

# What Can Internet Search Engines “Suggest” About the Usage and Usability of Popular Desktop Applications?

*Adam Fourney, Richard Mann, Michael Terry*  
HCI Lab, David R. Cheriton School of Computer Science  
University of Waterloo, Ontario, Canada  
{afourney, mann, mterry}@cs.uwaterloo.ca

## ABSTRACT

In this paper, we show how Internet search query logs can yield rich, ecologically valid data sets describing the common tasks and issues that people encounter when using software on a day-to-day basis. These data sets can feed directly into standard usability practices. We address challenges in collecting, filtering, and summarizing queries, and show how data can be collected at very low cost, even without direct access to raw query logs.

**ACM Classification:** H5.2 [Information interfaces and presentation]: User Interfaces. - Evaluation/Methodology

**General terms:** Human Factors, Design

**Keywords:** Internet search, Query log analysis

## INTRODUCTION

People often rely on search engines (e.g., Google<sup>1</sup>) to support their use of common desktop applications. For example, users may search for tutorials or pose questions relating to the tasks they are interested in performing. As such, query logs serve as an excellent record of the tasks and usability problems faced by an application’s user base. With proper analysis, these logs can yield rich, ecologically valid data that can feed into design and usability processes (e.g., heuristic evaluations, cognitive walkthroughs, and think-aloud studies). Compared to other means of collecting these data (e.g., surveys), the use of query logs is very fast and very cheap.

Query logs are highly valued in related fields of study. Most notably, logs are often used to improve information interfaces such as search engines (e.g., [3]) and website navigation (e.g., [5]). It has also been suggested that query logs be used to guide medical and social science research, with log analysis functioning as if “a survey were sent to millions of people asking them to, every day, write down what they were interested in, thinking about, planning, and doing” [6]. In the realm of programming, Joel Brandt et al. have also conducted extensive research exploring how software developers use web search when writing and debugging code [2].

<sup>1</sup><http://www.google.com>

To the best of our knowledge, however, query logs have not been used to characterize usage patterns and usability issues with rich desktop applications.

In the remainder of this abstract, we answer three challenges that must be addressed when mining usability information from search query logs. First, we describe a low-cost approach for public sampling of query corpora, noting that actual search query logs are almost never released publicly. This approach involves exploring the auto-completion and query suggestion services provided by many search providers (e.g., the type-ahead feature found in Google’s search bar). Next, we demonstrate how to filter sampled queries to select for those that are most likely to be task-oriented and useful for usability analysis. Finally, we describe methods for summarizing the data in order to provide a high-level overview to designers, UX engineers, and HCI researchers.

## HARVESTING SEARCH QUERIES

Search query logs are not made public by search providers. However, as was first reported by Ziv Bar-Yossef and Maxim Gurevich in [1], modern search engines (e.g., Google) provide indirect access to their logs through query suggestion services. These services automatically complete partially entered queries, providing the top 10 completions that match the input already provided. The suggestions, and their listing order, are derived from the prominence of the queries in the logs. Bar-Yossef and Gurevich used this approach to valuate keywords for advertising purposes.

We adapted this approach for exploring the usage and potential usability problems of popular software. As an example, consider query suggestions for the popular free software raster graphics program *GIMP* (short for “GNU Image Manipulation Program”). To complete the phrase “gimp how to”, Google suggests the following:

```
gimp how to ...  
1. make background transparent  
2. use layers  
3. cut out image  
4. make transparent  
5. crop  
6. curve text  
7. blur background  
8. change background  
9. change resolution  
10. draw a line
```

Simply entering the phrase “gimp how to” provides a very quick indication of some of the primary issues GIMP users encounter when using the software. To obtain a more complete sampling of the query logs, we generate the full tree of suggestions for a given prefix by expanding partial queries one character at a time (Figure 1).

```

gimp,
gimp a, gimp aa, gimp aaa, ...
...
gimp z, gimp za, gimp zaa, ...

```

Figure 1: Input sequence representing a depth-first traversal of Google’s prefix tree rooted at “gimp”.

A systematic search of query suggestions incorporating the term “GIMP” uncovers 14,559 additional unique query completions. Similar results were obtained for other popular open source applications (Table 1).

Application	# of Query Suggestions
ubuntu	122,242
firefox	74,795
audacity	6,517
inkscape	2,501

Table 1: Number of unique query suggestions provided by Google for some popular open source applications.

Currently, query suggestion interfaces do not report the frequency that each of the searches has been performed, thus a total ordering of all the suggestions is not possible. Instead, a partial ordering can be inferred from the structure of the generated search tree [1]. Returning to the GIMP example, “gimp transparent background” appears in the top 10 suggestions for the prefix “gimp t”. Thus, we can infer that this query is more popular than the 843 other suggestions found when generating the complete subtree for the “gimp t” prefix.

### FILTERING TASK-CENTRIC QUERIES

Once search suggestions have been harvested, one must employ filtering mechanisms to remove queries unrelated to tasks and usability issues. As noted in taxonomies of search queries (e.g., [4]), there are numerous classes of queries, each serving a different user need. We are interested in what Melanie Kellar et al. have termed “fact finding” queries [4], where “the goal of the user is to retrieve some short, specific information, possibly on one page.” Fact finding queries tend to be more specific than most, containing an average of 4.7 terms. Thus, a simple filtering heuristic is to select for queries that contain at least 4 terms.

Alternatively, task and usability-related queries can be selected for by considering only those suggestions that contain certain whitelisted keywords or phrases (e.g., “how to”, “tutorial”, “example”, “use”, “make”, “create”, “can”, “can’t”, etc.). As an example of the effectiveness of this simple approach, we randomly selected 100 of the most popular suggestions related to GIMP (sampling from the top two levels of the prefix tree). We manually coded each of these suggestions as either task-centric or not task-centric. Of the 49 task-centric suggestions, 37 (75%) queries either contained the phrase “how to”, or were expressed elsewhere in the corpus with the phrase “how to” incorporated. Only 10 (20%) of the non task-centric queries had this same property.

### SUMMARIZING AND VISUALIZING THE QUERY SPACE

After applying the filtering techniques outlined above, many thousands of unique queries may remain. These queries must be reduced, organized, and summarized to be of use. In the

<b>gimp</b>	
<b>image</b>	transparent, cut, out, two, change, size
<b>use</b>	tool, brush, clone, path, scissor, heal
<b>layer</b>	open, get, select, transparent, add, multiple
<b>transparent</b>	background, image, layer, white, color, erase
<b>brush</b>	install, add, use, download, get, load
<b>background</b>	transparent, change, remove, white, delete, add
<b>change</b>	color, background, size, font, image, text
<b>text</b>	change, curve, edit, rotate, bend, remove
<b>color</b>	change, eye, replace, hair, another, splash
<b>draw</b>	line, rectangle, curve, circle, arrow, shape
<b>add</b>	brush, font, border, background, layer, watermark
<b>picture</b>	cut, look, out, put, background, change
<b>install</b>	brush, plugin, font, script, gap
<b>font</b>	install, add, change, download, into, put
<b>photo</b>	black, edit, white, use, color, collage

Figure 2: A visualization of the query suggestions related to GIMP which contain the phrase “how to”. The leftmost column lists, in descending order, the words that occur most frequently in the query suggestions. Similarly, each row lists the words that co-occur most frequently with those listed in the left column.

simplest case, the data can be summarized by tabulating frequencies of terms, bigrams, and co-occurrences. Frequencies can then be presented using various visualization techniques such as tag clouds, word graphs, and the stylized table presented in Figure 2. These visualizations can help researchers identify popular topics and trends in the data, and can direct targeted in-depth exploration of the query logs. For example, Figure 2 suggests that GIMP users perform many searches relating to transparency, layers, and drawing primitives (e.g., straight lines, rectangles, circles, etc.). This latter problem (how to draw geometric primitives) is noteworthy because GIMP provides few tools for drawing simple shapes. The prevalence of these queries, coupled with the lack of these tools, demonstrates the viability of this approach in identifying unmet user needs.

### FUTURE WORK

Query log analysis holds great promise to inform the design of desktop applications. We are now pursuing more sophisticated methods for summarizing the data. For example, we are using data returned from executing the suggested queries to improve our clustering of results. We are also building interactive tools to simplify exploration of the query space.

### REFERENCES

1. Ziv Bar-Yossef and Maxim Gurevich. Mining search engine query logs via suggestion sampling. *Proc. VLDB Endow.*, 1(1):54–65, 2008.
2. Joel Brandt, Philip J. Guo, Joel Lewenstein, Mira Dontcheva, and Scott R. Klemmer. Two studies of opportunistic programming: interleaving web foraging, learning, and writing code. In *Proc CHI '09*, pages 1589–1598, New York, NY, 2009. ACM.
3. Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. Probabilistic query expansion using query logs. In *Proc WWW '02*, pages 325–332, New York, NY, USA, 2002. ACM.
4. Melanie Kellar, Carolyn Watters, and Michael Shepherd. A field study characterizing web-based information-seeking tasks. *J. Am. Soc. Inf. Technol.*, 58(7):999–1018, 2007.
5. W. Quesenbery, C. Jarrett, I. Roddis, S. Allen, and V. Stirling. Search Is Now Normal Behavior. What Do We Do about That. In *UPA 2008*, Baltimore, Maryland, USA, June 2008.
6. Matthew Richardson. Learning about the world through long-term query logs. *ACM Trans. Web*, 2(4):1–27, 2008.