Exploring the Query-Flow Graph with a Mixture Model for Query Recommendation

Lu Bai Jiafeng Guo Xueqi Cheng Xiubo Geng Pan Du
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{bailu, guojiafeng, gengxiubo, dupan}@software.ict.ac.cn, cxq@ict.ac.cn

ABSTRACT

Query recommendation has been recognized as an important tool that helps users in their information seeking activities. Many existing approaches leveraged the rich Web query logs to generate query recommendations. Recently, the query-flow graph, an aggregated representation of session information in query logs, 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 which are difficult to consume, or dominated by certain intent which cannot satisfy different user needs. In this paper, we propose to explore the query-flow graph with a mixture model for better query recommendation. Specifically, 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. In this way, we can well resolve the above two problems. Some primary experiments on real query logs show the effectiveness of our approaches as compared with baseline methods.

1. INTRODUCTION

Nowadays, query recommendation has been employed by most modern search engines as an important tool for helping users seek their information needs. Many approaches have been proposed to generate query recommendations by leveraging Web query logs, a rich resource recording the interactions between users and search engines. Different types of information in the query logs have been taken into account, including search results, clickthrough and search sessions.

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, 3, 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 search intent and thus the aggregated query-flow graph in fact is a mixture of multiple search intents. Most existing approaches [2, 3, 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 explore the query-flow graph with a mixture model 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. With this model, we can identify massive hidden intents of queries from the graph. We then apply an intent-biased personalized 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. For dangling queries, our random walk leverages the learned intents of queries as the prior distribution, so that it can find related recommendations even though the original query has no out-links. For ambiguous queries, the recommendation results under our model can naturally be clustered by intents and shown in a structured way. Therefore, recommendations from different intents are clearly separated for better understanding, and those from minor intents will also be covered to satisfy diverse user needs. Empirical experiments are conducted on a commercial query log, and the primary results show that our approach is promising.

2. RELATED WORK

Query recommendation is a widely accepted tool employed by search engines to help users express and explore their information needs. There have been extensive studies for query recommendation. For example, Mei et al. [6] proposed to use hitting time to recommend queries based on the clickthrough graph. Zhu et al. [10] 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. In

SIGIR Workshop on Query Representation and Understanding '11, July 24, 2011, Beijing, China

[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.

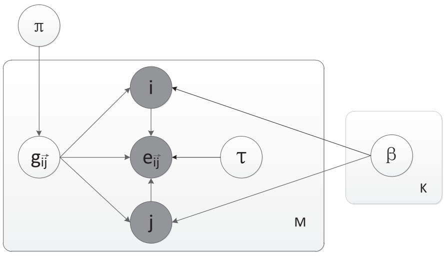

Figure 1: The graphic model of generation of query-flow graph

their later work [4], they projected the query-flow graph to low dimension Euclidean space through spectral projection, and then recommended nearby queries to the original one by calculating the similarity in this space. 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.

Mixture Models have shown great success in lots of domains including topic discovery, collaborative filtering, document classification and social network analysis. Two well-known mixture models are PLSA [5] and LDA [1], which have been proposed to model hidden topics of documents. Recently, there have been different mixture models applied on graphs for community discovery. For example, Newman et al. [7] proposed a probabilistic mixture model to discover the overlapped communities in graph. Ramasco et al. [8] introduced a more general mixture model on graph for the same purpose. Ren et al. [9] 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 modeling 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 is a novel representation of session information in query logs. It 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 $w_{\vec{ij}}$. 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.

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. 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 $e_{\vec{ij}} \in E$ denote a directed edge from query q_i to query q_j. We assume the following generative process for each edge $e_{\vec{ij}}$ in the query-flow graph:

1. Draw an intent indicator $g_{\vec{ij}} = r$ from the multinomial distribution π

2. Draw query nodes q_i, q_j from the same multinomial intent distribution β_r, respectively.

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

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.

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

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

By integrating over the search intents $g_{\vec{ij}}$, we can obtain the probability of a directed edge $e_{\vec{ij}}$ as follows

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

In this way, the likelihood of the graph G is

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

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 fact, $q_{ij,r}$ is the fraction of contribution from r-th search intent to the edge $e_{\vec{ij}}$'s generation.

Obviously, the expected log-likelihood of whole query-flow graph can be written as:

$$L = \sum_{i=1}^{N} \sum_{j:j\in C(i)} \sum_{r=1}^{K} w_{\vec{ij}} q_{\vec{ij},r} \log \left(\pi_r \beta_{r,i} \beta_{r,j} \tau_{\vec{ij},r} \right)$$

In M-step, we maximize the expected complete data log-likelihood which is

$$LL = \sum_{i=1}^{N}\sum_{j:j\in C(i)}\sum_{r=1}^{K} w_{\vec{ij}}q_{\vec{ij},r}\log\left(\pi_r\beta_{r,i}\beta_{r,j}\tau_{\vec{ij},r}\right) - \alpha(\sum_{r=1}^{K}\pi_r - 1)$$
$$- \sum_{r=1}^{K}\mu_r(\sum_{i=1}^{N}\beta_{r,i} - 1) - \sum_{i=1}^{N}\sum_{j:j\in C(i)}\sum_{r=1}^{N}\eta_{\vec{ij},r}(\tau_{\vec{ij},r} + \tau_{\vec{ji},r} - 1)$$

where for α, μ, η are lagrange multipliers. Taking the derivative with respect to π, β, τ respectively gives the M-step re-estimations as follows

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

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

$$\tau_{\vec{ij},r} = \frac{w_{\vec{ij}}q_{\vec{ij},r}}{w_{\vec{ij}}q_{\vec{ij},r} + w_{\vec{ji}}q_{\vec{ji},r}}$$

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

Given an original query and the query-flow graph, it is naturally led to apply a personalized random walk for query recommendation as in [2, 4]. As aforementioned, however, directly applying the traditional personalized random walk on query-flow graph may not well handle the dangling queries and ambiguous queries in recommendation. 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, define an intent-biased random walk over the query-flow graph G under the r-th search intent given the original query q_i determined by the following transition probability matrix

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

where M denotes the weight matrix of the query-flow graph with row normalized, 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, and then employed 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

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

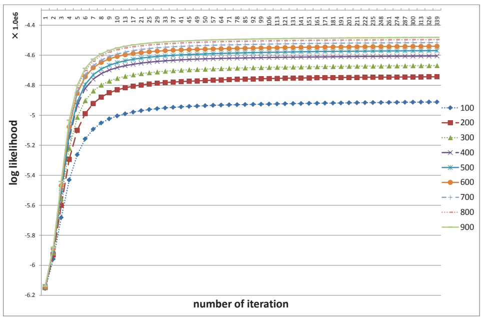

Figure 2: Log-Likelihood over Iterations under Different Mixture Numbers

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 this case, the recommendations for the dangling query will be mostly popular queries which may not related to original query. 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 three-month query log from a commercial search engine. After taking the non-English queries out, we convert the remaining queries into lower case and replace the non-alphanumeric character with white space. We split the query stream into query sessions using 30 minutes timeout, and construct the query-flow graph as described previously. 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 result graph consists of 16,980 distinct queries and 51,214 distinct edges.

4.2 Evaluation of Intents

In this section, we first show the learning performance of the proposed mixture model. Fig. 2 shows how the likelihood varies over iterations under different number of hidden intents. From the result, we can see the increase of likelihood turns slow when the 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

Table 1: Top Queries under Randomly Sampled 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

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 intent. We can see that the learned hidden intents can reveal very meaningful searching needs. The first column is about lyrics, the second is about cars, and the last is about poems. 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 an existing approach using traditional personalized random walk. For our intent-biased random walk, the parameter λ is set to 0.8, and ρ is set to 0.3.

Here we first 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, where 6 top ranked recommended queries are listed for each method. We can see the recommendations from our methods are much more related to the initial queries with intent biased. 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.

We further 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, which is very difficult for us to read. 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.

Table 3: Recommendations for Ambiguous Queries

Query = hilton		Query = we	
baseline	ours	baseline	ours
marriott	[hotel]	wwe	[finnacial]
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	[weather]
hampton inn	hotels com	bank of america	weather
sheraton	[celebrity]	yahoo mail	weather channel
hilton com	paris hilton	weather channel	accuweather
hotels com	michelle wie	wellsfargo	noaa
embassy suites	nicole richie	espn	[wrestling]
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

5. CONCLUSIONS

In this paper, we propose to explore the query-flow graph with a novel probabilistic mixture model for better query recommendation. Unlike previous methods, our model identifies the hidden search intents from the query-flow graph. A intent-biased random walk is then introduced to integrate the learned intents for recommendation. Experiment result shows the effectiveness of our approach. For the future work, we will conduct more detailed experiments to compare our model with other methods qualitatively and quantitatively.

6. ACKNOWLEDGMENTS

This research work was funded by the National Natural Science Foundation of China under Grant No. 61003166 and Grant No. 60933005.

References

[1] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003, 2003.

[2] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: model and applications. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 609–618, New York, NY, USA, 2008. ACM.

[3] 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*, WSCD '09, pages 56–63, New York, NY, USA, 2009. ACM.

[4] I. Bordino, C. Castillo, D. Donato, and A. Gionis. 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 '10, pages 515–522, New York, NY, USA, 2010. ACM.

[5] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. In *Machine Learning*, page 2001, 2001.

[6] Q. Mei, D. Zhou, and K. Church. Query suggestion using hitting time. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 469–478, New York, NY, USA, 2008. ACM.

[7] M. E. J. Newman and E. A. Leicht. Mixture models and exploratory analysis in networks. *PROC.NATL.ACAD.SCI.USA*, 104:9564, 2007.

[8] J. J. Ramasco and M. Mungan. Inversion method for content-based networks. *Phys. Rev. E*, 77(3):036122, Mar 2008.

[9] W. Ren, G. Yan, and X. Liao. A simple probabilistic algorithm for detecting community structure in social networks. 2007.

[10] X. Zhu, J. Guo, X. Cheng, P. Du, and H.-W. Shen. A unified framework for recommending diverse and relevant queries. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 37–46, New York, NY, USA, 2011. ACM.