People Search within an Online Social Network: Large Scale Analysis of Facebook Graph Search Query Logs

Nikita Spirin[1,2], Junfeng He[2], Mike Develin[2], Karrie Karahalios[1], Maxime Boucher[2]
Department of Computer Science[1] and Graph Search Team[2]
University of Illinois at Urbana-Champaign[1], Urbana, IL 61801 and Facebook[2], Menlo Park, CA 94025
{spirin2,kkarahal}@illinois.edu and {jfh,miked,maxime}@fb.com

ABSTRACT

Popular online social networks (OSN) generate hundreds of terabytes of new data per day and connect millions of users. To help users cope with the immense scale and influx of new information, OSNs provide a search functionality. However, most of the search engines in OSNs today only support keyword queries and provide basic faceted search capabilities overlooking serendipitous network exploration and search for relationships between OSN entities. This results in siloed information and a limited search space. In 2013 Facebook introduced its innovative Graph Search product with the goal to take the OSN search experience to the next level and facilitate exploration of the Facebook Graph beyond the first degree. In this paper we explore people search on Facebook by analyzing an anonymized social graph, anonymized user profiles, and large scale anonymized query logs generated by users of Facebook Graph Search. We uncover numerous insights about people search across several demographics. We find that named entity and structured queries complement each other across one's duration on Facebook, that females search for people proportionately more than males, and that users submit more queries as they gain more friends. We introduce the concept of a lift predicate and highlight how a graph distance varies with the search goal. Based on these insights, we present a set of design implications to guide the research and development of the OSN search in the future.

Keywords

graph distance; query log analysis; Facebook; people search

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering, Query formulation, Search process

1. INTRODUCTION

Online Social Networks (OSN) have revolutionized the way people communicate with each other by connecting millions of individuals on the same platform, e.g. Facebook on average had 802 million daily active users in March 2014[1]. Inside OSNs users constantly post status updates and share various aspects of their lives via photos, check-ins, and etc. This ease of content production coupled with the scale of OSNs have resulted in the exponential growth of information. Popular OSNs generate hundreds of terabytes of new data per day and contain billions of photos [3].

To help users cope with the immense scale and influx of new information, OSNs provide a search functionality. Typically, an OSN search engine allows users to submit keyword queries and retrieves OSN entities (people, photos, and etc.) based on a textual similarity of an entity description to a query with some adjustments for the entity popularity, recency of update, and proximity to a searcher. Examples include such OSNs as LinkedIn, Pinterest, and Foursquare. To facilitate discovery, OSNs usually provide a faceted search interface [42], which allows users to filter entities by expressing conditions on their attributes. However, until recently none of the existing OSNs provided an interactive interface for rapid and flexible formulation of structured queries or supported search for relationships between entities, e.g. "*Friends of my friends living in Shanghai*". Therefore, the search was still limited resulting in siloed information.

Human-Computer Information Retrieval (HCIR) [27, 28, 50] proposed a fresh paradigm to think about the search within an OSN – one which combines navigational and exploratory capabilities. HCIR aims to "*empower people to explore large scale information bases but demands that people also take responsibility for this control by expending cognitive and physical energy*", e.g. make an effort to write a structured query. Embracing this paradigm, in 2013 Facebook introduced its innovative Graph Search product with the goal to take the OSN search experience to the next level and facilitate exploration of the Facebook Graph beyond the first degree. Facebook Graph Search allows users to search for entities and relationships between entities and, following the terminology from [49], supports: (a) *interactive free-text queries*, like "*Lady Gaga*" (navigational queries for one entity by name); (b) *interactive structured queries*, like "*Photos of people who live in China*" (exploratory queries for filtering entities with conditions on attributes; the instances of a rich query grammar); (c) *one-shot free-text queries*, like "*query log mining*" (limited to users' status updates). All queries are separated into three different logs based on a query type.

Since Facebook Graph Search is the first search product of its kind operating at scale and used by millions of people,

[1]http://newsroom.fb.com/Key-Facts

CIKM'14, November 3–7, 2014, Shanghai, China.

http://dx.doi.org/10.1145/2661829.2661967 .

it is important to study its usage patterns and dynamics to guide the HCIR research and innovations in search within OSNs in the right direction going forward. In this paper we contribute by analyzing large scale anonymized query logs generated by users of Facebook Graph Search. We focus on the people search query logs because the people vertical is the most popular search vertical on Facebook. Only two query types, (a) and (b), were considered because the last one wasn't available in the logs provided for the analysis.

The main contributions of this paper are presented below. First, we analyze how named entity and structured queries co-exist within one search product and establish their complementary nature. This finding showcases the importance of each query type for enabling flexible search within OSNs. Second, we share unique insights about people search on Facebook related to a graph search distance via anonymized query logs and an anonymized social graph mining. Third, we perform a privacy-preserving demographic profiling and discover demographic-specific people search patterns. Finally, we present insights about structured grammar usage, which is unique to the Facebook Graph Search product.

In the next sections we cover related work, definitions, and background material; describe the data sets; and summarize high level properties of logs. Starting from Section 5, we share multiple insights about Facebook Graph Search usage. We present design implications in Section 8 and discuss limitations and conclusions of the work thereafter.

2. RELATED WORK

Research on people search has a long history. Early works studied ways to automatically assign reviewers to papers [12] or find topical experts [36] by representing people via documents they are associated with and framing people search as a traditional information retrieval or recommendation problem. However, as [40] noticed, it fundamentally changes the nature of the problem when the objects are people rather than documents. People form social relationships, and therefore, ranking people is qualitatively more complex than ranking textual documents. As a result, [40] proposed the concept of social matching to emphasize the social dimension. In turn, that led to the research and development of social matching systems, such as Referral Web [22], Expertise Recommender [30], and Aardvark [16], which return relevant people taking into account the social similarity between a candidate result and a searcher. [1] studied the fundamental properties of social networks making social search possible and concluded that "*where the data is incomplete or reflects non-hierarchical structure, tools that support social search should assist users by either providing a broader view of their local community or directly assisting users through a global analysis of the network data*" pointing out to the importance of search within OSNs. Recently, exploratory people search was investigated in the context of a People-Explorer project [15], which allows users to explicitly model their search preferences using sliders. Based on extensive experiments, the authors concluded that it is crucial to model task difference and user variance in people search.

Looking at the related work from a query log mining perspective, there is a large body of research dedicated to the analysis of web search engine usage. Starting from the influential taxonomy of web search queries [8], researchers studied query logs to understand how users search, analyzing the length [5,20,34,39], topical distribution [4,35], and temporal patterns [4] of queries. Query logs were also used to understand search sessions [21] and re-finding [38,44].

Because we are exploring people search, which is a type of vertical search operating in the people vertical, it is important to consider studies of vertical search engine logs. [31] studied queries issued to a blog search engine, and found that people were particularly likely to search for named entities, e.g. people and blogs on a topic of interest. [37] compared blog queries with news queries, observing that queries often refer to people and temporally relevant content. [39] compared microblog search and web search, and found that Twitter users similarly search for temporally relevant information and people. A study of a web people search query log [48] revealed that a significant number of users type just one query, that people search has lower click-through rates (CTR) compared to web search, and that the most popular results come from social media (OSNs). Recently, people search behavior and the role of graph distance in name and non-name queries was explored using the LinkedIn log [17]. It was reported that for name queries users primarily click on only one of the results and a shorter graph distance leads to higher CTR, while for non-name queries users are more likely to click on multiple results that are not among their existing connections, but with whom they have shared connections, i.e. the second degree connections.

Search logs were also used to uncover relationships between the search behavior and demographic characteristics of users. [6, 45] described the methodologies for usage of query logs and demographic profiles for search personalization. [47] presented the demographic-specific insights about search sessions and query topics. Several studies explored the influence of gender and age on search behavior. It is reported based on a series of interviews that males used search engines more than females in 2004 [13], but in 2012 the numbers equalized [32]. By analyzing web search engine usage, [25] found that females write longer queries than males. [10] investigated the search behavior of users retrieving information for children.

This paper complements and extends existing studies from three perspectives. First, while there is a large body of work on the topic of people search from an algorithmic side, there are only two people search query log studies [17,48] and only one of them is about search within an OSN [17]. We continue this line of work and answer previously untouched questions, e.g. "How does search behavior vary with age?". Additionally, we present novel insights about structured grammar usage unique to Facebook Graph Search, e.g. "What are the strategies for a name disambiguation?". Second, we extend the research on social search by studying query logs of an entity search system. Existing query log studies in the blogosphere [31], web [48], and Twitter [39] focused on document search systems. Third, our analysis uses logs generated by the system supporting several different query types, and therefore, we can do cross-type search behavior comparisons. Typically, search systems support only one query type.

3. BACKGROUND INFORMATION

In this section we describe the specifics of a user interface and provide definitions necessary to understand the paper.

3.1 User Interfaces

The search process starts by navigating to a Typeahead interface (Figure 1(a)). In the null-state, before any symbol

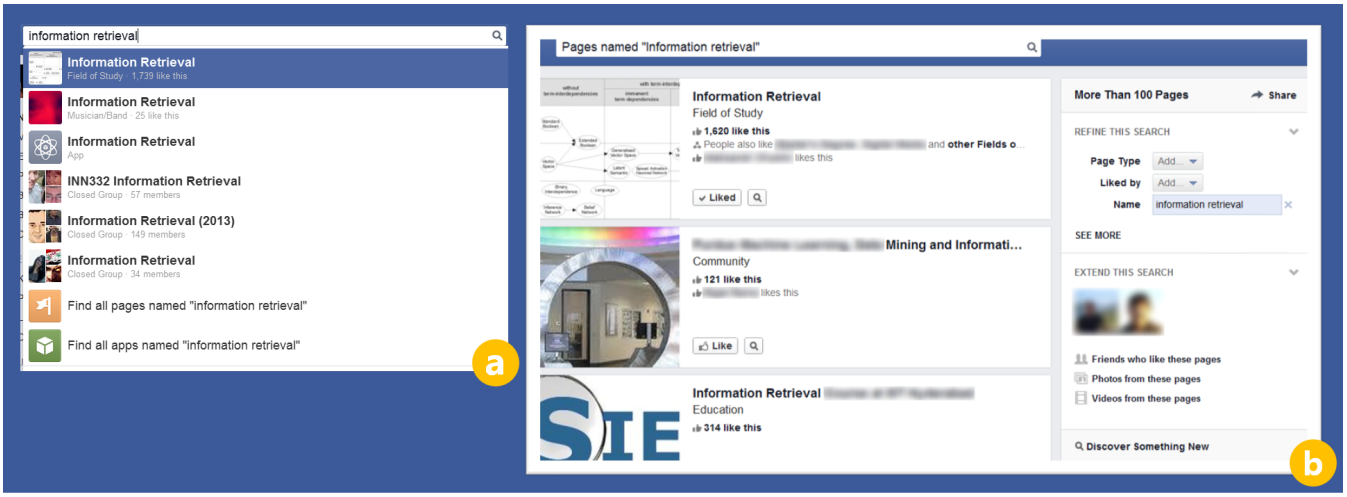

Figure 1: Search user interfaces: (a) Typeahead, which performs query suggestion for both named entity and structured grammar queries. It handles named entity queries by sending directly to a clicked entity page. (b) Browse, which presents results for structured grammar queries as a standard list with links and snippets.

is typed, Typeahead presents 7-8 query suggestions personalized for a searcher. These might include named entity queries or simple structured grammar queries. On every keystroke the Typeahead results get updated, and the user is presented with the input-dependent suggestions. At any point in time the user can pick a relevant result among the suggested options or keep typing. Suggestions, representing named entities, serve as search results directly, and upon click redirect to the corresponding entity page. Structured grammar queries lead to a standard search engine results page, called a Browse interface (Figure 1(b)), where the user is presented with a list/grid of entities matching the query conditions. For example, "*Lady Gaga*" is a named entity query, which upon click will navigate a searcher to the page maintained by Lady Gaga. "*People who like Information Retrieval and live in USA*" is an example of a grammar query, with three predicates (`people`, `like`, `residents`) and two entities (*USA*:Country, *Information Retrieval*:Field_of_Study). To effectively explore entities on Facebook, users may construct sophisticated structured grammar queries by concatenating predicate-entity pairs using a boolean `AND` operator. The search engine supports queries for entities in dozens of categories such as *Apps*, *Pages*, *People*, *Posts*, and others.

3.2 Definitions

Before we proceed to the analysis, it is useful to define some terminology. To make the definitions clear, where necessary we reference an example shown in Figure 2.

Person: In our case, people are represented as entities from: (a) a *User* category with all standard capabilities, such as friending, commenting and so forth; (b) human-like subcategories of a *Page* category such as *Athlete*, *Music Band*, *Politician*, and others – a broadcast-style account, typically used by celebrities to interact with their fans. One individual could have a *User* account and be an admin for several *Pages*. Unless otherwise stated, in the rest of this paper we focus on queries for Person from the *User* category. We only consider queries for Person from the *Page* category while discussing celebrity search in Section 6.

Celebrity: According to [29, 43], a celebrity is a "*highly visible in the media and overly public individual, who usually has emerged from the entertainment or sports industry and whose private life attracts greater public interest than the professional life*". For consistency, we define a celebrity as a Person with more than 10000 friends, fans, or followers.

Functional Predicate: In typed logic, F is a functional predicate with a domain type T and a codomain type U if, given any object X of type T, $F(X)$ is an object of type U [19]. The type of the predicate coincides with the codomain type. Similarly, a query type has the type of the result. Search grammar consists of numerous functional predicates, which perform typed mappings defined on the entities. For example, `friends` is a functional predicate that, given an entity of type *User*, produces a set of friends for this user and, hence, is a *User* predicate; `photos-in`, given an entity of type *Location*, produces a set of photos taken in that location and, hence, is a *Photo* predicate.

Semantic Query Template: Grammar queries consist of keywords, entities, and functional predicates, which can be combined using a boolean `AND` operator and functional superposition. An example query "*Photos of Alice and friends of Alice and males named Bob who live in California*" has a parsing tree shown in Figure 2(a). It has functional predicates as inner-nodes and keywords (a quoted string) and entities of types *User* and *State* as leaves. Because Facebook Graph Search is a highly personalized search engine, almost any two query parsing trees are different. However, semantics of these queries might be similar. For example, users may search for their friends by name, but names of the friends are likely to be different. Therefore, to study search patterns at a more general level, we categorize all grammar queries into factor classes [26] based on the structure of the corresponding parsing tree. First, we replace all leaves with the generic sentinel, e.g. "*Alice*", "*Bob*", and "*California*" are mapped to "*$*". Second, we sort tree nodes level-by-level using a lexicographic ordering on node names. Such factor classes are called Semantic Query Templates and each query goes to one class. An example is illustrated in Figure 2.

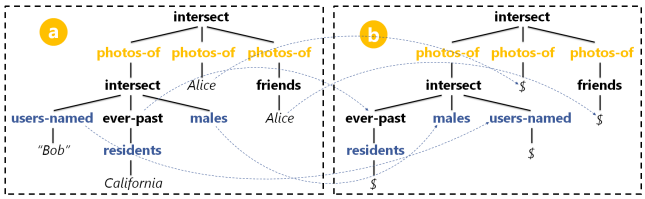

Figure 2: A grammar query (a) after a few transformations becomes a Semantic Query Template (b).

Graph [Search] Distance: The entities of all types (vertices) and various relationships between them (edges) form a Facebook Entity Graph (a Social Graph is an undirected subgraph of the Entity Graph made of only *User* vertices and `friends` edges). We use a traditional graph theoretic definition of the graph distance as the minimal number of edges connecting two given vertices [7]. We categorize all *User* queries into three major classes based on the graph distance between a searcher and a result:

- *Self*: queries for herself/himself;
- *Friend*: queries for friends, relatives, and many other entities connected to the searcher by an edge;
- *Non-friend*: out-of-network queries for friends of friends and many other entities not connected to the searcher.

For example, a user searching for a friend of a friend by name, e.g. "*John Smith*", performs a *Non-friend* query.

It is important to note that while named entity queries are not ambiguous based on the categorization above, the degree of a structured query is not strictly defined since it might contain multiple entities and various predicates. Therefore, for structured grammar queries we use the following distance calculation algorithm. We only consider grammar queries involving at least one *User* entity, such as "*Photos of* ***User1*** *and* ***User2***" or "*Places visited by* ***me***". We then look at each entity involved in a query and assign it a distance using the Social Graph and functional superposition of *User* predicates. Finally, we compute a bit vector with the three components, one for each of the three classes of graph distance, and normalize it by the number of non-zero components. Therefore, a grammar query might contribute to the weighted count for each of the graph distances.

For clarity, let us apply this algorithm to the query in Figure 2(a) by enumerating hypothetical searchers. Each *User* entity participates in a path from the root to this entity in the parsing tree: `intersect` → `photos-of` → `friends` → *Alice* and `intersect` → `photos-of` → *Alice*. If the searcher is Alice, the output vector is $(0.5, 0.5, 0)$, a half for herself and a half for friends of Alice. If the searcher is a friend of Alice, the output vector is $(0, 1, 0)$ because both paths are about Alice, who is one edge apart from the friend. If the searcher is not a friend of Alice, the output vector is $(0, 0, 1)$.

This and other custom data processing pipelines were implemented as MapReduce jobs [9]. Simple aggregation statistics were computed using Hive [41].

4. DATA SETS

We study people search behavior using four data sets collected at Facebook: (a) one which gives insight into the navigational search – an anonymized log of named entity queries; (b) another which gives insight into the exploratory search – an anonymized log of structured queries; (c) one which allows us to calculate graph distances – an anonymized Social Graph; (d) one which allows us to discover demographic-specific patterns – a set of anonymized User Profiles. Query logs were collected in the second half of 2013. The Social Graph and User Profiles were captured on 2013-10-17. The sizes of the data sets are given in Table 1.

Data Set	Attribute	Count
Named Entity Query (NEQ) Log	Users	$3M$
	Queries	$58.5M$
Structured Query (SQ) Log	Users	$3M$
	Queries	$10.9M$
Social Graph	Vertexes	$858M$
	Edges	$270B$
User Profiles	Users	$858M$

Table 1: Basic statistics about the data sets used.

4.1 Named Entity Query (NEQ) Log

The log contains named entity queries for Person from three million randomly sampled `en_US` (English, USA) users, who made at least one such query both in the month preceding the study and during the study periods. This allowed us to level novelty effects and observe a stable search behavior representative of a general population. Finally, after filtering we worked with 58.5 million queries. Each query log record includes such fields as an anonymized *id* of a searcher, an anonymized *id* of an entity to be searched, a time stamp, an entity type, and the query metadata.

4.2 Structured Query (SQ) Log

The log contains structured grammar queries of three million randomly sampled `en_US` (English, USA) users, who made at least one such query both in the month preceding the study and during the study periods. Unlike named entity queries, which are handled purely by Typeahead and always require a query writing or a suggestion selection, each structured grammar query and the corresponding Browse search engine results page (SERP) gets a unique URL, which can be shared and accessed without writing a query. We excluded such records from the log to make sure that it contains only authentic user-generated queries. Finally, after filtering we worked with 10.9 million structured grammar queries.

Participants in the Structured Query Log are independent from the participants in the Named Entity Query Log. Any overlaps are due to coincidence. Although the samples are independent and we don't join them, this doesn't prevent us from discovering general insights for both query types.

4.3 Social Graph and Demographic Profiles

We used a snapshot of the anonymized Social Graph made of 858 million entities and 270 billion edges. To gauge an understanding of search patterns for different demographic slices, we used an anonymized data set with the four user profile attributes: age, gender, number of friends, and celebrity status. Using information from the profiles, we performed checks for representativeness of our samples by comparing the key statistical properties of attribute value distributions for a sample and all 858 million profiles. The differences were insignificant because of the large sample size, which aligns with the reasoning on significance for big data [45].

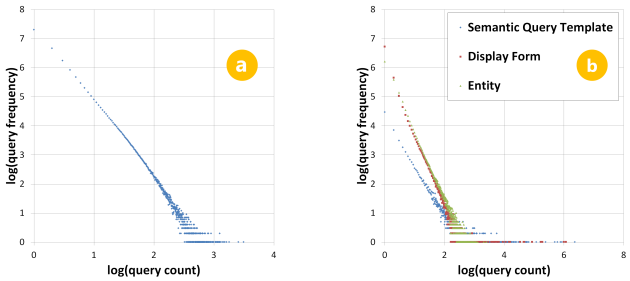

Figure 3: Power law graphs for query frequency of Named Entity (left) and Structured (right) Queries.

5. FIRST ORDER ANALYSIS

Characteristic of online usage behavior, we observed power law distributions for query popularity and user activity. Figure 3(a) shows the query frequency distribution in a log-log scale for NEQ, which follows a power law with the slope $\alpha = 2.63$. Figure 3(b) shows three query frequency distributions in a log-log scale for SQ. Each graph corresponds to a different definition of a unique grammar query: (a) one which at a display form level (the query string), e.g. "*Photos of **Alice** and **Bob***", has the slope $\alpha = 2.38$; (b) one which at an entity level, e.g. {***Alice***, ***Bob***}, has the slope $\alpha = 2.08$; (c) one which at a semantic template level, e.g. "*Photos of **$** and **$***", has the slope $\alpha = 1.15$. As we can see, most of the queries are issued only a few times; however, there are some very popular queries. The power law also holds for user activity with the slopes $\alpha = 2.43$ for NEQ and (a) $\alpha = 2.08$, (b) $\alpha = 1.90$, and (c) $\alpha = 1.13$ for SQ, which shows that there are some very avid searchers. This observation is predictable and aligns with existing log studies [2, 24, 48].

To shed light on the semantics of the most popular people queries, we computed the frequency distribution over Person subcategories of the *Page* category for the top-1000 celebrity *Page* queries for both NEQ and SQ, shown in Table 2. Both distributions are quite similar to each other and follow a power law. Musicians, public figures, and actors represent the three most popular celebrity *Page* categories.

Rank	NEQ	Perc.	SQ	Perc.
1	Musician/Band	32.2%	Musician/Band	27.7%
2	Public Figure	19.4%	Actor/Director	26.9%
3	Actor/Director	17.8%	Public Figure	19.1%
4	Entertainer	8.2%	Entertainer	8.4%
5	Artist	7.4%	Artist	6.4%
6	Athlete	7.3%	Athlete	5.2%
7	Character	2.5%	Character	2.3%
8	Comedian	2.2%	Politician	1.8%
9	Politician	1.9%	Author	1.3%
10	Author	1.1%	Comedian	0.8%

Table 2: Semantics of the top-1000 celebrity queries.

Contributing to the line of work on repeat queries [33,38], we looked at the query frequency from a different perspective and computed an average query repetition ratio per user, i.e. a fraction of unique queries out of all queries. Since in web search navigational queries are repeated differently from others [33], it is worth seeing how NEQ and SQ compare to each other in our scenario. We found that the repetition ratio for NEQ is 0.56. For each definition of a unique SQ, the repetition ratios are (a) 0.72, (b) 0.62, and (c) 0.47, respectively. Therefore, users *search for different people using SQ more than NEQ* and *repeat Semantic Query Templates to learn about different people*. Surprisingly, we found that the repetition ratios both for NEQ and SQ stay the same for all three classes of queries based on the graph distance. One interesting implication from it is that users repeatedly search for non-friends without adding them as friends. The repetition ratios stay the same for various demographic slices.

6. GRAPH SEARCH DISTANCE

Unique to this study are the insights about people search for various demographics and graph distances. The graph distance is the key parameter to quantify users' tendency for OSN exploration via search. The distribution over the graph distances is presented in Table 3. Other alternatives to quantify the OSN exploration via search are the number of unique queries or the repetition ratio. They have been considered in the previous section.

Query Type	Self	Friend	Non-friend
NEQ	0.6%	**57.6%**	41.8%
SQ	**5.2%**	31.2%	**63.6%**

Table 3: Query distribution over graph distances.

Users search for friends using NEQ and for non-friends using SQ. This demonstrates the importance and utility of having both query types to enable more effective exploration of the Social Graph. *Self* queries are negligible compared to an overall query volume, yet there is a significant difference between NEQ and SQ *Self* queries. Users search for themselves more using SQ. We think that this is because SQ are used to curate personal data published on Facebook, like "*My Photos*", while there are many other ways to navigate to a personal profile beyond using NEQ for yourself.

6.1 Influence of Demographic Characteristics on Graph Search Distance

Drilling down, we study people search behavior for different demographic slices. Among many available options, we picked the four attributes of the user profile: age, gender, number of friends, and celebrity status. Age and gender were used in multiple existing log studies [6, 11, 45–47], and it is interesting to compare their findings with the ones for Facebook. Number of friends and celebrity status are new attributes unique for this log study, which allow to capture the social dimension specific to search within OSNs.

We present a series of figures for each of the attributes in question. In the first column, we show fractions of the *Friend* queries out of all non-*Self User* queries for a set of bins; in the second column, we show the search trends for Named Entity *User* queries; in the third column – for Structured *User* queries. We focus on *Friend* and *Non-friend* queries, since *Self* queries are not common.

6.1.1 Age

Figure 4 presents how the graph distance varies with age. Looking at Figure 4(b), we see that *Friend* queries are more popular among NEQ for all age bins. On the contrary, according to Figure 4(c) the graph for SQ is bi-modal. While

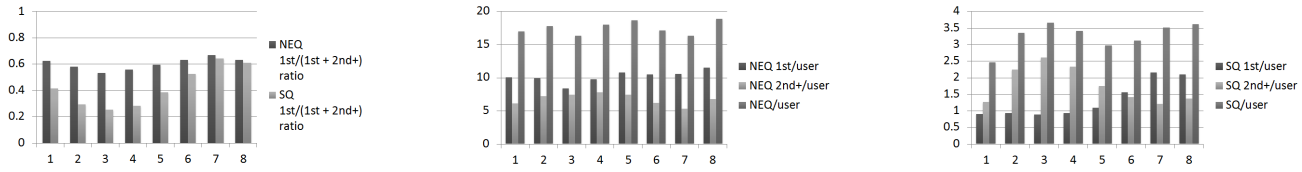

Figure 4: Search distance vs. Age (ten-year bins): (a) fraction of 1st degree queries out of all non-*Self User* queries; (b) average number of NEQ per user; (c) average number of SQ per user.

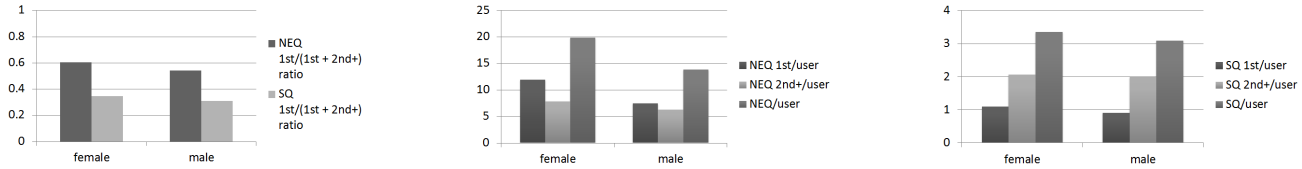

Figure 5: Search distance vs. Gender: (a) fraction of 1st degree queries out of all non-*Self User* queries; (b) average number of NEQ per user; (c) average number of SQ per user.

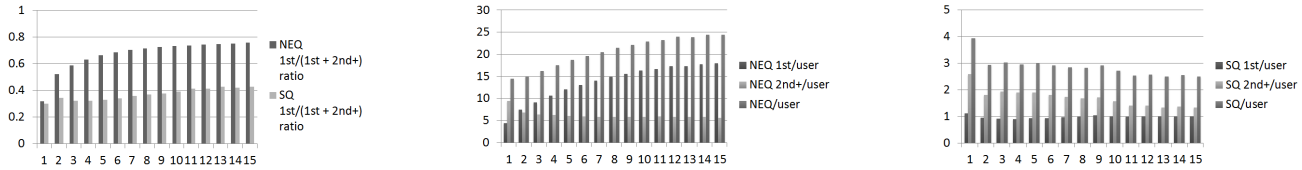

Figure 6: Search distance vs. Number of searcher's friends (100-friend bins): (a) fraction of 1st degree queries out of all non-*Self User* queries; (b) average number of NEQ per user; (c) average number of SQ per user.

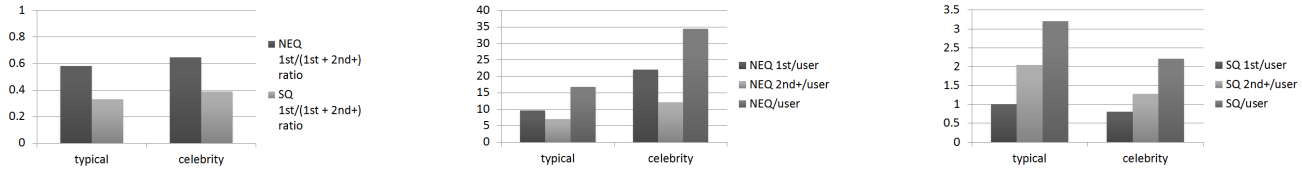

Figure 7: Search distance vs. Celebrity status:(a) fraction of 1st degree queries out of all non-*Self User* queries; (b) average number of NEQ per user; (c) average number of SQ per user.

Non-friend queries prevail for the younger group of users, *Friend* queries prevail for the older group of users. One can also notice a clear cyclic pattern in a combined Figure 4(a), which depicts the fractions of *Friend* queries out of all non-*Self User* queries for NEQ and SQ. Both graphs have one valley for users in their 30*s* and one peak for users in their 70*s*, which shows that the younger users more actively search for *Non-friends* and the older users more actively search for *Friends* relative to an average user. The graph for SQ has a higher variation than the graph for NEQ suggesting that the search personalization is more beneficial for SQ than NEQ.

6.1.2 Gender

It is reported in [13] that males searched more than females in 2004, but in 2012 the usage of search engines converged to the same level [32]. At the same time females communicate more and use the web less than males [18, 23]. We compared the search usage for female and male users in our case and found three insights. First, females write more queries than males (Figure 5(b,c)). Second, females more actively search for *Friends* using NEQ compared to an average user (Figure 5(a)). The fraction of *Friend* NEQ to the sum of *Friend* and *Non-friend* NEQ for females is 0.605, while for males it is 0.542. Third, according to Figure 5(a), we also found that males search more actively for *Non-Friends* using SQ compared to the mean for the population. The fraction of *Friend* SQ to the sum of *Friend* and *Non-friend* SQ for males is 0.328, while for females it is 0.348. The ideas based on these insights are discussed in Section 8.

6.1.3 Number of Friends

In Figure 6 we present how the graph search distance depends on the number of friends. First, note that the ratio of *Friend* to *Friend* and *Non-friend User* queries reaches

its saturation level at around 0.75 for NEQ and 0.43 for SQ (Figure 6(a)). Interestingly, the graph for SQ almost reaches its saturation already for the first bin (less than 100 friends), while the graph for NEQ grows steadily and saturates at around $10th$ bin (1000 friends). Second, the more friends a user has, the more *Friend* NEQ the user writes (Figure 6(b)). This suggests that users actively use NEQ to find information about their existing friends. On the contrary, the trend for *Non-friend* NEQ declines slightly with more friends. Therefore, we speculate that in this case people have already "friended" a good amount of users they want in their network. Third, it is worth noticing the volatility of graphs for different graph distances. The trend for *Non-friend* NEQ is flat, while *Friend* NEQ contribute to the growth of the query volume (Figure 6(b)). The trend for *Friend* SQ is flat, while the volume of *Non-friend* SQ changes depending on the number of friends (Figure 6(c)). This again suggests that NEQ are fundamental for searching for friends and SQ are fundamental for searching for non-friends. Together these two query types enable both navigational and exploratory people search on Facebook.

6.1.4 Celebrity Status

In Figure 7 we extend the analysis presented in the previous section and consider how celebrity status influences search behavior. This is an important question to study since search effectiveness and, hence, usage could be different in that extreme case (larger search space of friends). Because having 10000 friends is a rare event, there may only be a few celebrities in the original sampled query logs. Therefore, we considered a complete set of non-novice celebrity users of Facebook Graph Search and analyzed their anonymized queries during the same time intervals as for the original NEQ and SQ logs. According to Figure 7(b,c), on average celebrity users submit more NEQ and less SQ than typical users, which is in agreement with the findings presented in the previous section regarding the correlations between the number of queries and the number of friends. Moreover, from Figure 7(a) we conclude that the ratio of *Friend* queries to the sum of *Friend* and *Non-friend* queries for celebrities is biased towards the first degree connections compared to typical users. These findings suggest that celebrities are not as interested in exploring the graph and mostly use Facebook Graph Search for navigational purposes.

We deepen the analysis further by segmenting queries into celebrity and typical groups analogous to the approach we used for the searchers, i.e. we check whether a query contains an entity which has at least 10000 friends or fans. We computed the ratios of celebrity queries out of all non-*Self* $User$ queries for NEQ and SQ, two demographic groups (typical and celebrity), and graph distances (*Friends* and *Non-friends*). The results are presented in Table 4. It is worth highlighting that this experiment and the corresponding insights are about users searching for celebrities as $Users$ and not about users searching for celebrities as $Pages$.

According to Table 4, celebrities search more for other celebrities than typical users. To see whether this was a result of celebrities having more celebrity friends, and therefore, simply searching for friends, we calculated the average number of celebrity friends for a celebrity and found that it is 0.016[2]. The ratio of celebrity queries is higher than this, i.e. celebrities search for other celebrities disproportionally more than for typical users. Two other interesting findings are that both typical users and celebrities are more likely to search for a celebrity when they write a *Non-friend* query relative to a *Friend* query and using an NEQ relative to a SQ. Therefore, the surplus might be achieved by suggesting *Non-friend* celebrity NEQ to Facebook users in Typeahead.

[2]The same value for a typical user was infinitesimally small.

Type	Ratio	NEQ	SQ
Typical	Celebrity/Typical among friends	0.001	0.001
	Celebrity/Typical among non-friends	0.009	0.002
Celebrity	Celebrity/Typical among friends	0.167	0.067
	Celebrity/Typical among non-friends	0.247	0.123

Table 4: Relationship between a celebrity status of a queried entity and a celebrity status of a searcher.

7. STRUCTURED GRAMMAR USAGE

A prior query log study of a web people search engine [48] reports that users write queries with additional keywords to disambiguate a name of a person to be searched or to find relationships between people. The most used keywords were city names, jobs, and activities. However, the advanced functionality of that system was limited only to such keywords. We push this line of work forward and share insights about people search on Facebook, where users can write non-ambiguous grammar queries using a rich set of predicates.

7.1 Finding people using functional predicates

To understand how SQ length depends on the query popularity, we computed the average number of functional predicates for top-100 and top-1000 most frequent semantic query templates (SQ-A). We didn't use traditional definitions of the query length such as the number of letters or words [5, 20, 34] because structured queries consist of indelible entities and predicates. We additionally computed the lengths of SQ with at least one (a) `friends` predicate (SQ-F), (b) `friends-of-friends` composite predicate (SQ-FF), to understand how SQ length depends on the graph distance. The former serves as a proxy for *Friend* queries, while the latter for *Non-friend* queries. Table 5 summarizes the results.

SQ Type	top-100	top-1000
SQ-A	$\mathbf{1.64 \pm 0.59}$	2.00 ± 0.72
SQ-F	$\mathbf{1.01 \pm 0.61}$	1.70 ± 0.77
SQ-FF	$\mathbf{1.63 \pm 0.76}$	2.02 ± 0.89

Table 5: Average structured grammar query length measured as the number of functional predicates.

As we can see, shorter SQ is more popular because the average query length for top-100 is smaller than for top-1000. This is a predictable finding because shorter queries are likely easier to write. More interestingly, users write shorter queries when they search for the first degree connections. We have two ideas to explain this. First, the number of friends is much smaller compared to the number of non-friends, and hence, it takes less information to encode them (search entropy is lower). Second, according to [44],

"*repeat web queries are shorter and more effective*". Analogously, users might be more effective in formulating queries about their friends because they know more about them.

To understand what predicates people use to disambiguate SQ while searching for people, we computed top-10 predicates most frequently co-appearing in SQ with at least one *User* predicate having a clear distance semantics, i.e the same SQ-F and SQ-FF sets of queries. Table 6 summarizes the results. An exemplary SQ for the second column is "*Photos of **my friends** who like CIKM2014*"; for the third column – "***Friends of my friends** who are not my friends*".

Rank	in SQ-F	in SQ-FF	Lift predicates
1	*No predicates*	non-friends(me)	users-named
2	likers	residents	residents
3	residents	students	photos-of
4	employees	employees	friends(me)
5	members	likers	friends(X)
6	students	*No predicates*	non-friends(me)
7	females	users-named	females
8	visitors	friends(X)	photos-liked
9	home-residents	(employees, employer-location)	photos-by
10	(likers, likers)	home-residents	users-interested-in(males)

Table 6: Top-k predicates for different user queries.

The users disambiguate queries using predicates in the following groups: location – `visitors`, `residents`, `home-residents` (e.g. "*Friends who visited Dublin*"), affiliation – `employees`, `students`, `employer-location`, `members` (e.g. "*Employees of Tesla Motors*"), interest – `likers` (e.g. "*People who like Hadoop and Pig*"), gender – `females` (e.g. "*Females who are single*"), and relation to other users – `non-friends(me)` (e.g. "*Friends of Bob who are not my friends*"). They also submit queries for people without providing any predicate, and some queries that have predicates are more popular than queries without them. However, we found that on average, shorter queries are more popular. To investigate why some longer queries are more frequent than their shorter counterparts, we introduced the concept of a lift predicate.

A **Lift Predicate** is a predicate that, when used as part of a query, increases the frequency of the query compared to the query without this predicate. For example, if we have the query *"Users named Alice who live in California"* with the frequency 100 and the query *"Users who live in California"* with the frequency 50, then `users-named` is a lift predicate because the former query is more frequent than the latter.

The lift predicates are insightful because unlike in traditional query frequency analysis, which tells us *what* queries are popular, they help us understand *why* some queries are popular. Top-10 lift predicates are presented in the rightmost column of Table 6. In addition to the groups of predicates discussed above, users are interested in `photos`. We discuss how to leverage lift predicates in Section 8.

7.2 Functional Predicates and Graph Distance

In the previous section we studied what predicates people use to disambiguate their SQ in order to find other people. Below we study what people want to learn about other people and whether it depends on the graph distance.

Similar to the distance calculation algorithm described in Section 3.2, we consider only queries with at least one entity. For each entity in a query we (a) identify the distance from the searcher using the Social Graph or functional superposition of the *User* predicates, e.g. `friends(friends(me))` has a distance two; (b) find the closest non-*User* predicate, for which it serves as an argument unwrapping the functional embedding. For example, a query `photos-of(friends(me))` is a *Friend* query and the closest non-*User* predicate is `photos-of`; a query `videos-commented(123)` submitted by a friend of the user with the anonymous $id = 123$ is a *Friend* query and the closest non-*User* predicate is `videos-commented`; a query `photos-in(places-liked(me))` is a *Self* query and the closest non-*User* predicate is `places-liked`. The results for top-30 grammar predicates are presented in Figure 8.

Some predicates are "pure" because they can be applied only to the entities at a specific distance, e.g. `places-near`, while most of the predicates have all of the distance components. We observe a high variation from predicate to predicate and most of the predicates have a distribution over the graph distances deviating from the average distribution presented in Section 6. Users apply the `friends` predicate to non-friends, while interest-related predicates, e.g. `page-liked` and `videos-liked`, are biased towards friends. Users are also interested in knowing the locations of their first degree connections, e.g. `current-cities`. Job-related predicates, e.g. `employers`, are used more for non-friends, which aligns with Granovetter's theory of weak ties [14]. Some predicates with the related semantics, like media, have drastically different distributions: while `photos-of` is biased towards non-friends, `videos-of` is often used for friends.

8. DESIGN IMPLICATIONS

In this section, we discuss what our findings suggest for the design of next-generation search products within OSNs.

8.1 Supporting diverse information needs

Our key finding is that users search more for friends using NEQ and for non-friends using SQ (Section 6). Moreover, this behavior is consistent across numerous demographic slices (Section 6.1). Therefore, an interactive Typeahead interface supporting both NEQ and SQ facilitates navigation and exploration and makes information stored within Facebook useful and easy to search. Having these two query types tailored to a specific class of information needs within one system is beneficial for users of the OSN as these queries don't compete but rather complement each other. A similar idea was proposed based on the analysis of a LinkedIn people query log, where the authors found serious differences in post-search behavior for name and non-name queries.

8.2 Improving the quality of query suggestions

We noticed significant changes in search behavior for users with different demographics. We found that the number of *Friend* queries grows as users gain more friends, while the number of *Non-friend* queries slightly declines (Section 6.1.3), that celebrity users search for celebrities more than typical users (Section 6.1.4), that females and users, who are older than 60, are more interested in the first degree connections compared to the rest of the users in our sample (Sections 6.1.1,6.1.2), and several others. Therefore, we propose to further innovate around personalized search query suggestions given our demographics' distinct people search pat-

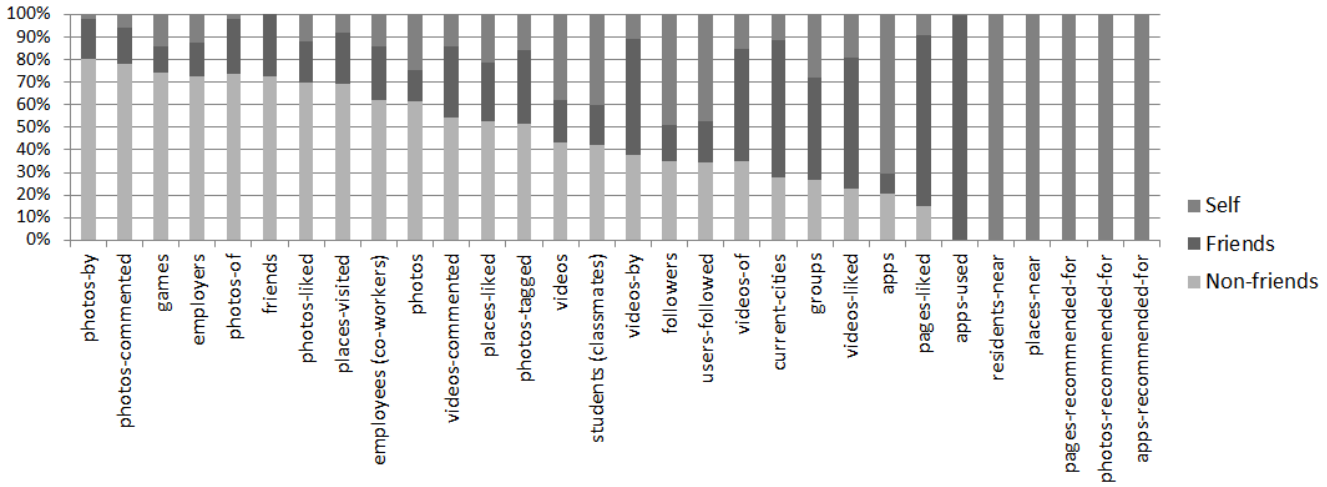

Figure 8: Distributions of graph distances for top-30 functional predicates. The distance is taken between a searcher and a user entity that serves as an argument for a functional predicate.

terns. It is worth mentioning that SQ usage behavior has a wider variation across different demographics, and hence, it makes sense to focus the efforts on that query type.

For example, we recommend building personalized grammar query suggestions at a Semantic Query Template level because we found that users reuse Semantic Query Templates to search for different entities. Moreover, users repeatedly search for non-friends without adding them to their friend network. This implies that query suggestions shouldn't be limited to friends only, but should also include some interesting distant vertices in the Social Graph.

We believe that the quality of search suggestions would improve using lift predicates by concatenating them to the *User* queries to facilitate entity disambiguation. Additionally, it would speed up the query writing process. Lift predicates are advantageous over frequent query suggestions because the former can boost the popularity of a new query, while the latter are limited to the set of existing queries.

We found that the distribution over graph distances varies from predicate to predicate. While some predicates are used primarily to explore information about friends, other predicates are used for non-friends. We propose ranking entities for a predicate using its graph distance distribution.

We discovered that users write shorter queries when they search for *Friends* and use more predicates to find *Non-friends*. Therefore, we propose generating interactive query suggestions by predicting an intended search distance and deciding whether (a) to add one more predicate to the original query to generate a list of longer queries or (b) to stop growing the query and iterate over the predicates applicable for the entity in question. For example, once a user specified a name of a friend, it makes sense to show other predicates used for friends, e.g. `pages-liked` or `videos-liked` (Figure 8). At the same time, if the user typed a name of a non-friend, it makes sense to extend the query using one of disambiguation or lift predicates, e.g. `employees` or `residents` (Table 6).

9. LIMITATIONS

We do acknowledge several limitations of this study. First, the paper uses proprietary data sets, which makes the reproducibility of the study possible only by a selected few. Yet we argue that the methodology underlying the reported analysis is general to allow similar analyses by other researchers with access to similar data sets. Moreover, it is the first large scale study of a search system providing structured querying capabilities for a casual user, which is important to distribute in the information retrieval community. Second, throughout the paper we consider three categories of queries based on the graph distance – *Self*, *Friend*, and *Non-friend*. However, we could have done more fine-grained analysis and calculated exact graph distances between entities by using more compute power. Third, some findings might depend on the concrete implementation of the Typeahead query suggestion algorithm, e.g. users might search for Non-friends more using SQ simply because they are shown such suggestions. However, the same applies to almost any query log study of a modern search engine. Finally, we used a quantitative approach, which knowingly has its own shortcomings. While it allows to uncover numerous data-driven insights at scale, it cannot uncover users' motivations and goals. Therefore, we can only speculate about the reasons underlying observed user behavior. Qualitative survey or interviews are necessary to backup our quantitative findings.

10. CONCLUSIONS AND FUTURE WORK

In this work we conducted a large scale analysis of Facebook Graph Search query logs. It is the first analysis that revealed insights on how demographic attributes and graph distance affect people search on Facebook. We presented a comprehensive overview of named entity and structured grammar queries usage and uncovered many new insights about people search within OSNs. The key takeaways from our study are two-fold. First, named entity and structured queries complement each other and it is crucial to have them both in one system to address a diverse set of information needs of users. Second, search behavior changes a lot for users with different demographics and it is important to model this variance to fulfil interests of everyone. We plan to extend this study with the qualitative research and understand user motivations underlying specific search behavior.

11. ACKNOWLEDGEMENTS

We thank the entire Facebook Graph Search Team for contributing their stellar work on various search subsystems, Shih-Wen Huang and anonymous reviewers for their valuable feedback on the drafts of this paper.

12. REFERENCES

[1] L. A. Adamic and E. Adar. How to search a social network. *Social Networks*, 27(3):187–203, 2005.

[2] R. Baeza-Yates and A. Tiberi. Extracting semantic relations from query logs. KDD '07.

[3] D. Beaver, S. Kumar, H. C. Li, J. Sobel, P. Vajgel, et al. Finding a needle in haystack: Facebook's photo storage. In *OSDI*, volume 10, pages 1–8, 2010.

[4] S. M. Beitzel, E. C. Jensen, A. Chowdhury, D. Grossman, and O. Frieder. Hourly analysis of a very large topically categorized web query log. In *Proceedings of SIGIR '04*.

[5] N. J. Belkin, D. Kelly, G. Kim, J.-Y. Kim, H.-J. Lee, G. Muresan, M.-C. Tang, X.-J. Yuan, and C. Cool. Query length in interactive information retrieval. In *Proceedings of ACM SIGIR '03*.

[6] B. Bi, M. Shokouhi, M. Kosinski, and T. Graepel. Inferring the demographics of search users: Social data meets search queries. In *Proceedings of WWW '13*.

[7] B. Bollobás. *Graph theory*. Elsevier, 1982.

[8] A. Broder. A taxonomy of web search. In *ACM Sigir forum*, volume 36, pages 3–10. ACM, 2002.

[9] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In *Proceedings of OSDI '04*.

[10] S. Duarte Torres, D. Hiemstra, and P. Serdyukov. An analysis of queries intended to search information for children. In *Proceedings of IIiX '10*.

[11] S. Duarte Torres and I. Weber. What and how children search on the web. In *Proceedings of CIKM '11*.

[12] S. T. Dumais and J. Nielsen. Automating the assignment of submitted manuscripts to reviewers. SIGIR '92.

[13] D. Fallows. Search engine users. *Pew Internet & American Life Project*, 2004.

[14] M. S. Granovetter. The strength of weak ties. *American journal of sociology*, pages 1360–1380, 1973.

[15] S. Han, D. He, J. Jiang, and Z. Yue. Supporting exploratory people search: A study of factor transparency and user control. In *Proceedings of CIKM '13*.

[16] D. Horowitz and S. D. Kamvar. The anatomy of a large-scale social search engine. In *Proceedings of ACM Conference on WWW '10*.

[17] S.-W. Huang, D. Tunkelang, and K. Karahalios. The role of network distance in linkedin people search. In *Proceedings of ACM SIGIR '14 Conference*.

[18] L. A. Jackson, K. S. Ervin, P. D. Gardner, and N. Schmitt. Gender and the internet: Women communicating and men searching. *Sex roles*, 44(5-6):363–379, 2001.

[19] B. Jacobs. *Categorical logic and type theory*, volume 141. Elsevier, 1999.

[20] B. J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information processing & management*, 36(2):207–227, 2000.

[21] R. Jones and K. L. Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. pages 699–708. ACM, 2008.

[22] H. Kautz, B. Selman, and M. Shah. Referral web: Combining social networks and collaborative filtering. *Commun. ACM*.

[23] A. Large, J. Beheshti, and T. Rahman. Gender differences in collaborative web searching behavior: an elementary school study. *Information Processing & Management*, 38(3):427–443, 2002.

[24] R. Lempel and S. Moran. Predictive caching and prefetching of query results in search engines. WWW '03.

[25] L. Lorigo, B. Pan, H. Hembrooke, T. Joachims, L. Granka, and G. Gay. The influence of task and gender on search and evaluation behavior using google. *Information Processing & Management*, 42(4):1123–1131, 2006.

[26] R. C. Lyndon, P. E. Schupp, R. Lyndon, and P. Schupp. *Combinatorial group theory*. Springer-Verlag Berlin, 1977.

[27] G. Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 2006.

[28] G. Marchionini. Toward human-computer information retrieval. *American Society for Information Science and Technology*, 2006.

[29] D. Marshall. *Celebrity and power: Fame in contemporary culture*. U of Minnesota Press, 1997.

[30] D. W. McDonald and M. S. Ackerman. Just talk to me: A field study of expertise location. In *Proceedings of ACM Conference on CSCW '98*.

[31] G. Mishne and M. de Rijke. A study of blog search. In *Proceedings of ECIR '06*.

[32] K. Purcell, J. Brenner, and L. Rainie. Search engine use in 2012. *Pew Internet & American Life Project*, 2012.

[33] M. Sanderson and S. Dumais. Examining repetition in user search behavior. In *Proceedings of ECIR '07*.

[34] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. In *Proceedings of SIGIR '99*.

[35] A. Spink, D. Wolfram, M. B. Jansen, and T. Saracevic. Searching the web: The public and their queries. *Journal of the American society for information science and technology*, 52(3):226–234, 2001.

[36] L. Streeter and K. Lochbaum. An expert/expert-locating system based on automatic representation of semantic structure. *IEEE Artificial Intelligence Applications*.

[37] A. Sun, M. Hu, and E.-P. Lim. Searching blogs and news: A study on popular queries. In *Proceedings of SIGIR '08*.

[38] J. Teevan, E. Adar, R. Jones, and M. A. S. Potts. Information re-retrieval: Repeat queries in yahoo's logs. In *Proceedings of SIGIR '07*.

[39] J. Teevan, D. Ramage, and M. R. Morris. #twittersearch: a comparison of microblog search and web search. In *Proceedings of WSDM '11*.

[40] L. Terveen and D. W. McDonald. Social matching: A framework and research agenda. *ACM Trans. Comput.-Hum. Interact.*

[41] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy. Hive: A warehousing solution over a map-reduce framework. *Proc. VLDB Endow.*, 2(2), Aug. 2009.

[42] D. Tunkelang. Faceted search. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 2009.

[43] G. Turner. *Understanding celebrity*. Sage, 2004.

[44] S. K. Tyler and J. Teevan. Large scale query log analysis of re-finding. In *Proceedings of WSDM '10*.

[45] I. Weber and C. Castillo. The demographics of web search. In *Proceedings of SIGIR '10*.

[46] I. Weber and A. Jaimes. Demographic information flows. In *Proceedings of CIKM '10*.

[47] I. Weber and A. Jaimes. Who uses web search for what: And how. In *Proceedings of WSDM '11*.

[48] W. Weerkamp, R. Berendsen, B. Kovachev, E. Meij, K. Balog, and M. de Rijke. People searching for people: Analysis of a people search engine log. In *Proceedings of ACM SIGIR '11*.

[49] S. Yogev, H. Roitman, D. Carmel, and N. Zwerdling. Towards expressive exploratory search over entity-relationship data. In *Proceedings of WWW '11*.

[50] G. Zenz, X. Zhou, E. Minack, W. Siberski, and W. Nejdl. From keywords to semantic queries-incremental query construction on the semantic web. *Web Semant.*, 7(3), 2009.