
This is the author’s final accepted version.

There may be differences between this version and the published version. You are advised to consult the publisher’s version if you wish to cite from it.

http://eprints.gla.ac.uk/158450/

Deposited on: 26 March 2018
Characterizing Search Behavior in Productivity Software

Horatiu Bota  
University of Glasgow  
Glasgow, Scotland, UK  
h.bota.1@research.gla.ac.uk

Adam Fourney  
Microsoft Research  
Redmond, Washington, USA  
adamfo@microsoft.com

Susan T. Dumais  
Microsoft Research  
Redmond, Washington, USA  
sdumais@microsoft.com

Tomasz L. Religa  
Microsoft Corporation  
Redmond, Washington, USA  
toreli@microsoft.com

Robert Rounthwaite  
Microsoft Corporation  
Redmond, Washington, USA  
robertro@microsoft.com

ABSTRACT

Complex software applications expose hundreds of commands to users through intricate menu hierarchies. One of the most popular productivity software suites, Microsoft Office, has recently developed functionality that allows users to issue free-form text queries to a search system to quickly find commands they want to execute, retrieve help documentation or access web results in a unified interface.

In this paper, we analyze millions of search sessions originating from within Microsoft Office applications, collected over one month of activity, in an effort to characterize search behavior in productivity software. Our research brings together previous efforts in analyzing command usage in large-scale applications and efforts in understanding search behavior in environments other than the web. Our findings show that users engage primarily in command search, and that re-accessing commands through search is a frequent behavior. Our work represents the first large-scale analysis of search over command spaces and is an important first step in understanding how search systems integrated with productivity software can be successfully developed.

KEYWORDS

Environment specific retrieval, command search

1 INTRODUCTION

In a 2007 column, Don Norman argued that graphical user interfaces would struggle to scale under the increasing complexity of software applications, and that scaling could be achieved by embracing rich search experiences [19]. This transformation was predicated on search interfaces gaining, or strengthening, an ability to directly execute actions in addition to retrieving documents. For example, on both Bing.com and Google.com, the query "stopwatch" both summons an interactive stopwatch, and retrieves documents about this timepiece. Likewise, on both the Windows 10 and macOS operating systems, the query "bluetooth" both accesses the device’s Bluetooth settings, and optionally retrieves web pages about Bluetooth technology. Finally, users of both Microsoft PowerPoint and Adobe Photoshop can issue the query "crop" to either execute the application’s crop command, or to retrieve documentation about this tool.

Core to these experiences, and to Norman’s vision, is that search systems effectively respond to queries where the searcher’s intention is to perform some action (web-mediated, or otherwise). Such search intents were first observed in the query logs of the AltaVista search engine [4, 21], and have varyingly come to be known as transactional queries [4], or resource interaction) queries [21]. Historically, transactional queries represent a minority of all web searches.

More recently, researchers have begun characterizing search interactions occurring via the virtual assistants that are integrated into modern operating systems [7, 15]. In this environment, log analysis has shown that roughly half of all queries result in the direct execution of an action (e.g., setting an alarm) [15]. As such, virtual assistants represent a mid-point along a spectrum of search systems that support direct execution of transactional queries.

In this paper, we provide the first large-scale log-based characterization of search interactions within the context of productivity software, namely Microsoft Office (Figure 1). Microsoft Office, and similar in-application search experiences (e.g., [3, 16]), represent an extreme point along the aforementioned spectrum – an under-studied search environment where most queries are intended to execute an action. To investigate this environment, we analyze the queries of a million Microsoft Office users, engaging in millions of search sessions over a one month period in 2017. With these data, we address the following research goals:
(RG1) Describing search behavior: In particular, we answer the following questions with respect to search in productivity software: (1.1) How is search distributed across queries, accessed commands, and users? (1.2) What types of search do users engage in and how frequently? (1.3) How do users engage in different types of search activity? Specifically, we want to characterize how users engage in command and control search in contrast to informational search.

(RG2) Describing search abandonment: Users frequently abandon their sessions without clicking on results returned by the search system. We set out to contrast abandoned search to searches in which commands are executed, and we highlight properties of search results and rankings that potentially influence abandonment.

(RG3) Describing re-ranking methods: The use of behavioral signals in improving search results ranking is widespread in modern web search [2, 25]. Therefore, we explore methods of using historical user interaction data and simple user engagement metrics in re-ranking command lists, with the intent of improving ranking quality in rich application search.

The rest of this paper is structured as follows: We review prior research, then describe the search experience that is integrated into many Microsoft Office products. We describe the specific instrumentation data we analyzed, and present results that address our research objectives. We conclude with a discussion of the implications of our findings for the development of in-application search experiences.

2 RELATED WORK

This work builds on three distinct threads of prior research which we review here. We begin by discussing transactional queries in the context of web search, and their generalizations to other domains. We then review research that characterizes command invocations in feature-rich software. Finally, we review past and ongoing efforts to integrate rich search experiences into feature-rich software.

2.1 Transactional Queries

Transactional queries were first characterized by Andrei Broder as web searches whose “intent is to perform some web-mediated activity” [4]. Broder interpreted this definition quite broadly, including queries intended to initiate e-commerce, as well as those intended to access phone directories, weather services, and other on-line databases. Upon examining 400 searches performed on the AltaVista web search engine, Broder reported that 30% of all queries met this definition. Other work has consistently placed transactional queries in the minority of all web searches, though the precise proportion varies by search engine [20], and by query topic [13].

In follow-up work [21], Rose and Levinson proposed a refinement to Broder’s taxonomy, subdividing transactional queries into four subcategories. Relevant to this work is the subcategory “interact”, whose queries include those where the searcher’s “goal is to interact with a resource” (e.g., a weather service, or a unit converter).

Again using AltaVista search logs, Rose and Levinson reported that between 4.6% - 6.0% of all queries met this narrower definition.

Web search engines have since moved to directly support Rose and Levinson’s interact queries by presenting tailored experiences directly on search engine results pages. In 2011, Chilton and Teevan reported that the Bing.com search engine supported nearly 100 such experiences [5]. Separately, Stamou and Efthimiadis studied donated search logs, and reported that 27% of web searches were intended to trigger these types of quick answers [22]. These trends prompted Don Norman to observe that “more and more, we type commands (into search engines), not search items”, and to further describe search as the “modern command language” [19].

Importantly, Norman’s observations were not confined to web search: desktop search, email search, and an early version of Microsoft Office search are explicitly mentioned in [19]. Broder’s definition of transactional queries naturally generalizes to these environments by simply lifting the constraint that the intended activities be web-mediated. Through this lens we consider work by Jiang et al., who analyzed queries posed to the Cortana virtual assistant [15], the successor of desktop search on the Windows 10 operating system. Jiang et al. reported that roughly half (47%) of all queries result in the direct execution of an action (e.g., setting an alarm, or playing a song) and thus meet the generalized definition of transactional queries. Moreover, this figure may underestimate the prevalence of transactional queries, as it excludes utterances that have a transactional intent, but which fail outside the capabilities of the virtual assistant [7, 24]. In this paper, we extend this generalization of transactional queries, and Broder’s query taxonomy, by characterizing search behaviors in Microsoft Office – one of several well-known applications to recently allow commands to be triggered through in-application search.

2.2 Command Invocations in Software

As noted above, Norman characterized search as the “modern command language” [19], and, in doing so, drew a direct comparison to a prior generation of command-line interfaces. Command-line systems have themselves been the subject of extensive study [10–12], and there are several relevant findings that we report here. First, as with web search [14], command invocations are known to follow an inverse power law distribution, with the most common commands occurring exponentially more often than less popular commands [11, 12]. Second, though each user will tend to frequent a small set of commands, there is little overlap in the command vocabularies between users [11]. Finally, even when users have the same commands or intentions in mind, they will often use different words or terminology to describe these intentions to the system, posing problems for command-line interfaces and search systems alike [10]. In this paper, we touch on each of these points, as they relate to commands issued in Microsoft Office via a search interface.

Graphical user interfaces (GUIs) are also command languages – albeit visual rather than textual. To this end, research has found that the aforementioned properties of command-line interfaces also apply to GUIs [17], and to the web search queries that users issue to support their use of those systems [6]. This has spawned work on two fronts: First, numerous command recommendation systems have been developed to help with command discovery, and to broaden users’ command vocabularies [9, 18, 26]. Second, to address the vocabulary problem, researchers have proposed integrating search experiences into graphical interfaces, including Microsoft
Office, so that a user’s free-text queries can be more flexibly matched to system commands. We describe these search systems next.

### 2.3 In-Application Search

Search is becoming a popular method of accessing commands, settings, and other functionality in modern feature-rich applications. In addition to Microsoft Office, Adobe’s creative products now feature a universal in-application search experience [16]. Likewise, the macOS operating system has long offered a search field in the system-wide help menu, providing its users with a means of searching and executing commands found in the top few levels of any application’s menuing system [3]. Similarly, search is the primary means of accessing settings in the Chrome web browser, and on all modern operating systems and platforms (Windows, macOS, iOS, Android, etc.)

Embedded in each of these in-application experiences is the need to return relevant commands, settings, and documents in response to user queries. In GUIs, there is often very little text associated with each command or setting, aggravating the aforementioned vocabulary problem, and limiting the effectiveness of simple ranking algorithms that depend on term or character overlap [6]. In the literature, several systems address this retrieval problem. For example, Fourney et al. [8] leveraged web search queries, together with on-line software tutorials, to better map keywords to system commands. Likewise, Adar et al., improved command search by learning a command-to-word embedding from millions of on-line tutorials [1]. In this work, we demonstrate how behavioral data can be used to improve command ranking.

In summary, our work extends prior research by characterizing transactional queries and command invocations in the context of a widely deployed suite of software applications. As such, our findings are derived from a comprehensive analysis of in-application query logs, and our work is the largest such study of its kind. We begin by providing a review of search functionality in Microsoft Office.

### 3 SEARCH IN MICROSOFT OFFICE

Microsoft Office is a collection of productivity software developed for knowledge workers, with millions of active users each month. Recent versions of Office1 include a unified search interface that allows users to issue free-form text queries to either find and quickly access commands, search through Office help documentation or retrieve Web results directly in the application interface. The unified search functionality is provided through a search box that prompts users to “Tell me what you want to do”, located above the application command menu (the ribbon).

In this paper, an in-application “search” starts when the user activates the query input area (the search box), and ends when the user clicks one of the results returned by the system, or deactivates query input by clicking outside the search area. This section provides an overview of search in Microsoft Office, and describes the types of results returned by the system.

#### 3.1 Result Types

Three types of results are accessible through Office search: commands, help documents and web results. We describe each below.

**Command Results.** Figure 1 shows an example of a Microsoft Word user issuing the query “pr” to the search system. Up to five command results are displayed immediately underneath the search box, and are updated continuously as users type. Results are ranked by their relevance to the user query, and need not be lexically similar to command names.

Command results returned by the search system can be of three sub-types: (i) **action commands** that initiate a process (for example, the “Print” command opens an additional dialog that users need to navigate in order to complete their task); (ii) **flag commands** which toggle a binary feature (such as the “Bold” command) without requiring any additional action from the user; (iii) **menu commands** which, on user click, open an additional side menu containing related sub-commands that the user can browse and click – in Figure 1, menu commands are displayed with a chevron icon on

1Starting with Microsoft Office 2016.
the right of their label. In addition, commands returned to users are disabled (greyed-out, not clickable) if the active system state is incompatible with their execution (e.g., the Group command is greyed-out when no items are selected for grouping). We investigate the role these command types serve in influencing search behavior in §5.5.

**Help Results.** Office search allows users to quickly retrieve help documentation using free-form text queries. Unlike commands, help results are not displayed directly beneath the search bar, but in a side panel where the user can refine their query and navigate returned documents. To activate the help search side panel, users need to click on the “Get Help” button shown below the command results list, as shown in Figure 1. Figure 2a shows an example of the help search side panel for the query “freeze” in Microsoft Excel.

**Web Results.** Users can also retrieve and display web documents directly in the application interface through Office search. The interaction pattern is similar to accessing help documents, in that web results are displayed in a side panel, where the user can explore returned documents. The web search side panel is activated by clicking on the “Smart Lookup” button below the command results list. Figure 2b shows an example of web documents returned for the query “slide master tutorial” in Microsoft PowerPoint.

### 3.2 Zero-query Interactions

After activating the search box, but before any query input, the search results area is populated with query suggestions (Figure 2c, bottom), and up to 5 of the most recent commands previously found and executed through search (Figure 2c, top).

### 3.3 Abandoned Search

Search sessions are regularly abandoned, with the user deactivating the search box (e.g., by clicking outside the search interface) without interacting with any of the results returned by the system. We explore abandonment in the context of in-application search in section 5.5 in more detail, but we highlight here the two types of search abandonment we observed in our analysis: (i) abandoned (query): search sessions in which the user explicitly types a query in the search box and decides not to interact with any of the results returned by the system; and (ii) abandoned (zero-query): search sessions in which the user activates the search area, but does not issue a query or interact with the search interface (e.g., recently used commands) before deactivating the search menu. We highlight these two types of abandonment (with and without query) because they are driven by different underlying user expectations with regard to system functionality.

### 4 DATA

To understand search in productivity software and address our specific research goals, we make use of Office instrumentation logs, which include: (i) information about the user: anonymized user identifier; (ii) information regarding application context: application version number, users’ system locale and language settings; (iii) information about search sessions: timestamps, user-typed queries, result rankings shown to user, and clicks.

From the instrumentation logs, we extracted English-language queries performed by Microsoft Office users who reside in the United States. Although search functionality is available in most Office products², we focus our analysis on three of the most popular applications in the suite: Word, Excel and PowerPoint. In addition, our data was obtained from a single Office build version, to ensure consistency of features available to users over incremental releases of Microsoft Office. Our data covers search events over a four week period, from the 29th of May to the 25th of June 2017. Finally, from these data, we randomly sampled one million users and their millions of queries.

### 5 RESULTS

We begin this section by describing the properties of query, command and user distributions in Office search (§5.1). We structure our results by providing a comparison between different types of in-application search (§5.2 – §5.4), followed by a closer look at characteristics of abandoned search (§5.5); lastly, we describe ways of using behavioral data to improve command ranking quality (§5.6).

#### 5.1 Query, Command and User Distributions

Similar to Web search [14], the distribution of queries observed in Office search is long-tailed, with a small number of query strings accounting for a large proportion of the search events, across applications. Figure 3a shows the cumulative distribution of distinct query strings observed in our sample, ordered by frequency. Roughly 10% of query strings account for 80% of searches in our sample. This

---

²Version 16 and higher.
trend is consistent across Office applications, suggesting that productivity search is similar across the three applications we studied. Figure 3b shows the cumulative distribution of commands executed via the search functionality. Similarly, the top 10% most popular commands executed via search account for roughly 70% of all search-issued commands — although the shapes of the two distributions suggest a heavier tailed query distribution. Finally, Figure 3c displays the cumulative distribution of searches across users, showing that the most active 10% of users account for roughly 50% of the searches in our sample.

5.2 Engagement Across Result Types

When users issue search queries in Office, the results may include commands, a link to Office help, or a link to Web search results. Among these three, command execution is the most likely outcome; command results are clicked 6.4 times as often as help documentation, and 32 times as often as web search. This indicates that most searches in Office are transactional, and are used to access — or re-access — commands.

Figure 4: Search session durations and query length.

Figure 4 shows the distribution of search session duration and query length across types of search outcomes. Search session duration is the duration in seconds between search bar activation (e.g., by a user click on the search bar) and search bar deactivation (e.g., by a user clicking on one of the results), and query length is the number of characters in the query at search bar deactivation.

The median duration of search sessions that terminate with a command execution is 7 seconds, whereas the median duration for search sessions that terminate with the activation of help or web search panels is 22 and 17 seconds respectively. Similarly, command search queries have a median length of 6 characters, whereas help or web search queries have median lengths of 12 and 11 characters respectively. These differences highlight that informational intent (i.e., the combination of help and web results) is typically expressed through longer queries and takes longer to formulate than command search. However, there is only a weak positive correlation between session duration and query length (Pearson’s $r = 0.39, p < 0.001$), suggesting that the typing effort is not the only factor behind the differences we observed. We now contrast command search to informational search in more detail.

5.3 Command (Transactional) Search

Given that command access is the most frequent search intent in Microsoft Office, we take a closer look at this activity. Table 1 shows the top 10 most frequent queries for each of the three applications in our dataset. Though each of these queries was explicitly typed by the searcher, user behavior is influenced by queries previously suggested via the no-query experience (as described in §3.2). These primed queries are over-represented in our dataset, and are denoted with †. We note that primed queries are abandoned at a much higher rate than other queries (%Abandoned), perhaps indicating users are exploring the system, rather than satisfying actual search needs. We discuss abandonment in detail in §5.5.

It is interesting to note that previous work on command execution in large-scale applications has shown a strong overlap among users in their access of the most popular commands, and low overlap for commands in the torso or tail of the corresponding command distribution [11, 17]. In contrast, our analysis of in-application search shows that there is low overlap in queries across users (%Users column; similarly reported with respect to command usage in [17]) — even with respect to the top searches — even though queries often directly match the names of commands. One possible interpretation is that users are leveraging in-application search to access idiosyncratic commands in their command vocabularies.

In addition to frequency, we report the proportion of users who issued each query (User Rank). This metric provides an indication of how popular a given query is across users. We highlight two examples: the query “undo” for Word, although it is the 5th most popular query by number of issues, it is the 15th most popular by the number of users who have issued the query at least once. Similarly, the query “group”, for PowerPoint, ranks 10th most popular by frequency of issues, but 17th most popular by number of users who have submitted it to the search system at least once. This shows that a smaller number of searchers make frequent use of these queries. We discuss these re-access and re-finding patterns in §5.4.

Finally, queries used to access commands vary widely. Table 2 presents a different perspective on command access, by showing the three most frequent commands executed through search, along with the five most popular queries used to retrieve these commands, and their relative frequency. There is less variation in the queries used to retrieve commands such as “Print” in Word or “Crop” in PowerPoint, where the command name also maps directly to the query used most frequently to retrieve it – 80% of retrievals of the “Crop” command being achieved via the query “crop”. On the other hand, Table 2 also shows examples of commands that are retrieved by a more varied set of queries, such as “Line Spacing" in Word or “Orientation” in PowerPoint. Because these commands require parameters (e.g., the amount of spacing between lines), users, in addition to using the command name as a query, may also specify parameters of the command as part of their queries. This behavior is informative for the development of search systems for productivity software, where command outcomes, rather than command descriptions are used for search (e.g., users issuing the query “blue text” rather than “change font color”).

5.4 Re-finding

Re-finding is an important aspect of search. As in Web search, where up to 39% of searches are driven by accessing previously retrieved resources [23], re-finding is frequent in Microsoft Office search. Not only do searchers frequently make use of the recently used commands list, they repeatedly issue the same queries. Table 1
### Table 1: Head queries per application. Priming queries are marked with the (†) symbol.

<table>
<thead>
<tr>
<th>Query</th>
<th>Most Popular Command</th>
<th>% Search Volume</th>
<th>% App Sessions</th>
<th>% Users</th>
<th>% Abandoned</th>
<th>% Requery</th>
<th># Query Rank</th>
<th># App Rank</th>
<th># Users Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Word</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>print†</td>
<td>PrintDefault</td>
<td>4.06%</td>
<td>5.95%</td>
<td>7.21%</td>
<td>71.99%</td>
<td>27.20%</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>write an essay†</td>
<td>Researcher</td>
<td>1.48%</td>
<td>2.36%</td>
<td>3.14%</td>
<td>99.48%</td>
<td>12.98%</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>word</td>
<td>WordCount</td>
<td>1.20%</td>
<td>1.12%</td>
<td>1.22%</td>
<td>2.86%</td>
<td>58.14%</td>
<td>3</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>spell</td>
<td>Proofing</td>
<td>1.00%</td>
<td>1.54%</td>
<td>1.77%</td>
<td>2.49%</td>
<td>27.78%</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>undo</td>
<td>Undo</td>
<td>0.88%</td>
<td>0.81%</td>
<td>0.85%</td>
<td>3.26%</td>
<td>60.22%</td>
<td>5</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>share my document†</td>
<td>Collaborate</td>
<td>0.84%</td>
<td>1.43%</td>
<td>1.96%</td>
<td>99.90%</td>
<td>4.28%</td>
<td>6</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>find</td>
<td>Find</td>
<td>0.80%</td>
<td>1.12%</td>
<td>1.17%</td>
<td>6.61%</td>
<td>40.65%</td>
<td>7</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>water</td>
<td>Watermark</td>
<td>0.80%</td>
<td>1.05%</td>
<td>1.13%</td>
<td>5.96%</td>
<td>41.69%</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>sp</td>
<td>Proofing</td>
<td>0.76%</td>
<td>1.10%</td>
<td>1.23%</td>
<td>2.52%</td>
<td>33.73%</td>
<td>9</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>page</td>
<td>PageNum</td>
<td>0.60%</td>
<td>0.81%</td>
<td>0.97%</td>
<td>12.11%</td>
<td>32.90%</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>Microsoft Excel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>header</td>
<td>HeaderAndFooter</td>
<td>2.09%</td>
<td>3.04%</td>
<td>3.32%</td>
<td>6.88%</td>
<td>39.17%</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>free</td>
<td>FreezePanes</td>
<td>1.70%</td>
<td>1.96%</td>
<td>2.11%</td>
<td>8.25%</td>
<td>52.57%</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>print</td>
<td>PrintDefault</td>
<td>1.44%</td>
<td>2.19%</td>
<td>2.53%</td>
<td>16.21%</td>
<td>32.90%</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>sort</td>
<td>Sort</td>
<td>1.18%</td>
<td>1.29%</td>
<td>1.44%</td>
<td>11.85%</td>
<td>53.38%</td>
<td>4</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>find</td>
<td>Find</td>
<td>1.07%</td>
<td>1.35%</td>
<td>1.27%</td>
<td>7.35%</td>
<td>54.82%</td>
<td>5</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>insert</td>
<td>InsertSheetRows</td>
<td>1.06%</td>
<td>1.14%</td>
<td>1.24%</td>
<td>23.88%</td>
<td>55.58%</td>
<td>6</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>freeze the top row†</td>
<td>FreezePanes</td>
<td>0.93%</td>
<td>1.43%</td>
<td>1.92%</td>
<td>99.78%</td>
<td>20.93%</td>
<td>7</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>insert row</td>
<td>InsertSheetRows</td>
<td>0.85%</td>
<td>0.93%</td>
<td>1.06%</td>
<td>8.62%</td>
<td>52.43%</td>
<td>8</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>freeze</td>
<td>FreezePanes</td>
<td>0.80%</td>
<td>1.16%</td>
<td>1.41%</td>
<td>13.41%</td>
<td>32.94%</td>
<td>9</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>insert a table†</td>
<td>InsertList</td>
<td>0.79%</td>
<td>1.34%</td>
<td>1.79%</td>
<td>99.77%</td>
<td>13.77%</td>
<td>10</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Microsoft PowerPoint</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>crop</td>
<td>Crop</td>
<td>2.76%</td>
<td>2.33%</td>
<td>2.47%</td>
<td>20.55%</td>
<td>64.18%</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>start presentation†</td>
<td>StartSlideshow</td>
<td>2.17%</td>
<td>3.76%</td>
<td>4.57%</td>
<td>99.55%</td>
<td>15.55%</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>change slide background†</td>
<td>FormatBackground</td>
<td>1.31%</td>
<td>2.19%</td>
<td>2.72%</td>
<td>99.42%</td>
<td>16.98%</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>portrait</td>
<td>Orientation</td>
<td>1.19%</td>
<td>1.94%</td>
<td>2.41%</td>
<td>7.27%</td>
<td>18.72%</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>de</td>
<td>DesignerPane</td>
<td>0.94%</td>
<td>1.04%</td>
<td>1.12%</td>
<td>4.15%</td>
<td>52.11%</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>master</td>
<td>SlideMaster</td>
<td>0.84%</td>
<td>1.02%</td>
<td>0.97%</td>
<td>1.61%</td>
<td>53.58%</td>
<td>6</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>design</td>
<td>DesignerPane</td>
<td>0.71%</td>
<td>0.82%</td>
<td>0.88%</td>
<td>2.59%</td>
<td>50.38%</td>
<td>7</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>change layout of slide†</td>
<td>MasterStyle</td>
<td>0.64%</td>
<td>1.09%</td>
<td>1.37%</td>
<td>99.66%</td>
<td>14.24%</td>
<td>9</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>group</td>
<td>Group</td>
<td>0.64%</td>
<td>0.69%</td>
<td>0.77%</td>
<td>35.03%</td>
<td>51.79%</td>
<td>10</td>
<td>15</td>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command Name</th>
<th>Query</th>
<th>Microsoft Word</th>
<th>Microsoft Excel</th>
<th>Microsoft PowerPoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proofing</td>
<td>spell</td>
<td>28.07%</td>
<td>FreezePanes</td>
<td>Orientation</td>
</tr>
<tr>
<td></td>
<td>sp</td>
<td>18.51%</td>
<td>free</td>
<td>portrait</td>
</tr>
<tr>
<td></td>
<td>spe</td>
<td>15.27%</td>
<td>freeze</td>
<td>orientation</td>
</tr>
<tr>
<td></td>
<td>spelling</td>
<td>6.35%</td>
<td>fr</td>
<td>landscape</td>
</tr>
<tr>
<td></td>
<td>spell check</td>
<td>6.25%</td>
<td>freeze</td>
<td>change to portrait</td>
</tr>
<tr>
<td>WordCount</td>
<td>word</td>
<td>42.96%</td>
<td>HeaderAndFooter</td>
<td>change orientation</td>
</tr>
<tr>
<td></td>
<td>wor</td>
<td>20.13%</td>
<td>header</td>
<td>design</td>
</tr>
<tr>
<td></td>
<td>wo</td>
<td>12.15%</td>
<td>footer</td>
<td>de</td>
</tr>
<tr>
<td></td>
<td>character count</td>
<td>4.86%</td>
<td>head</td>
<td>des</td>
</tr>
<tr>
<td></td>
<td>line</td>
<td>14.94%</td>
<td>foo</td>
<td>d</td>
</tr>
<tr>
<td></td>
<td>spacing</td>
<td>6.80%</td>
<td>3.71%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>single space</td>
<td>6.61%</td>
<td>3.23%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>double space</td>
<td>6.19%</td>
<td>5.82%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>double</td>
<td>4.78%</td>
<td>inser</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>inser a row</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>inser</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>insert rows</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command Name</th>
<th>Query</th>
<th>Microsoft Word</th>
<th>Microsoft Excel</th>
<th>Microsoft PowerPoint</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Most frequently used queries for executing top three most frequent commands executed through search.
shows, for a given query, the proportion of searches for which users had previously issued the same query within our sampling period (% Requery). There is variation among users and queries with respect to their re-querying rate, but on average, the priming queries are the least re-issued by users (and most likely to be abandoned).

The aforementioned view of re-querying is muddied by the fact that people varied in their use of the Office applications in general, and of the in-application search experience in particular (i.e. people with more sessions or queries had more opportunities for re-accesses). Figure 5(a) shows the proportion of re-accesses, but partitions users into cohorts based on the total number of queries they issued during the month we studied. We distinguish three separate re-access types: re-executing a command found through search ("Command"), re-issuing a query to the search system ("Query") or re-issuing a prefix or an extension of a previously issued query – for instance, the queries "pr" and "prin" – ("ApproxQuery"). For each of these types of search engagement, Figure 5(a) shows the distribution of re-access over repeated use of the search system. It is interesting to note that, for users with three distinct searches in our sample, the proportion of repeated queries is 9%, but the proportion of repeated commands is 17%, and that of approximately repeated queries is 26%. This suggests that users learn approximately equivalent queries (e.g., a set of prefixes) for accessing the same commands.

We also examine the distance (i.e. the number of search activations) between re-access instances. In this view, we include only search sessions that are indeed instances of re-access for a given user. As seen in Figure 5(b), 30% to 35% of repeated queries are identical to, or are approximations of, the previous query issued by the user. Moreover, a history window of 5 queries is sufficient to explain 70% of approximate query re-issues. Conversely, command re-access is distributed more evenly across the five most recently executed commands, primarily because the recently used command list provides easy access to these commands. Even so, almost 20% of the commands that are re-executed through search are repeats of the most recently executed command, and the top five most recent commands account for roughly 45% of command repeats. We now take a closer look at factors influencing search abandonment.

5.5 Search Abandonment

Like in web search [22], searches in Office are frequently abandoned. Abandonment can occur when a user inputs a query and elects not to click a result, or when a user activates the search bar without issuing a query and elects not to click on a recently used command or suggestion. In either case, a user may fail to click on a result either because the intended command is not listed, or, importantly, because it is present but the active system state is incompatible with its execution (i.e. the command is greyed-out). For example, in PowerPoint an object must be selected for the Crop and Group commands to be enabled, and thus the corresponding queries "crop" and "group" have high abandonment rates compared to similar queries (20.55% and 35.03% respectively, as seen in Table 1). We return to these system state errors, and their role in search abandonment, later.

5.5.1 Differences between command and abandoned search. Understanding the characteristics of abandoned search in contrast to successful command search is critical to developing rich search functionality that meet users’ expectations and needs. Figure 7 shows differences between command and abandoned search with respect to query length (7a), search session duration (7b), number of menu results shown in the search ranking, (7c) and number of disabled results shown in the search ranking (7d). The first row in Figure 7 shows differences in search session properties, such as query length and duration. Overall, for abandoned search sessions we observe longer queries (median query length 10 characters) and
worth noting that showing a disabled command is arguably better
don their searches or seek additional support. One hypothesis that
reflect more complex or obscure tasks, which may pose challenges
in both query formulation and task execution. As such, it is not
the menu result that leads to abandonment, but rather the more
complex task the user is engaged in. Even so, our findings suggests
that the type of commands returned to the user have an effect on search
abandonment.

5.5.2 Command types and search abandonment. Command re-
results returned by the search system in Microsoft Office are not
uniform with respect to their user interaction patterns. Section 3.1
reviews the different types of command results returned to users: ac-
ton, flag or menu commands. Given that, overall, a majority of
user clicks on command results are issued on the top-ranked result,
in this section we inspect the effects of top-ranked command type
on user search behavior.

The type of command displayed at top rank in the search results
returned varies with both user intent and with the distribution of
command types available in each of the applications we explored.
Table 3 shows the distribution of command types shown at top
rank. Although there is some variation across applications, the most
common types of results returned at top rank are action commands
and menu commands.

<table>
<thead>
<tr>
<th>Command</th>
<th>Informational</th>
<th>Abandoned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Flag</td>
<td>Menu</td>
</tr>
<tr>
<td>Word</td>
<td>49.88%</td>
<td>10.88%</td>
</tr>
<tr>
<td>Excel</td>
<td>44.45%</td>
<td>13.16%</td>
</tr>
<tr>
<td>PowerPoint</td>
<td>34.89%</td>
<td>21.29%</td>
</tr>
</tbody>
</table>

Table 3: Distribution of command types displayed at top-rank.

Figure 8 shows the distribution of search outcomes over command
types, controlled for how often each command type is displayed at top
rank.
which result lists to re-rank is not trivial, given that more than 70% of searches retrieved optimal rankings (i.e. clicked command 6%, in historical and oracle re-ranking query selection, respectively. So, all re-ranking strategies we explored increase the proportion of clicks at top-rank. Simply placing menu items in the ranking.

Table 4: Distribution of clicks over ranking positions.

<table>
<thead>
<tr>
<th>Selection</th>
<th>Strategy</th>
<th>Clicked result rank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>No re-ranking</td>
<td></td>
<td>70.54%</td>
</tr>
<tr>
<td>Historical</td>
<td>Result type</td>
<td>71.17%</td>
</tr>
<tr>
<td></td>
<td>Overall click-through</td>
<td>74.66%</td>
</tr>
<tr>
<td></td>
<td>Per-query click-through</td>
<td>77.93%</td>
</tr>
<tr>
<td>Oracle</td>
<td>Result type</td>
<td>76.77%</td>
</tr>
<tr>
<td></td>
<td>Overall click-through</td>
<td>82.65%</td>
</tr>
<tr>
<td></td>
<td>Per-query click-through</td>
<td>83.18%</td>
</tr>
</tbody>
</table>

Result type re-ranking. Based on our observation that searches in which menu commands are top-ranked seem to be abandoned at a higher rate, we re-rank results by explicitly placing menu commands below action or flag items in the ranking.

Per-query command click-through rate. User engagement with the search system allows us to observe command execution frequency and click-through rate for a given query. For a query and a list of results relevant to the query, we re-rank commands in descending order of their per-query click-through rate.

Overall command click-through rate. Similar to the previous strategy, we can compute global command click-through rates for individual commands, as a measure of how useful a command is overall. Given a list of results, we re-rank commands in descending order of their global click-through rate.

To evaluate the strategies outlined above, we split our data into training and test folds (each fold covering a non-overlapping period of two weeks). We use training data for two purposes: firstly, to compute overall and per-query command click-through rates; secondly, we use the training fold to select which queries (and corresponding rankings) potentially benefit from re-ranking. Selecting which result lists to re-rank is not trivial, given that more than 70% of searches retrieved optimal rankings (i.e. clicked command returned at top rank), and as such, re-ranking these searches would likely deteriorate the quality of their results. Thus, we compare two different methods to select instances of search in which to deploy our re-ranking strategies.

Historical query selection. We use training data to select queries that generated clicks on lower ranked results, on average, and we apply our re-ranking strategies only to those queries, as observed in our test data. Using this filtering method, we select 16.34% of the searches in our test fold for re-ranking.

Oracle query selection. Given that our logs contain the clicked result rank, we can identify all searches in which clicks were not issued on the top-ranked result, and apply re-ranking strategies to these searches. This selection method is not feasible in a real-world setting, where click rank is not known beforehand, but is an informative baseline with respect to the effectiveness of our re-ranking strategies. Using this filtering method, we select 29.46% of the searches in our test fold for re-ranking.

Table 4 shows the distribution of clicks over ranks for our re-ranking strategies. The existing system ranker returns the clicked result at top rank in more than 70% of searches in our test data. Even so, all re-ranking strategies we explored increase the proportion of clicks at top-rank. Simply placing menu type results lower in the ranking increases the proportion of top-rank clicks by 0.6% to 6%, in historical and oracle re-ranking query selection, respectively.

Together with our analysis of command types and their role in abandoned search, our findings suggest that placing menu items lower in the ranking does not deteriorate the quality of results returned to the user, and might even prevent abandonment. We hypothesize that users willing to navigate menu hierarchies to locate their intended command will do so even when the menu result – which collapses the relevant menu hierarchy – is at a lower rank; the converse might not be true, with users looking for quick access to an action (or flag) command perhaps being less inclined to review the sub-menu contents of top-ranked menu results in order to locate their intended command, and maybe more likely to abandon their search. We leave testing this hypothesis for future work.

Overall, the most effective re-ranking metric we explored is per-query command click-through rate. Re-ranking based on this metric increased top-rank clicks by 7% to 13% over historical and oracle query selection methods, respectively. Even though integrating behavioral signals into search results ranking is widespread in search algorithms for the web [2, 25], their application to in-application command search is under-explored and our work provides an overview of simple re-ranking strategies based on user interaction data as a first step towards integrating behavioral signals into command and control search.

6 CONCLUSIONS

Rich search experiences have become available across a variety of surfaces: from the web, to personal information repositories, feature-rich applications and operating systems. These novel search experiences transform both the way users engage with complex applications, and their expectations of search systems in general.

Our study provides a characterization of user behavior in an under-studied area of information access: search in productivity software. Millions of users actively engage with productivity software each month. As search interactions become integrated into their workflows, understanding search behavior is necessary for developing systems that can effectively respond to users’ queries.

Users primarily engage in finding commands through the search interface available in Microsoft Office (RG1.2). The distributional properties of observed queries are similar to other search domains, in that a large proportion of search volume is generated by a small proportion of queries (RG1.1). However, our results show that, unlike command access, there is low overlap in queries across users – even for frequently observed query strings. One possible interpretation is that users are leveraging in-application search to access idiosyncratic commands in their command vocabularies. We also show that queries used to access commands vary widely, and that action outcomes (e.g., “portrait”) rather than command names (e.g., “change orientation”) are commonly used to retrieve parameterized commands (RG1.3). Users frequently engage in command re-access through search, not only by using the zero-query interface available in Microsoft Office, but by explicitly reissuing queries to the system. For frequent users of search, on average, more than 26% of queries are approximate repetitions of a previously issued query. Moreover, up to 70% of repeated queries occur within a window of five most recent user queries. Together with our characterization of
observed queries, our findings regarding re-access are informative for the development of command retrieval systems (RG1).

Identifying instances of search in which users are unable to access intended commands is a crucial element of improving search quality. Our work on abandonment shows that successful command search and abandoned searches vary with respect to their behavioral properties, such as query length and search duration, and with respect to result ranking characteristics, such as the number of disabled or menu commands shown in the results list (RG2). Furthermore, we find that different command types influence search outcomes, and show that menu commands are prevalent at top rank in abandoned searches. Our findings suggest that ranking sub-commands, as opposed to collapsible menus, directly in the results list might benefit users trying to execute multi-step actions.

Finally, our work on re-ranking commands using metrics derived from historical interaction data shows that behavioral signals can be used to improve command rankings (RG3). More work is required to merge previous efforts on command recommendation and the use of behavioral data in command retrieval, and through our study we provide a direction for future endeavors in this space.

7 ACKNOWLEDGMENTS

We would like to thank Madeline Kleiner for insightful discussions which helped to shape this research in its early stages, and motivate many of the research questions that were pursued in this paper.

REFERENCES


