Query Recommendation by Modelling the Query-Flow Graph

Lu Bai, Jiafeng Guo, and Xueqi Cheng

Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{bailu,guojiafeng}@software.ict.ac.cn,
cxq@ict.ac.cn

Abstract. Query recommendation has been widely applied in modern search engines to help users in their information seeking activities. Recently, the query-flow graph has shown its utility in query recommendation. However, there are two major problems in directly using query-flow graph for recommendation. On one hand, due to the sparsity of the graph, one may not well handle the recommendation for many dangling queries in the graph. On the other hand, without addressing the ambiguous intents in such an aggregated graph, one may generate recommendations either with multiple intents mixed together or dominated by certain intent. In this paper, we propose a novel mixture model that describes the generation of the query-flow graph. With this model, we can identify the hidden intents of queries from the graph. We then apply an intent-biased random walk over the graph for query recommendation. Empirical experiments are conducted based on real world query logs, and both the qualitative and quantitative results demonstrate the effectiveness of our approach.

1 Introduction

Nowadays, query recommendation has been recognized as an important tool that helps users seek their information needs. Many approaches have been proposed to generate query recommendations by leveraging query logs. Different types of information in the query logs have been taken into account, including search results (11), clickthrough (12) and search sessions (5).

Recently, the query-flow graph (2) has been introduced as a novel representation of session information in query logs. It integrates queries from different search sessions into a directed and homogeneous graph. Nodes of the graph represent unique queries, and two queries are connected by a directed edge if they occur consecutively in a search session. The Query-flow graph has shown its utility in query recommendation (2–4).

However, there are several problems in directly using the query-flow graph for recommendation as in existing approaches. Firstly, due to the information sparsity, lots of dangling queries which have no out-links exist in the query-flow graph[1]. Therefore, recommendation approaches based on random walks (2, 3) over the directed graph may not well handle such dangling queries. Moreover, queries are often ambiguous in their

[1] In our experiment, we observe that the dangling queries account for nearly 9% of the total queries, which is not negligible in real application.

M.V.M. Salem et al. (Eds.): AIRS 2011, LNCS 7097, pp. 137–146, 2011.

search intent and thus the aggregated query-flow graph in fact is a mixture of multiple search intents. Most existing approaches (2–4) do not take into account the ambiguous intents in the query-flow graph when generating recommendations. Therefore, for ambiguous queries, one may either produce recommendations with multiple intents mixed together which are difficult for users to consume, or provide recommendations dominated by certain intent which cannot satisfy different user needs.

In this paper, we propose to model the query-flow graph for better query recommendation. Specifically, we introduce a novel mixture model for the query-flow graph. The model employs a probabilistic approach to interpret the generation of the graph, i.e., how the queries and the transitions between queries are generated under the hidden search intents. We then apply an intent-biased random walk over the graph for query recommendation. In this way, we can well resolve the recommendation problems for dangling queries and ambiguous queries in using query-flow graph.

We conducted empirical experiments based on a collection of query logs from a commercial search engine. Both the qualitative and quantitative results demonstrate the effectiveness of our approach as compared with existing baseline methods.

2 Related Work

2.1 Query Recommendation

Query recommendation is a widely accepted tool employed by search engines to help users express and explore their information needs. Beeferman et al. (1) applied agglomerative clustering algorithm over the clickthrough bipartite graph to identify related queries for recommendation. Ma et al. (7) developed a two-level query recommendation method based on both the user-query graph and the query-URL graph. Zhu et al. (13) generated diverse query recommendations based on the query manifold structure.

Recently, query-flow graph was introduced by Boldi et al. (2), and they applied personalized random walk (2, 3) over the query-flow graph to recommend queries. Unlike previous work on query-flow graph, our approach explores the query-flow graph with a mixture model for query recommendation, so that we can well resolve the recommendation problems for dangling queries and ambiguous queries in using query-flow graph.

2.2 Mixture Models

Recently, there have been different mixture models applied on graphs for community discovery. For example, Newman et al. (8) proposed a probabilistic mixture model to discover the overlapped communities in graph. Ramasco et al. (9) introduced a more general mixture model on graph for the same purpose. Ren et al. (10) described a mixture model for undirected graph, where each edge in the graph is assumed to be from the same community. Inspired by the above work, we propose a novel mixture model to interpret the generation of the query-flow graph under multiple hidden intents.

3 Our Approach

In this section, we first briefly introduce the query-flow graph. We then describe the proposed mixture model in detail, which learns the hidden intents of queries by modelling

the generation of the query-flow graph. Finally, we show how to leverage the learned intents for better query recommendation with an intent-biased random walk.

3.1 Query-Flow Graph

The query-flow graph integrates queries from different search sessions into a directed and homogeneous graph. Formally, we denote a query-flow graph as $G = (V, E, w)$, where $V = Q \cup \{s, t\}$ is the set of unique queries Q in query logs plus two special nodes s and t, representing a starting state and a terminal state of any user search session. $E \subseteq V \times V$ denotes the set of directed edges, where two queries q_i and q_j are connected by an edge if there is at least one session of the query log in which q_j follows q_i. w is a weighting function that assigns to every pair of queries $(q_i, q_j) \in E$ a weight $w_{\overrightarrow{ij}}$. The definition of the weight w may depend on the specific applications. In our work, we simply consider the weight to be the frequency of the transition in the query log. Fig. 1 shows a query-flow graph that is constructed from the a set of search sessions.

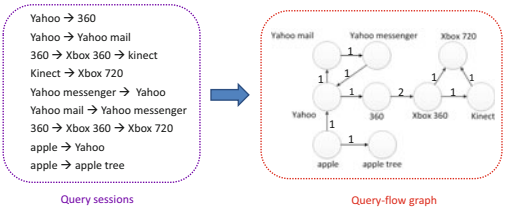

Fig. 1. A simple query-flow graph constructed from query sessions

3.2 Mixture Model on Query-Flow graph

We propose a novel mixture model to interpret the generation of the query-flow graph. In essentials, our model is based on the following assumption: Queries are generated from some hidden search intents, and two queries occurred consecutively in one session if they are from the same search intent. The above assumption is quite natural and straightforward. Typically, users submit a query to search according to their potential information needs (i.e., search intent). Users may consecutively reformulate their queries in a search session until their original needs are fulfilled (or exit with a failure). Therefore, without loss of generality, queries occurred consecutively in a search session can be viewed as under the same search intent.

Specifically, given a query-flow graph G which consists of N nodes and M directed edges, we assume the graph G is generated under K potential search intents, where each intent is characterized by a distribution over queries. Let $e_{\overrightarrow{ij}} \in E$ denote a directed edge from query q_i to query q_j. We then assume the following generative process for each edge $e_{\overrightarrow{ij}}$ in the query-flow graph:

1. Draw an intent indicator $g_{\overrightarrow{ij}} = r$ from the multinomial distribution π.
2. Draw query nodes q_i, q_j from the same multinomial intent distribution β_r, respectively.

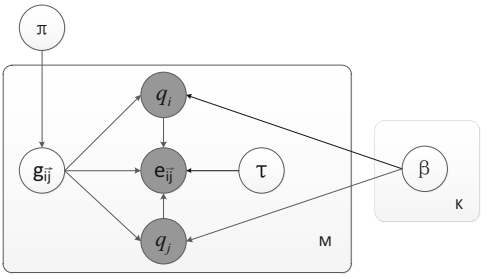

Fig. 2. The graphic model of generation of query-flow graph

3. Draw the directed edge $e_{\vec{ij}}$ from a binomial distribution $\tau_{\vec{ij},r}$ under a hidden intent $g_{ij} = r$.

Here, the K-dimensional multinomial distribution π reflects the proportion of different search intents over the whole query-flow graph, the multinomial distribution β over queries describes the hidden search intents, and the binomial distribution τ captures the probability of the edge direction between two queries under a given search intent. The Fig. 2 shows the graphic model of generating query flow graph.

Based on the above process, the probability of an observed directed edge $e_{\vec{ij}}$ belonging to the r-th search intent can be obtained by

$$\Pr(e_{\vec{ij}}, g_{\vec{ij}} = r|\pi, \beta, \tau) = \pi_r \beta_{r,i} \beta_{r,j} \tau_{\vec{ij},r}$$

In this way, the likelihood of the whole query-flow graph G is

$$\Pr(G|\pi, \beta, \tau) = \prod_{i=1}^{N} \prod_{j:j\in C(i)} \left(\sum_{r=1}^{K} \pi_r \beta_{r,i} \beta_{r,j} \tau_{\vec{ij},r} \right)^{w_{\vec{ij}}} \tag{1}$$

where $w_{\vec{ij}}$ denotes the weight of edge $e_{\vec{ij}}$, and $C(i)$ denotes the set of nodes pointed by query q_i.

The parameters to be estimated in our model are π, β, and τ. We maximize the likelihood shown in Equation (1) to estimate these parameters. The sum in the bracket makes the direct estimation difficult, but with the help of Expectation Maximization (EM) algorithm the problem can be solved easily.

As we can see, the hidden variables in our mixture model are intent indicators $g_{\vec{ij}}$. In E-step, the posterior probabilities of hidden variables are calculated as

$$q_{\vec{ij},r} = \Pr(g_{\vec{ij}} = r|e_{\vec{ij}}) = \frac{\Pr(e_{\vec{ij}}, g_{\vec{ij}} = r)}{\Pr(e_{\vec{ij}})} = \frac{\pi_r \beta_{r,i} \beta_{r,j} \tau_{\vec{ij},r}}{\sum_{r=1}^{K} \pi_r \beta_{r,i} \beta_{r,j} \tau_{\vec{ij},r}}$$

In M-step, we update the parameters by the following formulas:

$$\pi_r = \frac{\sum_{i=1}^{N} \sum_{j:j\in C(i)} w_{\vec{ij}} q_{\vec{ij},r}}{\sum_{r=1}^{K} \sum_{i=1}^{N} \sum_{j:j\in C(i)} w_{\vec{ij}} q_{\vec{ij},r}}, \quad \tau_{\vec{ij},r} = \frac{w_{\vec{ij}} q_{\vec{ij},r}}{w_{\vec{ij}} q_{\vec{ij},r} + w_{\vec{ji}} q_{\vec{ji},r}}$$

$$\beta_{r,i} = \frac{\sum_{j:j\in C(i)} w_{ij}^{\rightarrow} q_{ij,r}^{\rightarrow} + \sum_{k:i\in C(k)} w_{ki}^{\rightarrow} q_{ki,r}^{\rightarrow}}{\sum_{i=1}^{N} \left(\sum_{j:j\in C(i)} w_{ij}^{\rightarrow} q_{ij,r}^{\rightarrow} + \sum_{k:i\in C(k)} w_{ki}^{\rightarrow} q_{ki,r}^{\rightarrow} \right)}$$

The E-step and M-step are repeated alternatively until the log-likelihood does not increase significantly. Note that the EM algorithm will not necessarily find the global optimal. We resolve this by trying several different starting points to get an good solution in practice.

3.3 Intent-Biased Random Walk

As aforementioned, directly applying the traditional personalized random walk on query-flow graph into recommendation may not well handle the dangling queries and ambiguous queries. Here we further introduce our intent-biased random walk to recommend queries based on the learned intents above. The basic idea of our model is to integrate the learned intents of queries into the prior preference of the personalized random walk, and apply the random walk under different search intent respectively.

Formally, let W denote the weight matrix of the query-flow graph G with row normalized. An intent-biased random walk over the query-flow graph G under the r-th search intent given the original query q_i is then determined by the following transition probability matrix $A_{i,r}$, which is defined as following:

$$A_{i,r} = (1 - \lambda)W + \lambda \mathbf{1} P_{i,r}$$

where λ denotes the teleportation probability, and $P_{i,r}$ denotes the preference vector of intent-bias random walk under the r-th intent defined as

$$P_{i,r} = \rho \cdot \mathbf{e}_i^T + (1 - \rho) \cdot \beta_r$$

where $\mathbf{e}_i^T$ is the vector whose entries are all zeroes, except for the i-th whose value is 1, β_r is our learned r-th intent distribution over queries, and $\rho \in [0, 1]$ is the weight balancing the original query and its intent.

The intent-biased random walk has a unique stationary distribution $R_{i,r}$ such that $R_{i,r} = A_{i,r}^T R_{i,r}$ (called the personalized PageRank score relative to q_i under the r-th intent). Such a personalized PageRank can be computed using the power iteration method. We can then employ the personalized PageRank score to rank queries with respect to q_i for recommendation.

We apply our intent-biased random walk under each intent of query q_i, and obtain the corresponding recommendations. Finally, the recommendations are grouped by intent and represented to users in a structured way, where the intent groups are ranked according to the intent proportion of the given query q_i calculated by

$$\Pr(r|i) \propto \Pr(r) \Pr(i|r) = \pi_r \beta_{r,i}$$

As we can see, if we set the parameter ρ to 1, our intent-biased random walk will degenerate into the traditional personalized random walk as applied in previous work (2, 4). Obviously, under such a personalized random walk, we may not obtain any recommendations for dangling queries. To avoid non-recommendation, one may add a

small uniform vector to the preference vector (i.e., teleportation to every node). However, in that case, the recommendations for the dangling query will be mostly popular queries which may not related to original query at all. While in our model, we set ρ less than 1 so that we can smooth the preference vector with the learned intents, which can provide rich background information of the original query in a back-off way. If the original query is a dangling query, the preference vector will be reduced to the corresponding intent distribution so that we can still obtain related recommendations for the original query.

Moreover, previous approaches usually applied the personalized random walk on the graph for recommendation without addressing the hidden intents. In this way, for ambiguous queries, they may either produce recommendations with multiple intents mixed together which are difficult for users to consume, or provide recommendations dominated by certain intent which cannot satisfy different user needs. In our model, we can naturally generated recommendations for the original query with respect to its different intents. The structured recommendation results would be easy to understand and diverse search intents can be covered.

4 Experiments

4.1 Data Set

The experiments are conducted on a 3-month query log from a commercial search engine. We split the query stream into query sessions using 30 minutes timeout, and construct the query-flow graph as section 3.1 described. To decrease the noise in search sessions, we get rid of those edges with frequencies lower than 3. We then draw the biggest connected component of the graph for experiment. After these steps, the obtained graph consists of 16,980 distinct queries and 51,214 distinct edges.

4.2 Evaluation of Intents

Fig. 3 shows how the log likelihood varies over iterations under different number of hidden intents. According to Fig. 3, the increase of log likelihood turns slow when the

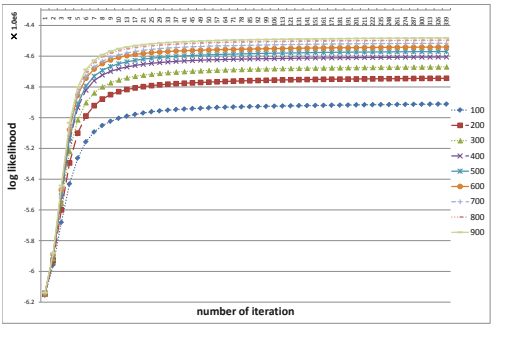

Fig. 3. Log-likelihood over iterations under different mixture Numbers

Table 1. Top 10 queries for 3 randomly sampled learned intents

lyrics	cars	poems
lyrics	bmw	poems
song lyrics	lexus	love poems
lyrics com	audi	poetry
a z lyrics	toyota	friendship poems
music lyrics	acura	famous love poems
azlyrics	nissan	love quotes
lyric	infiniti	sad poems
az lyrics	mercedes benz	quotes
rap lyrics	volvo	mother s day poems
country lyrics	mercedes	mothers day poems

Table 2. Recommendations for dangling queries

Query = "yamaha motor"		Query = "radio disney"	
baseline	ours	baseline	ours
mapquest	yamaha	mapquest	disney
american idol	honda	american idol	disney channel
yahoo mail	suzuki	yahoo mail	disney com
home depot	kawasaki	home depot	disneychannel com
bank of america	yamaha motorcycles	bank of america	disney channel com
target	yamaha motorcycle	target	disneychannel

intent number is larger than 600. It indicates that the mixture model with 600 hidden dimensions is basically sufficient to capture the potential search intents over this graph. Larger number of intents are very probably to be redundant and may cause the problem of over-fitting. Therefore, we set the intent number to 600 in our experiments.

We randomly sample 3 learned intents to demonstrate the effectiveness of our mixture model, as shown in Table 1. For each intent, we list the top 10 ranked queries according to their probabilities under the corresponding intent. The first, second, and third columns are about lyrics, cars, and poems, respectively. The labels of each intent are created by human judge for illustration.

4.3 Evaluation of Query Recommendation

In this part, we evaluate the recommendation performance of our approach by comparing with traditional personalized random walk (2). For our intent-biased random walk, the parameter λ is set to 0.8, and ρ is set to 0.3.

Qualitative Comparison. We take the randomly selected dangling queries "yamaha motor" and "radio disney" as examples to demonstrate the effectiveness of our approach. The recommendation results from our approach and baseline method are demonstrated in the Table 2 . We can see the recommendations from our methods are much more related to the initial queries . On the contrary, the recommendations from baseline method are mostly queries that are popular in the whole date set but unrelated to the original queries. This is because for the dangling queries, the traditional random walk based approaches can only find recommendations with the help of the uniform teleport.

Table 3. Recommendations for Ambiguous Queries

Query = "hilton"		Query = "we"	
baseline	ours	baseline	ours
marriott		wwe	
expedia	marriott	wells fargo	wells fargo
holiday inn	holiday inn	weather	bank of america
hyatt	sheraton	wellsfargo com	wellsfargo
hotel	hampton inn	we channel	wamu
mapquest	embassy suites	tna	
hampton inn	hotels com	bank of america	weather
sheraton		yahoo mail	weather channel
hilton com	paris hilton	weather channel	accuweather
hotels com	michelle wie	wellsfargo	noaa
embassy suites	nicole richie	espn	
residence inn	jessica simpson	usbank com	wwe
choice hotels	pamela anderson	wwe com	tna
marriot	daniel dipiero	www wellsfargo com	wrestleview
hilton honors	richard hatch	bankofamerica com	ecw

We also compared our approach with the baseline method on ambiguous queries. We randomly selected two queries with multiple hidden search intents based on our learned model as shown in the Table 3. We can see that structured query recommendations can be provided by our approach for ambiguous queries. Take the query "we" as an example, the top three categories of recommendations provided by our approach correspond to "financial", "weather" and "wrestling", respectively. The labels here are also human annotated for illustration. However, the baseline method only produces one recommendation list which is a mixture of several intents. Query "hilton" is another interesting example with multiple intents. In this case, the recommendations generated by the baseline method are dominated by queries related to the hotel. In contrast, our approach can obtain two categories of recommendations, one about the hotel and the other about the celebrity. Therefore, our approach may better satisfy users' needs by covering diverse intents of the query.

Quantitative Comparison. Furthermore, we conducted quantitative experiments for evaluating the performance of our recommendation approach. As aforementioned, our proposed approach naturally provides *structured* query recommendation to users . While the baseline method using personalized random walk provides the traditional *list-based* query recommendations to users. We thus follow the way proposed in (6) to compare the performances of different recommendation methods by users' click behaviour.

We randomly sampled 100 queries as our test set. For each query, top 15 recommendations are used for performance comparison. For each recommendation, human judges are required to label how likely he/she would like to click it with a 6-point scale (0, 0.2, 0.4, 0.6, 0.8, 1) as the willingness measure. 3 human judges were asked to participate the labelling process .

We also adopted the *Clicked Recommendation Number (CRN)*, *Clicked Recommendation Score(CRS)*, and *Total Recommendation Score(TRS)* as our evaluation measures. For each query q, let $R = \{r_1, \ldots, r_k\}$ denote the k recommendations generated by a certain method, and $L = \{l_1, \ldots, l_k\}$ denote the corresponding label scores on these

Table 4. Comparisons between Our Approach and Baseline Approach

	Baseline	Ours
Average CRN	4.09	4.21(+2.9%)
Average CRS	0.598	0.652(+9.0%)
Average TRS	0.181	0.194(+7.1%)

recommendations, where k is the size of recommendations. The three measures for query q are then defined as follows

$$CRN_q = |\{r_i | l_i > 0, i \in [1,k]\}|, \quad CRS_q = \frac{\sum_{i=1}^{k} l_i}{CRN_q}, \quad TRS_q = \frac{\sum_{i=1} l_i}{k}$$

where $|*|$ denote the set size. As we can see, CRN reflects the adoption frequency of query recommendations, CRS shows the preference on adopted recommendations, and TRS indicates the effectiveness of overall query recommendations.

Table 4 shows the quantitative evaluation results of the two approaches. The numbers in the parentheses are the relative improvements of our approach over the baseline method. The results show that by providing structured query recommendations based on our intent-biased random walk, we can largely improve both the click number and click willingness on recommendations.

5 Conclusions

In this paper, we propose to explore the query-flow graph for better query recommendation. Unlike previous methods, our novel mixture model identifies the hidden search intents from the query-flow graph. An intent-biased random walk is then introduced to integrate the learned intents for recommendation. Experimental results show the effectiveness of our approach. For the future work, it would be interesting to try to combine the query words, search session and clickthrough information in a unified model to help generate better query recommendations.

Acknowledgments. This research work was funded by the National High-tech R&D Program of China under grant No. 2010*AA*012500, and the National Natural Science Foundation of China under Grant No. 61003166 and Grant No. 60933005.

References

1. Beeferman, D., Berger, A.: Agglomerative clustering of a search engine query log. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2000, pp. 407–416. ACM, New York (2000), http://doi.acm.org/10.1145/347090.347176
2. Boldi, P., Bonchi, F., Castillo, C., Donato, D., Gionis, A., Vigna, S.: The query-flow graph: model and applications. In: Proceeding of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, pp. 609–618. ACM, New York (2008), http://doi.acm.org/10.1145/1458082.1458163

3. Boldi, P., Bonchi, F., Castillo, C., Donato, D., Vigna, S.: Query suggestions using query-flow graphs. In: Proceedings of the 2009 Workshop on Web Search Click Data, WSCD 2009, pp. 56–63. ACM, New York (2009), http://doi.acm.org/10.1145/1507509.1507518
4. Bordino, I., Castillo, C., Donato, D., Gionis, A.: Query similarity by projecting the query-flow graph. In: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, pp. 515–522. ACM, New York (2010), http://doi.acm.org/10.1145/1835449.1835536
5. Cucerzan, S., White, R.W.: Query suggestion based on user landing pages. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2007, pp. 875–876. ACM, New York (2007), http://doi.acm.org/10.1145/1277741.1277953
6. Guo, J., Cheng, X., Xu, G., Shen, H.: A structured approach to query recommendation with social annotation data. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM 2010, pp. 619–628. ACM, New York (2010), http://doi.acm.org/10.1145/1871437.1871518
7. Ma, H., Yang, H., King, I., Lyu, M.R.: Learning latent semantic relations from clickthrough data for query suggestion. In: Proceeding of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, pp. 709–718. ACM, New York (2008), http://doi.acm.org/10.1145/1458082.1458177
8. Newman, M.E.J., Leicht, E.A.: Mixture models and exploratory analysis in networks. Proc. Natl. Acad. Sci. USA 104, 9564 (2007), doi:10.1073/pnas.0610537104
9. Ramasco, J.J., Mungan, M.: Inversion method for content-based networks. Phys. Rev. E 77(3), 036122 (2008)
10. Ren, W., Yan, G., Liao, X., Xiao, L.: Simple probabilistic algorithm for detecting community structure. Phys. Rev. E 79, 036111 (2009), http://link.aps.org/doi/10.1103/PhysRevE.79.036111
11. Sahami, M., Heilman, T.D.: A web-based kernel function for measuring the similarity of short text snippets. In: Proceedings of the 15th International Conference on World Wide Web, WWW 2006, pp. 377–386. ACM, New York (2006), http://doi.acm.org/10.1145/1135777.1135834
12. Yi, J., Maghoul, F.: Query clustering using click-through graph. In: Proceedings of the 18th International Conference on World Wide Web, WWW 2009, pp. 1055–1056. ACM, New York (2009), http://doi.acm.org/10.1145/1526709.1526853
13. Zhu, X., Guo, J., Cheng, X., Du, P., Shen, H.W.: A unified framework for recommending diverse and relevant queries. In: Proceedings of the 20th International Conference on World Wide Web, WWW 2011, pp. 37–46. ACM, New York (2011), http://doi.acm.org/10.1145/1963405.1963415