Detecting Promotion Campaigns in Query Auto Completion

Yuli Liu†, Yiqun Liu†, Ke Zhou‡, Min Zhang†, Shaoping Ma†, Yin Yue#, Hengliang Luo#

†Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science & Technology, Tsinghua University, Beijing, China

‡Yahoo! Research, London, U.K.

#Samsung R&D Institute China - Beijing

yiqunliu@tsinghua.edu.cn

ABSTRACT

Query Auto Completion (QAC) aims to provide possible suggestions to Web search users from the moment they start entering a query, which is thought to reduce their physical and cognitive efforts in query formulation. However, the QAC has been misused by malicious users, being transformed into a new form of promotion campaign. These malicious users attack the search engines to replace legitimate auto-completion candidate suggestions with manipulated contents. Through this way, they provide a new malicious advertising service to promote their customers' products or services in QAC. To our best knowledge, we are among the first to investigate this new type of Promotion Campaign in QAC (PCQ). Firstly, we look into the causes of PCQ based on practical commercial search query logs. We found that various queries containing certain promotion intents are submitted multiple times to search engines to promote their rankings in QAC. Secondly, an effective promotion query detection framework is proposed by promotion intent propagation on query-user bipartite graph, which takes into account the behavioral characteristics of promotion campaigns. Finally, we extend the query detection framework to promotion target detection to identify the consistent promotion target which is the inherent goal of the promotion campaign. Large-scale manual annotations on practical data set convey both the effectiveness of our proposed algorithm, and an in-depth understanding of PCQ.

Keywords

Promotion campaign; Query auto completion; Spam Detection

1. INTRODUCTION

As a popular feature of search engines, Query Auto Completion (QAC) suggests possible completions of the partial queries submitted by users [1]. QAC exists to help users formulate more effective queries in less time and with less effort of interacting with the search engine, leading to a more enjoyable user experience [2, 7]. Currently, most QAC processes do not adopt personalization techniques [6] (e.g. when the user profile is not available), which means that for a given query prefix, many users are presented with the same set of suggestions. Therefore, to promote certain products or services, malicious users organize promotion campaigns via manipulating QAC services. The visibility of the promotion campaigns in QAC will thus increase, due to the fact that search users are inevitably see the QAC suggestions when typing a specific query even the suggestions have been manipulated by spammers.

For almost all major search engines, QAC candidates are generated and ranked based on users' query logs [3, 4, 5]. The malicious users can therefore deceive search engines through submitting multiple fake query streams to manipulate the QAC rankings and promote certain targets in suggestions. Considering the large number of users who are exposed to QAC and the fact that many users may be misguided to the spam Website if she clicks the promotion suggestions, Promotion Campaign in QAC (PCQ) becomes a critical spamming activity in malicious search engine marketing services.

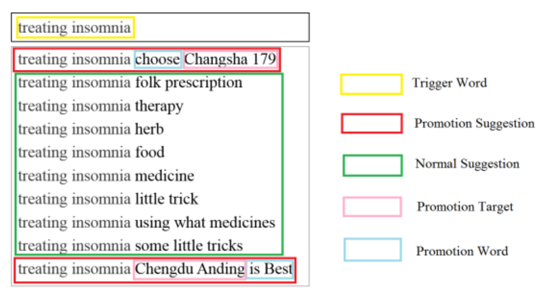


Figure 1: A translated example of promotion campaign in query auto-completion (QAC) of a commercial search engine.

Figure 1 shows a real-world example of PCQ in which the users are presented with several malicious auto-completion candidates after issuing the query prefix "treating insomnia" (治疗失眠). The prefixes themselves have no promotion intention, but certain prefixes (denoted as *trigger word*, such as "treating insomnia") can trigger *promotion suggestions*. There are two promotion suggestions that are triggered by the same trigger word with different *promotion targets* (i.e. product or service name) in this suggestion list, each of which consists of two parts: prefix and auto completion suffix. For both cases, the promotion targets are shown in the suffix parts of the QAC suggestions. The *promotion words* "choose" (到), "is best" (资深) in the suffix are used to express apparent promotion intention, and the promotion targets "Changsha 179" (长沙 179) and "Chengdu Anding" (成都安定) are two private practices in the cities of Changsha and Chengdu in China, respectively, who want to promote their treatment plans. These private practices are illegal ones and are not permitted to promote through legitimate ways such as sponsored search. Therefore, they turn to the malicious promotion markets and choose PCQ to promote their products or services.

From this example we can see that PCQ aims to generate promotion query suggestions and rank them to top positions in the QAC lists. For most cases, promotion suggestions do not help the users to access their desirable information and therefore are obstacles in users' search processes. Through analysis into query

CIKM'16, October 24–28, 2016, Indianapolis, IN, USA.
© 2016 ACM. ISBN 978-1-4503-4497-5/16/09...$15.00.
DOI: http://dx.doi.org/10.1145/XXXXXXX.XXXXXXX

behavior logs, we found that by submitting multiple promotion queries (manipulated queries that contain promotion intents), promoters/spammers can deceive search engine to recommend suggestions that contain promotion intents, thus facilitating promotion campaigns. They provide improper information to the users, make QAC less effective, and can be detrimental to the credibility of the Web search engines [28].

In our collected promotion data set (described in Sec.4.3.3), about 30% of the promoted suggestions are ranked within top 3 positions in their corresponding QAC rankings. This means that users' search experience would significantly be affected as they have high chances to examine and click on those highly ranked suggestions [2]. Considering the fact that many crowd sourcing and E-commerce Web sites such as Tuibaila[1] and Xianglong[2] are providing PCQ services on almost all major Chinese search engines, it is important to detect these activities and avoid their damage to search experiences. However, to the best of our knowledge, few attentions have been paid to the PCQ phenomena in existing works.

According to our study into the QAC spamming activities, we found that Chinese search engines (such as Baidu, Sogou, etc.) are much more affected by PCQ than English search engines. It is probably due to the fact that the platforms that provide PCQ services are mainly in the Chinese Web. However, we also noticed a number of existing studies [19, 25] that show efforts in manipulating QAC results of English search engines such as Google. The mechanism in these efforts are quite similar with the PCQs in the Chinese Web, which includes submitting large amounts of queries to search engines with the help of crowd-sourcing workers (mainly recruited via MTurk). It means that PCQ problem may also happen to English search engines if spammers decide to use the manipulation of QAC to achieve promotion goals. Therefore, we believe that the promotion campaign in QAC is an urgent problem that should raise researchers' awareness and need to be addressed.

To shed light on the PCQ detection problem and help search engines better serve users' information needs, we presents the first study on promotion campaign detection in QAC. The main contributions of this paper are three-folds:

- We make a comprehensive analysis in PCQ, which is a newly observed phenomenon of promotion campaign in QAC through both market side and search log side analysis.
- We propose a novel framework for promotion query detection from query logs, based on propagating the promotion intents on a query-user bipartite graph. A large number of manual annotations are also collected to verify its effectiveness and enhance our understanding of this phenomenon as well.
- To detect promotion campaigns more effectively, we extend the framework to identify promotion targets, which are the consistent and inherent goal of promotion campaign. Based on this extended framework, two countermeasures are proposed: advance precaution and real-time identification.

2. QAC PROMOTION CAMPAIGNS

In this section, we focus on analyzing in details the promotion campaigns in QAC (i.e. PCQ). An overview of PCQ is described in Sec.2.1, followed by details of the data and annotations in Sec.2.2, and quantitative data analysis in Sec.2.3.

[1] http://www.zhongxincompany.com/

[2] https://shop107660749.taobao.com/index.htm

2.1 Overview

Since most major search engines rely on user behavior logs to generate candidate queries for both query suggestions and QAC services [3, 4, 27], spammers thus generally issue a lot of promotional queries to spam the query logs so as to manipulate the suggested recommendations.

According to our investigation into QAC promotion campaigns, we found that promotion activities are usually performed according to the following procedure (see Fig.2). At first, the customer of PCQ usually design one or more trigger words (e.g. "treating insomnia" or "insomnia") and submit them to the spammers. With the goal of having promotion suggestions (e.g. "treating insomnia, choose Changsha 179") appear in the corresponding QAC lists when search users input the trigger words in search boxes, the spammers use each trigger word and corresponding promotion target (e.g. "Changsha 179") to generate a number of promotion suggestions (queries). Usually, dozens of queries are designed to promote one target (as shown in Sec.2.3). Since the queries are repeatedly submitted by multiple spammers to search engines to pollute the user behavior logs, QAC generation algorithms of search engines would consider the huge amount of promotion queries as a good source of evidence to generate suggestions. Promotion suggestions thus will be presented in the QAC ranking lists while users input the trigger words in the search box.

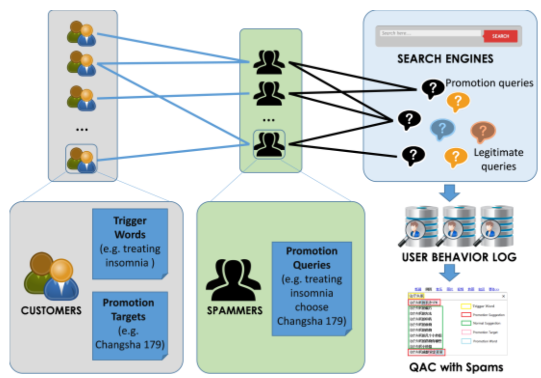


Figure 2: Promotion campaign in QAC (PCQ).

From the above procedure we can see that these spamming activities can be stopped in at least two different phases in the generation of QAC ranking lists. In the early phase, once the search engine notices that a large number of queries are submitted which contain promotion intents, they should be able to detect these promotion query streams and stop considering them while generating query suggestions (i.e. advance precaution). In the late phase, before the QAC lists are shown to users, a filtering step can be performed to ensure that no promotion suggestions are provided (i.e. real-time identification). For both phases, it requires the detection of query words with promotion intents to stop the spamming activities. Before we propose algorithms for detecting PCQ (Sec.3), we first study its characteristics by exploiting query logs and manual annotations.

2.2 Data

2.2.1 Query Logs

We choose the query log dataset from a popular commercial search engine for quantitative analysis of PCQ, and later modeling and evaluation. The query logs used in this work have the following attributes. **Query:** the phrase that a user searched. **ID:** cookie id that can be adopted to identify different users. **Query**

results: a set of most relevant results (Web pages) returned by the search engines when the query was issued. **Click:** An indicator for each query result showing whether it was clicked. **Time:** the timestamp when the query or the click event happened.

To effectively analyze the phenomenon of PCQ, we select experimental dataset from the entire query logs (terabytes) that covers the period from May 18 to July 5 in the year of 2015, which lasts for 7 weeks using the following strategy. Firstly, we select 27 successful PCQ cases from hundreds of relevant promotion markets that provide PCQ service, which are used by the market operators to show the effectiveness of PCQ. We also manually verified that these 27 promotion cases still trigger promotion suggestions in QAC in the commercial search engine on May 25, 2015. Secondly, we select a set of filter words consisting of either trigger words ("treating insomnia") or promotion targets ("Changsha 179"). Thirdly, we keep only those query log entries that contain any one of the filter words in their queries (e.g. "treating insomnia medicines", "treating insomnia to Changsha 179" or "Changsha 179 hospital"). And finally, we eliminate the queries that are submitted less than 10 times in the time period of 7 weeks. Through this method, the created data set contains both normal (non-promotion) and promotion queries regarding the content topics that may be polluted by spammers.

To investigate the dynamic nature of PCQ, we divide the seven weeks' filtered dataset into 7 independent datasets (denoted as D_1, D_2, D_3, ..., D_7) on a weekly basis[3]. For example, the dataset D_1 covers the period from May 18 to May 24, consists of about 1.2 million query entries, with around 73.2% containing only trigger word, 14.6% containing only promotion target, and the remaining 12.2% containing both. After constructing those weekly datasets, we perform a data annotation process to investigate the PCQ phenomena. Figure 3 illustrates the data collection and annotation process in which we constructed 7 filtered query log datasets (D_1, D_2, ... D_7) and 7 annotated query sets (denoted as Q_1, Q_2, Q_3, ..., Q_7) (described in the following section).

Apart from the above filtered datasets and annotated query sets, to evaluate the effectiveness of our framework on unfiltered real-world data, we also select 3 days' (July 3 to July 5) entire query logs (without any pre-processing) as the test data, which is denoted as D_{test} that contains over 27 million query entries. The datasets used in this paper are publicly available[4].

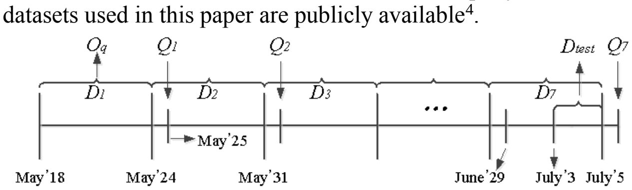


Figure 3: Chronology of the data collection and annotation.

2.2.2 Annotation

For the constructed query log datasets, we ask three annotators to annotate PCQ activities in them. Since annotating all the queries in the dataset is not feasible, we select the top 3000 (unique) potential promotion queries as the observed set O_q identified by the proposed PCQ detection algorithm (described in Sec.3) that performs on D_1. Then we annotate O_q on Monday, May 25, 2015, as our first annotated query set Q_1. After that, we manually annotate the same set of the 3000 queries O_q using the same annotation procedure at a fixed time each week (i.e. on Monday morning) for different time periods to obtain Q_2, Q_3,...,Q_7.

Table 1: Annotation of PCQ activities in the query logs

Annotation Task	Example	
Candidate query	治疗失眠长沙 179 最好 (treating insomnia, Changsha 179 is best)	治疗失眠的药物 (treating insomnia medicines)
(a) Is the query with promotion intent?	Yes	No
(b) Extract promotion target if "Yes" to (a)	Changsha 179	N/A
(c) Does the prefix of the query trigger promotion suggestions in QAC	Yes	Yes
(d) Label trigger word if "Yes" to (c)	治疗失眠(treating insomnia)	治疗失眠 (treating insomnia)
(e) Record number of characters of (d)	4 characters	4 characters
(f) Record promotion suggestions if any	治疗失眠到成都安定 (treating insomnia, choose Chengdu Anding)	
(g) Extract promotion targets from (f)	成都安定(Chengdu Anding)	

The annotation process (with two examples shown in Table 1) is as follows. Firstly, the annotator is instructed to determine whether a given candidate query has promotion intent about a target (a product or a service). If so, he/she should label the candidate query as *promotion query* (Table 1a) and extract corresponding promotion target entity from it (Table 1b). Otherwise, the query is labeled as a normal (non-promotion) query (Table 1a, b). After that, the annotator is asked to type the query literally to the search engine from which we collect query logs data, to check whether there are QAC suggestions recommended by the search engine when certain characters are typed. If certain promotion suggestions (e.g. "treating insomnia, choose Chengdu Anding") appear after typing certain characters (i.e. trigger words, e.g. "treating insomnia") in QAC, three actions will be taken. Firstly, the annotator labels the query candidate itself (e.g. "treating insomnia medicines") as *trigger query* (Table 1c), i.e. the query's prefix contains trigger word that can trigger promotion suggestions. Secondly, the annotator records the corresponding trigger word (Table 1d) and its length (in Chinese character) (Table 1e). Thirdly, the annotator records all the unique QAC promotion suggestions (Table 1f) and extracts promotion targets from those promotion suggestions (Table 1g). Otherwise if no promotion suggestions are displayed, the annotator continues to enter the next character to further check if it can trigger.

To summarize, at the end of the annotation process, for each candidate query, we obtain the annotated query set Q_i consists of seven labeled fields shown in Table 1. As we show as an example previously, the candidate query ("treating insomnia medicines") in query logs can be annotated as trigger query even if it is firstly annotated as normal (non-promotion). Besides, different queries with the same trigger word as prefix will trigger the same promotion suggestions (as the examples show).

After the annotation process, we assess the agreement among the annotators to see whether the annotation results are reliable. Figure 4(a) shows the agreement in the labeling of promotion query (whether the query is with promotion intent), trigger query (query prefix triggers the PCQ) and the length of the trigger word. We can observe that the Cohen's Kappa coefficient of labeling

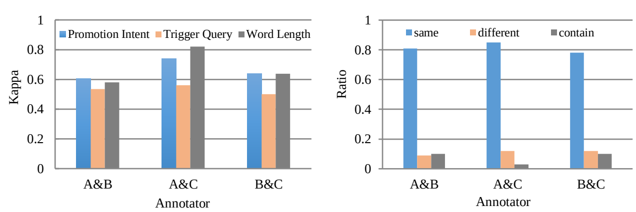
[3] The choice to create weekly datasets is because it takes about 7 days to achieve a new promotion campaign according to the advertising service market research. This enables us to investigate the dynamism of promotion campaigns.

[4] http://www.thuir.cn/group/~yqliu/

promotion intent between annotators A, B and C are relatively high (good agreement according to [26]), which means that the promotion queries are easy to be detected manually. Comparatively, the annotator agreement of trigger query is slightly below that of promotion intent, with moderate agreement (between 0.4 and 0.6). It implies that typing the query verbatim and inspecting the promotion suggestions are relatively more difficult tasks. Characters typed successionally or rapid skimming in the input process can both cause erroneous judgments. However, if a query is labeled as a trigger query, the same trigger word (same length) is very likely to be recorded by different annotators, which is indicated by good Kappa agreement of the trigger word length. It also means that the promotion suggestions will be displayed when specific trigger word is entered.

Figure 4(b) shows the annotator agreement in identifying the promotion targets from the promotion queries (Table 1a, b). The legend “same” means that two annotators identify the same target from a query, “contain” represents a target t_1 identified by an annotator is longer than another’s target t_2, and t_1 contains all terms in t_2. If two annotators identify two different targets (neither “same” nor “contain”) from a query or only one of them labels the query as promotion query, we denote the labeling results of the two annotators as “different”. As we can see, most of the targets identified by different annotators are the same, and the ratios of “contain” and “different” are relatively low (below 20%).

We use majority voting to merge the results of all the annotations to obtain the final annotation labels. Namely, for example, if two or all of the annotators label a query as promotion query, we regard it as a promotion query. However, to determine the promotion targets, if all three annotators disagree with each other, we conduct an additional annotation by another annotator to select the target with the minimum number of characters while being able to represent the name of a product or a service clearly.



(a) Agreement in query (b) Agreement in target

Figure 4: Annotator agreement in QAC promotion detection.

2.3 Analysis of PCQ Activities

The properties of promotion query and suggestion, and temporal dynamics of PCQ are analyzed respectively in this section.

2.3.1 PCQ Characteristics

We analyze the characteristics of the annotated promotion campaign queries. The detailed statistics are presented in Table 2. For Q_1, 1235 out of 3000 candidate queries (41%) are labeled as promotion queries while the remaining are labeled as normal (non-promotion) ones. In addition, 1440 out of those 3000 queries (48%) are labeled as trigger queries (i.e. query prefix triggers promotion suggestions) while the rest does not trigger any promotion suggestions in labeling process. One interesting observation is that 285 out of the 1765 normal queries are labeled as trigger queries (e.g. “treating insomnia medicines”). In other words, about 16% of the normal queries in our dataset have been contaminated with the promotion campaign while in the process of typing the query which starts with “treating insomnia”, without the promotion intention, the search users will be expected to see the promotion campaigns (e.g. “treating insomnia, choose

Table 2: Statistics of promotion campaigns in Q_1

Item	Statistics
# Promotion / Normal Queries	1235 (41%) / 1765 (59%)
# Trigger / Non-trigger Queries	1440 (48%) / 1560 (52%)
# Normal & Trigger Queries	285 (16% of Normal Queries)
# Promotion & Non-Trigger Queries	73 (6% of Promotion Queries)
# Promotion & Trigger Queries	1162 (94% of Promotion Queries)
# Queries per Target	30.6
# Suggestions per Target	39.4
# Users per Promotion Query	62.1
# Promotion Queries per User	3.6

Changsha 179”) in QAC suggestions. Note that there are only a small number of promotion queries (73) found to be non-trigger queries, which means that a majority of promotion queries are related with QAC events.

We can also observe that each promotion target is promoted by about 30 promotion queries (Table 1b) on average while the average number of promotion suggestions that displays for the same promotion target (Table 1g) is also high (39.4). This is consistent with what we show in Sec. 2.1, i.e. to increase the visibility of a promotion target, a promotion customer usually designs multiple trigger words, with each trigger word corresponding to a few promotion suggestions/queries. Therefore dozens of such queries are designed to promote one target.

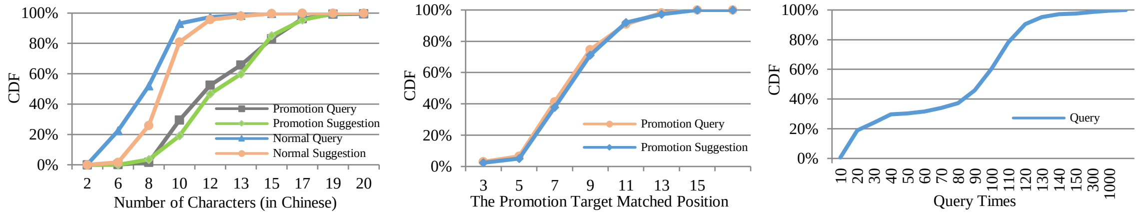
Finally, we found that a user submits 3.6 unique promotion queries on average. The PCQ is provided by the spamming advertising market as a service to customers. Therefore, to serve multiple customers, a promoter needs to submit multiple batches of queries with different targets to promote the campaigns. We also found that, to avoid being noticed by the search engine and increase the popularity of the promotion query, a promotion query is executed on average by 62.1 users (promoters/spammers).

Figure 5 shows the length distributions of the trigger words and the corresponding promotion suggestions. We can see that most of the trigger words (56.7%) have four characters (in Chinese). Besides, 2716 promotion suggestions are recommended for those four-word trigger words, which account for almost 60% of the promotion suggestions. As Figure 5 shows, each trigger word corresponds to around three promotion suggestions on average.

The cumulative distribution functions (CDF) of the three properties of PCQ (i.e. character length, target entity position, and promotion query frequency) are shown in Figure 6. Figure 6(a) presents the differences in lengths of promotion and normal queries / suggestions. We can see that the promotion queries and suggestions have more characters than normal ones. Besides, few of the promotion suggestions and queries have less than 6 characters. As Figure 6(b) shows, almost 95% of the promotion targets appear after the 5th character of the string of the promotion queries or promotion suggestions. Moreover, the distributions of promotion queries and suggestions are in good agreement in both Figure 6(a) and Figure 6(b), which indicates that promotion queries help to display promotion suggestions.

Figure 6(c) illustrates that promotion queries are mainly with intermediate query frequencies (i.e. 80-110 in our dataset). The promoters would not submit promotion queries too frequently to avoid being conspicuous while the appropriate amount of query frequencies can achieve promotion campaigns in QAC.

In addition, we conduct comparative analysis of the user behavioral characteristics between the normal and promotion queries. Figure 7 shows the comparison of the click ratio between the normal and promotion query behaviors. For a given query, the click ratio is calculated as the fraction of query entries with at



(a) Length of query / suggestion. (b) Character index of promotion targets. (c) Query times of promotion query

Figure 6: Cumulative distributions of promotion suggestions in length, character index position, and frequencies.

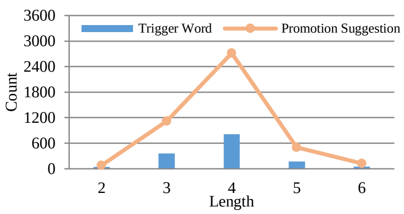


Figure 5: Distribution of promotion suggestions in Q_1 and trigger words of different length.

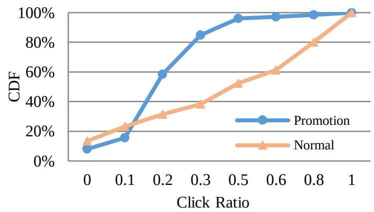


Figure 7: Comparison of promotion and normal suggestions on query click ratio behavior.

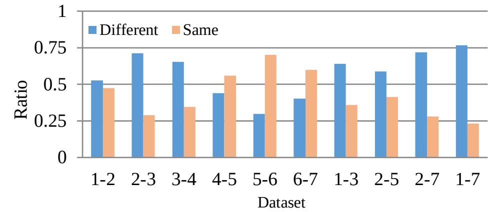


Figure 8: Dynamic changes of promotion suggestions.

least one result click to all entries. We can observe that, compared to the normal query behaviors, promotion queries generally lead to lower click ratios, which means that fewer search results are clicked for the queries with promotion intents.

2.3.2 PCQ Temporal Dynamics

In this section, we analyze the temporal characteristics of PCQ. As Figure 8 shows, in two comparing annotated sets Q_1 and Q_2 shown as "1-2", for a given promotion target, we compare the promotion suggestions of Q_1 and Q_2, which promote the target. Finally, we find that more than 50% of the promotion suggestions in Q_1 and Q_2 are different on average. This means that the promotion suggestions may differ a lot at different time even when they promote the same target. Besides, with time interval increasing the difference increases (e.g., "1-7"). This phenomenon elaborates that the promotion suggestion is continuously changing. Since QAC ranking changes over time [3], in order to keep the promotion targets showing in QAC results, the promoters need to keep maintaining (changing) promotion queries, which makes the promotion suggestions change over time.

Figure 9 is used to illustrate the relationships between the change in promotion campaign and in corresponding query logs. Taking "1-2" as an example, we collect all of the targets in promotion suggestions (Table 1g) from two annotated sets Q_1 and Q_2. We name the promotion targets that are contained in Q_2 but not appear in Q_1 as *emerging targets*. Such targets did not appear for a lot of times in queries of D_1 (shown as legend A) but many times in D_2 (shown as legend B), which is a common phenomenon in all comparisons. This illustrates that during one week prior to new promotion suggestions (with emerging targets) showing, many corresponding promotion queries are submitted to search engine. Besides, for the *disappearing targets* that only exist in Q_1, the number of queries that contain these targets in D_1 (shown as legend C) is bigger than that in D_2 (shown as legend D). This result means that if the promotion queries (with disappearing targets) are not sufficient, the promotion suggestions that display corresponding targets will disappear.

We can draw the following conclusions from Figure 9: i) promoters achieve promotion goals by submitting sufficient promotion queries that contain specific targets to search engines; ii) If the number of a certain series of queries are declining or disappearing, the corresponding promotion targets will disappear

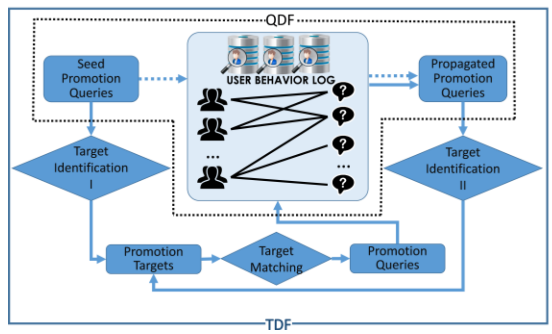


Figure 10: Detection of promotion campaign in QAC.

in QAC; iii) It takes time to achieve a promotion goal, thus the promoters usually start the promotion campaigns at least a week before the emerging targets show in QAC.

Through these detailed analysis, we obtain a better understanding of the promotion campaigns. The promotion query is the channel that facilitates the generation of promotion campaign. The promotion query also has to be diverse and continuously changing in order to promote a target.

3. PCQ DETECTION FRAMEWORK

Figure 10 presents the flowchart of our detection framework. For promotion Query Detection Framework (QDF) depicted within the dashed arrows, we first select a small set of seed promotion queries from crowd sourcing efforts or E-commerce Web sites. Then, we construct a "user-query" bipartite graph based on user query logs. Finally, we propose a propagation algorithm to diffuse the spamming scores of seed promotion queries on the bipartite graph to detect additional queries that contain promotion intents. We can eliminate PCQ in the early phase by rejecting these query streams or by post-filtering them in the result rankings of QAC.

To conduct further research of catching the inherent goal (i.e. the promotion target) of PCQ, we extend the proposed QDF to the promotion Target Detection Framework (TDF) depicted by the solid arrows in Figure 10. At first, we rely on target identification I (manual annotation) to identify seed promotion targets from seed promotion queries. After the step of target matching, we collect all the promotion queries from query logs that contain promotion targets in theirs suffixes. Then, we use the matched promotion queries to drive the "user-query" bipartite graph proposed in QDF to propagate to more queries. More promotion targets will thus be identified from diffused queries by target identification II (our target identification algorithm). The diffused promotion targets will continue to match more promotion queries and conduct propagation in bipartite graph until the iteration terminates.

After the target detection framework (TDF) is performed, as a unified working flow, using detected promotion targets we can conduct early phase solution of the PCQ by removing or not considering the queries that are most likely to possess promotion intents, whose suffixes contain promotion targets. In relatively late phase, we can also identify the recommended suggestions that contain promotion targets in their suffixes to stop them from appearing in QAC in real-time.

3.1 Promotion Query Detection

In this section, we describe in details our QDF algorithm. As mentioned in Sec.2.2.1, we first select a set of 27 successful promotion cases provided by the QAC advertising service markets (similar to those in Figure 1). From those successful promotion cases, we obtain their corresponding explicit promotion suggestions S. Given the query log dataset D_1 (Sec.2.2.1), we select a subset of query entries that match those promotion suggestions from S as seed promotion queries. Through this method, we obtain 12 seeds (i.e. unique seed promotion queries).

We propose a "user-query" bipartite graph propagation algorithm based on the following assumptions:

Assumption 1: *a promoter conducts multiple promotion campaigns (i.e. multiple unique queries) for different customers.*

Assumption 2: *a promotion query may be submitted by a number of promoter/user identities.*

Based on the assumptions (as we verified in Sec.2.3.1), we can diffuse the spam scores of seed promotion queries on the "query-user" bipartite graph. Specifically, the query logs can be viewed as a bipartite graph, with query vertices and user vertices at two sides, and each edge represents one unique query log entry connecting its query and user vertices. The seed queries can be used as the initial seeds to drive the propagation process on the bipartite graph. To construct the bipartite graph, we determine the weight of each edge as follows.

Here, we define an edge weight matrix W_{uq}, where u represents a user, and q represents a query. In addition, we also define two node weight matrixes W_u and W_q on the user level and the query level respectively, based on the characteristics of the promotion campaign. Our algorithm is shown in Algorithm 1, where $w_{uq}(u_j,q_i)$ represents the frequency that query q_i is submitted by u_j, $N(u_j)$ and $N(q_i)$ respectively represent the query frequency of user u_j and the frequency of query q_i, and N_j is the set of all the neighbors of vertex u_j. The weight $w_q(q_i)$ of vertex q_i is calculated based on the behavior characteristics of the promoters:

$$w_q(q_i) = 1 + \frac{N(q_i) - C(q_i)}{N(q_i)} + \sum_{i=1}^{N(q_i)} \frac{I(t_{i+1} - t_i)}{N(q_i) - 1} \tag{1}$$

where $C(q_i)$ is the number of query records that have clicks on the search results. As Figure 7 shows, the promotion queries maintain relatively low click ratio. Besides, we sort the logs of the same query in chronological order, and assuming if the interval between two adjacent logs is less than ε, the indicator function $I(\cdot)$ returns 1, otherwise 0. Based on the labeled queries in Q_l, we found that 70% of promotion queries' interval results calculated by the third part of equation 1 are higher than 0.5. However, this is the case for only 10% of the normal queries. This implies that promoters (spammers) submit the promotion queries at more regular

Algorithm 1: Promotion intent propagation on bipartite graph

Input: Q_s: Selected seed promotion query set;
U: The user set;
Q: The query set;

for $q_i \in Q_s$ **do** $p_q(q_i) = 1$
for $q_i \in Q$ **do** $p_q(q_i) = 0$
while not converged **do**
  for $u_j \in U$ **do**
    $p_u(u_j) = w_u(u_j) \times \sum_{i \in N_j} \frac{w_{uq}(u_j, q_i)}{N(u_j)} \times p_q(q_i)$
  for $q_i \in Q$ **do**
    if $q_i \in Q_s$ **then** $p_q(q_i) = 1$
    else $p_q(q_i) = w_q(q_i) \times \sum_{j \in N_i} \frac{w_{uq}(q_i, u_j)}{N(q_i)} \times p_u(u_j)$

Output: $p_q(q_i)$ for all queries

intervals. Utilizing the same methodology, we can also obtain the user node weight $w_u(u_j)$.

After the algorithm terminates, each query q_i and user u_j respectively receive a score of spam probability $p_q(q_i)$ and $p_u(u_j)$, which denote the probability of a query or a user being a promotion query or a promoter, respectively. We then rank the diffused queries in descending order of this spam probability, and select the top 3000 queries as the observed query set O_q. As described in Sec.2.2.2, by manual annotation, 1235 promotion queries are detected and added to the promotion query set Q_p.

3.2 Promotion Target Detection

Our promotion Query Detection Framework (QDF) algorithm can detect the queries that contain promotion intents. However, as mentioned above, in order to increase the visibility of promotion target in QAC, multiple promotion queries are designed by spammers to promote a target (as shown in Sec.2.3.1). This increases the difficulty of promotion query detection. Promotion target (i.e. the entity to be promoted) is the inherent goal of spammers, which consistently appear in a batch of promotion queries. Therefore, if the promotion targets can be effectively detected, we can use these targets to identify all of the corresponding queries that promote them in the query logs. In this section, we present the target detection framework (TDF).

To achieve this, we need to first accurately identify the promotion target from the query phrase. We regard the target as a name entity and use the open source toolkit CRF++[5] based on the model of Conditional Random Fields (CRF) to extract the target entities. To train a CRF++ model, we randomly select 500 promotion queries from Q_p, and apply part of speech and target tagging on those selected queries to generate the training documents for CRF++. Based on the trained model, we obtain the predicted probability p_w that indicates the likelihood of a word in the query phrase being a promotion target.

Promotion targets tend to be present in the auto-completion suffix of query suggestions, which means that a target entity located at the latter part of a query exhibits a higher probability of being promotion target (as Figure 6(b) illustrates). To improve the accuracy of target recognition, we calculate the statistical position probabilities p_f, p_m, and p_b that represent the probability of a promotion target resides in the front, middle, and back of the query, respectively. Through statistical analysis of the promotion query set Q_p, the position probabilities are estimated as: p_f =

[5] https://taku910.github.io/crfpp/

0.053, $p_m = 0.412$, $p_b = 0.535$. We thus adopt these probabilities as the weight to polish the predicted probabilities from the CRF++ model, i.e. according to the character position of each word in the query, each predicted probability p_w of the word is multiplied by the corresponding statistical position weight to obtain the ultimate promotion target probability p_c. Then given a query and all the possible words within the query, we choose the word that has the largest p_c as the target entity.

After the procedure of target entity recognition, we can perform our extended framework TDF as shown in Figure 10. Figure 10 illustrates that TDF is an iterative framework with cyclical process. We aim to propagate to get more promotion targets using a small set of seed targets that are extracted from seed queries Q_s using manual identification. Specifically, 12 unique seed targets are used to drive the extended framework with an initial spam score of 1. The new seed promotion queries Q_{s1} that contain seed targets are selected from query logs with spam scores that are set to be identical to the position probabilities according to the character-based position of seed targets matched in the queries. The Q_{s1} then drives propagation process based on the bipartite graph propagation in Algorithm 1. The algorithm of promotion target detection is shown in Algorithm 2.

The score of target identified from diffused queries is calculated by incorporating all the promotion probabilities of the queries that contain this target entity. Similar to promotion query detection, each target entity is assigned a probability score when the algorithm terminates, with a larger numeric score denotes a higher likelihood that a target entity is a promotion target. At the last propagation iteration, we count the number of times that a target entity appears in the propagated queries. To promote one target, multiple promotion queries need to be executed, which means that the promotion targets should appear multiple times. By analyzing the top 400 target entities, we find that 92% of the entities that appear less than 3 times are not contained in promotion queries. So we drop all the target entities whose frequency is less than 3.

4. EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the performance of proposed detection frameworks, compared with a set of baselines (Sec.4.1). We use the query log datasets, and the annotated query sets (as described in Sec.2.2) for the evaluation.

4.1 Experiment Setups

4.1.1 Baseline Methods

In order to evaluate the effectiveness of our proposed method, we implement two types of baseline approaches (i.e., learning-based method and link-based method) to compare against.

The most commonly used spam detection approach is to utilize machine learning to train a classifier to distinguish spam from normal objects [29]. The differences among the classifiers mainly lie in the features used to represent the data. Therefore, we adapt previous spam detection work to our context (detecting promotion query), and utilize Support Vector Machine (SVM) to train three different models based on content based features only, behavior based features only, and a combination of both content and user features respectively, as our three baselines. We describe those features in more details.

Content based features: We adapt the features described in [29] that utilizes lexical patterns and part-of-speech patterns to effectively identify deceptive messages. We regard the promotion query as deceptive text content, and then extract those content-based features to represent each query.

User Behavior based features: Previous research [13] found that features based on user behavior patterns can detect Web spam effectively. Through tracking query logs, we can observe that a query can be executed multiple times by one user or several users. Following this previous work, we extract 8 user behavior features for each query, including the mean query time interval between the query issuing of a single user that submits multiple times, the mean time interval of all the users' query submissions, mean interval time between query and click, the number of users that search the query, the largest number of all the users' query frequencies, the average query frequency per user, two features quantifying click ratio and query time interval that calculated according to the second and third part of equation 1 respectively.

Algorithm 2: Promotion target detection

Input: T_s: Selected seed target set;
Q_{s1}: New seed queries;
$p_p : p_f, p_m, p_b$;
1: **for** $t_i \in T_s$ **do** $p(t_i) = 1$
2: **for** $q_i \in Q_{s1}$ **do** $p(q_i)$ is assigned according to target position
3: **while** not converged **do**
4: obtain queries Q_o that contain any targets
5: **for** $q_k \in Q_o$ **do**
6: $p(q_k) = p(t_j) \times p_p$
7: obtain queries Q_p by a propagation
8: **for** $q_j \in Q_p$ **do**
9: identify and store target t_j and get $p(t_j)$
Output: $p(t_i)$ for all the target entities

Besides learning-based detection method, we also choose TrustRank [14] as a link-structure analysis based baseline, which has proved effective at detecting Web spam. The user IDs in the query logs are used to connect queries. For example, if a user u submitted query q_i and query q_j, then we build an edge between q_i and q_j. After establishing connections between all the queries, we model the query logs as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of a set $\mathcal{V}$ of N queries (vertices) and a set $\mathcal{E}$ of undirected links (edges) that connect queries.

4.1.2 Evaluation Methodology

We randomly select 3000 queries from D_1 and label them as promotion or non-promotion, then train each classifier model described in Section 4.1.1 using the 3000 labeled queries. Based on the graph $\mathcal{G}$, we employ the same seed queries used in the QDF to drive TrustRank algorithm. To evaluate all the methods, we perform them (i.e., four baselines as described in Section 4.1.1 and QDF, as shown in Table 3) on dataset D_2. QDF-W does not consider node weight and is used to illustrate the effectiveness of node weight derived from the behavioral characteristics mentioned in Section 3.2.

In order to fairly compare those approaches, we utilize the widely used pooling method [30] to gather the queries for judgments. Essentially, we select the top 1000 queries identified as promotional by each method (six query sets each contains 1000 queries) and use the 6000 queries to form a result pool. After eliminating the duplicated queries, we manually label the remaining queries in the pool as the test dataset Q_t which contains 1227 promotion queries and 2343 normal ones.

4.2 Performance of QDF

The AUC results of different methods on dataset Q_t are shown in Table 3. We can observe that not surprisingly, the classifier that combines both content and user behavior features can achieve better performance than using solely one type of features. The content-based classifier achieves the worst performance, which

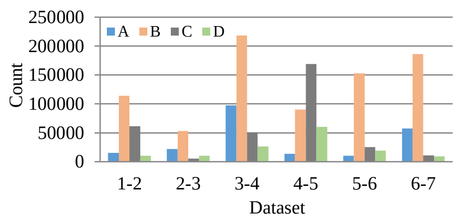


Figure 9: Relations between changes in promotion target and changes in corresponding query behaviors.

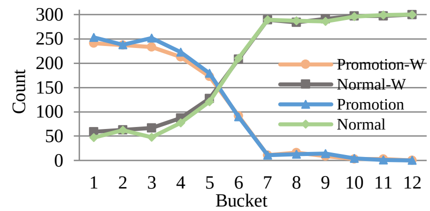


Figure 11: Detection performance (in terms of the number of promotion queries in buckets) of proposed QDF algorithms.

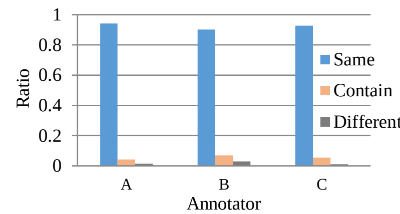


Figure 12: Performance of the target entity recognition model.

may be due to the fact that queries are in general too short for content-based method to perform effectively. To achieve promotion goal, the promoters show some abnormal behaviors inevitably. Therefore, user behavior based classifier obtains relatively good performance. The link analysis based approach TrustRank performs better than the classifier approach (only less effective than our methods). This indicates that considering the relationships between users and queries can be helpful in PCQ detection. Our proposed QDF methods make full use of the assumptions (Sec.3.1) based on analyzing the user-query relationships, and then propagate the promotion intents on bipartite graph. Therefore, both the QDF methods with and without node weight achieve very high AUC.

Table 3: Detection performance (in AUC) of two different proposed QDF methods compared to baselines.

Approaches	AUC	Differences (with QDF)
Classifier (Content)	0.756	-21.5%
Classifier (User)	0.807	-16.4%
Classifier (Content + User)	0.831	-14.0%
TrustRank	0.915	-5.6%
QDF-W (no Node Weight)	0.962	-0.9%
QDF (with Node Weight)	0.971	–

We select the top 3000 queries as the observed set O_q after the QDF terminates. To perform more detailed analysis of QDF, we segment the O_q into 10 buckets according to the spamming score ranking of each query, and then count the number of promotional and normal queries in each bucket. The distributions of each bucket's promotion query and normal query for the QDF algorithm are shown in Figure 11, where one is propagating without node weights (denoted as "-W"), and the other is propagating with node weights. As we can see, in buckets 1, 2, and 3, most of the queries contain promotion intents (with the detection precision of 82%), while in buckets 7, 8, 9, and 10 almost all of the queries are normal in both of the experimental results. The fraction of normal queries increases as the detected query spamming score decreases, which means that the query with the high spamming score maintain high probability to be a promotion query. As we can see in buckets 1-4, our QDF with the node weight method detects more promotion queries than the method without the node weight. It implies that the node weight based on behavioral characteristics is helpful in promotion campaigns detection.

4.3 Performance of TDF

As mentioned above, using TDF, two countermeasures against PCQ can be implemented, which are advance precaution and real-time identification. Promoters use promotion query as their spamming channel to fulfill their ultimate goal on promoting targets. Therefore we regard the query or suggestion whose suffix contains promotion target as a promotion query or a promotion suggestion. After detecting the promotion targets, we can remove the promotion queries detected by promotion targets from the query logs to restrain the spamming channels. By this way, we can prevent promotion campaigns in advance. Besides, based on the detected targets, we can apply appropriate restriction to promotion suggestions that contain promotion targets in QAC when the search engine recommends suggestions. Through this method, we can identify promotion suggestions in real-time. We describe the performance of those two countermeasures as below.

4.3.1 Performance of target entity recognition

Before we conduct our Target Detection Framework evaluation, we first validate the performance of our target entity recognition method. We select 500 promotion queries obtained from the propagation of the query logs in dataset D_1 to train the model of CRF++. To ensure the reliability of our method, we perform QDF on the new dataset D_7 and manually label 250 promotion queries after the propagation terminates as the target identification test set Q_t. Two methods are conducted to identify targets from each of the promotion queries in Q_t, one is based on manual annotation while the other is based on our trained target entity recognition model. We ask three annotators to perform targets labeling and individually compare the annotation results with the target set obtained from our trained model. The evaluation results are shown in Figure 12.

We can see that about 96% of the query targets identified from our trained model (CRF++) are almost the same with the manual annotations (from any annotator) for more than 90% of the cases, while only a few of the identified query targets are different. This demonstrates that our target entity recognition method is effective.

4.3.2 Early phase detection

We conduct the target detection framework on the most recent query log dataset D_7. After the detection terminates, we rank the diffused targets in descending order of the predicted spam probability. To evaluate the performance of the TDF, we select three query sets from D_7: low-frequency ranges from 14 to 30 query frequency, mid-frequency ranges from 31 to 100, and high-frequency ranges from 200 to 900 queries, while each set contains 300 cases. We then use the early phase detection method (advance precaution) to detect promotion queries from those 900 query set (denoted as Q_7'). As almost all of the promotion targets appear after the 4th character of the promotion queries or promotion suggestions (Fig.6 (b)), therefore we deem a query as a promotion query if it contains a promotion target after the forth character.

Due to space limitation, we only present in Table 4 the evaluation results of top 50 targets where the performance is 0.85 in terms of F-measure. Note that if we choose the top 10 promotion targets to detect promotion queries, we can obtain very high precision (close to 1.0) but relatively low recall (around 0.4). With the increase of the number of selected targets, the recall gradually increases but the precision decreases. We obtain the best F-measure when the number of selected targets is 50.

4.3.3 Late phase detection

After we label the selected 900 queries, we also manually collect the suggestion lists triggered by them. By annotating, we obtain the labeled suggestions with 549 promotion suggestions (13%) that contain the corresponding ranking information in QAC and 3,636 normal suggestions (87%). We discover that a large portion (about 30%) of the promotion suggestions appear in high QAC rankings (within top 3). It demonstrates that QAC service is seriously contaminated by PCQ. We then apply our late phase solution method (real-time identification) to detect the promotion suggestions. Again, we deem a QAC suggestion as a promotion suggestion if the suggestion contains promotion target after the forth character.

Similar with the early phase solution, with a small number of high-ranking targets, we can get higher precision but relatively lower recall. With the increasing of the number of targets, the recall value gradually increases. When we select the top 50 targets we obtain the best detection results in terms of F-measure (0.84), as shown in Table 4. To summarize, we demonstrate that TDF in both early and late phases performs well.

Table 4: TDF performance in two detection phases

	Precision	Recall	F-measure
Early phase	0.899	0.807	0.851
Late phase	0.900	0.800	0.847

4.3.4 Comparison with QDF

To compare the performance of QDF with the TDF framework (early phase) in terms of detecting promotion queries, we perform the evaluation on the created query set Q_7' (Sec.4.3.2). We empirically set a threshold θ, for QDF, i.e. if a query's QDF predicted spam score is higher than θ, we deem it as a promotion query. The detection results on query sets with various query frequencies are presented in Table 5.

In general, both QDF and TDF perform well in detecting promotion queries with different frequencies', which indicates that our framework is effective and robust. Besides, we demonstrate that compared with QDF, the performance of TDF is better, which means that using TDF can detect more promotion queries than QDF. Through TDF, we can achieve two countermeasures against PCQ, and the experimental results show that our framework is very helpful in eliminating the promotion campaigns in QAC. Through TDF, we identify the inherent goal of the promotion campaigns. No matter how promotion queries and suggestions evolve, the promotion campaigns can be eliminated because their promotion goals remain relatively stable.

Table 5: Comparison of performance between QDF and TDF in detecting promotion queries of different query frequencies. The bold number represents the TDF results.

Query Frequency	Precision	Recall	F-measure
200-900	0.810 / **0.835**	0.741 / **0.737**	0.773 / **0.783**
31-100	0.843 / **0.891**	0.810 / **0.889**	0.826 / **0.890**
14-30	0.789 / **0.840**	0.742 / **0.769**	0.765 / **0.803**

4.4 Performance on Unabridged Query Logs

To better observe and analyze the promotion campaigns in QAC, we filter the query logs. Experiments in Sections 4.2 and 4.3 show that our framework performs well in detecting PCQ on the filtered datasets (i.e., $D_1, D_2, \ldots D_7$). To validate that our framework can achieve effective detection results in unprocessed real-world query logs, we use the unabridged dataset D_{test} (3 days' entire query logs mentioned in the last paragraph of Section 2.2.1) to evaluate QDF and TDF in this section.

Again, we use the seed promotion queries and promotion targets that extract from D_1 to drive the propagation process. After QDF experiment, we also rank all the queries in descending order of their spam scores, and then select the top 3000 queries for annotation. In this section, our goal is to evaluate our framework's performance, therefore we ask the three annotators to only label the promotion intents (Table 1a). After labeling, 1561 promotion queries are identified. We find that only 41.5% of these queries are the same with the promotion queries in Q_1. Using the labeled queries, we calculate that the AUC is 0.906. Table 6 presents the results for both QDF and TDF. For TDF, late phase detection cannot be evaluated due to the corresponding suggestions are not collected, therefore we only present the TDF results in the early phase. As we can see, although our framework performs a little worse on unfiltered dataset (Table 6) comparing with pre-processed datasets (Table 5), it still achieves acceptable results and can eliminate PCQ effectively.

Table 6: Detection performance on D_{test}.

	Precision	Recall	F-measure
QDF	0.781	0.802	0.791
TDF	0.829	0.800	0.814

5. RELATED WORK

Three lines of researches are closely related to the PCQ detection problem we describe in this paper: Web spam detection, online promotion campaign detection, and QAC performance evaluation.

Web Spam Detection. Web spamming techniques can be grouped into two categories: content and link spamming [8]. Previous works introduced several content-based features to detect Web spam pages such as term features [9], linguistic features [10], textual features [11] and HTML patterns [12]. These features are difficult to apply to the detection of QAC promotion campaigns because they aim to manipulate the ranking of query suggestions rather than Web pages, which are much shorter and lack of structure information. Link spammers create hyper-link structures to optimize scores of promotion targets in the hyper-link structure analysis algorithms [13]. A variety of trust and distrust propagation algorithms such as TrustRank [14] and Truncated PageRank [15] prove to be effective in terms of demoting the spams. These provide valuable lessons about propagation algorithms but they cannot be directly used in detecting QAC spamming activities, either.

Online Promotion Campaign Detection. There exist a large number of promotion campaigns in social media platforms. These promotion activities are usually in the form of coordinated free text campaigns and the amount has recently been growing in significance. Lee et al. [16] studies the problem of detecting these campaigns in twitter with a content–driven framework. After that, a scalable framework is proposed to detect both spam posts and promoting campaigns by Zhang et al [17], which tries to identify accounts that post URLs for similar promotion purposes. Recently, many promotion campaigns also rely on CQA platforms to misguide users. To fight this kind of spamming activities, Li et al [18], focus on the promotion channels that lead to actual spamming content and propose a propagation algorithm to detect possible promotion campaigns.

Promotion campaigns in Query Auto Completion (QAC) scenarios are different from the campaigns in social media platforms because the promotion suggestions are created indirectly by submitting malicious queries containing target information to search engines. Therefore, we need to use query logs to study spammers' cheating strategies and conduct detection accordingly. Besides, new detection strategies are required because it can be extremely difficult for us to extract various features from the promotion suggestions due to their inherent character of short length [6].

QAC Performance Evaluation. To find out whether QAC help users to fulfill their information needs, Shokouhi [4] and Yossef [20] consider the submitted queries as ground-truth and use retrieval performance metrics to measure the performance of QAC rankings. Shokouhi et al. [3] and Strizhevskaya et al. [21] leverage the aggregated query frequencies to construct an oracle QAC list for each prefix and then compare the actual lists with the oracle ones. Kharitonov et al. [22] propose a model of user interactions with QAC based on click behavior modeling in Web search scenarios. Based on this model, they propose two metrics, e-Saved and p-Saved, for evaluating the quality of QAC ranking lists. Manual judgments [23] and alternative evaluation procedures [24] are also proposed to evaluate the performance of query suggestions. The QAC quality evaluation methods can provide insights into the quality of the legitimate generated suggestions. However, it can only detect promotion suggestions after they appear in the QAC lists because it requires user interaction data (query or click) to evaluate the ranking performance. It therefore cannot be adopted to avoid the bad influence of PCQ to search users.

6. CONCLUSION AND FUTURE WORK

In this paper, we introduce the new emerging problem of promotion campaigns in QAC. With comprehensive manual annotations of a number of detected promotion queries, we analyze the properties and dynamic characteristics of PCQ to get a better understanding of this malicious phenomenon. Based on the analysis into the cause for these campaigns, we propose a promotion Query Detection Framework (QDF). To catch the inherent goal of PCQ, we further propose an extended promotion Target Detection Framework (TDF) based on QDF to avoid promotion campaigns from the target level. Based on extensive experiments, we show the effectiveness of the two proposed detection frameworks.

This work is a first attempt towards studying and detecting the PCQ phenomena, and there is much room for further improvements. For example, we can further study the characteristics of PCQ, to model the behaviors of the promoters while submitting queries and improve the performance of our framework. Besides, analyzing the syntactic patterns of promotion queries to develop specific entity recognition algorithm may also further improve promotion target identification performance.

7. REFERENCES

[1] Bhatia, S. Majumdar, D. and Mitra, P. 2011. Query suggestions in the absence of query logs. In *SIGIR'11*, ACM, 795-804.

[2] Hofmann, K. Mitra, B. Radlinski, F. and Shokouhi, M. 2014. An Eye-tracking Study of User Interactions with Query Auto Completion. In *CIKM'14*, ACM, 549-558.

[3] Shokouhi, M. and Radinsky, K. 2012. Time-sensitive query auto-completion. In *SIGIR'12,* ACM, 601-610.

[4] Bar-Yossef, Z. and Kraus, N. 2011. Context-sensitive query auto-completion. In *WWW'11*. ACM, 107-116.

[5] P. Boldi, F. Bonchi, C. Castillo, D. Donato, and S. Vigna. Query suggestions using query-flow graphs. In *Proceedings of the 2009 workshop on Web Search Click Data*. pages 56–63, 2009.

[6] Shokouhi, M. 2013. Learning to personalize query auto-completion. In *SIGIR'13*, ACM, 103-112.

[7] Cao, H. Jiang, D. Pei, J. He, Q. Liao, Z. and Chen, E. et al. 2008. Context-aware query suggestion by mining click-through and session data. In *SIGKDD'08*, ACM, 875-883.

[8] Castillo, C. and Davison, B. D. 2011. Adversarial Web search. *Foundations & Trends in Information Retrieval, 4*(5), 377-486.

[9] Ntoulas, A. Najork, M. Manasse, M. and Fetterly, D. 2006. Detecting Spam Web Pages through Content Analysis. In *WWW '06*, ACM.

[10] Piskorski, J. Sydow, M. and Weiss, D. 2008. Exploring linguistic features for Web spam detection: A preliminary study. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web* (Beijing, China, April 22, 2008).

[11] Attenberg, J. and Suel, T. 2008. Cleaning search results using term distance features. In *AIRWeb'08*, ACM, 21-24.

[12] Urvoy, T. Chauveau, E. Filoche, P. and Lavergne, T. Tracking Web spam with HTML style similarities. *ACM Transactions on the Web*. February, 2008.

[13] Liu, Y. Cen, R. Zhang, M. Ma, S. and Ru, L. 2008. Identifying Web spam with user behavior analysis. In *AIRWeb'08*, 9-16.

[14] Krishnan, V. and Raj, R. 2006. Web Spam Detection with Anti-Trust-Rank. In *AIRWeb'06*, 37-40.

[15] Becchetti, L., Castillo, C., Donato, D., Leonardi, S., & Baeza-Yates, R. 2006. Using rank propagation and probabilistic counting for link-based spam detection. In *Proc. of WebKDD* (Vol. 6).

[16] Lee, Kyumin, et al. 2011. Content-driven detection of campaigns in social media. In *CIKM'11*, ACM, 551-556.

[17] Zhang, X. Zhu, S. and Liang, W. 2012. Detecting Spam and Promoting Campaigns in the Twitter Social Network. 2012 IEEE 12th *International Conference on Data Mining* (Vol.5, pp.1194-1199). IEEE Computer Society.

[18] X, Li. Y, Liu. M, Zhang. S, Ma. X, Zhu. and J, Sun. 2015. Detecting Promotion Campaigns in Community Question Answering. In *AAAI'15,* 2348-2354.

[19] Lakhani, R. Google Instant Search Autocomplete Manipulation, 2011. http://www.ecreativeim.com/blog/2011/03/google-instant-search-autocomplete-manipulation/

[20] Strizhevskaya, A. Baytin, A. Galinskaya, I. & Serdyukov, P. 2012. Actualization of query suggestions using query logs. In *WWW'12*, ACM, 611-612.

[21] Kharitonov, E. MacDonald, C. Serdyukov, P. and Ounis, I. 2013. User Model-based Metrics for Offline Query Suggestion Evaluation. In *SIGIR'13*, ACM, 633-642.

[22] Bhatia, S., Majumdar, D., & Mitra, P. 2011. Query suggestions in the absence of query logs. In *SIGIR'11*, ACM, 795-804.

[23] Liu, Y. Song, R. Chen, Y. Nie, J. Y. and Wen, J. R. 2012. Adaptive query suggestion for difficult queries. In *SIGIR'12*, ACM, 15-24.

[24] Mei, Q. Zhou, D. and Church K. 2008. Query suggestion using hitting time. In *CIKM'08,* ACM, 469–478.

[25] Starling, L. How to remove a word from Google autocomplete. 2014, http://searchengineland.com/using-autocomplete-hijack-local-search-results-improve-online-reputation-199568R.

[26] Landis, J R. and Koch, G G. The measurement of observer agreement for categorical data. *Biometrics* 33.1(1977):159-174.

[27] Jones, R. Rey, B. Madani, O. and Greiner W. Generating query substitutions. 2006. In *WWW'06*, ACM, 387–396.

[28] Lazar, J. Meiselwitz, G. & Feng, J. Understanding Web credibility: a synthesis of the research literature. *Foundations and Trends® in Human-Computer Interaction,* 2007, 1:139-202.

[29] Qazvinian, V. Rosengren, E. Radev, D R. et al. 2011. Rumor has it: identifying misinformation in microblogs. In *EMNLP'11*, 1589-1599.

[30] Sanderson M. Test Collection Based Evaluation of Information Retrieval Systems [J]. *Foundations & Trends® in Information Retrieval*, 2010, 4(6):247-375.