGs$).

3.1.2 Ordinary Dataset

For comparison, we construct an ordinary Q&A dataset by randomly collecting Q&As from the CQA platforms. Using the method of manual annotation to sample normal Q&As is not applicable due to the fact: (i) it is difficult for judges to ascertain which contents are deceptive and which ones are legitimate without any preliminary knowledge (e.g. the promotion campaign information from the crowdsourcing sites); and (ii) for comprehensive analysis, a large number (tens of thousands) of Q&As need to be labeled which is time consuming and labor intensive.

In CQA platforms, such as Yahoo! Answers or Baidu Zhidao, each question has a hashed unique numeric ID (i.e., q_{id}) according to the posted time (the number of digits in q_{id} is different in different periods) and the URL of the corresponding question page is bonded with the ID. Inspired by this fact, the ordinary dataset is collected through: (i) we obtain the unlabeled q_{id} set $Uq = \{q_{id} \in N^+ | Cq_i - 5 \leq q_{id} \leq Cq_i + 5, q_{id} \neq Cq_i, Cq_i \in Cq\}$, where Cq represents the set of $+Q_s$ ' ID. (ii) we crawl all the questions whose IDs are in the set Uq but not in Cq and all their answers which are not in $+A_s$.

The ordinary dataset is collected in this way because: (i) we want to ensure that the unlabeled questions are selected relatively randomly. Since a $+Q$ can be posted at any time, the corresponding ordinary Q&As can be considered randomly selected; (ii) we want most of the unlabeled questions ordinary. Most the contents in CQA are ordinary [15], so few of the randomly crawled Q&As are malicious; (iii) we want to try our best to simulate a practical CQA spam detection task scenario, in which the algorithm has to separate spams from normal contents that are generated within a same time period. Since the IDs are generated according to the time stamps (not in chronological order due to they are hashed), we can ensure that the contents in the ordinary dataset are generated within the similar time periods as the deceptive ones.

To verify that most contents in the ordinary dataset are not related with spamming activities, we randomly select 500 questions with their 896 answers for manual verification. With the instruction of sampled deceptive Q&As (i.e., preliminary knowledge), assessors can better annotate Q&As. After annotation, five out of the 896 answers (less than 0.6%) are labeled by a group of three assessors as deceptive, and no question is labeled as deceptive. This result shows that the ordinary dataset can be treated as normal Q&As, and the negligible promotion ones in them have minimal impact on the experimental results.

3.2 Statistics of Collusive Activities

Table 1 shows the statistics of combined two datasets as described above. There are 66K users in collusive spamming dataset and 470K users in ordinary dataset. Besides, according to the number of crowdsourcing tasks (i.e. 2.6K promotion campaigns), we obtain the same number of $+QG$ and $+AG$. The number of the ordinary questions is nearly 10 times as many as the deceptive ones, which is reasonable because we believe that most contents on CQA are legitimate ones.

In Table 2, we provide detailed statistics with respect to the collusive spamming crowd-sourced workers. $CWer$ / Group means

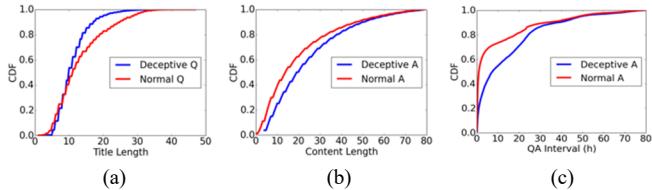


Figure 2: Comparisons of individual attribute distributions.

the average number of crowd worker (*CWer*) in a group (i.e. promotion campaign). As Content / *CWer* and User / *CWer* show, on average, each *CWer* creates about 90 CQA accounts and posts 55 questions in CQA. A crowd worker can create up to 11K CQA accounts to pollute the CQA platforms. This means that *CWers* cause severe damage to CQA and detecting spam accounts directly is not effective, because even if we block them, the *CWer* can continue registering new accounts. As *CWer* × Group shows, each *CWer* joins in about 10 tasks (promotion campaigns) on average. This implies that to gain more profits a *CWer* may post many spam contents to promote commercial targets in different tasks.

3.3 Individual Attributes Analysis

Based on the collected dataset, we can make comparative analysis on individual attributes between deceptive and ordinary Q&As. Table 3 depicts the comparisons between the proportions of deceptive and normal Q&As containing the corresponding attributes. As we can observe, fewer (about 18%) deceptive contents (+*Qs* and +*As*) are posted by the anonymous users than the normal ones (about 30%). Deceptive questions are more inclined to be responded by deceptive answers. In fact, more than half of +*Qs* have only one answer and most of them are solved (i.e., have best answer), while the corresponding percentages of -*Qs* are lower. Compared to the normal questions, fewer +*Qs* give awards for answers and fewer of them have descriptions (detailed information for the question title), but more of them are posted with tags. Compared to -*As*, more deceptive answers are alone (i.e., no other answers for the same question) and more of them are selected as the best answers. Besides, more +*As* are the first answer of their questions and almost no +*As* receive comments. A small fraction of +*As* are also posted by masters (high-level answerers identified by CQA platforms).

We hypothesize that the sentiment between deceptive contents and normal contents are different. To verify that, given a question or an answer, we calculate its scores distributions on 7 different sentiments such as “dislike”, “like” and “neutral”, the higher the score, the stronger the corresponding sentiment [21]. We present the sentiments that with the highest score (Top1) and the second-highest score (Top2) from each content. As table 4 shows, the largest proportion of Top1 sentiment is “like” in +*Q* (36%) and +*A* (50%), but in -*Q* and -*A*, they are “dislike” (23%) and “like” (29%) respectively. For Top2 sentiments, although all types of contents present “neutral” sentiment, the proportions show differences. These results show that most deceptive questions tend to disclose “like” sentiment other than “dislike”, and more +*As* prefer “like” sentiment than -*As*.

Figure 2 further shows the comparisons of individual-level attributes between deceptive contents and normal ones, in terms of title length, content length and QA time interval (time interval between a answer and its question). As Figure 2(a) indicates, about 50% of +*Qs*’ title length is in the range of 10 to 20, but 40% in -*Qs*. From Figure 2(b), we observe that deceptive answers’ length tends to shorter than the normal answers. Besides, +*As*’ QA time intervals are usually longer than that of -*As* (shown in Figure

Table 1: Statistics of the datasets. The prefix of plus sign (+) means deceptive, and the minus sign (-) means normal.

+User	-User	<i>CWer</i>	+ <i>Q</i>	- <i>Q</i>	+ <i>A</i>	- <i>A</i>	+ <i>QG</i>	+ <i>AG</i>
66K	470K	725	40K	397K	61K	689K	2.6K	2.6K

Table 2: Statistics about crowd worker.

	Max	Mean	Min
<i>CWer</i> / Group	19	2.7	2
Question / <i>CWer</i>	7,381	55.5	1
User / <i>CWer</i>	11,303	90.5	2
<i>CWer</i> × Group	650	9.8	1

Table 3: Comparisons of individual-level attributes.

Question Attributes	+ <i>Q</i>	- <i>Q</i>	Answer Attributes	+ <i>A</i>	- <i>A</i>
Anonymous	0.18	0.29	Anonymous	0.19	0.25
Only one answer	0.54	0.29	Alone	0.36	0.05
No answers	0.02	0.21	Master posted	0.12	0.35
Solved	0.91	0.32	Best	0.58	0.20
With tags	0.56	0.05	First posted	0.41	0.16
Has awards	0.03	0.29	Has comment	0.07	0.22
Has descriptions	0.08	0.41	–	–	–

Table 4: Statistics about the sentiment of Q&As.

	+ <i>Q</i>	- <i>Q</i>	+ <i>A</i>	- <i>A</i>
Top1	0.36 (like)	0.23 (disgust)	0.50 (like)	0.29 (like)
Top2	0.8 (neutral)	0.57 (neutral)	0.79 (neutral)	0.78 (neutral)

2(c)), due to the QA time interval request in task description.

From the above analysis, it is clear that the types and ranges of individual attributes between *Qs* and *As* are asymmetric (different). Besides, we can find that there are certain differences between deceptive contents and normal ones. Based on solely these individual attributes, we construct our combined factor graph model to detect deceptive questions and answers. However, this model does not perform well empirically (as shown in Table 9, Sec.6). This indicates that the individual-level attributes are not sufficient for collusive spamming activities detection, since each human-generated deceptive content is inherently distinct and collusive activities is more deceptive. Generally, collusive manipulated Q&As may share identifiable synergic attributes inevitably, such as the common theme (promotion campaign) and the same keywords. Therefore, capturing and inferring collusive (group-level) attributes might be important, which can be achieved by detecting group distributions in Q&As (Sec.4).

4. GROUP DETECTION AND ANALYSIS

To analyze the collusive spamming activities in CQA, the first major step is proposing an effective group detection method. In this section, we describe a group detection method (*GDM*) to facilitate clustering questions and answers respectively. Through detected groups, we want to analyze the collusive spamming activities and extract identifiable attributes from them. For simplicity, if more than half of the questions in a question group detected by *GDM* are deceptive, we consider it a detected deceptive question group (+*DQG*). If not, we regard it as a detected normal question group (-*DQG*). Similarly, the detected deceptive answer group (+*DAG*) and the detected normal answer group (-*DAG*) can be obtained.

4.1 Group Detection Method

As described above, both question and answer groups need to be detected, so we build two independent undirected graphs: question graph $G^Q = (V^Q, E^Q)$ and answer graph $G^A = (V^A, E^A)$, where V^Q is a set of $|V^Q| = M$ questions and V^A is a set of $|V^A| = N$ answers, E^Q and E^A are edge sets of question-question and answer-answer relationships. The major difficulty in constructing the graph is how to determine the edges between any two vertexes (i.e., two question in G^Q or two answers in G^A), because there are massive number of contents and any two of them are relative independent. To tackle this, we create a criterion for determining the edges, using question graph G_Q construction as an example, the neighbor set $N(Q_i)$ of question Q_i in G^Q is obtained as follows:

$$N(Q_i) = \left\{ Q_j \middle| \alpha \frac{V(Q_i) \cdot V(Q_j)}{\|V(Q_i)\| \cdot \|V(Q_j)\|} + \beta \frac{I(Q_i) + I(Q_j)}{L(Q_i) + L(Q_j)} > \theta \right\} \quad (1)$$

The Q_j belongs to a question set $W(Q_i)$ in which all the questions are posted in a time window after the post time of Q_i . The task (promotion campaign) in the crowdsourcing platforms has deadline for submissions (as shown in Figure 1). Because we want the deceptive contents of a task to be allocated into a group, the time restriction need to be take into account. If the time span between two questions is too long, then it is not necessary to calculate their relationship because they are less likely to belong to the same task. The time window conforms to the time restriction, and it simplifies the calculation because for a question Q_i , we only need to estimate the edges between Q_i and others in $W(Q_i)$.

To estimate the connection strength, we use the topic probabilities calculated by the topic model LDA [3] to represent the theme distributions of each question. After trying several parameter settings, we found that using 20 topics is a reasonable setting considering both efficiency and effectiveness. Therefore, each content is represented by a vector set in 20-dimensional space. The question topic vectors $V(Q_i)$ and $V(Q_j)$ represent the topic distributions of question Q_i and Q_j . Besides, to promote a commercial target, the deceptive questions in a task are very likely to contain the same words (the keywords). To catch this, we calculate the *tf-idf* value of each word in a question at first. Then, we loop through each word in Q_i one by one and obtain the common words which is also contained in Q_j . Finally, we sum the corresponding *tf-idf* values of the common words. $I(Q_i)$ and $I(Q_j)$ denotes the sum of all the common words' *tf-idf* values in Q_i and Q_j respectively. $L(Q_i)$ and $L(Q_j)$ is the length of Q_i and Q_j in terms of word count. The parameters α and β denote the weight of theme similarity and word similarity respectively and $\alpha + \beta = 1$.

Similarly, we can estimate the edges between answers and build answer graph G^A . After getting these two graphs, we use a high-quality smart local moving (SLM) algorithm for large-scale modularity-based community detection [26] to detect QG and AG respectively, which has been proved effective and efficient in a diverse set of graphs even for very large networks. A popular approach to community detection is based on the idea of optimizing a modularity function which is an NP-hard problem. Many different heuristic algorithms have been proposed for modularity optimization [32]. SLM algorithm relies on a well-known local moving heuristic in a more sophisticated way, and it therefore produces more accurate results.

Communities detected by the algorithm are clusters of closely connected nodes within a network. Please be noted that not all groups are deceptive ones since normal users may also ask similar questions simply due to common interests.

Table 5: Rand Index results of our group detection approach.

α	$+Q$	$+A$	$+A^\#$
0.8	0.959	0.957	0.951
0.6	0.970	0.961	0.964
0.4	0.981	0.973	0.970
0.2	0.978	0.989	0.963
0	0.972	0.976	0.956

Table 6: Statistics of detected group.

DQG	DAG	$+DQG$	$+DAG$
120K	91K	2.2K	1.6K

Table 7: Analysis about ground truth deceptive groups and detected deceptive groups.

	$+QG$			$+DQG$			$+AG$			$+DAG$		
	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min
User	316	11.5	1	131	12.4	1	397	20.2	1	466	22.7	1
Con	471	15.4	1	199	15.6	2	491	24.6	2	498	26.0	1
Time	92	3.2	1.5	110	4.5	1.1	104	3.4	0.1	153	4.6	0.9

Table 8: Group-level attributes comparisons.

	$+QG$	$+DQG$	$-DQG$	$+AG$	$+DAG$	$-DAG$
# $+C$ / Group	1.00	0.97	0.04	1.00	0.97	0.07
# $-C$ / Group	0.00	0.03	0.96	0.00	0.03	0.93
# $+U$ / Group	1.00	0.98	0.24	1.00	0.97	0.22
# $-U$ / Group	0.00	0.02	0.76	0.00	0.03	0.78
# $QAer$ / Group	0.008	0.006	0.15	0.007	0.003	0.10
# QA -Time	0.51	0.44	0.12	0.47	0.47	0.11
# Best / Group	0.91	0.89	0.33	0.66	0.70	0.25

4.2 Group Detection Performance

In the following section, we analyze the relative importance of the theme similarity and word similarity by corresponding results. We also evaluate another method for answer group detection, which does not need to construct G^A , but cluster answers according to their corresponding questions' group information.

By comparing the ground truth group distributions and deceptive contents' detected group distributions, we can quantitatively evaluate GDM . We use *Rand Index* (*RI*), a well-known metric for evaluating the quality of clustering when the ground truth is known, which has a value between 0 and 1, with 0 indicating that the two data clusters do not agree on any pair of points and 1 indicating that the data clusters are exactly the same. Due to the lack of normal contents' ground truth group information, we cannot evaluate GDM 's performance for detecting normal groups directly. If the proposed GDM perform well on deceptive group detection, then to a certain degree, it shows acceptable ability on CQA contents clustering.

We perform GDM many times by giving θ different values and find that the group detection performance is the best when $\theta = 0.48$. Table 5 presents the *RI* results of GDM with $\theta = 0.48$ and different α values. In general, our algorithm performs well ($RI > 0.9$). When α is 1, namely, only using theme similarity to estimate edges, the results are the worst. As α grows, the value of RI increases firstly and then decreases. When $\alpha = 0.4$ for $+Qs$ ' group detection and $\alpha = 0.2$ for $+As$ ' group detection, the detected group distributions of $+Qs$ and $+As$ are most consistent with the ground truth $+QG$ and $+AG$. This indicates that both theme and word similarity are useful for estimating the edges, and the weight of word similarity is higher. Overall, using GDM to obtain the answers' group

information is better than the method that follows question group detection results to cluster answers directly (as shown in $+A^\#$). This is because that, the question's length is usually shorter than answer's (as Figure 2(a) and 2(b) show), therefore, the similarity between questions are harder to be measured than answers. If a question is clustered to a wrong group, all its answers will also be wrongly allocated.

As the RI results show, we select $\alpha = 0.4$ (i.e., $\beta = 0.6$) and $\alpha = 0.2$ to facilitate question graph and answer graph constructing respectively. Performing GDM on the graphs, we obtain 120K detected question groups (DQG) and 91K answer groups (DAG) in total. If a group contains more deceptive contents than normal ones, we deem it as deceptive group. Through this way, we obtain 2.2K detected deceptive question group ($+DQG$) and 1.4K detected deceptive answer group ($+DAG$), which is shown in Table 6.

Table 7 presents the statistical analysis about the ground-truth deceptive groups and the detected deceptive groups. As it shows, each $+QG$ contains about 12 users on average, which is close to that of $+DQG$. The average number of users in $+AG$ and $+DAG$ are both about 20. The maximum of users in deceptive answer group is about 400. In $+QG$ and $+DQG$, the number of contents (Con) in per group is about 15, which is less than the corresponding number in $+AG$ and $+DAG$. There is no limit to the size of a group of GDM , therefore the minimum size (i.e., content count) is 1. The mean time spans ($Time$) of $+QG$ and $+AG$ are both 3 days. Therefore, in GDM mentioned above, we set the time window to be 3 days to conform the time restriction. In detected group $+DQG$ and $+DAG$, the mean time span is longer. This is because two contents with relatively long time interval may be connected through intermediate contents. The statistics information between ground truth groups and corresponding detected groups are similar, which further illustrates that the detected deceptive contents' group distributions are reasonable.

4.3 Group Attributes Analysis

As mentioned, it is important to extract distinctive attributes for identifying deceptive contents from CQA. We start with a comparison analysis on group-level attributes. In Table 8, we can see that in question and answer group comparisons, all the attributes between ground truth groups ($+QG$ and $+AG$) and detected deceptive groups ($+DQG$ and $+DAG$) are similar, which means that the attributes extracted from detected deceptive groups are consistent to the ground truth groups. On the other hand, the differences of group-level attributes between the detected normal groups ($-DQG$ and $-DAG$) and deceptive ones are significantly different. This implies that we can use the detected groups to extract identifiable attributes to help detecting deceptive contents.

If a CQA user posts any deceptive contents (Q&As), we deem it as a deceptive user ($+U$); on the other hand, it is a normal user ($-U$). Similarly, $+C$ denotes deceptive content and $-C$ is the opposite. Besides, if a user is related to two questions in a QG , that is, posts one and answers another one, we call it a $QAer$. And if a user gives an answer in a AG , meanwhile, posts another answer's question, it is a $QAer$ too. The QA -*Time* means a group's similarity degree on the time interval between a question and its answer. For Qs , the *Best* means solved questions (i.e., has the best answer), and for As , it denotes the best answers.

As we can see in Table 8, in deceptive groups, on average, the ratios of $+C$ and $+U$ are close to 1, and the corresponding ratios of $-C$ and $-U$ are near to 0. However, for normal question and answer groups, the corresponding ratios are completely reversed. Because all the contents in ground truth groups are deceptive, the positive user and content ratios are 1. The mean ratios of $QAer$ in

the normal groups are higher than the positive groups, to evade being detected, a $CWer$ avoids repeating an account in a task (i.e., rarely post both Q and A in a task).

Given a content (Q or A), we can obtain its QA time interval according the posted time of its corresponding A or Q. Therefore, each group can calculate a QA -*Time* by $2 \times t_{num} / (C_{num} \times (C_{num} - 1))$, where t_{num} represents the number of similar intervals that the difference between two contents' QA time intervals is less than 2 hours, and C_{num} is a group's size. The deceptive groups have higher QA -*Time* than the normal ones, because the deceptive contents in a group are organized and regular. As mentioned in Sec.3.2.1, most of $+Qs$ select a $+A$ as the best answer. Therefore deceptive groups have higher ratios of "Best" contents.

If a task's submissions are clustered into the same group by our proposed GDM algorithm, but the group also contains many normal contents, the identifiable attributes may not be extracted due to the mix of different types of contents. The comparison analysis of attributes in Table 8 shows that GDM can aggregate deceptive contents corresponding to the ground truth groups, and also separate normal ones from them to a large degree. As our statistics demonstrate, almost 79% $+Qs$ and 67% $+As$ are in $+DQG$ and $+DAG$, which means that most of deceptive contents are clustered together while few normal ones are included, i.e., deceptive contents and normal ones are separated. It is important to exploit them as crucial group attributes for detecting deceptive contents. As shown empirically in Sec.6, incorporating detected group attributes can dramatically improve the deceptive Q&A detection.

5. DECEPTIVE Q&A DETECTION

In this section, we propose a framework of deceptive Q&A detection, exploiting the individual (Sec.3.3) and the detected group (Sec.4.3) Q&A attributes. The target of our framework is to distinguish deceptive and normal Q&As, which means that we want to infer the label set Y^Q for V^Q and the label set Y^A for V^A . There are two options to build our model: 1) regarding G^Q and G^A as two independent graphs, and proposing two independent factor graph models FGM^Q and FGM^A for Y^Q and Y^A inferring respectively; 2) utilizing the naturally existing interactions between two graphs to integrate them as a unified graph $G = (V, E)$, where V represents all the Q&As and E is a set of E^Q, E^A and question-answer edges. Based on the integrated graph G , we can propose a combined factor graph model $CFGM$ to uniformly infer the entire label set Y for V . The combined model $CFGM$ is able to incorporate different attributes and correlations. For any models, we first sample a part of nodes as training set and the remaining as test set, then the corresponding model infers each of the remaining node's probability distributions of being deceptive or normal. Our goal is to train a supervised classification model.

5.1 Independent Factor Graph

Take FGM^Q for example, which only uses the attributes and correlations in graph G^Q . Figure 3 shows the graphical representation. The set of question nodes $V^Q = \{Q_1, Q_2, \dots, Q_M\}$ in G^Q is mapped to a factor node set $Y^Q = \{y_1^Q, y_2^Q, \dots, y_M^Q\}$ in question factor graph FG^Q . Using the known factor set in training set, FGM^Q infers whether an unknown node is spam or non-spam.

For each question's label node y_i^Q , we combine the individual-level attributes into attributes vector s_i^Q and g_i^Q for group-level attributes vector. In addition to attributes, we define the correlations (structural factor) in G^Q to bridge the factor node in FG^Q . Corresponding to the attributes and correlations, we define the following three factors:

- Individual-level attribute factor: $f_s(s_i^Q | y_i^Q)$ is the probability

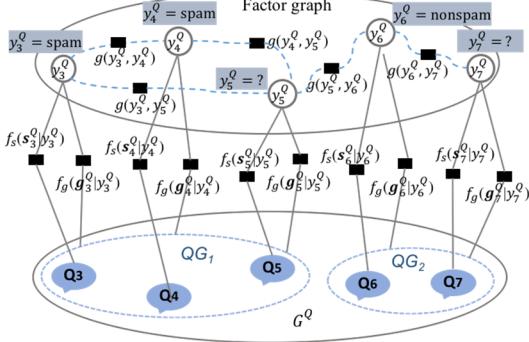


Figure 3: Graphical representation of the FGM^Q model.

of generating the individual-level attribute vector s_i^Q in G^Q given the label of factor node y_i^Q .

- Group-level attribute factor: given the factor node y_i^Q , $f_g(g_i^Q | y_i^Q)$ represents the probability of generating the group-level attributes that are extracted from the question group which the question Q_i belongs.
- Correlation factor: $g(y_i^Q, C(y_i^Q))$ denotes the correlations between Q s, where $C(y_i^Q)$ is the set of correlated Q s to y_i^Q .

For confirming the correlated questions set, we have two basic intuitions. First, any two questions in the same group may have a correlation. As analyzed in Sec.4.2, two questions in a group are likely have the same label. Second, the questions posted by the same user may have a correlation. The spammers and spam contents have strong connections, since the spammers tend to perform spamming activities [27]. Therefore, if a question Q_j in the same QG or posted by the same user with Q_i , we add Q_j into $C(y_i^Q)$.

Given question network G^Q , the objective of our model is to maximize the formation probability of the questions in the network, i.e., $P(Y^Q | G^Q)$ which is factorized as:

$$P(Y^Q | G^Q) = \prod_i f_s(s_i^Q | y_i^Q) \cdot f_g(g_i^Q | y_i^Q) \cdot g(y_i^Q, C(y_i^Q)) \quad (2)$$

Given answer graph G^A , the FGM^A model can also be represented like Figure 3. Similarly, we can define the joint distribution $P(Y^A | G^A)$ over the answer factor node set Y^A , which is also factorized into two types of attributes factors (individual-level and group-level) and correlation factors to bridge the answers.

5.2 Combined Factor Graph Model

Given the probability of $P(Y^Q | G^Q)$ and $P(Y^A | G^A)$, the conditional distribution over the combined graph G is factorized as:

$$\begin{aligned} P(Y | G) &\propto P(Y^Q | G^Q) \cdot P(Y^A | G^A) \\ &\propto \prod_i f_s(s_i | y_i) \cdot f_g(g_i | y_i) \cdot g(y_i, C(y_i)) \end{aligned} \quad (3)$$

where Y represents the set of factor nodes that mapped from all the Q&As in G . The attributes of questions and answers can both be divided into two levels: individual and group. However, the graph G^Q and G^A are asymmetric as analyzed in Sec.3, therefore the type and number of feature factors are different between the two graphs. In the combined factor graph model $CFGM$, all the factor nodes can be treated uniformly [6].

There exist many naturally relations between G^Q and G^A , which can be used by the combined model $CFGM$. Besides the corre-

tions in G^Q and G^A mentioned above, we consider other two correlations between QA based on the following intuitions. First, the deceptive questions are usually answered by deceptive answers as described in Sec.3. Second, spammers tend to post spam questions and answers. Therefore, if y_i represents a question Q_i in G , the answers of Q_i and other answers posted by the same user who submit Q_i will be added into $C(y_i)$.

5.3 Model Learning and Inference

The learning and inferring process of different models are the same due to they are all based on the factor graph model. In this section, we take $CFGM$ for example to describe how to learn and infer our model. The factors in Eq. (3) can be instantiated in different ways. In this work, we use exponential-linear functions. Thus, the attribute factors of $CFGM$ can be defined as

$$f_s(s_i | y_i) = \frac{1}{Z_\chi} \exp\{\chi^T \Phi(y_i, s_i)\} \quad (4)$$

$$f_g(g_i | y_i) = \frac{1}{Z_\delta} \exp\{\delta^T \Theta(y_i, g_i)\} \quad (5)$$

where χ and δ is a weighting vector, Φ and Θ is a vector of feature functions. Similarly, we define the correlation factor as

$$g(y_i, C(y_i)) = \frac{1}{Z_\lambda} \exp\left\{\sum_{y_j \in C(y_i)} \lambda^T \Psi(y_i, y_j)\right\} \quad (6)$$

where Ψ can be defined as a vector of indicator functions.

The parameters to be estimated are $\theta = (\chi, \delta, \lambda)$. We learn the parameters through maximizing the logarithm of the likelihood function $P(Y|G, \theta)$. For presentation simplicity, we concatenate all factor functions in Eqs. (4), (5), and (6) for a content node y_i as

$$\mathbf{h}(y_i) = (\Phi(y_i, s_i)^T, \Theta(y_i, g_i)^T, \sum_{y_j} \Psi(y_i, y_j)^T)^T \quad (7)$$

The joint probability defined in Eq. 3 can be rewritten as

$$\begin{aligned} P(Y | G) &= \frac{1}{Z} \prod_i \exp\{\theta^T \mathbf{h}(y_i)\} \\ &= \frac{1}{Z} \exp\left\{\theta^T \sum_i \mathbf{h}(y_i)\right\} = \frac{1}{Z} \exp\{\theta^T \mathbf{h}\} \end{aligned} \quad (8)$$

where $Z = Z_\chi Z_\delta Z_\lambda$ is a normalization factor, \mathbf{h} is the aggregation of factor functions over all Q&A nodes. Based on this equation, the log-likelihood objective function can be written as:

$$O(\theta) = \log \sum_{y_i \in Y} \exp\{\theta^T \mathbf{h}\} - \log Z \quad (9)$$

To solve the log-likelihood function, we adopt a gradient descent algorithm (or Newton-Raphson algorithm) [25]. The gradients for each θ are derived as:

$$\frac{\partial O(\theta)}{\partial \theta} = \mathbb{E}_{p_\theta(Y^L | G)} \mathbf{h} - \mathbb{E}_{p_\theta(Y^L | G)} \mathbf{h} \quad (10)$$

where the first term is the expectation of factor function \mathbf{h} given the known data distribution of the combined network (Y^L represents the sampled labeled Q&As), and the second term represents

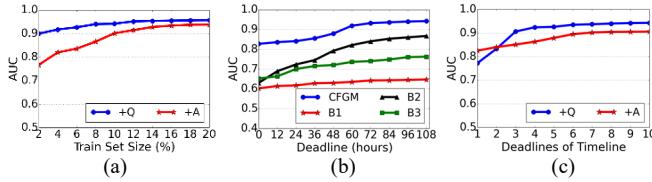


Figure 4: Evaluation of CFGM (AUC) with respect to different training set sizes, detection deadline and time periods.

the expectation of factor function under the distribution $P_\theta(Y|G)$ learned by the model.

It is intractable to directly calculate the marginal probability in the second term of Eq. (10). In this work, we use loopy belief propagation (LBP) [20] to approximate the gradients based on the following process. First, perform LBP to calculate corresponding marginal distributions. Second, update each parameter to maximize the objective function. The learning process performs the LBP algorithm twice in each iteration, one is for estimating the marginal distribution of unknown variables and the other for marginal distribution over all cliques [25]. Finally, each parameter is updated with the learning rate η :

$$\theta_{\text{new}} = \theta_{\text{old}} + \eta \cdot \frac{\partial(\theta)}{\theta} \quad (11)$$

After we obtain the learned parameters $\theta = (\chi, \delta, \lambda)$, we infer the factor node labels Y^T in test set. All the nodes in test set are assigned with labels that can maximize the marginal probabilities with the estimated parameters:

$$Y^* = \arg \max_{Y^T} P(Y|G) \quad (12)$$

We again utilize the LBP algorithm to compute the marginal probability of each content node $P(y_i|Y^T, G)$ and then predict the type of a node as the label with largest marginal probability. The marginal probability is then taken as the prediction confidence.

6. EXPERIMENTS

In this section, we present the experiments to evaluate the effectiveness of our proposed approach using our collected test set.

6.1 Experimental Setup

Seven types of group-level attributes and ten types of individual-level attributes are considered as the feature factors in our model. All of the attributes are described in Sec.3 and Sec.4. In addition, we categorize all the correlations mentioned above into two types: 1) content-based: two contents in the same group or having the QA relations (i.e., one answer responds another question) will be connected; 2) user-based: if two contents are posted by the same user, then they should be correlated.

We randomly select 10% of deceptive Q&As (i.e., only about 4K +Qs and 6K +As) as the training set for +Q and +A detection and the remaining as test set. In experiments, we evaluate the performance of deceptive question and answer detection respectively, rather than treating detected deceptive questions' corresponding answers as deceptive. We cannot regard +Qs' answers as positive directly. As our datasets show, not all the candidate deceptive answers are deceptive, because +Qs may attract normal users to answer them (as Sec.3.1 describes). By treating +Q and +A as two separate detection tasks, we can evaluate the proposed model comprehensively and also show its scalability.

Table 9: Comparisons between our methods and baselines.

Method	Question Detection				Method	Answer Detection			
	Pre	Rec	F-m	AUC		Pre	Rec	F-m	AUC
<i>CFGM-GS</i>	0.76	0.84	0.80	0.83	<i>CFGM-GS</i>	0.71	0.86	0.78	0.80
<i>CFGM-G</i>	0.76	0.86	0.81	0.85	<i>CFGM-G</i>	0.73	0.86	0.79	0.84
<i>CFGM-U</i>	0.81	0.90	0.85	0.91	<i>CFGM-U</i>	0.76	0.91	0.83	0.89
<i>FGM</i> ^Q	0.79	0.87	0.83	0.84	<i>FGM</i> ^A	0.74	0.86	0.80	0.85
<i>CFGM</i>	0.85	0.91	0.88	0.95	<i>CFGM</i>	0.78	0.92	0.84	0.90
<i>B1</i>	0.63	0.60	0.61	0.65	<i>B1</i>	0.68	0.65	0.67	0.69
<i>B2</i>	0.76	0.85	0.80	0.87	<i>B2</i>	0.74	0.75	0.74	0.81
<i>B3</i>	0.78	0.85	0.81	0.79	<i>B4</i>	0.70	0.71	0.70	0.73
–	–	–	–	–	<i>B5</i>	0.79	0.82	0.80	0.86

6.2 Baseline Approaches

We compare our proposed approach with the following methods for detecting deceptive contents in CQA:

Baseline1 (B1) [21]: We adapt the content-based features described in the approach that utilizes lexical patterns and part-of-speech patterns to effectively identify deceptive messages in the Microblogs environment by Bayes classifier. We think this approach can also be applied in the CQA platforms, due to the detected contents are both deceptive.

Baseline2 (B2) [15]: It proposes a propagation algorithm to diffuse promotion intents on an “answerer-channel” bipartite graph and detect possible spamming activities in CQA. Most of the promotion channels such as URLs, telephone numbers and social media accounts have been disabled in CQA. Therefore, we only take the idea of label propagation algorithm to build a “content-user” bipartite graph based on the common assumption that spam users tend to post spam contents [27, 17]. In the “content-user” bipartite, the questions and answer are treated uniformly and there is an unweighted edge between a content (question or answer) and its poster (asker or answerer). The sampled deceptive contents are used as the labeled seed to drive the algorithm.

Baseline3 (B3) [14]: To distinguish high-quality questions from low-quality ones, it uses the question-related and asker-related features to construct graphs and train the classifiers. Question-related features are extracted from question text including subject and content; asker-related features come from askers’ profiles. We deem the low-quality questions as deceptive questions.

Baseline4 (B4) [5]: To detect commercial answers in CQA, it applies logistic regression as the learning method by integrating semantic analysis, posters’ track records, and the special features of CQA websites.

Baseline5 (B5) [24]: It extracts several features from the questions, the answers, and the users who provided them to address the challenge of evaluating answer quality. Based on the extracted QA features, logistic regression model is used for predicting the quality of an answer. We hypothesize the extracted features can also facilitate the deceptive answer detection.

CFGM-G: Comparing to *CFGM*, it only removes the group-level attributes, which is constructed to illustrates the necessity of group detection.

CFGM-GS: It uses the proposed combined factor graph model, but the group-level attributes and individual sentiments attributes are not integrated in it. Through this method, we want to analyze whether there is difference between legitimate Q&As and deceptive ones.

CFGM-U: To show whether user-based correlations is useful for our model, the user-based correlations (mentioned in Sec.6.1) are not used in this approach compared to *CFGM*.

The approaches of *B1* and *B2* can either detect deceptive questions or deceptive answers. The *B3* are compared with the performance of our model's deceptive question detection, and *B4* and *B5* are used for deceptive answer detection. All the methods mentioned in this paper use the same training and test set.

6.3 Experimental Results

6.3.1 Classification Performance

Table 9 shows the performance of deceptive question and answer detection with different methods on four metrics: Precision (Pre), Recall (Rec), F-measure (F-m) and AUC.

As we can see, the *B1 baseline* achieves the worst performance, which means that using content-based features only is not effective in deceptive content detection. *Baseline B2* uses the mutual reinforcement-based relations between spammers and spam contents, and obtains relatively good results (better than *B1*) both on $+Q$ and $+A$ detection. This indicates that the assumption of “spam users tend to post spam contents” is reasonable. However, due to the insufficient information (only user-content relations are used), it cannot achieve better performance than the methods that with more representative attributes, such as *B3* and *B5*. Although *B4 baseline* aims at detecting deceptive answers in CQA (same with our goal), it does not perform well. This may be because it is unsuitable to detect the collusive spamming contents. Besides, the performance of *B3* and *B5* are not as well as our method. That implies that evaluating CQA content’s quality cannot be applied in deceptive content detection directly.

Besides, comparing *CFGM-GS* and *CFGM-G*, we can find that the individual sentiment attributes are helpful but the performance difference is marginal. As the comparisons of *CFGM-G* and *CFGM* demonstrate, the performance improvements given the exploitation of group-level attributes are noteworthy both on $+Q$ and $+A$ detection. By comparing *CFGM-U* and *CFGM*, we find that removing user-based correlations will decrease the performance to some extent. The combined model *CFGM* performs better than the independent models *FGM^Q* and *FGM^A*, because it integrates more sufficient correlations such as the QA relations and the relations between the *Qs* and *As* that are posted by the same user.

As the above results indicate, our independent models can effectively detect $+Q$ and $+A$ respectively, which means that the proposed model has strong applicability. To further evaluate the sensitivity of our framework to the training data, we vary the size of the training set from 2% deceptive contents to 20%, and track the corresponding classification results (AUC). As Figure 4(a) shows, not surprisingly, as the size of training set increases, the detection performance tends to rise in the beginning, but then stabilize at around 14%. This indicates that *CFGM* is effective in deceptive content detection and even only hundreds of training data can aid the algorithm to gain promising performance.

6.3.2 Early Detection

As mentioned, detecting deceptive contents at the very early phase is crucial to ensure the user experience of CQA. We conduct two types of early detection evaluation: 1) Early deceptive question detection, according to the average time span of QA in our dataset, we make ten deadlines in chronological order. Given a detection deadline, all the information after the deadline is invisible during the test stage [31]. We select the contents that in the 10% training set and before the deadline as the new training set, and all the others are treated as test set. 2) We aim to utilize the early known deceptive contents (Q&As) to detect the subsequent unknown deceptive contents ($+Qs$ and $+As$). In some cases, there are only some relatively old labeled data. We aim to validate whether

our method can predict new coming contents' labels based on the old training set. To do this, we set another ten deadlines according to the timeline of all our dataset. Each deadline has a digital marker, the smaller of the marker, the earlier of it. We regard the Q&As after a deadline as new unknown contents, and the others are old ones. Given a deadline, we select 10% of the contents that before it as the training set.

Figure 4(b) presents different methods' results of first type of early detection, the earlier of the deadline, the less information are used. It exhibits that with deadline delaying, the performances on $+Q$ detection gets better, even at the earliest deadline the AUC value is acceptable (e.g. AUC of 0.8 within 12 hours). Besides, our method outperforms other baselines at any early stage. Figure 4(c) shows our method's performance on the second type of early detection. As it exhibits, in the first half of the figure, with the deadline time node growing, the AUC of both $+Q$ and $+A$ detection increases in a fast manner. In the other half, the growth slows a bit and tends to stabilize. Our method also performs better than other baselines whose results have been omitted; due to the lack of space, the small figure cannot exhibits them well. All these results demonstrate that our proposed approach can achieve early detection effectively.

7. CONCLUSIONS

In this paper, we study the problem of the crowdsourcing manipulated content (i.e., collusive deceptive content) detection. To tackle this problem, we define the group in CQA platforms according to the crowdsourcing tasks (promotion campaigns). The question and answer graphs are built respectively according to contents' theme similarity and word similarity. Based on the two graphs, we detect question and answer groups respectively, and find that our proposed group detection method can effectively detect groups and extract corresponding group attributes.

Given various extracted attributes (individual-level and group-level) and correlations (content-based and user-based), we propose a combined factor graph model (*CFGM*) to learn to infer whether a question or an answer is deceptive. An efficient algorithm is proposed to learn model parameters and to infer the labels of unknown contents. Experimental results on a real-world dataset validate the effectiveness of the proposed model. The *CFGM* can achieve reasonable performance of detecting deceptive contents, even with very small size of training set. Besides, the proposed model performs effectively on two levels of early detection, which can inhibit the broadcast of deceptive information timely.

Detecting the collusive deceptive contents facilitates CQA to be more credible and effective, and represents a new research direction in CQA spam content detection. As future work, it is interesting to study how to define another concept of collusive groups and study the collusive behaviors on the other level. Besides, it is also interesting to apply our model on the other platforms such as Microblogs and online review websites, which have also been polluted by the malicious crowd workers.

8. ACKNOWLEDGMENTS

This work was supported by Natural Science Foundation (61622208, 61532011) and National Key Basic Research Program (2015CB358700) of China.

9. REFERENCE

- [1] Agichtein, E. Castillo, C. Donato, D. Gionis, A. and Mishne, G. 2008. Finding high-quality content in social media. *International Conference on Web Search and Web Data Mining*, (WSDM'08). ACM, 183-194.
- [2] Baltadzhieva, A. 2015. Question quality in community ques-

- tion answering forums. *AcM Sigkdd Explorations Newsletter*, 17(1), 8-13.
- [3] Blei, D. M. Ng, A. Y. and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 993-1022.
- [4] Cao, C. Caverlee, J. Lee, K. Ge, H. and Chung, J. 2015. Organic or Organized?: Exploring URL Sharing Behavior. *ACM International on Conference on Information and Knowledge Management*. ACM.
- [5] Chen, C. Wu, K. Srinivasan, V. and Kesav, B. R. 2013. The best answers? think twice: online detection of commercial campaigns in the cqa forums. *Ieee/acm International Conference on Advances in Social Networks Analysis and Mining*. 458-465.
- [6] Dong, Y. Zhang, J. Tang, J. Chawla, N. V. and Wang, B. 2015. CoupledLP: Link Prediction in Coupled Networks. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 199-208. ACM.
- [7] Fayazi, A. Lee, K. Caverlee, J. and Squicciarini, A. 2015. Uncovering Crowdsourced Manipulation of Online Reviews. *The, International ACM SIGIR Conference*, 233-242.
- [8] Feng, S. Xing, L. Gogar, A., and Choi, Y. Distributional footprints of deceptive product reviews. In *AAAI*, 2012, 98-105.
- [9] Jindal, N. and Liu, B. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining* (WSDM '08). ACM, 2008, 219-230.
- [10] Kim, S. Chang, H. Lee, S. Yu, M. and Kang, J. 2015. Deep Semantic Frame-based Deceptive Opinion Spam Analysis. *The, ACM International on Conference on Information and Knowledge Management*, 1131-1140.
- [11] Lee, K., Caverlee, J., Cheng, Z., and Sui, D. Z. Campaign extraction from social media. *ACM Trans. Intell. Syst. Technol.*, 5(1):9:1–9:28, Jan. 2014.
- [12] Lee, K., Eo, B. D., and J. Caverlee. Seven months with the devils: A long-term study of content polluters on twitter. In *ICWSM*, 2011.
- [13] Lee, K., Caverlee, J., Cheng, Z., and Sui, D. Z. 2011. Content-driven detection of campaigns in social media. *ACM Conference on Information and Knowledge Management* (CIKM'11). ACM, 551-556.
- [14] Li, B. Jin, T. Lyu, M. R. King, I. and Mak, B. 2012. Analyzing and predicting question quality in community question answering services. *International Conference Companion on World Wide Web*. ACM, 775-782.
- [15] Li, X. Liu, Y. Zhang, M. Ma, S. Zhu, X. and Sun, J. 2015. Detecting Promotion Campaigns in Community Question Answering. In *IJCAI*, 2348-2354.
- [16] Lim, E.-P., Nguyen, V.-A., Jindal, N., Liu, B. , and H. W. Lauw. 2010. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (CIKM '10). ACM, 938-948.
- [17] Lu, Y. Zhang, L., Xiao, Y. and Li, Y. 2013. Simultaneously detecting fake reviews and review spammers using factor graph model. *ACM Web Science Conference*, 225-233.
- [18] Mukherjee, A., Kumar, A. , Liu, B. , Wang, J. , Hsu, M. , Castellanos, M. , and Ghosh, R. Spotting opinion spammers using behavioral footprints. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (KDD '13). ACM, 632-640.
- [19] Mukherjee, A., Liu, B., & Glance, N. 2012. Spotting fake reviewer groups in consumer reviews. *International Conference on World Wide Web*. ACM, 191-200.
- [20] Murphy, K. P., Weiss, Y., and Jordan, M. I. Loopy belief propagation for approximate inference: An empirical study. In *UAI'99*, pages 467–475,1999.
- [21] Qazvinian, V. Rosengren, E. Radev, D. R. and Mei, Q. 2011. Rumor has it: identifying misinformation in microblogs. *Conference on Empirical Methods in Natural Language Processing* (EMNLP'11). 1589-1599.
- [22] Ott, M., Choi, Y., Cardie, C., and J. T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *HLT*, 2011.
- [23] Sandulescu, V. and Ester, M. 2015. Detecting Singleton Review Spammers Using Semantic Similarity. In *Proceedings of the 24th International Conference on World Wide Web* (WWW '15 Companion). ACM, 971-976
- [24] Shah, C. and Pomerantz, J. 2010. Evaluating and predicting answer quality in community QA. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '10). ACM, 411-418.
- [25] Tang, W. Zhuang, H. and Tang, J. 2011. Learning to Infer Social Ties in Large Networks. *Machine Learning and Knowledge Discovery in Databases*, 381-397.
- [26] Waltman, L. Eck, N. J. V. 2013. A smart local moving algorithm for large-scale modularity-based community detection. *European Physical Journal B*, 86(11), 1-14.
- [27] Wu, F. Shu, J. Huang, Y. and Yuan, Z. 2015. Social Spammer and Spam Message Co-Detection in Microblogging with Social Context Regularization. *ACM International on Conference on Information and Knowledge Management*, 1601-1610. ACM.
- [28] Xu, C. Zhang, J. Chang, K. and Long, C. 2013. Uncovering collusive spammers in Chinese review websites. The, *ACM International Conference on Information and Knowledge Management*, 979-988.
- [29] Xu, C. 2013. Detecting collusive spammers in online review communities. In *Proceedings of the sixth workshop on Ph. D. students in information and knowledge management*. ACM. 33-40.
- [30] Zhang, K. Wu, W. Wu, H. Li, Z. and Zhou, M. 2014. Question Retrieval with High Quality Answers in Community Question Answering. *ACM International Conference*, 1-6.
- [31] Ma, J. Gao, W. Mitra, P. Kwon, S. Jansen, B. J. Wong, K. F. & Cha, M. 2016. Detecting Rumors from Microblogs with Recurrent Neural Networks. In *IJCAI*.
- [32] Fortunato, S. 2010. Community detection in graphs. *Physics Reports*, 486(3–5), 75– 174.