Query Representation and Understanding

July 28, 2011, Beijing, China

Organizers

Hang Li (Microsoft Research Asia)
Gu Xu (Microsoft Bing)
Bruce Croft (University of Massachusetts Amherst)
Michael Bendersky (University of Massachusetts Amherst)
Understanding the user's intent or information need that underlies a query has long been recognized as a crucial part of effective information retrieval. Despite this, retrieval models, in general, have not focused on explicitly representing intent, and query processing has been limited to simple transformations such as stemming or spelling correction. With the recent availability of large amounts of data about user behavior and queries in web search logs, there has been an upsurge in interest in new approaches to query understanding and representing intent.

This is the second workshop on query representation and understanding at SIGIR. The first workshop was held at SIGIR 2010. These workshops have the goal of bringing together the different strands of research on query understanding, increasing the dialogue between researchers working in this relatively new area, and developing some common themes and directions, including definitions of tasks, evaluation methodology, and reusable data collections.

This year, the workshop will include invited talks, poster session and panel discussions. Three invited talks by both academic and industrial researchers will provide the audience with an overview of three broad research areas that are relevant to workshop activities. The poster session will include five short oral presentations, as well as posters, and provide time for the authors to exchange views and ideas with the workshop participants. In the panel discussion, panelists will report some findings based on initial studies on the development of data collections conducted before the workshop. Discussions will be made from the panelists, as well as from workshop participants. The goal of the discussions is to promote a creation of a data collection that will enable competition and innovation in the area of query representation and understanding.

We would like to thank the program committee members for the great efforts in reviewing all the submissions, the SIGIR 2011 committee for their support, and all the authors for their contributions.

Hang Li
Gu Xu
Bruce Croft
Michael Bendersky
Workshop Organizing Committee
Workshop Organization

Organizers: Hang Li (Microsoft Research Asia)  
Gu Xu (Microsoft Bing)  
Bruce Croft (University of Massachusetts Amherst)  
Michael Bendersky (University of Massachusetts Amherst)

Program Committee: Alex Kotov (University of Illinois at Urbana-Champaign)  
Craig MacDonald (University of Glasgow)  
Daxin Jiang (Microsoft Research)  
Donald Metzler (ISI, University of Southern California)  
Fernando Diaz (Yahoo! Research)  
Fuchun Peng (Microsoft Bing)  
Jiafeng Guo (ICT, Chinese Academy of Sciences)  
Jian-Yun Nie (University of Montreal)  
Krisztian Balog (Norwegian University of Science and Technology)  
Le Zhao (Carnegie Mellon University)  
Matthias Hagen (Bauhaus University Weimar)  
Min Zhang (Tsinghua University)  
Oren Kurland (Technion, Haifa)  
Patrick Pantel (Microsoft Research)  
Pu-Jen Cheng (National Taiwan University)  
Rodrygo Santos (University of Glasgow)  
Vanja Josifovski (Yahoo! Research)  
Wessel Kraaij (Radboud University Nijmegen)  
Xiaobing Xue (University of Massachusetts)  
Yi Chang (Yahoo! Labs)  
Yumao Lu (Microsoft Bing)
Workshop Program

08:30 – 08:50 Poster Setup

08:50 – 10:00 Invited Talk by Nick Craswell

**Query Understanding for Relevance Measurement**
Nick Craswell (*Microsoft Bing*)

**Abstract**
Understanding the user needs underlying a query can be very difficult, even for a human relevance judge. When evaluating our algorithms, particularly those with a sophisticated query model, it may be wise to use real queries and a notion of relevance that is aligned with real user needs. I will present two lines of work in this area. One is the TREC Web Track, where we attempt to incorporate real Web tasks, real queries and a diverse set of user intents for each query. Click-based clustering or crowdsourcing have been used to identify possible intents. The other line of work is click-based experimentation using result interleaving. Compared to TREC methods, interleaving can detect more subtle and personalized preferences. It is sensitive enough to get significant results from tens of users who install a browser toolbar. Analysis of these different approaches is according to statistical power, ease/availability of use, and fidelity to real user preferences.

10:00 – 10:30 Coffee Break

10:30 – 11:00 Accepted Talks I

**Using Web Snippets and Query-logs to Measure Implicit Temporal Intents in Queries**
*Ricardo Campos (LIAAD – INESC Porto, LA)*
*Alípio Mário Jorge (LIAAD – INESC Porto, LA)*
*Gaël Dias (HULTIG, University of Beira Interior)*

**Complex Network Analysis Reveals Kernel-Periphery Structure in Web Search Queries**
*Rishiraj Saha Roy (IIT Kharagpur, Kharagpur, India)*
*Niloy Ganguly (IIT Kharagpur, Kharagpur, India)*
*Monojit Choudhury (Microsoft Research India, Bangalore, India)*
*Naveen Kumar Singh (NIT Durgapur, Durgapur, India)*

**Investigation of Web Query Refinement via Topic Analysis and Learning with Personalization**
*Lidong Bing (CUHK)*
*Wai Lam (CUHK)*
11:00 – 12:10 Invited Talk by Ricardo Baeza-Yates

Multi-faceted Query Intent Prediction
Ricardo Baeza-Yates (Yahoo! Research)

Abstract
In this presentation we report results for automatic classification of queries in a wide set of facets that are useful to the identification of query intent. Our hypothesis is that the performance of single-faceted classification of queries can be improved by introducing information of multi-faceted training samples into the learning process. We test our hypothesis by performing supervised and unsupervised multi-faceted classification of queries based on the combination of correlated facets.

Our experimental results show that this idea can significantly improve the quality of the classification. Since most of previous works in query intent classification are based in single facets, these results are a first step to an integrated query intent classification model. This is joint work with Liliana Calderon and Cristina Gonzalez.

12:10 – 13:45 Lunch Break

13:45 – 14:05 Accepted Talks II

Toward a deeper understanding of user intent and query expressiveness
Debora Donato (Yahoo! Labs)
Pinar Donmez (Yahoo! Labs)
Sunil Noronha (Yahoo! Labs)

Exploring the Query-Flow Graph with a Mixture Model for Query Recommendation
Lu Bai (ICT, CAS, Beijing, P.R. China)
Jiafeng Guo (ICT, CAS, Beijing, P.R. China)
Xueqi Cheng (ICT, CAS, Beijing, P.R. China)
Xiubo Geng (ICT, CAS, Beijing, P.R. China)
Pan Du (ICT, CAS, Beijing, P.R. China)

14:05 – 15:15 Invited Talk by Maarten de Rijke

Using Linked Open Data for Understanding Queries (and Other Short Text Segments)
Maarten de Rijke (University of Amsterdam)

Abstract
One way of capturing what it is that queries are about is to map them to concepts in the linking open data cloud. In the talk, I will compare various methods for addressing this task, using a mixture of information retrieval and machine learning techniques. Features used include query features, concept features as well as search-history features. Simply
performing a lexical match performs poorly, and so does using retrieval by itself, but complemented with a learning to re-rank approach, we obtain significant improvements. Time permitting, I will describe ongoing work on an extension of these ideas for capturing the about-ness of tweets. On top of the features used for understanding queries, we explore the use of tweet-specific features, based on hash tags, re-tweets, user mentions, etc.

The talk is based on joint work with Edgar Meij, Marc Bron, Laura Hollink, Bouke Huurnink and Wouter Weerkamp.

15:15 – 16:00 Coffee Break and Poster Discussion

16:00 – 17:30 Panel Discussion
Using Web Snippets and Query-logs to Measure Implicit Temporal Intents in Queries

Ricardo Campos  
LIAAD – INESC Porto, LA  
HULTIG, University of Beira Interior, Polytechnic Institute of Tomar, Portugal  
ricardo.campos@ipt.pt

Alípio Mário Jorge  
LIAAD – INESC Porto, LA  
DCC - FCUP  
University of Porto, Portugal  
amjorge@fc.up.pt

Gaël Dias  
HULTIG, University of Beira Interior, Portugal  
DLU/GREYC, University of Caen, France  
ddg@di.ubi.pt

ABSTRACT
Understanding the user's temporal intent by means of query formulation is a particular hard task that can become even more difficult if the user is not clear in his purpose. For example, a user who issues the query Lady Gaga may wish to find the official web site of this popular singer or other information such as informative or even rumor texts. But, he may also wish to explore biographic data, temporal information on discography release and expected tour dates. Finding this information, however, may prove to be particularly difficult, if the user does not specify the query in terms of temporal intent. Thus, having access to this data, will allow search mechanisms to improve search results especially for time-implicit queries. In this paper, we study different approaches to automatically determine the temporal nature of queries. On the one hand, we exploit web snippets, a content-related resource. On the other hand, we exploit Google and Yahoo! completion engines, which provide query-log resources. From these resources, we propose different measures to understand the temporal nature of queries. We compare these measures by analyzing their correlation. Finally, we conduct a user study to temporarily label queries.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – Query formulation, Search Process

General Terms
Algorithms, Experimentation.

Keywords
Temporal Information Retrieval, Implicit Temporal Queries, Query Classification, Temporal Query Understanding.

1. INTRODUCTION
The temporal intent of queries may be explicit or implicit. Explicit temporal queries are the most obvious ones, carrying explicit temporal evidence stated by the user. Some examples are SIGIR 2011, Iraq War 2003 or even future temporal queries such as Football World Cup 2014. Despite an apparent timeless nature, implicit temporal queries embody inherent temporal evidence. They consist of a set of keywords implicitly related to a particular time interval, which is not explicitly specified by the user. Some examples are Tour de France, Miss Universe or Haiti earthquake.

Understanding the temporal nature of a query, namely of implicit ones, is one of the most interesting challenges [3] in Temporal Information Retrieval (T-IR). However, few studies have attempted to answer questions like “How many queries have a temporal intent?”; “How many of those are temporally ambiguous?” and “Do they belong to some prevalent category?”. If we are able to answer to these questions, we may estimate how many queries will be influenced by a prospective temporal approach. A further automatic identification of temporal queries would enable to apply specific strategies to improve web search results retrieval. However, inferring this information is a hard challenge. First, different semantic concepts can be related to a query. An example is the query Scorpions that may be the rock band, the arachnid or the zodiac sign, each one with a different temporal meaning. Second, it is difficult to define the boundaries between what is temporal and what is not and so is the definition of temporal ambiguity. Third, even if temporal intents can be inferred by human annotators, the question is how to transpose this to an automatic process. One possible solution to date time-implicit queries is to seek for related temporal references over complementary web resources, such as document collections (e.g., web pages, web snippets, news articles, web archives) or web usage data (e.g., web query logs). In this paper we first summarize the three types of temporal queries. We then propose two different studies for the classification of implicit temporal queries based on the temporal value of web snippets and the temporal value of web query logs.

2. TYPES OF TEMPORAL QUERIES
Unlike explicit temporal queries, where the temporal nature is clearly defined by the user, implicit temporal queries can have a variety of inherent temporal nature. Similarly to [8], we classify time-implicit queries into one of the three following categories:

Type A - ATemporal: those not sensitive to time, i.e., queries not temporally related, like for instance make my trip.

Type B - Temporal Unambiguous: queries that take place in a very concrete time period like Haiti Earthquake which occurred in 2010.

Type C - Temporal Ambiguous: queries with multiple instances over time, such as Football World Cup, which occurs every four years. But also, queries such as Pascal, in the sense of the philosopher, where the user can be interested in different time segments of his life (e.g. birth date, date of death).

3. IMPPLICIT TEMPORAL QUERIES
The extraction of temporal information is usually based on a metadata-based approach upon time-tagged controlled collections, such as news articles, which are informative texts typically annotated with a timestamp [6] [8] [9] [10]. This information can be particularly useful to date relative temporal expressions found in a document (e.g., today) with a concrete date (e.g., document creation time). However, it can be a tricky process if used to date implicit temporal queries as referred in [12]. Indeed, the time of the document can differ significantly from the actual content of the document. An example is a document published in 2009 but which contents concern 2011. For instance, if a document is
related to the time-implicit query Miss Universe (see Figure 1), then there exists a high probability to associate the document temporal information to the document timestamp i.e. 2009 instead of 2011.

Figure 1: Use of Timestamp to date Miss Universe Query.

In this case, a content-based approach, which would extract temporal information from the content of the document, would obviously be the most appropriate solution to determine whether a query is temporally implicit or not. Another approach is to timestamp the queries based on similar year-qualified queries (e.g., Miss Universe 2009, Miss Universe 2010) stored in web query logs. Both differ in how they deal with query-dependency: while a web content approach simply requires the set of web search results, an approach based on web query logs implies that some versions of the query have already been issued.

3.1 Web Snippets

One of the most interesting approaches to date implicit temporal queries is to rely on the exploration of temporal evidence within web pages. As claimed in [3], this is an interesting future research direction, for which there is not yet a clear solution [1] [2]. Our main purpose in this section is to check if web snippets can be used to date time-implicit queries based on the temporal information existing in the title, in the text (generally known as snippet) and in the link (URL) of the web snippet. To this end, we conducted an experiment [5] where we studied the temporal value of web snippets1. We executed a set of 450 implicit queries extracted from Google Insights for Search Collection2, which registers the hottest and rising queries performed worldwide, defining a retrieval of 200 results per query. For each of the queries we computed three measures, TSnippets(.,) , TTitle(.,) and TUrl(.,), which assess how strong a query is temporally related. All represent the ratio between the number of respective items retrieved with dates divided by the total number of retrieved results. Results show that on average 10% of the web snippets retrieved for a given implicit query, contain year dates, of which 23% have more than one date. In this paper, we introduce a new insight to this approach. We propose to calculate the Pearson correlation coefficient between each of the dimensions (TSnippets(.,) , TTitle(.,) and TUrl(.,)) which assess how strong a query is temporally related.

Our next step is to manually classify each query with regard to its temporal intent based on the values of TSnippets(.,), TTitle(.,) and TUrl(.,). We first start by classifying the query in accordance to its concept ambiguity following the approach of [13] who defines three types of concept queries: ambiguous, broad and clear. Results in Table 1 show that most of the queries are ambiguous in concept, followed very closely by clear queries, which do not offer any doubt in terms of their meaning and by a small set of broad queries.

<table>
<thead>
<tr>
<th>Conceptual Classification</th>
<th>Number Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambiguous</td>
<td>220</td>
</tr>
<tr>
<td>Clear</td>
<td>176</td>
</tr>
<tr>
<td>Broad</td>
<td>54</td>
</tr>
</tbody>
</table>

Table 1: Concept and Temporal Classification of Queries.

Second, we aim at classifying the queries with regard to its temporal value. However, given that each concept of a query can have a different temporal dimension, we only focus on the temporal classification of clear concept queries. To this end, we computed, for each of the 176 clear queries (e.g., Toyota recall, lady gaga, Dacia duster, hairstyles), a temporal ambiguity value. Given the fact that dates occur in different proportions in the three items i.e. titles, snippets and urls, we value each feature differently through $ω$ (18.14% for TTitle(.,) , 50.91% for TSnippets(.,) and 30.95% for TUrl(.,)), where $f$ is the function regarding the corresponding item based on equation (1).

$$TA(q) = \sum_{f}^{n} ω_f \cdot f(q), \quad I = \{TSnippets(.,), TTitle(.,), TUrl(.,)\}$$

With this measure we can define a simple model for the automatic temporal classification of queries. A query is ATemporal if $TA(.)$ is below 10%, otherwise the query is defined as Temporal. Our experiments, strictly depending on the value of the year dates found in the web snippets, show that of all clear concept queries, 25% have implicit temporal intent. The remaining 75% are ATemporal queries. Moreover, we showed that year dates occur more frequently in response to queries belonging to the categories of sports, automotive, society and politics.

In order to evaluate our simple classification model, we conducted a user study. Using the same 176 clear concept queries, we asked three human annotators to judge their temporality. Human annotators were asked to consider each query, to look at web search results and to classify them as ATemporal or Temporal. Judgments were made assuming that the human annotator did not have any kind of specific temporal purpose in the execution of the query. The basic idea was to check if human annotations were correlated to our simple classification methodology just by looking at the set of web search results, even if there was a total absence of temporal intent. An inter-rater reliability analysis using the Fleiss Kappa statistics [7] was performed to determine consistency among annotators. Results showed a value of 0.89, meaning an almost perfect agreement between the raters. Overall, results pointed at 35% of implicit temporal queries from human annotators, while only 25% were given by our methodology. Then, we used the same test (Fleiss Kappa) to determine the consistency among each of individual annotators and the system decision. Results showed an average Kappa of 0.40 with a

---

1 Available at [http://www.ccc.ipt.pt/~ricardo/software] [17th June, 2011]

2 [http://www.google.com/insights/search] [17th June, 2011]
percentage of overall agreement of approximately 75%. A more thorough analysis also showed that most of the differences in the automatic classification of the system and the human annotator rating were related to electronic device queries, which introduce noise in the automatic definition of year dates (e.g., Nikon d3000). Overall, we may conclude that the occurrence of year dates in web snippets by itself is not sufficient to temporally classify these kinds of specific queries and that complementary information, such as the number of instances or the number of different dates should be considered in future approaches. Furthermore, this information can also be very useful in order to improve the results set returned to the user, either by adjusting the score of a document in response to an implicit temporal query, or by proposing alternative temporal search results exploration, through timelines or temporal clusters. We propose to do this by clustering the k top web snippets retrieved from the execution of any given query in [4]. The use of web documents to date queries not containing any temporal information can be however a tricky process. The main problem is related to the difficulties underlying the association of the year date found in the document and the query. An elucidative example is shown in Figure 3 where the dates appearing in the text and in the URL are not related with the query Miss Universe.

Miss Universe was held this year in Bahamas, 2006, was an incredible year, but everybody is waiting for the FIFA South Africa Football World Cup. From www.nbc.com/wc2010

Figure 3: Use of web contents to date Miss Universe Query.

3.2 Web Query Logs

Another approach to date implicit temporal queries is to use web query logs. The temporal activity of a search engine can be recorded in two different ways: (1) from an infrastructural perspective, i.e., date and time of the request and (2) from the user activity dimension, i.e., user search query such as Football World Cup 2010. This can be seen as a Web Usage Mining approach. This latter information can particularly be useful to infer temporal intents in queries not containing a specific year such as Tour de France, based on similar year-qualified queries such as Tour de France 2010. However, one of the problems of this approach lies in the fact that the number of year-qualified queries is too reduced. Indeed in [5], we showed that explicit temporal queries represent 1.21% of the overall set of a well-known query log collection (AOL Collection). Most of them belong to the categories automotive (21.96%), entertainment (9.48%) and Sports (8.15%). An additional problem is that we may have to deal with queries that have never been typed e.g. Blaise Pascal 1623 (his birthday year date). This tends to get even worse in specific applications that lack massive user interaction. Another problem is that query logs are hard to access outside big industrial laboratories. Moreover, web query logs are not adapted to concept disambiguation. For most part of the queries this is a real problem that would result in inconclusive information to the user. Consider for example the query Euro and suppose a web query log that has some related explicit temporal queries like Euro 2008 or Euro 2012. Having access to this information makes it possible to assign the query with the dates 2008 or with the future date 2014. Yet, this information is insufficient to disambiguate the query in its several meanings, causing the system to date the query Euro with temporal information regarding the European football world cup, when in fact it could be related to the European currency. One of the first works in this line was proposed by [11] who presented an interesting approach based on the access to a query log with frequency information. In order to temporally qualify a query, the authors introduced a weighted measure that considers the number of times a query q is pre- and post-qualified with a given year y as shown in Equation 2.

\[ W(q, y) = \#(q, y) + \#(y, q) \]  

A query is then considered implicitly year-qualified if it is qualified by at least two different years. Moreover, the authors introduce a confidence measure to confirm that the query has an implicit temporal intent. This value is defined in Equation 3 where the sums on the denominator of the equation go all over pre- and post-qualifications of the query q.

\[ a(q) = \frac{\sum q \cdot w_c(q, y)}{\sum q \cdot w_c(q, y) + \sum q \cdot w_c(y, q)} \]  

If the query is always qualified with a single year then \( a(q) = 1 \). Overall results show that 7% of the queries have implicit temporal intent. These values contrast with the 25% that we present in our study based on web content analysis.

3.2.1 Yahoo! and Google Temporal Value

In this section, we aim at quantifying the temporal value of a query in Yahoo! and Google web logs and compare it with web snippets by means of a Pearson correlation coefficient with a confidence interval for paired samples. We already showed that the number of explicit temporal queries existing in web logs can be very small but we must also take into account that the simple fact that a query is year-qualified does not necessarily mean that it has a temporal intent. An illustrative example is the query make my trip which is substantially more qualified with words than with temporal features. In order to measure this value, we introduce two measures TLogYahoo(.) and TLogGoogle(.) similarly to TTitle(.), TSnippets(.) and TURL(.). To compute these values, we rely on Google and Yahoo! auto-complete query search feature, which constitutes a powerful mechanism to understand the temporality of a given query based on the fact that it displays user queries supported by real user search activities. TLogGoogle(.) and TLogYahoo(.) are defined in Equation 4 and 5 respectively as the ratio between the number of queries suggested with year dates divided by the total number of suggested retrieved queries.

\[ TLogGoogle(q) = \frac{\# Suggested \; Queries \; Retrieved \; With \; Dates}{\# Suggested \; Queries \; Retrieved} \]  

\[ TLogYahoo(q) = \frac{\# Suggested \; Queries \; Retrieved \; With \; Dates}{\# Suggested \; Queries \; Retrieved} \]  

For example, if we type in the query bp oil spill, both Yahoo! and Google search engines suggest a set of 10 queries, of which only one, in this case in Yahoo! search interface (see left hand part of Figure 4) includes a year date. This means that for the query bp oil spill TLogYahoo(.) would be 10% and TLogGoogle(.) 0% (see right hand part of Figure 4).

Figure 4: Auto-complete Query Suggestion for the query Bp Oil Spill in Yahoo! (on the left) and Google (on the right).
Based on the values obtained for each of the 176 clear concept queries, we calculated the Pearson correlation coefficient (see Table II) to compare the temporal value of web snippets by means of TSnippets(), TTitle() and TUrl() values with the temporal value of web logs by means of TLogGoogle() and TLogYahoo().

Table II: Pearson Correlation Coefficient.

<table>
<thead>
<tr>
<th></th>
<th>TLogYahoo</th>
<th>TTitle</th>
<th>TSnippet</th>
<th>TUrl</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLogYahoo</td>
<td>0.63</td>
<td>0.61</td>
<td>0.52</td>
<td>0.48</td>
</tr>
<tr>
<td>TLogGoogle</td>
<td>0.69</td>
<td>0.63</td>
<td>0.44</td>
<td></td>
</tr>
</tbody>
</table>

Final results show that best correlation values occur between TTitle() and TLogGoogle() with a value of 0.69 and between TSnippet() and TLogGoogle() with 0.63. This means that as dates appear in the titles and snippets, they also tend to appear, albeit in a more reduced form, in the auto-complete query suggestion of Google.

An additional analysis led us to conclude that the temporal information is more frequent in web snippets than in any of the query logs of Google and Yahoo! (see Figure 5 and Figure 6). Overall, while most of the queries have a TSnippet() value around 20%, TLogYahoo() and TLogGoogle() are mostly near to 0%.

5. ACKNOWLEDGMENTS
This research was partially funded by the Portuguese Foundation for Science and Technology through a PhD grant with Reference SFRH/BD/63646/2009 and the VIPACCESS project with Reference PTDC/PLP/72142/2006. Other funds also came from the ACM Special Interest Group on Information Retrieval (SIGIR).

6. REFERENCES
Complex Network Analysis Reveals Kernel-Periphery Structure in Web Search Queries

Rishiraj Saha Roy∗and Niloy Ganguly
IIT Kharagpur
Kharagpur, India - 721302.
{rishiraj, niloy}@cse.iitkgp.ernet.in

ABSTRACT

Web search queries have evolved into a language of their own. In this paper, we substantiate this fact through the analysis of complex networks constructed from query logs. Like natural language, a two-regime degree distribution in word or phrase co-occurrence networks of queries reveals the existence of a small kernel and a very large periphery. But unlike natural language, where a large fraction of sentences are formed only using the kernel words, most queries consist of units both from the kernel and the periphery. The long mean shortest path for these networks further show that paths between peripheral units are typically connected through nodes in the kernel, which in turn are connected through multiple hops within the kernel. The extremely large periphery implies that the likelihood of encountering a new word or segment is much higher for queries than in natural language, making the processing of unseen queries much harder than that of unseen sentences.

Categories and Subject Descriptors
H.3.3 [Information Search and Retrieval]: Query Formulation

General Terms
Measurement, Experimentation

Keywords
Web search queries, Co-occurrence networks, Two-regime power law, Kernel and peripheral lexicons

1. INTRODUCTION

Web users communicate their information need to a search engine through queries. The fact that search engines do not really understand or process natural language grammar drives the average Web user to specify their queries in a language that has a structure far simpler than natural languages, but perhaps more complex than the commonly assumed bag-of-words model. In fact, Web search queries define a new and fast evolving language of its own, whose dynamics are governed by the behavior of the search engine towards the user and that of the user towards the engine. Earlier works have shown that in general, direct application of grammatical frameworks and concepts used in Natural Language Processing to understand queries has not been very productive. The linguistic structure of queries has significant differences and similarities from that of natural languages and has to be discovered and understood from the first principles. Not much linguistic or statistical analysis has been done to understand this structure, mainly because, query log data is not publicly available. Moreover, the absence of a generic query structure across all domains make fully automated techniques for linguistic analysis infeasible.

In recent times, complex networks have provided an elegant framework for understanding and analyzing evolving linguistic structures. For example, modeling syntactic and semantic features of a language, consonant inventories and their dynamics can all be accomplished through the study of linguistic networks [1]. Of special interest are the word co-occurrence networks, where each distinct word is considered a node and edges represent co-occurrence of two words in the same sentence [2]. Similarly, in our models, each query is viewed as a sentence. We use two notions of co-occurrence — local and global. In the local co-occurrence networks, immediate word neighborhood is considered important and an edge is added between two words only if they occur within a distance of two (i.e. separated by zero or one word) in a query [2]. Global networks assume order independence; edges are added between words if they are found in the same query. Restriction of a network is used to prune the edges which might occur purely by chance [2]. To be precise, let $i$ and $j$ be two distinct words from the corpus. Let $p_i$, $p_j$ and $p_{ij}$ be the probabilities of occurrence of $i$, $j$ and the bigram $ij$ in the data. Then, in a restricted network, an edge exists if $p_{ij} > p_i p_j$.

Several researchers have argued that queries are bags-of-segments, and not bags-of-words [3, 4]. For example, the query "australian open 2011 home page" is semantically equivalent to the queries "australian open 2011" "home page" and "home page" "australian open 2011", but not to page 2011 open australian home. Therefore, we also study the co-
occurrence networks of segments, where nodes are distinct segments instead of words, with the same principles used for edge creation. Query segmentation has been performed based on a technique that uses query logs as the only resource for training [4]. This method is fundamental to our goal because the use of document-based resources for query segmentation risks projection of natural language structure onto queries. For example, the likely segmentation of a word co-occurrence networks built from natural language sentences 

"bill" by our method. While "a fake" "bill." This is segmented as

The last two networks were built analogously from natural

queries collected through Bing Australia (http://www.natcorp.ox.ac.uk/) [2]. The degree distribution (DD), represented by $p(k)$, is the fraction of nodes in the network with degree equal to $k$; it is very popularly used as a primary topological indicator of complex networks. Nevertheless, often the Cumulative Degree Distributions (CDD), i.e. the degree $k$ versus the fraction of nodes with degree greater than or equal to $k$, $P(k)$, is preferred over DD because the former is more robust to the presence of noise in the data. The negative derivative of CDD gives DD. Fig. 1 shows the CDD of $QUSN_g$ and $QRSN_g$. The two-regime power-law nature of the distribution is very clear from the plot, and in fact, holds for all the eight networks. Hence, we will not report the measurements on each network individually.

### 1.1 Two-regime Power Law

The degree distribution (DD), represented by $p(k)$, is the fraction of nodes in the network with degree equal to $k$; it is very popularly used as a primary topological indicator of complex networks. Nevertheless, often the Cumulative Degree Distributions (CDD), i.e. the degree $k$ versus the fraction of nodes with degree greater than or equal to $k$, $P(k)$, is preferred over DD because the former is more robust to the presence of noise in the data. The negative derivative of CDD gives DD. Fig. 1 shows the CDD of $QUSN_g$ and $QRSN_g$. The two-regime power-law nature of the distribution is very clear from the plot, and in fact, holds for all the eight networks. For easy visualization, the best-fit power law lines are also shown. In Table 1 we report the best-fit exponents $\gamma_1$ and $\gamma_2$ for the upper and lower regimes of the DD for the networks. $k_{cross}$ refers to the degree where the crossover between the two regimes takes place.

This is a strikingly similar behavior between word co-occurrence networks build from natural language sentences

<table>
<thead>
<tr>
<th>Network</th>
<th>Expansion</th>
<th>C</th>
<th>$C_{rand}$</th>
<th>$d$</th>
<th>$d_{rand}$</th>
<th>$k_{cross}$</th>
<th>$\gamma_1$(DD)</th>
<th>$\gamma_2$(DD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUWN$_l$</td>
<td>Query Unrestricted Word Net (local)</td>
<td>0.57</td>
<td>$4.24 \times 10^{-5}$</td>
<td>7.08</td>
<td>4.24</td>
<td>1000</td>
<td>$-1.82$</td>
<td>$-3.11$</td>
</tr>
<tr>
<td>QRWN$_l$</td>
<td>Query Restricted Word Net (local)</td>
<td>0.28</td>
<td>$3.59 \times 10^{-5}$</td>
<td>9.45</td>
<td>4.48</td>
<td>1000</td>
<td>$-1.82$</td>
<td>$-3.30$</td>
</tr>
<tr>
<td>QUWN$_g$</td>
<td>Query Unrestricted Word Net (global)</td>
<td>0.61</td>
<td>$5.48 \times 10^{-5}$</td>
<td>7.00</td>
<td>3.92</td>
<td>1500</td>
<td>$-1.81$</td>
<td>$-2.89$</td>
</tr>
<tr>
<td>QRWN$_g$</td>
<td>Query Restricted Word Net (global)</td>
<td>0.48</td>
<td>$4.57 \times 10^{-5}$</td>
<td>7.12</td>
<td>4.14</td>
<td>1500</td>
<td>$-1.89$</td>
<td>$-2.85$</td>
</tr>
<tr>
<td>QUUSN$_l$</td>
<td>Query Unrestricted Segment Net (local)</td>
<td>0.44</td>
<td>$2.43 \times 10^{-5}$</td>
<td>7.19</td>
<td>4.62</td>
<td>1500</td>
<td>$-1.97$</td>
<td>$-3.13$</td>
</tr>
<tr>
<td>QRSN$_l$</td>
<td>Query Restricted Segment Net (local)</td>
<td>0.36</td>
<td>$2.29 \times 10^{-5}$</td>
<td>7.31</td>
<td>4.72</td>
<td>1500</td>
<td>$-2.00$</td>
<td>$-2.82$</td>
</tr>
<tr>
<td>QUUSN$_g$</td>
<td>Query Unrestricted Segment Net (global)</td>
<td>0.47</td>
<td>$2.85 \times 10^{-5}$</td>
<td>7.12</td>
<td>4.39</td>
<td>2000</td>
<td>$-1.98$</td>
<td>$-3.17$</td>
</tr>
<tr>
<td>QRSN$_g$</td>
<td>Query Restricted Segment Net (global)</td>
<td>0.41</td>
<td>$2.71 \times 10^{-5}$</td>
<td>7.22</td>
<td>4.46</td>
<td>2000</td>
<td>$-2.01$</td>
<td>$-3.25$</td>
</tr>
<tr>
<td>NUWN$_l$</td>
<td>NL Unrestricted Word Net (local)</td>
<td>0.69</td>
<td>$1.55 \times 10^{-4}$</td>
<td>2.63</td>
<td>3.03</td>
<td>$\approx 2500$</td>
<td>$-1.50$</td>
<td>$-2.70$</td>
</tr>
<tr>
<td>NRWN$_l$</td>
<td>NL Restricted Word Network (local)</td>
<td>0.44</td>
<td>$1.55 \times 10^{-4}$</td>
<td>2.67</td>
<td>3.06</td>
<td>$\approx 2500$</td>
<td>$-1.50$</td>
<td>$-2.70$</td>
</tr>
</tbody>
</table>

Figure 1: Cumulative degree distributions for segment networks ($QUSN_g$ and $QRSN_g$).
2. INSIGHTS FROM THE NETWORKS

In [2], the topological properties of word co-occurrence networks have been explained in terms of well-known linguistic theories. In absence of any prior linguistic study of Web search queries, here we make an attempt to explain the similarities and differences between query and natural language networks by drawing analogy to linguistic theories and their applicability in the context of queries.

2.1 Deviation from Small-world Property

Small-world networks are found in many natural, technological and social systems and have been shown to exhibit interesting properties [5]. Cancho and Solé show that word co-occurrence networks are small-world and proposes that this might facilitate retrieving words quickly from the mental lexicon [2]. Three conditions must be satisfied in order for a network to be small-world: Sparseness ($|V| >> k$), $d \approx d_{rand}$ and $C >> C_{rand}$ [5]. For our networks, $k$ ranges between 17.69 and 28.96. Thus, the sparseness and clustering coefficient constraints are satisfied (Table 1). However, $d$ ($\approx 7$) is 1.71 times greater than $d_{rand}$ (Table 1), which is too large to be considered a small-world. Note that in contrast word co-occurrence networks from Standard English feature much smaller $d$ (2.63 and 2.67 for unrestricted and restricted word networks respectively) [2]. This high diameter and resulting sparseness for query co-occurrence networks, as we shall see shortly, has interesting implications. High sparseness is also reflected in the sharp contrast in average degrees, which are 74.20 and 70.13 for unrestricted and restricted word networks in Standard English [2]. However, it is interesting to note that unlike the word co-occurrence networks, query networks do not reflect the structure of the mental lexicon and therefore, need not be constrained by factors needed to facilitate fast word retrieval. Rather these networks reflect underlying cultural connections where variation and diversification are the prime drivers.

2.2 Kernel and Peripheral Lexicons

The emergence of two regimes in the DD of word co-occurrence networks has been explained by the existence of distinct kernel and peripheral lexicons. Extending this analogy, one can argue that the language of Web search queries is characterized by the existence of two different types of words or segments (henceforth referred to as units): Units popular in queries acquire high degrees and constitute the kernel lexicon (K-Lex), while rarer ones, with degree much lower than those in the kernel, form the peripheral lexicon (P-Lex). $k_{ran}$ is observed to be the same for unrestricted and restricted network pairs, which is also observed for Standard English [2].

Composition of K-Lex and P-Lex. In natural language, the kernel and peripheral lexicons are defined as domain-independent and domain-specific vocabularies. We observe that for queries, the kernel units are generic and often domain independent; i.e., these units can be used in queries from almost all or at least a very wide range of domains. Examples of kernel units include Australia, download, best, and names of popular cities and games. In contrast, the peripheral units are present in queries only from specific domains.

We found that units in queries can be thought to be of two categories depending upon their role in the query. Heads represent the central information need in a query (lord of the rings, barack obama, high schools). Modifiers, on the other hand, are specified by the user to indicate their requirements more precisely (lyrics, wikipedia, near).

We observe that most of the units in the K-Lex can act as modifiers (e.g., how to, download, video). Very popular head units such as melbourne, ipod and nokia also occur in the K-Lex. These heads occur in the K-Lex due to their high frequency of occurrence in the region from where the logs are sampled, but interestingly, can also act as modifiers in many contexts. For example, users often specify an error they are facing or an accessory they need, providing ipod and nokia as the query context, respectively. The P-Lex, on the other hand, is constituted of mostly heads (goldlocks and the three bears, leukemia, harvard university). Some modifiers (e.g., accessories, cast, who is) are also present in the P-Lex. The point to note is that these modifiers are very specific. For instance, accessories go with mostly electronic devices and automobiles, cast with movies and who is with names of people. However, this logic breaks down for some modifiers like lyrics which is a specific modifier that can only occur for song queries, but is present in the K-Lex. The reason for this is that entertainment, especially songs, constitutes a major chunk of all queries.

To complete the analogy, we note that in natural language, K-Lex consists of function words (e.g., is, of, and) and very common content words (e.g., eat, sleep, flower), and P-Lex consists of domain specific content words (e.g., programming, Buddhism, Titanium).

2.3 Queries against Natural Language

We observe several insightful differences between the prop-
properties of co-occurrence networks for natural language (Standard English) and those for queries. We highlight these individually in the subsequent paragraphs.

The kernel-to-periphery ratio is much higher in queries than that for English. For Standard English, the ratio of the sizes of P-Lex to K-Lex is \( \approx 85 \), while for queries it ranges between 250 and 850 (average kernel and periphery sizes are 1285 and 648615 respectively). The presence of a very large periphery means that in a query, one is highly likely to encounter previously unseen words, something which is much less probable in natural language. This is because users continuously try to provide “discriminating” (instead of popular) words and phrases to narrow down the search space.

Periphery stands by itself. A large diameter \((d \approx 7)\) for query networks implies that one must also traverse intra-peripheral edges to reach one peripheral unit from another. Interestingly, such linkages emerge from tail queries. These units may not have a direct connection to the kernel and hence do not provide the search engine with enough known context when first encountered. This makes processing of rare queries harder than processing of new words in natural language sentences. This is because, new words always occur in the context of known words that provide enough context for word understanding. For example, in query networks, we have the following shortest path: \( \text{airedale terrier} \langle< \text{tumor} \langle< \text{download} \langle< \text{prison break} \). Here the first two and the last segments are from the periphery and the rest from the kernel. Clearly, the query constituted of the first two segments “airedale terrier” “tumor” is hard to understand and process automatically.

The kernel network is much less tightly coupled than in English. The kernel network (KN) is defined to be formed by nodes from the kernel lexicon only, and the edges that go between them [2]. The CDD of the KN for query networks falls off as power law (Fig. 2) with exponent \( \gamma_{KN} \) ranging between \(-2.74 \) and \(-2.96 \), and the average clustering coefficient is 0.59 (fully connected networks will have a CC of 1); thus, unlike word-occurrence networks for natural language, KN for queries does not exhibit a clique-like behavior. In natural language, sentences are long, and contain stopwords like the and in the same sentence. This is primarily responsible for the creation of cliques in the kernel network. However, queries, are mostly devoid of stopwords, and modifiers belonging to different classes rarely occur in the same query – for instance, top modifiers like pdf and simdb do not share an edge.

The peripheral network consists of a large number of small disconnected components. The degree distribution in the peripheral network (PN, defined analogously as KN) decays much more slowly (inset, Fig. 2), due to the presence of a large number of approximately equal-sized disconnected components. The largest component of the PN contains \( \approx 20000 \) nodes, which, however, is very large compared to the corresponding network in natural language.

Most edges run between kernel and periphery for queries, whereas intra-kernel edges dominate in natural language. From the distribution of the types of edges \((\approx 98.76\% \) between kernel and periphery, \( \approx 0.48\% \) intra-kernel, \( \approx 0.76\% \) intra-periphery), we can see that the high degrees in the kernel are significantly due to contributions from peripheral units. As mentioned earlier, a typical query generally has a rare unit as a distinguishing feature.

Socio-cultural factors govern the Kernel-Periphery distinction in queries. While for natural language the divide between the K-Lex and P-Lex emerges primarily due to cognitive/syntactic and only marginally due to socio-cultural factors, for queries, the division emerges mainly due to socio-cultural factors and only marginally due to syntactic factors. For instance, lyrics is in the K-Lex and genes in the P-Lex only because lyrics search is far more popular than gene search; this is purely a socio-cultural phenomenon and has nothing to do with the syntactic structure of the query language. However, the fact that Australian cities, download or wiki are in K-Lex is more an effect of the structure of queries; though strictly speaking, almost everything about queries is socio-cultural; there is hardly anything cognitive here. Query logs are a representation of the collective information need of the people of a particular geography and demographics.

3. CONCLUSIONS AND FUTURE WORK
The contribution of this paper lies in establishing that word co-occurrence patterns in Web search queries and natural languages are strikingly similar, yet very different from each other. While like natural language, queries reflect a Kernel-Periphery distinction implying an underlying linguistic structure, at the same time, unlike natural languages, query networks lack small-world property which has been argued to be essential in natural language for quickly retrieving words from the mind. More importantly, the underlying implications of these differences are that it is much more difficult to understand the context of a word or a segment in a query. A tightly-knit kernel connected to every peripheral node forms the basis for contextual inferencing in natural language both by man and machine. In queries, the absence of such a kernel and a much larger periphery results in a high surprise factor for several peripheral units. The capability of peripheral units to exist by themselves makes POS identification hard in queries. Understanding these fundamental differences may open up an area where researchers would more formally characterize query grammar. We believe that the application of complex networks to query representation holds immense potential, of which this work is just a beginning.

4. REFERENCES
Investigation of Web Query Refinement via Topic Analysis and Learning with Personalization

Lidong Bing Wai Lam

Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
Shatin, Hong Kong
{lilbing, wlam}@se.cuhk.edu.hk

ABSTRACT
We investigate the benefits of latent topic analysis and learning with personalization for Web query refinement. Our proposed framework exploits a latent topic space, which is automatically derived from a query log, to leverage the semantic dependency of terms in a query. Another major characteristic of our framework is an effective mechanism to incorporate personal topic-based profile in the query refinement model. Moreover, such profile can be automatically generated achieving personalization of query refinement. Preliminary experiments have been conducted to investigate the query refinement performance.

Categories and Subject Descriptors
I.5.1 [Pattern Recognition]: Models—Statistical

General Terms
Algorithms

Keywords
query refinement, personalization, bookmark data, query log

1. INTRODUCTION
Web query refinement aims at reformulating a given Web query to improve search result quality. There are three broad types of refinement, namely, substitution [8, 13], expansion [1, 4, 15, 6], and deletion [9, 10, 16]. Besides these broad types, some other fine-grained types include stemming [12], spelling correction [3], abbreviation expansion [14], etc. In [5], a linear chain Conditional Random Field model is employed to deal with these fine-grained refinement operations. For the broad types of refinement mentioned above, a common approach is to generate some candidate queries first, and then a scoring method is designed to assess the quality of these candidates. For example, Wang and Zhai proposed a contextual model by investigating the context similarity of terms in history queries [13]. Two terms with similar context are used to substitute each other in candidate query generation. Then a context based translation model is employed to score the candidate queries. Jones et al. employed hypothesis likelihood ratio to identify those highly related query phrases or term pairs in user sessions [8]. The above existing methods make use of history queries in a query log, and exploit the term context information to generate term pairs and score new candidate queries. One shortcoming of existing context-based methods is that they cannot deal with some ambiguous terms effectively especially when a term has very diverse contexts in history queries.

This paper focuses on term substitution in query refinement. Our proposed framework also consists of two phases, namely, candidate query generation and candidate query scoring. For candidate query generation, we employ a method similar to the term substitution pattern mining presented in [13], which generates a candidate query by substituting one term in the input query and is capable of keeping the semantic relation between the candidate and the input query. For candidate query scoring, we propose a framework that considers semantic dependency of latent topics of term sequence in a given query. Our proposed model exploits a latent topic space, which is automatically derived from a query log, to leverage the semantic dependency of terms in a query. When we score a candidate query, the latent topic sequence of the query is used as hidden evidence to guide the semantic dependency assessment. Another major characteristic of our framework is an effective mechanism to incorporate personal topic-based profile in the query refinement model. Moreover, such profile can be automatically generated from a query log achieving personalization of query refinement. Our final hybrid scoring model combines latent topic evidence and a bigram-based language model. Preliminary experiments have been conducted to investigate the query refinement performance.

2. CANDIDATE QUERY SCORING
Since we currently focus on the investigation of query term substitution, our candidate query scoring method aims at comparing queries of the same length.
2.1 Latent Topic Analysis

A typical record in query log can be represented as a 4-tuple (anonymous user id, query, clicked-url, time). One direct method for using query log data for latent topic analysis is to treat each clicked-url as a single document unit. This approach will suffer from the data sparseness problem since most URLs only involve very small number of queries. Instead of adopting such a simple strategy, we aggregate all the queries related to the same host together and construct one pseudo-document for each host. For example, the pseudo-document of “www.mapquest.com” consists of the queries “mapquest”, “travel map”, “driving direction”, and so on. Some general Web sites such as “en.wikipedia.org” are not suitable for latent topic analysis because they involve large amount of queries as well as many query terms, and cover very diverse semantic aspects. To tackle this problem, we first sort the host pseudo-documents in descending order according to the number of distinct query terms they have. Then, the top ranked pseudo-documents are eliminated in our latent topic discovery process.

We employ the standard Latent Dirichlet Allocation (LDA) algorithm [2] to conduct the latent semantic topic analysis on the collection of host-based pseudo-documents. In particular, GibbsLDA package is used to generate the set of latent topics. Let $Z$ denote the set of latent topics. Each topic $z_i$ is associated with a multinomial distribution of terms. The probability of each term $t_k$ given a topic $z_i$ is denoted by $P(t_k|z_i)$.

2.2 Latent Topic Based Scoring with Personalization

A candidate query $q : t^1 \cdots t^n$ is composed of a sequence of terms, where the superscript indicates the position of the term and $t^r (1 \leq r \leq n)$ may take any admissible term in $T$ where $T$ denotes the vocabulary set. The topic of $t^r$ is denoted by $z^r$ which denotes an admissible topic in $Z$. We investigate a Hidden Markov Model (HMM) as shown in Figure 1 to score the candidate query. The query terms are observable and represented by filled nodes. The latent topics are unobservable and represented by empty nodes. Different from a common application of HMM that solves the tagging or decoding problem, we make use of this model to compute the marginal joint probability of the term sequence denoted as $P(t^{1:n})$ for scoring the candidate query. Let $z^{1:n}$ denote the topic sequence. The candidate query score can be computed by:

$$P(t^{1:n}) = \sum_{z^{1:n}} P(t^{1:n}, z^{1:n})$$

Due to the dependency structure of the model, the joint probability of the topic sequence and term sequence can be expressed as:

$$P(t^{1:n}, z^{1:n}) = \prod_{r=1}^{n} P(t^r|z^r) \prod_{r=2}^{n} P(z^r|z^{r-1})$$

This model involves the term emission probability from a topic $P(t_k|z_i)$ which can be regarded as considering the relationship of terms and topics. Another model parameter $P(z_j|z_i)$ captures the relationship of two topics in the modeling and scoring process. This pairwise topic relation-

ship offers a means to incorporate different strategies of considering topic contexts governing neighboring terms in the candidate query. One strategy is to favor neighboring terms in a query sharing similar topic context. To facilitate this objective, we design a scheme that favors the pairwise topic probability if the semantic content of the two topics has high degree of similarity.

To provide a better quality for query refinement, we investigate the capability of personalization. Different users have different personalized preference which can be revealed by the fact that their queries concentrate on a particular set of topics. For example, teenagers usually issue queries related to basketball, computer game, and so on. Therefore, each user has an inherent preference on different topics. This personalized interest can be encapsulated in a topic-based profile. Then the profile can be taken into account in the calculation of the topic-based score. We describe how to incorporate a topic-based profile in the scoring model.

Let $\Pi^u = \{\pi_1^u, \pi_2^u, \cdots, \pi^n_u\}$ denote the profile of the user $u$, where $\pi_i^u = P(z_i|u)$ is the probability that the user $u$ prefers the topic $z_i$. Considering the probability constraint, we have $\sum_i \pi_i^u = 1$. Equation 2 can be modified as:

$$P(t^{1:n}, z^{1:n}) = \prod_{r=1}^{n} P(t^r|z^r) \prod_{r=2}^{n} P(z^r|z^{r-1})$$

where $\pi_i^u = P(z_i|u)$. Thus, when we score a query, the preference of $u$ is taken into account.

To compute efficiently the score, we make use a dynamic programming technique commonly adopted for tackling the computational requirement of HMM models. The forward variable $\alpha_r(i)$ is defined as:

$$\alpha_r(i) \triangleq P(t^{1:r}, z^r = z_i),$$

which is an intermediate score of the partial query $t^1 \cdots t^r$ given the topic $z_i$ at the position $r$. When $r = 1$, we set $\alpha_1(i) = \pi_i^u P(t^1|z_i)$. The recursive calculation of $\alpha$ is:

$$\alpha_r(i) = \sum_{j \in Z} \alpha_{r-1}(j) P(z^r = z_i|z^{r-1} = z_j) P(t^r|z^r = z_i),$$

where $P(z^r = z_i|z^{r-1} = z_j)$ is the pairwise topic probability at the position $r$.

The topic-based score $S_i(q)$ is calculated by summing over all possible $z_i$ of $\alpha_r(i)$:

$$S_i(q) = P(t^{1:n}) = \sum_{z_i \in Z} \alpha_r(i).$$
2.3 Model Parameter Design
The model parameter $P(t_k|z_i)$ can be readily obtained from the probability of a term given a topic in the LDA analysis mentioned above.

For the parameter $P(z_j|z_i)$ corresponding to the pairwise topic relationship, we consider the objective of query refinement. The topics of terms in the same query tend to remain consistent from semantic point of view because of the unique search intention of the user for the given query. Different strategies can be developed to achieve this objective. As a preliminary investigation, we examine the degree of semantic similarity of the pair of topics. Basically, the more similar between two latent topics, the higher is this similarity measure. Another option for calculating $P(z_j|z_i)$ is a bigram-based score of $q$, and calculated as:

$$S_b(q) = \frac{\lambda \log S_b(q) + (1 - \lambda) \log S_b(q)}{\lambda \log S_b(q)} + (1 - \lambda) \log S_b(q),$$

where $S_b(q)$ is a bigram-based score of $q$, and calculated as:

$$S_b(q) = \prod_{i=1}^{n} P(t_i|t_i^{-1}).$$

$P(t_i|t_i^{(0)})$ is set to $P(t_i^{(1)})$. $\lambda$ is the parameter for controlling their relative weights.

3. PRELIMINARY EXPERIMENTS

3.1 Experimental Setup
We use the AOL query log [11] from 1 March, 2006 to 31 May, 2006. The raw data contains a lot of noise, so some cleansing operations are performed, such as navigation query removal and stop words removal. We adopt a hybrid method to detect user session boundary [7] and remove those sessions without any clicking. The data set is split into two sets, namely, the history set and the test set. The history set contains the first two months’ log and the test set contains the third month’s log. The pseudo-documents for latent topic analysis are constructed with the history set. We select hosts involving at least 5 queries and we remove the top 0.1% of the hosts according to the distinct query terms they have. Finally, 189,670 pseudo-documents are obtained and the number of topics is set to 30. The value of $\lambda$ used is 0.4 for the personalized model and 0.2 for the non-personalized model, with which the best performance is achieved. The bigram language model in our hybrid scoring method is estimated from the queries in the history set of the query log, and the Dirichlet smoothing technique [17] is employed to tackle the problem of data sparseness.

For conducting the comparison, we implement a context based term association (CTA) method presented in Wang and Zhai [13]. We denote the query scoring step in CTA as “CTA-SCR”. We generate the contextual and translation models for the top 100,000 terms in the history set, and apply the threshold in [13] to filter out the noise.

We conduct an automatic method by utilizing the session information to evaluate the performance of the refinement models. In a search session, when users feel unsatisfied with the results of current query, they may refine current query and search again. We differentiate two kinds of queries, namely, satisfied queries and unsatisfied queries. In a user session, the query which causes at least one URL clicked and is located in the end of the session is called a satisfied query. The query which is located just ahead of the satisfied query in the same user session is called an unsatisfied query. We collect a set of unsatisfied queries for conducting the experiment and their corresponding satisfied queries are treated as the benchmark answer for the refinement task. The scoring method will return a ranked list of the candidate queries. Then we evaluate the performance of the method at top $m$. If the true answer can be found in the top $m$ candidates, that query is considered as successful. Accuracy is defined as the total number of successful queries divided by the total number of test queries.

For generating user profile, we randomly select 400 users who have more than 100 sessions in the history set. Then the queries issued by the same user in the history set are aggregated together to generate user profile. For each user, we select one of his unsatisfied queries from the test set which has at least 3 terms, and use this query as the input of the refinement models.
Figure 2: Performance of different scoring methods.

3.2 Results

The performance of different scoring methods is given in Figure 2. “Our model” is the framework presented in this paper combining personalized topic scoring and bigram scoring. “Without Personalization” is our model without personalization. “CTA-SCR” is a baseline method as described above. “Bigram” is another baseline model using a pure bigram method. It can be observed that our framework that considers personalization achieves the best performance. It indicates that our method can rank good suggestions of query refinement higher. We also find that with user profiles, the topic-based scoring part is more reliable and it plays a more important role. “CTA-SCR” performs better than the pure bigram method, but not as effective as our method.

4. CONCLUSIONS AND FUTURE WORK

We present a framework for performing term substitution in Web query refinement based on a personalized topic-based hybrid scoring method. Our method can detect and consider the semantic dependency of terms in queries. From the experimental results, we observe that taking both the semantic dependency and personalization into account can help offer better query refinement quality.

In this preliminary work, the model parameters, namely, emission probability and transition probability, are directly estimated from the results of latent topic analysis. In our future work, we intend to employ Expectation Maximization (EM) algorithm to conduct more precise parameter estimation.

5. REFERENCES


Toward a deeper understanding of user intent and query expressiveness

Debora Donato
Yahoo! Labs
Sunnyvale, CA, USA
debora@yahoo-inc.com

Pinar Donmez
Yahoo! Labs
Sunnyvale, CA, USA
pinard@yahoo-inc.com

Sunil Noronha
Yahoo! Inc
Sunnyvale, CA, USA
noronha@yahoo-inc.com

ABSTRACT
Understanding the intents of the search users have recently been a popular research topic among Information Retrieval community. Accurate inference of the intent is a major goal of today’s search engines. Ranking the relevant Web documents, targeting better advertisements, assisting the user with better suggested queries are just a few applications where understanding the intent plays a key role. In order to better understand how users formulate their intents into queries, we conducted a preliminary experimental study in which users agreed to be recorded during their search activity. Monitoring the user behavior and discussing the process with them at the end of their search activities allows us to gather some fascinating insights about intents.

Keywords
user intent understanding, search tasks, empirical studies on user behavior, intent complexity

1. INTRODUCTION
Inferring users’ intents has been a popular topic among Web and Information Retrieval researchers. But what is an intent? In his seminal work, Broder [1] defines an intent as “the need behind a query”. However this definition can not be leveraged for designing effective search engines that are aware of the mental model of the user. To the best of our knowledge, not much effort has been spent in order to deeply understand the mental path that connects this need to its textual representation in the form of queries.

In the attempt to reveal the basic nature of intent, namely what is in the user’s mind when they enter a search query, we conducted an experimental study [5] on 10 volunteers who agreed to give a verbose description of the intent of their queries during the search process. Such a study was quite informative about the nature of the intent and allowed us to gain deep insights into how intents are related to each other. A better understanding of the above aspects are fundamental in order to arrive to an operational definition of intent necessary to develop next generation tools for inferring user intents and refine the guidelines for annotators to make better decisions.

Previous studies are mainly focused on intent modeling or on clustering intents into categories or taxonomies. The tacit assumption on which most of the earlier work relies is that human annotators can easily infer the main intent of each query. However, whether or not human judges may infer user’s intents by observing their queries and/or the URLs they browse is still an open question. In the attempt of answering to such a question, we realize that more fundamental matters need to be investigated first. In particular, we need to understand i)what constitutes an intent and ii) what are the factors that affect the user to articulate what is in her mind in the form of queries.

Our qualitative analysis [4] has revealed that there are at least three main factors to consider when we look at the expressiveness of the queries; in other words, the competence of a set of queries to articulate the user intent:

1. the complexity of the task the user wants to accomplish; as explained in Section 3 such a complexity is measured in terms of the number of dimensions and choices that the user needs to explore at the time she engages with the search engine;
2. the number of dimensions already explored; such a factor determines how much effort needs to be spent by the user to complete the task;
3. the specificity of what she is looking for; each task can involve general or specific aspects of the task, which are influenced by the user’s past experiences.

Far from being exhaustive, the current analysis is radically novel and opens up future discussions about the relationship between queries and their underlying intents.

The rest of the paper is organized as follows. In Section 2 we explain the empirical study in detail and we present some of the collected data1. Section 3 presents the answers to the two main questions about intent definition and expressiveness of the queries and presents some further insights about intents. Section 4 discusses how our findings can be leveraged by search engines in order to satisfy users’ intents.

2. EMPIRICAL STUDY
In order to elaborate a definition of intents and mine the factors that influence query expressiveness, we asked 10 par-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$10.00.

1The complete list of queries can not be reported due to space constraints but all the conclusions reported here also apply to the remaining of the user sessions.
Participants to perform a number of search tasks. Each participant agreed to be video-recorded during her search session and let us collect quantitative data, that is, queries, search results, clicked urls, landing pages, dwell times, and so on. Before joining the video-session, the participants were asked to list up to 10 things they were interested in conducting a search about. Before starting the session they were instructed to use the search engine of their choice (Google, Y!, Bing, etc.) and they were given 55 minutes to perform their search tasks. It was not required to complete each search topic on the list since we wanted the participants to take as much time as needed to find the information they were looking for. During this time frame, the participants were encouraged to perform their searches as they would have normally done; e.g. reviewing any Web pages or information as needed, making changes, trying multiple times, etc. We used the “Think Aloud” protocol [2] to collect their thoughts throughout the search process. The participants’ sessions were recorded by a moderator who only interfered to remind them to keep talking out loud or to instruct them to move onto the next search topic on their list. At the end of the session, each participant was interviewed by the moderator to gather qualitative insights about what the user intent was.

3. OBSERVATIONS

We report the insights obtained by the analysis of 18 search tasks. Our findings are derived by a cross comparison between the set of the issued queries and the participants’ detailed explanations on what their actual intent was. Participant’s descriptions were definitely instrumental to understand the real intent of the queries and to assess the extent at which the queries express such an intent. Given space constraints, we can give compact summaries of only 5 of the 18 tasks:

- **Peruvian literature**: This was a very comprehensive session. The volunteer explained that all the different queries were related to the same unique intent. The intent was to replace some readings from a text book with easier and shorter readings from the Web. The list of things the teacher has in mind at the time of the query are the list of chapters and the topic of each chapter, which is related to a specific culture, and each query reflects that structure of the book.

- **Bugaboo stroller**: The participant wants to sell an used bugaboo stroller. Hence, she wants to determine the right price for it, considering that she has bought a number of accessories that she can add to the basic model. In none of the queries she mentioned that she want to sell as opposed to buy a stroller. It is communicated only verbally to the mediator but not to the search engine.

- **San Francisco restaurants**: The participant’s daughter’s 21st birthday is approaching. She is looking for places to eat in San Francisco. She is specifically looking for a restaurant she has never been before. She does not have a restaurant in mind, but she rather wants to see what is out there. Furthermore, she is seeking to find something fun to do after dinner that night. These are the list of things she has consciously in her mind at the time of the query.

- **A’s spring training**: The participant will have a trip to Las Vegas soon and would like to do research on things to do there. She is interested to know what others have done that she may be interested in doing if she has not done so already.

One major observation resulted from this exercise is that intent is always composed of two components: known and unknown. This is a simple observation in hindsight but it has not been well understood and exploited in the IR models. There is a part of the intent known to the user (from past experience, education, etc.) and the rest constitutes the unknown part. It is the unknown part that the user searches for. The size of the two components with respect to each other varies from intent to intent and user to user.

We provide more detailed discussions on this subject in Sections 3.1 and 3.2

3.1 Deeper Analysis of the Search Sessions

We studied the user search sessions in detail with respect to answering the following fundamental questions:

1. **What is the complexity of the search task?**
   - Does the search task consist of multiple factors to consider?
   - How much information needs to be conveyed to satisfy the searcher?
2. **What is the known component of the intent?**
3. **What is the unknown component of the intent?**
4. **How much effort does the searcher have to put to spot what she is looking for or complete the search task?**
   - Does a simple answer satisfy the user? e.g. someone searching for the weather forecast in Boston on a particular day will be satisfied by a simple answer; hence minimal effort is needed. - Does the searcher have to go through a number of different results to make a decision? e.g. someone looking to buy a car needs to investigate a number of different models, prices, mileage, etc.; hence major effort is required.
5. **Is the intent articulated well by the set of queries?**
   - If the intent is perfectly predictable from the search queries, then we conclude that the intent is totally expressed (TE).
   - If the intent is mostly clear from the search queries in the session, then we conclude that the intent is mostly expressed (ME).
   - If the intent can be guessed partially with significant confidence, then we conclude that the intent is fairly expressed (FE).
   - If there is little signal to predict the intent from the search queries, then we conclude that the intent is totally unclear (MU).
   - If the underlying intent can not be inferred from the search queries, then we conclude that the intent is totally unclear (TU).

Table 1 summarizes the answers for each of the above questions. It is worth to stress that such findings were in-
Figure 1: Expressiveness of the set of queries used in each task with respect to the dimensionality of the problem (X axis) and the effort to completion (Y axis).

ferred looking both at the search activity and at the users’ verbose description of what they are trying to achieve. Search tasks are classified according to a number of different aspects or dimensions the user needs to browse in order to be satisfied. Some tasks, like buying a house, are naturally complex since a lot of information needs to be conveyed to the user before she actually purchases the house. The complexity of the task is directly related to the effort for analyzing all the dimensions of the task. Effort is maximal if the user is at the beginning of his search task, i.e. if the unknown part of the intent is much larger than the known part. This is not always the case: in the example Vacation to Cabo, the user has already bought his holiday package so he has already analyzed all the possible dimensions (i.e., flights, hotels, sightseeing tours) and the effort to complete his current need (to know whether there is a better deal) is minimal. Motivated by the notion of where the users are positioned with respect to how much is unknown to them, the effort can be minimal, fair or maximal. The last aspect we considered is the generality of the task. In some examples like the bugaboo stroller or things to do in SF, the main aspect of the unknown part is related to the personal experience of the users or concerns very specific attributes of a more complex object. In the previous cases: we define the task specific. All the other tasks are considered to be general.

Figure 1 shows how generality and effort influence the expressiveness of the set of queries issued to satisfy an intent. If the task is very general the intent is always expressed (ME, FE) regardless of the effort taken to complete the task. Surprisingly the intent is poorly expressed in the most of the cases in which the search is specific. In such cases the expressiveness is also conditioned on the effort: the intent is poorly expressed in the case of major effort required whereas the expressiveness increases when the effort to completion is minimal.

3.2 Insights

During the lab study, we have discussed a number of useful insights that help us understand how users form their intents, and the characteristics of each intent. In this section, we discuss these insights in detail with supporting evidence from the search sessions.

- **Intents have a structure that reflects the user’s mental model (cognitive objects)** [2] at the time of their search for information. The structure may consist of several objects and the relationships between these objects and past experiences.

  In the Peruvian literature session, a latin literature teacher searches for readings to include into her course, and she relies on the structure of the text book to shape her search. The book contains chapters, each chapter refers to other books, each book has an author and authors have poems and articles. She follows the same structure when she forms her queries. Hence, they are not reformulations of the same intent or multiple intents, but rather individual pieces of the structure which as a whole defines her intent.

- **There can be a conjunction of intents in which one intent is combined with another in order to complete a task.** One intent may be prerequisite of another intent; hence they are executed in conjunction.

  In the bugaboo stroller example, the participant’s intent is to sell a used bugaboo stroller in order to buy a double jogging stroller. Hence, some of her queries serve for the selling intent, e.g. “prices for used bugaboo strollers”, and the rest serve for the buying intent; e.g. “double jogging stroller”.

- **Intent may not solely be limited to what someone wants/needs, but also what someone does not want/need.**

  In the example where the participant is looking for things to do in Las Vegas, he is specifically interested in the activities he has not done before during his past trips to Vegas. Similarly, in the example where another participant is looking for restaurants in San Francisco, she wants to find a restaurant that she has never been before. This important detail has not been explicitly conveyed to the search engine in neither case. Since search engines are tailored to retrieve most popular results, these participants will likely not be satisfied by the results. They will need to spend extra effort to spot what they were looking for.

- **Not all queries convey the intent.** Among the queries that are issued with the same underlying intent, there are some queries that do not express the intent. Depending on how articulate the user is, the ratio of such queries may vary.

  In the air traffic example, the searcher issues a total of 14 queries from which only 5 convey the intent whereas the rest is related but missing the fact that she is interested in knowing the job description and duties of a traffic management coordinator.

4. DISCUSSIONS

Our lab experiment has yielded a number of important findings and given us insights to re-visit how search engines should be designed. Depending on whether the user is seeking general or specific information, how well the queries express the intent varies. It is not enough to consider a single query in the session to predict the intent even in the case of well-articulated intents. Furthermore, the position of the user with respect to how much she knows about the search topic changes the criteria for her satisfaction. Although personalization is an emerging topic, the current search engines

---

5See http://donade.net/papers/2011/QueryRepresentationUnderstanding/queries.pdf for the example
do not differentiate the retrieval models according to how much information the user has acquired thus far. Ideally, the search engines could alter the results after the user visits a number of links and comes back to the result page. The change should clearly depend on how much the user’s knowledge has changed after her visit to the clicked sites. This is certainly a challenging task but it could make a big difference in terms of user satisfaction. Furthermore, the lab studies such as the one we described in this paper could be leveraged to determine if there are any common patterns in the unknown components of similar intents. If this is the case, then we can build models to capture these patterns focusing on the unknown (missing) piece of information.

The notion of result set modification based on the user’s current knowledge is only one component of a truly personalized search experience. We have observed that the search tasks are broader and more specific than an one-size-fits-all interface can adapt. We believe that personalization is far more beyond than simple information about the user’s gender, age, location, etc. True personalization should take into account an entire spectrum of information about the user; e.g., past actions and history, preferences, cultural and economical information, and so on. We further believe that mere diversification of the search results is inadequate to compensate for the search engine’s lack of knowledge of what is in the user’s mind. Every user is unique; hence, a general diversification scheme is far from satisfactory to truly cover all possible alternatives.

5. REFERENCES


Acknowledgments
We would like to thank Brian Ashbaugh and Prasad Kantamneni for their help during the early stages of this research.

Table 1: Analysis of the Sessions

<table>
<thead>
<tr>
<th>Task</th>
<th>Complexity</th>
<th>Known</th>
<th>Unknown</th>
<th>Effort</th>
<th>Clarity</th>
<th>generality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peruvian literature</td>
<td>complex</td>
<td>content of a course book, latin literature</td>
<td>simpler and shorter readings than those in the book</td>
<td>minimal</td>
<td>MU</td>
<td>specific</td>
</tr>
<tr>
<td>Things to do in SF</td>
<td>complex</td>
<td>restaurants been before</td>
<td>restaurants not been before, things to do after dinner</td>
<td>major</td>
<td>MU</td>
<td>specific</td>
</tr>
<tr>
<td>College scholarships</td>
<td>complex</td>
<td>daughter starting UC</td>
<td>college scholarships</td>
<td>major</td>
<td>ME</td>
<td>general</td>
</tr>
<tr>
<td>A’s spring training</td>
<td>complex</td>
<td>Oakland’s spring training in Phoenix</td>
<td>lodging, schedule, what to do</td>
<td>fair</td>
<td>FE</td>
<td>general</td>
</tr>
<tr>
<td>Bareroot roses</td>
<td>simple</td>
<td>the kind of roses</td>
<td>taking care of them</td>
<td>fair</td>
<td>ME</td>
<td>general</td>
</tr>
<tr>
<td>NFL playoffs 2008</td>
<td>simple</td>
<td>nfl playoffs 2008</td>
<td>expert picks, who will win</td>
<td>major</td>
<td>MU</td>
<td>specific</td>
</tr>
<tr>
<td>Vacation to Cabo</td>
<td>complex</td>
<td>trip already booked</td>
<td>the best deal possible</td>
<td>fair</td>
<td>TU</td>
<td>specific</td>
</tr>
<tr>
<td>Job Hunt</td>
<td>complex</td>
<td>wants a job</td>
<td>where to look, interview preparation</td>
<td>major</td>
<td>MU</td>
<td>specific</td>
</tr>
<tr>
<td>Buying a car</td>
<td>complex</td>
<td>buy a Lexus in 2-4 years</td>
<td>their future appearance, pros and cons</td>
<td>major</td>
<td>MU</td>
<td>specific</td>
</tr>
<tr>
<td>Iphone apps</td>
<td>simple</td>
<td>there exists apps for iphone</td>
<td>all apps, how to do things on iphone</td>
<td>minimal</td>
<td>ME</td>
<td>general</td>
</tr>
<tr>
<td>Samsung lcd</td>
<td>simple</td>
<td>brand and size he wants</td>
<td>models, reviews, prices</td>
<td>minimal</td>
<td>ME</td>
<td>specific</td>
</tr>
<tr>
<td>Estate for sale</td>
<td>complex</td>
<td>san jose houses for sale</td>
<td>prices, pictures, whom to contact</td>
<td>major</td>
<td>ME</td>
<td>general</td>
</tr>
<tr>
<td>Xbox</td>
<td>simple</td>
<td>xbox 360</td>
<td>news, future releases</td>
<td>major</td>
<td>ME</td>
<td>general</td>
</tr>
<tr>
<td>Bogaboo stroller</td>
<td>simple</td>
<td>bogaboo model owned, accessories</td>
<td>price to sell</td>
<td>minimal</td>
<td>MU</td>
<td>specific</td>
</tr>
<tr>
<td>Things to do in Vegas</td>
<td>complex</td>
<td>trip to vegas</td>
<td>interesting things not done before</td>
<td>major</td>
<td>FE</td>
<td>specific</td>
</tr>
<tr>
<td>Best restaurants in bay area</td>
<td>simple</td>
<td>review sites</td>
<td>particular review site</td>
<td>major</td>
<td>FE</td>
<td>general</td>
</tr>
<tr>
<td>2009 cars</td>
<td>simple</td>
<td>there will be new models for 2009</td>
<td>how the models will look like</td>
<td>minimal</td>
<td>FE</td>
<td>general</td>
</tr>
<tr>
<td>Car rental</td>
<td>complex</td>
<td>rental car reservation already done</td>
<td>coupons, deals</td>
<td>minimal</td>
<td>FE</td>
<td>specific</td>
</tr>
</tbody>
</table>
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 a directed 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. 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 or 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 our experiment, we observe that the dangling queries account for nearly 9% of the total queries, which is not negligible in real application.

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

Figure 1: The graphic model of generation of 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 detection. 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 $w_{ij}$. 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 $e_{ij} \in E$ denote a directed edge from query $q_i$ to query $q_j$. We assume the following generative process for each edge $e_{ij}$ in the query-flow graph:

1. Draw an intent indicator $g_{ij} = r$ from the multinomial distribution $\pi$
2. Draw query nodes $q_i, q_j$ from the same multinomial intent distribution $\beta_r$, respectively.
3. Draw the directed edge $e_{ij}$ from a binomial distribution $\tau_{ij,r}$.

Here, the $K$-dimensional multinomial distribution $\pi$ reflects the proportion of different search intents over the whole query-flow graph, the multinomial distribution $\beta$ over queries describes the hidden search intents, and the binomial distribution $\tau$ 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 $e_{ij}$ belonging to the $r$-th search intent can be obtained by

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

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

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

In this way, the likelihood of the graph $G$ is

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

(1)

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

The parameters to be estimated in our model are $\pi, \beta,$ and $\tau$. 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 $g_{ij}$. In E-step, the posterior probabilities of hidden variables are calculated as

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

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

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

$$
L = \sum_{i=1}^{N} \sum_{j \in C(i)} \sum_{r=1}^{K} w_{ij} q_{ij,r} \log \left( \pi_r \beta_{i,r} \beta_{j,r} \tau_{ij,r} \right)
$$
In M-step, we maximize the expected complete data log-likelihood which is

\[
LL = \sum_{i=1}^{N} \sum_{j \in C(i)} \sum_{r=1}^{K} w_{ij} q_{ijr} \log \left( \pi_{i} \beta_{r} \beta_{j} T_{ijr} \right) - \alpha \sum_{r=1}^{K} \tau_{r} - 1
\]

where for \( \alpha, \mu, \eta \) are lagrange multipliers. Taking the derivative with respect to \( \pi, \beta, \tau \) respectively gives the M-step re-estimations as follows

\[
\pi_{i} = \frac{\sum_{r=1}^{R} \sum_{j \in C(i)} w_{ij} q_{ijr}^{a}}{\sum_{r=1}^{R} \sum_{j \in C(i)} \sum_{r \in C(j)} w_{ij} q_{ijr}^{a}}
\]

\[
\beta_{jr} = \frac{\sum_{r \in C(j)} w_{ij} q_{ijr}^{a} + \sum_{k \in C(j)} w_{ik} q_{ikr}^{a}}{\sum_{r \in C(j)} \left( \sum_{j \in C(i)} w_{ij} q_{ijr}^{a} + \sum_{k \in C(j)} w_{ik} q_{ikr}^{a} \right)}
\]

\[
\tau_{jr} = \frac{w_{ij} q_{ijr}^{a} + w_{ik} q_{ikr}^{a}}{w_{ij} q_{ijr}^{a} + w_{ik} q_{ikr}^{a}}
\]

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

\[
A_{i,r} = (1 - \delta)M + \delta P_{i,r}
\]

where \( M \) denotes the weight matrix of the query-flow graph with row normalized, \( \delta \) denotes the teleportation probability, and \( P_{i,r} \) denotes the preference vector of intent-bias random walk under the \( r \)-th intent defined as

\[
P_{ij,r} = \rho \cdot e_{j}^{T} + (1 - \rho) \cdot \beta_{r}T_{ijr}
\]

where \( e_{j}^{T} \) is the vector whose entries are all zeroes, except for the \( r \)-th whose value is 1, \( \beta \) 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 is then applied to the query-flow graph for recommendation without addressing the hidden intents. In this way, for ambiguous queries, the results can be produced 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...
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.

<table>
<thead>
<tr>
<th>Query = yamaha motor</th>
<th>Query = radio disney</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>ours</td>
</tr>
<tr>
<td>mapquest</td>
<td>mapquest</td>
</tr>
<tr>
<td>yahoo mail</td>
<td>yahoo mail</td>
</tr>
<tr>
<td>bank of america</td>
<td>bank of america</td>
</tr>
<tr>
<td>target</td>
<td>target</td>
</tr>
<tr>
<td>baseline</td>
<td>baseline</td>
</tr>
<tr>
<td>ours</td>
<td>ours</td>
</tr>
<tr>
<td>yamaha</td>
<td>disney</td>
</tr>
<tr>
<td>suzuki</td>
<td>disney com</td>
</tr>
<tr>
<td>kawasaki</td>
<td>disneychannel com</td>
</tr>
<tr>
<td>disneychannel com</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Recommendations for Dangling Queries

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 $T$ is set to 0.8, and $\rho$ 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 data 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 select 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.

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 quantitatively and quantitatively.

6. ACKNOWLEDGMENTS

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

Table 3: Recommendations for Ambiguous Queries

<table>
<thead>
<tr>
<th>Query = hilton</th>
<th>Query = we</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>ours</td>
</tr>
<tr>
<td>marriott</td>
<td>[hotel]</td>
</tr>
<tr>
<td>expedia</td>
<td>marriott</td>
</tr>
<tr>
<td>holiday inn</td>
<td>holiday inn</td>
</tr>
<tr>
<td>hyatt</td>
<td>hotel</td>
</tr>
<tr>
<td>mapquest</td>
<td>embassy suites</td>
</tr>
<tr>
<td>hampton inn</td>
<td>hotels com</td>
</tr>
<tr>
<td>sheraton</td>
<td>[celebrity]</td>
</tr>
<tr>
<td>[celebrity]</td>
<td>paris hilton</td>
</tr>
<tr>
<td>residence inn</td>
<td>nicole richie</td>
</tr>
<tr>
<td>choice hotels</td>
<td>jessica simpson</td>
</tr>
<tr>
<td>hilton honors</td>
<td>pamelad anderson</td>
</tr>
<tr>
<td>[weather]</td>
<td>[wrestling]</td>
</tr>
<tr>
<td>bank of america</td>
<td>bank of america</td>
</tr>
<tr>
<td>[weather]</td>
<td>weather channel</td>
</tr>
<tr>
<td>[weather]</td>
<td>weather channel</td>
</tr>
<tr>
<td>[wrestling]</td>
<td>wwe</td>
</tr>
<tr>
<td>[financial]</td>
<td>wells fargo</td>
</tr>
<tr>
<td>[financial]</td>
<td>bank of america</td>
</tr>
<tr>
<td>[financial]</td>
<td>wells fargo</td>
</tr>
</tbody>
</table>

References


