An Active Learning Approach to Recognizing Domain-Specific Queries From Query Log

Weijian Ni, Tong Liu(✉), Haohao Sun, and Zhensheng Wei

College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, Shandong, China
niweijian@gmail.com, liu_tongtong@foxmail.com, shhlat@163.com, zhensheng_wei@163.com

Abstract. In this paper, we address the problem of recognizing domain-specific queries from general search engine's query log. Unlike most previous work in query classification relying on external resources or annotated training queries, we take query log as the only resource for recognizing domain-specific queries. In the proposed approach, we represent query log as a heterogeneous graph and then formulate the task of domain-specific query recognition as graph-based transductive learning. In order to reduce the impact of noisy and insufficient of initial annotated queries, we further introduce an active learning strategy into the learning process such that the manual annotations needed are reduced and the recognition results can be continuously refined through interactive human supervision. Experimental results demonstrate that the proposed approach is capable of recognizing a certain amount of high-quality domain-specific queries with only a small number of manually annotated queries.

Keywords: Query classification · Active learning · Transfer learning · Search engine · Query log

1 Introduction

General search engines, although being an indispensable tool in people's information seeking activities, are still facing essential challenges in producing satisfactory search results. One challenge is that general search engines are always required to handle users' queries from a wide range of domains, whereas each domain often having its own preference on retrieval model. Taking two queries "steve jobs" and "steve madden" for example, the first query is for celebrity search, thus descriptive pages about Steve Jobs should be considered relevant; whereas the second one is for commodity search, thus structured items of this brand should be preferred. Therefore, if domain specificity of search query was recognized, a targeted domain-specific retrieval model can be selected to refine search results [1,2]. In addition, with the increasing use of general search engines, search queries have become a valuable and extensive resource containing a large number of domain named entities or domain terminologies, thus domain-specific

© Springer International Publishing AG 2017
L. Chen et al. (Eds.): APWeb-WAIM 2017, Part II, LNCS 10367, pp. 18–32, 2017.
DOI: 10.1007/978-3-319-63564-4_2

query recognition can be viewed as a fundamental step in constructing large scale domain knowledge bases [3].

Domain-specific query recognition is essentially a query classification task which has been attracting much attention for decades in information retrieval (IR) community. Many traditional work views query classification as a supervised learning problem and requires a number of manually annotated queries [4,5]. However, training queries are often time-consuming and costly to obtain. In order to overcome this limitation, many studies leveraged both labeled and unlabeled queries in query classification [6,12]. The intuition behind is that queries strongly correlated in click-through graph are likely to have similar class labels.

In this paper, inspired by semi-supervised learning over click-through graph in [6,7], we propose a new query classification method that aims to recognize queries specific to a target domain, utilizing search engine's query log as the only resource. Intuitively, users' search intents mostly remain similar in short search sessions and most pages concentrate on only a small number of topics. This implies the queries frequently issued by same users or retrieve same pages are more likely to be relevant to the same domain. In other words, domain-specificity of each queries in query log follows a manifold structure. In order to exploit the intrinsic manifold structure, we represent query log as a heterogenous graph with three types of nodes, i.e., users, queries and URLs, and then formulate domain-specific query recognition as transductive learning on heterogenous graph.

The performance of graph-based transductive learning is highly rely on the set of manually pre-annotated nodes, named as seed domain-specific queries in the domain-specific query recognition task. We further introduce a novel active learning strategy in the graph-based transductive learning process that allows interactive and continuous manual adjustments of seed queries. In this way, the recognition process can be started from an insufficient or even noisy initial set of seed queries, thus alleviating the difficulty of manually specifying a complete seed set for recognizing domain-specific queries. Moreover, through introducing interactive human supervision, the seed set generated during the recognition process tend to be more informative than the one given in advance, and is beneficial to improve the recognition performance.

We evaluate the proposed approach using query log of a Chinese commercial search engine. We provide in-depth experimental analyses on the proposed approach, and compare our approach with several state-of-the-art query classification methods. Experimental results conclude the superior performance of the proposed approach.

The rest of the paper is organized as follows. Section 2 describes the graph representation of query log. Section 3 gives a formal definition of domain-specific query recognition problem together with the details of the proposed approach. Section 4 presents the experimental results. We discuss related work in Sect. 5 and conclude the paper in Sect. 6.

2 Graph Representation of Query Log

In modern search engines, the interaction process between search users and search engine is recorded as so-called query log. Despite of the difference between search engines, query log generally contains at least four types of information: users, queries, search results w.r.t. each query and user's click behaviors on search results. Table 1 gives an example of a piece of log that is recorded for an interaction between a user and search engine.

In this work, we make use of heterogenous graph, as shown in Fig. 1, to formally represent the objects involved in the search process. More specifically, a tripartite graph composed of three types of nodes, i.e., users, queries and URLs is constructed according to the interaction process recorded in query log. There are two types of links (shown by dashed line and dotted line) in the tripartite graph that indicate query issuing behavior of search users and click-through behavior between queries and URLs, respectively. In addition, the timestamps of query issuing behaviors are attached on each links between the corresponding user and query.

Based on the graph representation, the inherent domain-specificity manifold structure in query log implies that the strongly correlated queries, through either user nodes or URL nodes, are highly likely to be relevant to the same domain. Therefore, with a set of manually annotated domain-specific queries

Table 1. Query log example

Field	Content	Description
UserId	*bc3f448598a2dbea*	The unique identifier of the search user
Query	*piglet prices sichuan*	The query issued by the user
URL	*alibole.com/57451.html*	URL of the webpage retrieved by the query
Timestamp	*20111230114640*	The time when the query was issued
ViewRank	*4*	The rank of the URL in search results
ClickRank	*1*	The rank of the URL in user's click sequence

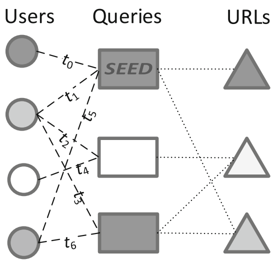

Fig. 1. User-Query-URL tripartite graph representation

(e.g., depicted as a *SEED* gray rectangle in Fig. 1), the domain-specificity of other queries can be derived through transductive learning on the tripartite graph (e.g., in Fig. 1, the gray level of each nodes indicates the mass of domain-specificity propagated from the seed node).

In the next section, we will describe the graph-based transductive learning process in more details.

3 Domain-Specific Query Recognition

3.1 Problem Definition

Formally, let $\mathcal{G}_{tri} = \langle U \cup Q \cup L, E^{(UQ)} \cup E^{(QL)} \rangle$ be the tripartite graph of query log, where $U = \{u_1, \cdots, u_{|U|}\}$, $Q = \{q_1, \cdots, q_{|Q|}\}$ and $L = \{l_1, \cdots, l_{|L|}\}$ denote the set of search users, queries and click-through URLs, respectively.

The links in $E^{(UQ)} \cup E^{(QL)}$ are weighted according to strength of the relation. Intuitively, if a user repeatedly issued the same query or only a few users issued that query, the relation between them would be strong. We thus calculate the weight of link between $u_i \in U$ and $q_j \in Q$ as follows:

$$W_{i,j}^{(UQ)} = \frac{N_{i,j}^{(UQ)}}{\sum_{i'=1}^{|U|} N_{i',j}^{(UQ)}} \cdot log \frac{|Q|}{\sum_{j'=1}^{|Q|} I_{i,j'}^{(UQ)}} \tag{1}$$

where $N_{i,j}^{(UQ)}$ denotes the times u_i issued q_j. $I_{i,j'}^{(UQ)}$ is an indicator function that equals to 1 if these is a link between u_i and q_j in $\mathcal{G}_{tri}$, and 0 otherwise.

The weight of links in $E^{(QL)}$ dependents on frequency and rank of the URL clicked w.r.t the query and the number of the queries bringing click-through on the URL. Similarly as Eq. 1, the weight of link between $q_j \in Q$ and $l_k \in L$ can be calculated by URL frequency w.r.t. the query and inverse query frequency of the URL:

$$W_{j,k}^{(QL)} = \frac{N_{j,k}^{(QL)}/R_{j,k}}{\sum_{k'=1}^{|L|} N_{j,k'}^{(QL)}/R_{j,k'}} \cdot log \frac{|Q|}{\sum_{j'=1}^{|Q|} I_{j',k}^{(QL)}} \tag{2}$$

where $N_{j,k}^{(QL)}$ and $R_{j,k}$ denote the times and rank of l_k clicked w.r.t. q_j, respectively. $I_{j,k}^{(QL)}$ is an indicator function that equals to 1 if these is a link between q_j and l_k in the graph, and 0 otherwise.

In addition, Each link in $E^{(UQ)}$ is associated with a set of timestamps $T_{i,j} = \{t_{i,j}\}$, where $t_{i,j}$ is the time when user u_i issued query q_j. Note that $|T_{i,j}| = N_{i,j}^{(UQ)}$.

As for a target domain, suppose vectors $\mathbf{f} \in [0,1]^{|U|}$, $\mathbf{g} \in [0,1]^{|Q|}$ and $\mathbf{h} \in [0,1]^{|L|}$ denote the predicted domain-specificity of each user, query and URL in query log, respectively. The closer the value is to 1, the more confident the corresponding object is relevant to the target domain. Besides, vector

$\mathbf{y} \in \{0,1\}^{|Q|}$ denotes the pre-annotated domain-specificity of queries. Specifically, if query q_j is manually selected as a domain-specific one, i.e. seed query, $y_j = 1$; otherwise, $y_j = 0$. Using the above notations, the problem of learning domain-specificity is defined as follows:

Problem Definition I (Domain-specific query recognition): *Given a tripartite graph $\mathcal{G}_{tri}$ with its associated link weight matrices $\mathbf{W}^{(UQ)}$ and $\mathbf{W}^{(QL)}$ for a query log, and a set of manually specified seed queries $\mathbf{y}$, the aim is to estimate $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{h}$ as the prediction of domain-specificity of each objects in query log.*

3.2 Tranductive Learning on Tripartite Graph

The assumption for learning domain-specificity of objects in query log is that the domain-specificity distribution exhibits strong manifold structure, i.e., two objects tend to have similar domain-specificity if they are strongly associated with each other. Besides, the domain-specificity learned should be consistent with that pre-annotated on the seed queries. We formally design the following objective function:

$$\begin{aligned}\mathcal{O}_{tri}(\mathbf{f},\mathbf{g},\mathbf{h}) = {} & \alpha \cdot \sum_{i=1}^{|U|}\sum_{j=1}^{|Q|} W_{ij}^{(UQ)}(f_i - g_j)^2 + \beta \cdot \sum_{j=1}^{|Q|}\sum_{k=1}^{|L|} W_{ij}^{(QL)}(g_j - h_k)^2 \\ & + \gamma \cdot \sum_{j=1}^{|Q|}\sum_{j'=1, j'\neq j}^{|Q|}\sum_{i=1}^{|U|} \frac{W_{ij}^{(UQ)} \cdot W_{ij'}^{(UQ)}}{\Delta_{i,j,j'}^{\tau}}(g_j - g_{j'})^2 \\ & + \delta \cdot \sum_{j=1}^{|Q|} I_j^{(seed)}(g_j - y_j)^2\end{aligned} \tag{3}$$

where $I_j^{(seed)}$ is an indicator function that equals to 1 if query q_j is specified as a seed query, and 0 otherwise. $\Delta_{i,j,j'}$ is the minimum timespan of u_i issuing q_j and $q_{j'}$ and is calculated as:

$$\Delta_{i,j,j'} = \min_{t \in T_{i,j}, t' \in T_{i,j'}} |t - t'| \tag{4}$$

In Eq. 3, α, β, γ and δ ($\alpha, \beta, \gamma, \delta \geq 0$ and $\alpha + \beta + \gamma + \delta = 1$) are parameters that balance the contributions of the four items in the objective function. $\tau \geq 0$ is the parameter controlling the shrinkage effect on query weights. The larger τ gives more penalty to pair of queries with long issuing timespan, which are more likely to belong to different search sessions.

The first two items in Eq. 3 evaluate how smooth are the predictions of each objects in query log w.r.t. the manifold structured in the tripartite graph $\mathcal{G}_{tri}$. The third item evaluate the smoothness of predicted query domain-specificity w.r.t. search session. The forth item evaluates how the predicted domain-specificity of seed queries fit with that pre-annotated and it can be viewed as a soft constraint for the predictions on the seed queries.

Based on the objective function in Eq. 3, the optimal domain-specificity of each user, query and URL can be derived through the following optimization problem:

$$\begin{aligned} \min_{\mathbf{f},\mathbf{g},\mathbf{h}} \quad & \frac{1}{2}\,\mathcal{O}_{tri}(\mathbf{f},\mathbf{g},\mathbf{h}) \\ \text{s.t.} \quad & \mathbf{f} \in [0,1]^{|U|},\ \mathbf{g} \in [0,1]^{|Q|},\ \mathbf{h} \in [0,1]^{|L|} \end{aligned} \tag{5}$$

As for the optimization problem in Eq. 5, it is possible to derive a closed form solution. However, the closed form solution requires matrix inversion operations, which will be computationally inefficient since there are generally huge amount of objects, i.e., unique users, queries and URLs in query log. Therefore, we propose an efficient iterative algorithm to approximately solve the optimization problem in Eq. 5 based on Stochastic Gradient Descent (SGD). The basic idea is to iteratively update the domain-specificity of each objects towards the direction of negative gradient of $\mathcal{O}(\mathbf{f},\mathbf{g},\mathbf{h})$, when observing each $\langle$user, query, URL$\rangle$ trinity in query log. More precisely, through computing the derivatives of $\mathcal{O}(\mathbf{f},\mathbf{g},\mathbf{h})$ w.r.t. $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{h}$, we derive the update formulas for $\langle u_i, q_j, l_k\rangle$ as follows:

$$\begin{aligned} f_i &\leftarrow f_i - \mu \cdot \alpha \cdot W_{ij}^{(UQ)}(f_i - g_j) \\ g_j &\leftarrow g_j - \mu \cdot \big(\alpha \cdot W_{ij}^{(UQ)}(g_j - f_i) + \beta \cdot W_{jk}^{(QL)}(g_j - h_k) \\ &\qquad + \gamma \cdot \sum_{j'=1, j'\neq j}^{|Q|} \sum_{i=1}^{|U|} \frac{W_{ij}^{(UQ)} \cdot W_{ij'}^{(UQ)}}{\Delta_{i,j,j'}^{\tau}}(g_j - g_{j'}) \\ &\qquad + \delta \cdot I_j^{(seed)}(g_j - y_j)\big) \\ h_k &\leftarrow h_k - \mu \cdot \beta \cdot W_{jk}^{(QL)}(h_k - g_j) \end{aligned} \tag{6}$$

where μ is the learning rate.

From the update formulas in Eq. 6, it can be seen that in each iteration the domain-specificity of each object is updated by taking into account domain-specificity of its associated objects in query log; in other words, domain-specificity of each object propagates along the tripartite graph during the optimization process. This implies that the optimization process is guided by the manifold structure on domain-specificity.

3.3 Active Learning Strategy

In the above domain-specificity learning problem, the seed queries $\mathbf{y}$ have a direct effect on the prediction accuracy. If noisy queries (i.e., queries irrelevant to the target domain) are included in the seed set, the mistake will propagate along the graph during the learning process, which has negative influence on recognition precision; whereas limited or unrepresentative seed set cannot guarantee the coverage of recognized domain-specific queries. In order to construct a high-quality seed query set with the least human annotation efforts, we introduce an active learning strategy into the domain-specificity learning process.

In recent years, many active graph-based transductive learning approaches have been proposed [8–11]. Most of the existing efforts in the literature aim to develop active learning algorithms for general graph data, irrespective of the characteristics of graphs in particular applications. In this work, instead of employing existing approaches, we propose a novel graph-based active learning algorithm, specially tailored for domain-specificity learning task.

The proposed active learning algorithm works in a batch mode: a number of the informative queries are selected and annotated in each round of active learning. The possible informative queries are divided into two types: informative domain-specific queries and informative domain-irrelevant queries, which are named as `DS-set` and `DI-set` respectively. Formally, given two informativeness criterion functions $\mathcal{I}^+ : 2^Q \rightarrow \mathbb{R}$ and $\mathcal{I}^- : 2^Q \rightarrow \mathbb{R}$, the aim is to identify a `DS-set` $Q^{(DS)} \subseteq Q$ and a `DI-set` $Q^{(DI)} \subseteq Q$ ($Q^{(DS)} \cap Q^{(DI)} = \emptyset$) such that $\mathcal{I}^+(Q^{(DS)})$ and $\mathcal{I}^-(Q^{(DI)})$ are maximized, respectively.

There are two keys to the active domain-specificity learning algorithm: (1) informativeness criterion and (2) query selection algorithm. We will give the details in the following subsections.

Informativeness criterion. In order to measure the informativeness brought by annotating domain-specificity of a set of queries, three factors: prediction reliability, redundancy and authority, of each query are taken into account.

Prediction reliability. Intuitively, domain-specificity prediction accuracy can be promoted if mistakenly predicted queries were corrected and added into seed set. The informativeness criterion thus should prefer the queries that are unreliably predicted. However, it is hard to evaluate prediction reliability due to the lack of ground truth of domain-specificity for the queries beyond seed set. In this work, we make an assumption that *Queries from the same domain statistically have higher lexical similarity than that from different domains.* Thus `DS-set` should prefer the queries has low lexical similarity with seed queries while predicted as domain-specific; whereas `DI-set` should prefer the queries has high lexical similarity with seed queries while predicted as domain-irrelevant.

Redundancy. With limited annotation budget, it is better to select diverse rather than redundant queries for domain-specificity judgement because undesirable redundant queries in `DS-set` and `DI-set` will lead to unnecessary repetitive annotation efforts.

Authority. As for graph-based transductive learning, each node in the graph generally has different levels of importance. A central node generally has more influence on other part of the graph than non-central ones, because the domain-specificity of a central node can be more easily propagated along the graph during the learning process. The informativeness criterion should thus prefer the queries with high authority in the tripartite graph.

Synthesizing the above three factors, the informativeness of a query set X selected as `DS-set` and `DI-set` is calculated as:

$$\mathcal{I}^+(X; S) = \sum_{q \in X} wt(q) \Big(\sum_{p \in S} \big(1 - sim(q,p)\big) - \eta \cdot \sum_{(o \in X) \wedge (o \neq q)} sim(q,o) \Big) \quad (7)$$

$$\mathcal{I}^{-}(X;S) = \sum_{q \in X} wt(q) \Big(\sum_{p \in S} sim(q,p) - \eta \cdot \sum_{(o \in X) \wedge (o \neq q)} sim(q,o) \Big) \tag{8}$$

where S is the current seed set, and $sim(\cdot,\cdot)$ is the lexical similarity between pair of queries. Following the prediction reliability assumption, $1 - sim(q,p)$ and $sim(q,p)$ are used as rough measures of the reliability of predicting p as domain-irrelevant or not. $sim(q,o)$ is used to measure the redundancy of the selected query set. $wt(q)$ is the function quantifying authority of each query q. We simply apply Google's PageRank on the tripartite graph and take the ranking score of each node as $wt(q)$. η is the parameter balancing the contributions of prediction reliability and redundancy in the informativeness criterion.

Query selection algorithm. Given seed set S, selecting `DS-set` and `DI-set` composed of k queries can be formulated as the following optimization problems:

$$Q^{(DS)} = \underset{(X \subseteq P^{+}) \wedge (|X| = k)}{\operatorname{argmax}} \; \mathcal{I}^{+}(X;S) \tag{9}$$

$$Q^{(DI)} = \underset{(X \subseteq P^{-}) \wedge (|X| = k)}{\operatorname{argmax}} \; \mathcal{I}^{-}(X;S) \tag{10}$$

where P^{+} and P^{-} are the query pools that consist of queries predicted as domain-specific and domain-irrelevant using current seed set S, respectively. Since the prediction of domain-specificity learning algorithm is a rank of all the queries according to the predicted domain-specificity w.r.t. the target domain, P^{+} and P^{-} can be practically constructed by fixed number of the queries in the top and rear of the rank list, respectively.

Essentially, the optimization problems in Eqs. 9 and 10 are knapsack packing problems and NP-hard in general. We develop a polynomial time greedy heuristic solution. The overall algorithm can be found in Algorithm 1. In what follows, for brevity we shall only consider the `DI-set` selection problem in Eq. 10. The same conclusions can be easily derived for the `DS-set` selection problem in Eq. 9.

Algorithm I. (Greedy Algorithm for DI-set Selection)

```
Input:  Seed set $S$;
        Query pool $P^{-}$, in which each queries are predicted as irrelevant
        to the target domain;
        The number of queries $k$ selected for annotation;
Output: The set of queries for annotation $Q^{(DI)}$, subject to $|Q^{(DI)}| = k$
Initialize $Q^{(DI)} \leftarrow \emptyset$
while $|Q^{(DI)}| \leq k$ do
    foreach $q \in P^{-} - Q^{(DI)}$ do
        $\mathcal{I}^{-}(q) = wt(q) \cdot \left( \sum_{p \in S} sim(q,p) - \gamma \sum_{q' \in Q^{(DI)}} sim(q,q') \right)$
    end
    $q^{*} = \operatorname{argmax}_{q \in P^{-} - Q^{(DI)}} \mathcal{I}^{-}(q)$
$Q^{(DI)} = Q^{(DI)} \cup \{q^{*}\}$
end
```

Simply speaking, Algorithm 1 iteratively selects the most informative query q^* out of the query pool $P^- - Q^{(DI)}$ and adds it into the result set $Q^{(DI)}$ (line 2–8). The computational complexity of the algorithm is $O(|P^-| \cdot |Q^{(DI)}|^2)$. Practically, the size of query pool and `DI-set` (usually varies from tens to thousands) are far less than the total number of queries in query log. Thus algorithm 1 can scale well to real-world query log with millions of objects.

Although Algorithm 1 is an approximate solution to the optimization problems in Eq. 10, it is guaranteed to have a fix error bound because of the submodular object function $\mathcal{I}^-(X; S)$. Due to space limitation, we only give the error bound of Algorithm1 in the following theorem:

Theorem I: *Let X be the query set selected using Algorithm I, and $Q^{(DI)}$ be the optimal solution of the problem in* Eq. 10, *then,*

$$\mathcal{I}^-(X; S) \geq \left(1 - \frac{1}{e}\right) \cdot \mathcal{I}^-(Q^{(DI)}; S) \tag{11}$$

4 Experiments

4.1 Experiment Settings

Query log. We performed experimental evaluation of the proposed approach using a publicly available query log of a Chinese commercial search engine[1]. In the experiment, we selected *Science:Agriculture* in the open web directory DMOZ[2] as the target domain. Before used for domain-specific query recognition, the query log was sampled by filtering out a number of queries that are obviously irrelevant with the domain of agriculture. Table 2 shows the statistics of the query log corpus used in the experiment.

Evaluation method. For a specified target domain, it is always practically hard to find a comprehensive set of queries specific to the domain that is qualified to evaluate the coverage of the recognized results. In the experiments, we thus mainly focus on evaluating domain-specific query recognition approaches in terms of precision. In particular, we evaluated the proposed approach and baselines in terms of Precision@n (or P@n for short). P@n measures the percentage of true domain-specific queries in the top-n recognized results. Given a

Table 2. Dataset statistics

Node type	Number
User	1,608,222
Query	3,971,977
URL	7,341,534

[1] http://www.sogou.com/labs/dl/q.html.

[2] https://www.dmoz.org/.

recognized domain-specific query list $r = \langle t_1, \cdots, t_n \rangle$, P@$n$ is calculated as:

$$\text{P@}n = \frac{1}{n} \cdot \sum_{i=1}^{n} I_i^{(specific)} \tag{12}$$

where $I_i^{(specific)}$ is an indicator function that equals to 1 if t_i is judged as specific to the target domain, and 0 otherwise.

In the experiment, correctness of recognized domain-specific queries are manually justified by two volunteers. Given a recognized query, the volunteers are asked to answer "yes" or "no" depending on their judgments on whether the query is relevant to the target domain – DMOZ *Science:Agriculture*. If they gave different answers, domain-specificity of the query would be finally determined by an agriculture expert, i.e., a master student majored in agriculture hired for the experiment. In all the following experiments, top-500 queries recognized by each methods were manually examined for evaluation.

4.2 Parameter Sensitivity Analysis

The parameters that control the contributions of different items in the learning objective, i.e., α, β, γ and δ in Eq. 3 are the main parameters of the proposed approach. We perform sensitivity analysis of these parameters. Firstly, we evaluated the impact of the fitness constraint in Eq. 3 on domain-specific query recognition, while keeping the relative contributions of user-side, URL-side and session smoothness constraints (the first three items in Eq. 3) as constant. In particular, we fixed $\alpha = \beta$ and $\gamma = 0$ empirically, and varied δ from 0 to 1 with the step of 0.1. Secondly, we evaluated the relative impact of user-side and URL-side smoothness constraints in Eq. 3, while keeping session smoothness constraint and fitness constraint as constant. In particular, we fixed $\delta = 0.2$ and $\gamma = 0$ empirically, and varied $\frac{\alpha}{\alpha+\beta}$ from 0 to 1 with the step of 0.1. Thirdly, we evaluated the impact of session smoothness constraint, while keeping relative impact of other constraints in Eq. 3 as constant. In particular, we fixed $\alpha = \beta = \delta$ empirically, and varied γ from 0 to 1 with the step of 0.1.

In each of the three experiments, we utilized a fixed seed set with 200 manual annotated domain-specific queries. For the sake of simplicity, we didn't introduce the proposed active learning strategy in the learning process. Tables 3, 4 and 5 show domain-specific queries recognition results in terms of P@n with different parameters.

From Table 3, it can be seen that domain-specific query recognition performance is quite stable across a wide range of δ (from 0.2 to 0.7). Besides, when δ is quite small ($\delta < 0.1$), the performance drops heavily. It indicates the importance of seed queries to graph-based transductive learning.

From Table 4, it can be seen that smaller α ($\frac{\alpha}{\alpha+\beta} < 0.5$) achieves better domain-specific query recognition performance in terms of P@n ($n \leq 50$), whereas the performance in terms of P@n ($n > 100$) drops heavily when $\frac{\alpha}{\alpha+\beta} < 0.3$. Besides, larger α ($0.8 \leq \frac{\alpha}{\alpha+\beta} \leq 0.9$) achieves relatively consistent

Table 3. Domain-specific query recognition results with varying δ

δ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
P@10	0	0.20	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
P@50	0	0.38	0.88	0.88	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	0.72
P@100	0.02	0.16	0.5	0.80	0.80	0.75	0.75	0.82	0.78	**0.90**	0.41
P@200	0.01	0.11	0.27	0.58	0.59	0.59	0.63	0.63	0.65	0.61	0.23
P@500	0.01	0.10	0.20	0.35	0.35	0.35	**0.38**	0.34	0.37	0.12	0.08

Table 4. Domain-specific query recognition results with varying $\frac{\alpha}{\alpha+\beta}$

$\frac{\alpha}{\alpha+\beta}$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
P@10	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.80	0.80	0.90	0.90	0.50
P@50	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	0.86	0.82	0.90	0.88	0.60
P@100	0.80	0.80	0.80	0.76	0.78	0.78	0.81	**0.83**	0.73	0.76	0.61
P@200	0.74	0.74	0.69	0.67	0.61	0.65	0.65	**0.76**	0.73	0.75	0.55
P@500	0.45	0.55	0.44	0.47	0.35	0.37	0.40	0.45	0.50	**0.54**	0.33

Table 5. Domain-specific query recognition results with varying γ

γ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
P@10	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.80	0.40	0.40	0.40	0	0
P@50	**0.92**	**0.92**	0.82	0.78	0.58	0.48	0.28	0.30	0.14	0.08	0
P@100	0.79	**0.87**	0.79	0.61	0.55	0.51	0.21	0.21	0.19	0.12	0
P@200	0.60	**0.67**	0.61	0.54	0.42	0.38	0.18	0.16	0.12	0.11	0
P@500	0.38	**0.43**	0.36	0.35	0.24	0.22	0.13	0.11	0.08	0.06	0

results over a large range of n. The probable reason is that for domain-specificity learning, manifold structure embedded in the relations between queries and URLs is more precious than that between queries and users, because the topic of a page generally focus on a few domains whereas a user may have a variety of interests involving a number of domains. On the other hand, the numbers of unique URLs is much larger than that of users in the query log (as shown in Table 2), making the relations between queries and URLs more sparse and helpless in discovering wider range of domain-specific queries.

From Table 5, it can be seen that the proposed approach performs not as well when $\gamma > 0.3$, so session smoothness constraint may be not so important as other constraints in optimizing the objective function in Eq. 3. However, when $\gamma = 0.1$ and 0.2, we can see improvements on P@n ($n = 100, 200, 500$) over that achieved when $\gamma = 0$. It indicates that session information still contributes in estimating domain-specificity of queries, especially when larger number of candidate queries are taken into account.

4.3 Effectiveness of Active Learning

We verified effectiveness of the proposed active learning algorithm for graph-based transductive learning. In this experiment, we started from an initial seed set consists of 50 domain-specific queries specified by an agriculture expert, and then repeatedly selected a batch of 30 queries for which the agriculture expert was inquired to annotate. Then the new queries were added into the seed set. In order to avoid the bias caused by single trial, the agriculture expert was asked to specify 200 agriculture queries as a query pool, and we conduct the active learning experiments five times, each of which was based on an initial seed set consists of 50 queries randomly sampled from the query pool. The average over the five trials was used for evaluation. Figure 2 shows the results of domain-specific query recognition based on active learning. We also show the result achieved by graph-based transductive learning without active learning strategy, which as labeled as "200 (fixed)".

We can see that domain-specific query recognition performance in terms of P@n continuously improves while new queries are selected and added into seed sets, especially when the seed set already constructed is of smaller size. We also see that the results based on actively selected 200 seed queries are roughly better than those based on 200 fixed seed queries. More accurately, the improvements are about 6.5%, 6.7% and 5.4% in terms of P@100, P@200 and P@500, respectively. This indicates that the proposed active learning algorithm is helpful in selecting informative seeds for graph-based transductive learning and thus improving the performance of domain-specific query recognition.

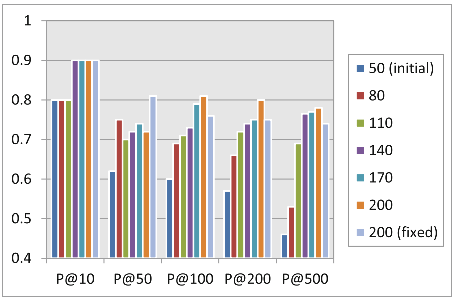

Fig. 2. Domain-specific query recognition results of active learning

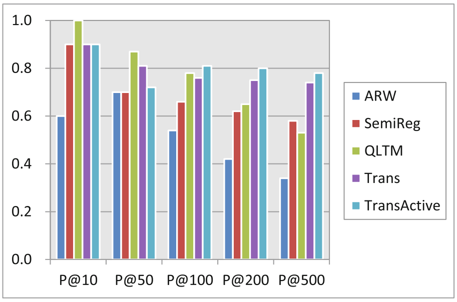

Fig. 3. Comparison of domain terminology extraction results

4.4 Comparison with Baseline Methods

To demonstrate the effectiveness of the proposed approach, three state-of-the-art methods exploiting inherent structure of query log for query classification, i.e., ARW [12], SemiReg [6] and QLTM [13], were chosen as baselines for comparison. All the baselines are outlined in related work section.

To make a fair comparison, all the methods made use of the pre-specified 200 agriculture queries as the only training data. We also investigated performance of the proposed approach without/with active learning strategy (abbr. Trans and TransActive, perspectively). The comparison of the proposed approach with baselines is presented in Fig. 3.

It can be seen that the proposed approach (Trans) outperforms all the baselines in term of P@n ($n = 100, 200, 500$) and the improvements are significant especially when n gets larger: about +24% from the best baselines (QLTM) in terms of P@500. Among all the methods, QLTM performs best when n is small ($n = 10, 50$); however, the performance drops when n gets larger, even worse than SemiReg when $n = 500$. QLTM is able to exploit multi-dimensional latent relations inherent in query log and actually works under supervised learning paradigm. Although supervised learning can leverage latent query features well, but it suffers from the facts that the pre-annotated queries are rather limited and the domain-specific queries are highly sparse. In this experiment, there are only 200 positive queries and the percentage of Agriculture-specific queries in the sampled query log amounts to only about 0.1%.

Besides, among all the graph-based semi-supervised learning methods (ARW, SemiReg, Trans and TransActive), TransActive performs best in term of all the evaluation measures. This provides strong evidence that active learning is capable of guiding the semi-supervised learning process, thus beneficial to the domain-specific query recognition task.

5 Related Work

Query classification, often referred to as search intent learning [6,13,18], query topic mining [14], search task learning [15] and etc., has been extensively studied in IR community for decades. From the learning techniques perspective, existing work on query classification can be put into three categories: supervised, unsupervised and semi-supervised.

It is natural to view query classification as supervised learning; however, it is also challenging to leverage supervised learning techniques in query classification as search query is often short and ambiguous. Therefore, one key to supervised query classification is to enrich feature representation of queries. Shen et al. used the retrieved pages as expansion of the query [4]. Lee et al. found the statistics of click distribution of a query is helpful to identify the underlying search goal [5].

When there is no query category predefined, query classification will turn out to be an unsupervised learning task. Clustering techniques have been extensive utilized in query classification. For example, Hu et al. conduct clustering on the clicked URLs of queries and took each URL cluster as a subtopic of a query [14]; Li et al. proposed a clustering framework with multiple kernel to identify synonymous query intent templates [16]; Qian et al. used incremental clustering method to group clicked URLs in log stream and took each URL group as constant or bursty query intents [17].

Recently, graph-based semi-supervised learning techniques have been extensively used in query classification tasks, especially for those using query log as the resource. Fuxman et al. [12] modeled click-through graph of query log as Markov Random Fields and employed absorbing random walks to compute probability of a query to belong to a pre-defined class. Li et al. [6] formulated query classification as semi-supervised learning on click graphs, with a content-based classifier regularization to avoid erroneous propagation. In order to enhance classification accuracy, the click-through graph of query log is expanded in many existing work. For example, Jiang et al. [13] proposed a query log topic model to derive latent relations between search queries, URL, session and term. Query classifiers are learned using latent relations and further combined using several strategies such as maximum confidence and majority voting.

Our work follows semi-supervised learning theme, but differs in the introduction of active learning strategy. To our best knowledge, no previous work leverages active learning in query classification problem. Moreover, multiple types of objects, i.e., users, queries, URLs and sessions, in query log are simultaneously integrated in query classification task in a more principled way through using tripartite graph representation.

6 Conclusion and Future Work

In this paper, we propose a novel approach to recognize domain-specific queries from general search engine's query log. There are mainly two advantages of the proposed approach. Firstly, the manifold structure inherent in query log is fully exploited through using heterogenous graph representation of query log. Secondly, a novel active learning strategy is introduced into graph-based transductive leaning process to reduce the human annotation efforts and continuously refine the recognition results. Experimental results on real-world query log demonstrate the effectiveness of the proposed approach. Future work includes evaluating the proposed approach on more target domains.

Acknowledgement. This work is partially supported by Chinese Natural Science Foundation (61602278), Shandong Province Higher Educational Science and Technology Program (J14LN33) and China Postdoctoral Science Foundation (2014M561949).

References

1. Arguello, J., Diaz, F., Callan, J., Crespo, J.F.: Sources of evidence for vertical selection. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 315–322 (2009)
2. Giachanou, A., Salampasis, M., Paltoglou, G.: Multilayer source selection as a tool for supporting patent search and classification. Inf. Retrieval J. **18**(6), 559–585 (2015)
3. Yan, X., Liu, Y., Fand, Q., Zhang, M., Ma, S., Ru, L.: Domain-specific terms extraction based on web resource and user behavior. J. Softw. (in Chinese) **24**(9), 2089–2100 (2013)
4. Shen, D., Pan, R., Sun, J.-T., Pan, J.J., Wu, K., Yin, J., Yang, Q.: Query enrichment for web-query classification. ACM Trans. Inf. Syst. **24**, 320–352 (2006)

5. Lee, U., Liu, Z., Cho, J.: Automatic identification of user goals in Web search. In: Proceedings of the 14th International Conference on World Wide Web, pp. 391–400 (2005)
6. Li, X., Wang, Y., Acero, A.: Learning query intent from regularized click graphs. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 339–346 (2008)
7. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. Adv. NIPS **16**(16), 321–328 (2004)
8. Zhu, X., Lafferty, J., Ghahramani, Z.: Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In: ICML 2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining (2003)
9. Gu, Q., Zhang, T., Han, J.: Batch-mode active learning via error bound minimization. In: Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence, pp. 300–309 (2014)
10. Shi, L., Zhao, Y., Tang, J.: Batch mode active learning for networked data. ACM Trans. Intell. Syst. Technol. **3**(2), 1–25 (2012)
11. Ji, M., Han, J.: A variance minimization criterion to active learning on graphs. In: Proceedings of the 15th International Conference on Artificial Intelligence and Statistics, pp. 556–564 (2012)
12. Fuxman, A., Tsaparas, P., Achan, K., Agrawal, R.: Using the wisdom of the crowds for keyword generation. In: Proceeding of the 17th International World Wide Web Conference, pp. 61–70 (2008)
13. Jiang, D., Leung, K.W.T., Ng, W.: Query intent mining with multiple dimensions of web search data. World Wide Web **19**(3), 475–497 (2016)
14. Hu, Y., Qian, Y., Li, H., Jiang, D., Pei, J., Zheng, Q.: Mining query subtopics from search log data. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 305–314 (2012)
15. Ji, M., Yan, J., Gu, S., Han, J., He, X., Zhang, W.V., Chen, Z.: Learning search tasks in queries and web pages via graph regularization. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 55–64 (2011)
16. Li, Y., Hsu, B.J.P., Zhai, C.: Unsupervised identification of synonymous query intent templates for attribute intents. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp. 2029–2038 (2013)
17. Qian, Y., Sakai, T., Ye, J., Zheng, Q., Li, C.: Dynamic query intent mining from a search log stream. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp. 1205–1208 (2013)
18. Ren, X., Wang, Y., Yu, X., Yan, J., Chen, Z., Han, J.: Heterogeneous graph-based intent learning with queries, web pages and Wikipedia concepts. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, pp. 23–32 (2014)

Springer

http://www.springer.com/978-3-319-63563-7

Web and Big Data
First International Joint Conference, APWeb-WAIM 2017,
Beijing, China, July 7–9, 2017, Proceedings, Part II
Chen, L.; Jensen, C.S.; Shahabi, C.; Bin, W.; Lian, X.
(Eds.)
2017, XXI, 362 p. 136 illus., Softcover
ISBN: 978-3-319-63563-7