Situating Web Searching in Data Engineering: Admissions, Extensions, Repairs, and Ownership

By

Daniel Steven Griffin

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Information Science

in the

Graduate Division

University of California, Berkeley

Committee in charge:

Professor Deirdre K. Mulligan, Co-Chair

Professor Steven Weber, Co-Chair

Professor Marion Fourcade

Professor Jenna Burrell

Fall 2022

Abstract

Situating Web Searching in Data Engineering: Admissions, Extensions, Repairs, and Ownership

by

Daniel Steven Griffin

Doctor of Philosophy in Information Science

University of California, Berkeley

Professor Deirdre K. Mulligan, Co-Chair

Professor Steven Weber, Co-Chair

When does web search work? There is a significant amount of research showing where and how web search seems to fail. Researchers identify various contributing causes of web search breakdowns: the for-profit orientation of advertising driven companies, racial capitalism, the agonistic playing field with search engine optimizers and others trying to game the algorithm, or perhaps 'user error'. Suggestions for making web search work for more people more of the time include: regulations aimed at competition or the design of the search interface; changing the conception of, metrics for, and evaluation of relevance; allowing subjects of search queries some space of their own on the results pages to speak back; proposals for public search engines; and better-informed users of search.

I take a different tack. Rather than focusing on identifying and remediating points of failure, I seek to learn from successful searchers how they make search work. So, I look to data engineers. I closely examine the use of web search in the work practices of data engineering, a highly technical, competitive, and fast changing area. Data engineers are heavily reliant on general-purpose web search. They use it all the time and it seems to work for them. The practical success I report is not determined by some solid 'gold standard' metrics or objective standpoint, but by how they have embraced web search and present it as useful and more importantly essential to their work. It is success for their purposes: in gradations, located in practice, and relative to alternatives.

Through interviews and document analysis informed by digital ethnography, I use theories from situated learning and sociotechnical systems to explore how and why search works for data engineers. I draw from feminist science and technology studies, the sociology of expertise, situated learning theory, and organizational sociology to explore and position my four core findings.

First, I find that personal knowledge of the technical mechanisms of search plays a limited role in data engineers successful searching. Exploring why and how web search works for data engineers allowed me to probe the role of knowledge about the mechanisms of search. Contrary to dominant literature that views individual ignorance of search mechanisms as contributing to failed searches and search literacy as a necessary, if independently insufficient, path towards mitigating search failures and the harms to

which they contribute, I find little evidence that data engineers' personal knowledge of the mechanisms of search contributes to their successful use of it.

Data engineers receive little formal on-the-job training or mentoring on how to use web search successfully. Data engineers describe web search as a solitary exercise in which they receive little formal guidance. Moreover, data engineers describe web search as a solitary practice. The absence of formal training is surprising given the professions' admitted heavy reliance on web search. In addition to the absence of formal training, data engineers report little discussion about search practices or collaboration in searching and some discomfort with their heavy reliance. However, I find one form of talk about search, what I call "search confessions"—statements, often hyperbolic, about one's reliance on web search—to be pervasive and a key way in which the community of data engineers legitimate their heavy reliance on web search and develop and express shared norms about how to use search well.

Second, rather than personal knowledge, I find that occupational, professional, and technical components of their work practices contribute to their successful use of search. Expertise embedded in these components of data engineers' web search practices improve two key search processes: query generation and results evaluation. The work practices of data engineers also decouples the immediate effects of searching from organizational action.

To extend the description of successful web search practices, I address how data engineers confront search failure. I look at how they turn to ask colleagues questions when web searching fails and find them performing repair. These sites of coordination and collaboration post-search failure also provide opportunities for broader knowledge sharing and a space to legitimate their work and expertise, both individually and as a profession.

If it is normal and acceptable to rely so heavily on search, it may be a surprise that there is so little talk about searching. Data engineers regularly present search as an individual responsibility—they search by themselves or on their own and desire to keep their searching private. This individual responsibility exemplifies the extent to which the firms employing data engineers do not use data from web search activity to better know and control the search practices. My findings did not reveal technology-enabled management of web search practices. I analyze the absence of firm management of search and the solitary and secretive search practices as a product of organizational reliance on data engineers to flexibly learn on the fly. The privacy of search generally protects the resources (time, attention, and reputation) of individual data engineers to pursue the distributed searching and learning on the fly they are tasked with.

In the conclusion, I advance two further arguments before developing provocations grounded in the key takeaways. While web searching for data engineering is generally put to successful use, I show how the effective use of web search is supported by and limited to a dependence on the knowledge of others and how uneven access to community norms and knowledge limit who is effective. They key takeaways center on how web search in data engineering is continually re-legitimated; extended beyond the search

box and the search results page; did not hinge on personal knowledge of the technical mechanisms of web search; is entangled with notions of responsibility, credit and blame, for knowledge; and the intentional application of technique to influence search activity, did not make an appearance.

Being 'better-informed search users' for data engineers means being situated in practices around search with embedded expertise and reinforced values that support their uses of web search. For the data engineers I talked with, organizational and occupational factors including the structures of the technology, workplace interactions, and norms—all well outside of and stretching well before and after the moments of typing a query into a search box or reviewing a search results page—make search work.

To dad. I'm having fun and learning lots.

Contents

Acknowledgements

I would like to thank my committee for their support throughout this process and for seeing me through to the finish. Thank you to Deirdre Mulligan for inviting me to look at web search back in 2016. Now I cannot stop looking. I am extremely grateful for your support for my curiosity and for your example of how to do research that impacts society and mentor new scholars to do the same. I deeply appreciate you reading through so many rough drafts, engaging with my ideas in their formative stages, and for showing me I could do better. Thank you, Deirdre, for shepherding me over the finish line. Thank you to Steve Weber for so many rich conversations as I started my PhD and for helping me formulate my questions and consider why they matter. I still regularly go back to meeting notes from years ago. Thank you Jenna Burrell for your guidance in methods, especially for letting me participate in the Twitter paper research, and for your constant enthusiastic encouragement. Thank you Marion Fourcade for joining me on this journey. Thank you for your close attention and for the many suggestions to connect with the literature, both the old and brand new.

Thank you to my research participants who willingly talked so much about this thing that people are hesitant to talk about. Please consider this dissertation an offer to talk again. Thank you also to the many friends and classmates who helped connect me with people to talk to.

I'm grateful for the feedback and support over the years in the Doctoral Research and Theory Workshop. I received feedback in 2019 on an initial sketch of this research in a prototype idea session and then later a draft proposal at (which at the time I called "Shaping the shop floor: characterizing coder (re)search labor within cognitive assemblages")[1]. Thanks to Coye Cheshire for showing us how to practice with care to our work and care for each other.

Thank you to Jake Goldenfein for keeping our paper going across the years and around the world, and thank you Sebastian Benthall for connecting us (and for our conversations).

Special credit for so much of this research and my development as a researcher goes to Anne Jonas, Richmond Wong, Elizabeth Resor, and Emma Lurie. You've helped me think through and write through so many questions and dilemmas, across Slack, Zoom, and Twitter direct messages. The requests for help with a search or a citation were always met with more than just an answer, but connection and support. Thank you Anne and Elizabeth for the weekly calls. I'm glad they will continue. Thank you Richmond for always appreciating my speculations and for helping me explore new literatures. Thank you Emma for taking a chance on the side project, it was incredibly enriching to work with you and generative for

[1] I had directed the proposal towards a call for papers for Data & Society's Algorithms on the Shop Floor Workshop (Elish & Watkins, 2019). I was not selected as a research participant—sharing this as praxis—but am grateful for how the call shaped my research and for the papers workshopped that I have cited below.

this dissertation.

Thank you to the rest of the folks at the School of Information who have made this adventure so rewarding. Thank you to the other faculty members. Thank you to Anno Saxenian for your example of scholarship and service. I'm grateful for Marti Hearst's foundational work on web search. Particularly grateful to Paul Duguid for his deep engagement with scholarship and all of us students, particularly his mentorship in Classics and such supportive feedback on papers and chapters. His published work was also always a constant guide and inspiration.[2] Thank you Bob Glushko for all your personal encouragement. A big thank you to the staff at the School of Information, especially Catherine Cronquist Browning, Inessa Gelfenboym Lee, Kevin Heard, Gary Lum, and Jonathan Henke. Thank you Caitlin Appert for the retweets. I'm grateful for the chance to participate in the Center for Long-Term Cybersecurity, the Algorithmic Fairness and Opacity Group, and the Center for Technology, Society & Policy.

I've been at the school too long to name everyone who has played a role in shaping my journey and this research but I'm so grateful for the community, and particularly the reflexive and intentional approach of the PhD students who came before and have come after. Thank you to Nick Doty for his constant example. Thank you also to Galen Panger, Noura Howell, Elaine Sedenberg, Jen King, Nick Merrill, Sarah Van Wart, and Andy Brooks for showing the way. Thank you to the cohort ahead of me for your friendship and your example of mutual support—Anne Jonas, Max Curran, and Shazeda Ahmed. Thank you Andrew Chong, Jon Gillick, Nitin Kohli, Guanghua Chi, Jeremy Gordon, Zoe Kahn, Doris Lee, Ji Su Yoo, and Sijia Xiao for the conversations and companionship. (Thank you to Mike Berger for conversations over runs that encouraged me to see myself applying to the PhD program.) Thank you also to the many other students and classmates over the last several years who I cannot wait to thank in person, especially to Vijay, Andy, Nikhil, and Anand.

I must also thank the National Science Foundation for supporting Grant # 1650589 which funded some of my studies, along with the GI Bill (which I will take this opportunity to note was racist in design and effects (Katznelson, 2005)).

As you will see in the dissertation itself, searching and asking questions is a massively collaborative project, with much collaboration of the sort that sadly escapes notice. I cannot possibly name all those scholars I've interacted with on social media and who have been so welcoming. I'm so thankful for the interactions with and examples set by Safiya Noble, Francesca Tripodi, and Jutta Haider. Online and off, I'm grateful to the many people who have provided search terms, provided the inspiration and impetus to search, asked why these questions mattered, and helped evaluate what I was finding from my own searches. I'm sure

[2] I cannot help but note Paul's name in the Acknowledgements of Lave & Wenger (1991)—"that rare colleague whose editorial involvement became akin to coauthorship" (p. 26)—, a central building block for this dissertation.

I am forgetting others who deserve credit. I must take the blame, though I hope you'll take away something about blame from this dissertation.

Thank you to my friends and family for your constant encouragement and for entertaining my constant questions about how you searched the web. Thank you Andrew Reifers for being there and for talking with me about my fascination with search. Thank you Rhonda Griffin for showing a willingness to ask questions and always supporting my curiosity.

Most of all, I'd like to thank Emily Witt for such tremendous support. This would not have been possible without your help. Encouraging me and talking me through every stage of this research, teaching me so much about how to ask questions and listen, and always a model of a thoughtful champion of careful purposeful research and our shared responsibilities. You shaped my questions and conclusions. Thank you for all the proofreading and brainstorming these last several weeks. And thank you for taking over more than your share of bath times, bed times, diaper times, and dishes these last few months. You inspire me and make me better. Thank you Wilder for your curiosity and Otis for joining me in the baby carrier for late night writing sessions.

1 Introduction

"Probably 90% of my job is Googling things," Christina said. Then she chuckled.

Amar, seemingly also amused, laughed as he said, "I've really never thought about it myself, even though it's kind of like 90% of my job to just look things up."

And Noah, too, described searching the web as a central aspect of doing the work of a data engineer:

> If I found out that some coworkers were doing it a lot less than me, I would actually wonder if they weren't doing their job as effectively as they could. Which is not to suggest that I'm the greatest googler[3] of all time. But I consider it a core of doing my job. You have to be able to search.

The quotes above are from interviews conducted for this research. Christina, Amar, and Noah are data engineers.[4] They work in enterprise software, social media, and media streaming. Like many who write code, and all the data engineers that I talked with across a range of industries, they heavily rely on general-purpose web search engines to do their job.

I take a close look—through interviews and digital ethnography—at this heavy reliance. How are they able to search? Does this work for them? How does it work for them? How do they learn to search the web as data engineers? Do they really "just google it"? Are their employers OK with this? What can we learn from their success?

I do find that data engineers have generally been successful in making use of web search at work. I present this case not only of data engineers' heavy reliance, but a successful heavy reliance. Though I do not suggest they are successful in every search or that every data engineer finds success in their ways of searching. Rather, I find data engineers' work practices incorporate and facilitate searching the web and this contributes to successful work performance. So the practical success I describe, and seek to understand, is not determined by some solid 'gold standard' metrics or objective standpoint, but by how they have embraced web search and present it as useful (or at least a "satisfactory accomplishment" (Thompson, 1967)) and essential, with little complaint. It is success for their purposes: it works in gradations, located in practice, and relative to alternatives (see de Laet & Mol (2000)). My focus has not been on the boundary line of success or failure. My focus is on what data engineers do to make their use of search successful, success in their eyes.

Why look at the use of general-purpose web search engines? People go to search engines to

[3] While "Googler" is sometimes used to refer to Google employees, it is also commonly used, rendered here in lowercase, to refer to someone who uses Google or other search engines.

[4] All names of my research participants are pseudonyms.

find out all sorta of information from the most mundane or trivial to the deeply significant. People use search engines as they seek to determine if they are pregnant (Kraschnewski et al., 2014), or to navigate unwanted pregnancy (Guendelman et al., 2022; Mejova et al., 2022), or pregnancy loss (Andalibi & Bowen, 2022; Andalibi & Garcia, 2021). People use search to find and make sense of health information (Mager, 2009), including whether to have their newborns receive the lifesaving Vitamin K shot (DiResta, 2018). People use search engines to learn about recycling programs (Haider, 2016) and sustainability (Haider et al., 2022). People use search engines to "fact check" the news (Tripodi, 2018) or to "just" google it (Toff & Nielsen, 2018). People use search engines to critique politicians (Gillespie, 2017) and to find which one to vote for (Mustafaraj et al., 2020). People use search engines for everyday life (Haider & Sundin, 2019) and to learn about things of the most pressing societal relevance (Sundin et al., 2021).

Search engines shape the web—what we find and what people will write (Introna & Nissenbaum, 2000). But, web search is a relatively new technology. People continue to negotiate its role, map out its limitations, and imagine alternative designs and practices. I took up this research in the context of high profile failures of search engines and of those searching and discussions about how web search, or the people using it, might do better. Search engines promote racist and sexist representations of people (Dave, 2022; Noble, 2018; Urman & Makhortykh, 2022). Search engines sometimes lend credibility to false and hateful beliefs (Mulligan & Griffin, 2018). Search engines are manipulated to promote propaganda (Tripodi, 2022b).

I should qualify some of my comments above. I use a sociotechnical lens, drawing on many approaches, to see searching as a product of the interplay between the technology and the people. Search engines, including the technology and the people managing and making it, *and the people that use them* do those things. Search engines and their uses are shared creations. People, at the search engines, the websites, the regulatory agencies and politicians offices, the newspapers and school house, and the searchers, together, make search engines what they are. People make or allow racist and sexist representations of others. People say or let others say, "Google told me so!" (Narayanan & De Cremer, 2022). People manipulate others or permit such manipulation. I follow others that point to how "search engines, and Google's powerful position in particular, are negotiated and stabilized in social practices." (Mager, 2012)

This is a case study showing and analyzing successful web search practices. Rather than join the extensive literature documenting ways that web search engines are harmful, fail, are mistaken and misused, or abused, I describe where web search is made to work. I do not discuss how people should search the web in the abstract. I discuss how these people situated web search to be useful in their work.

First, some complicating context. On the face of it, web search for data engineers is a private

and solitary activity. Data engineers' direct web search activity is generally not shared with peers, monitored or managed by their supervisors in performance evaluations, nor facilitated by special tools that might guide or correct them. The direct web search activity, which I will refer to as the ostensible web search, is not publicized or socialized. They are not applying their coding skills to their searching tasks. They are not sharing the data about their searching activity. Nor, it seems, are their managers surveilling their searching performance.

For data engineers, the key ingredients in situating web search practices for success are:

- admitting searching as a tool appropriate for the work,
- practices for repairing fruitless searches,
- supporting query generation and results evaluation, and
- providing privacy for searching.

These ingredients promote the situated learning of search and promote, rely on, and align with search as extended.

What does it mean to say search is extended? Earlier work distinguished the search results and the results-of-search (Mulligan & Griffin, 2018). The search results are the ranked webpages, the snippets describing them, the advertisements, and the other content returned by the search engine for a query. The results-of-search are not the set of pages, but the results of the search itself. In the search breakdown described in (Mulligan & Griffin, 2018), where Holocaust-denier search pages ranked at the top for the query [did the holocaust happen], the results-of-search included "what searchers experience Google as communicating about those sites" (p. 571). This distinction was developed with reference to Bucher (2017)'s "algorithmic imaginary"—"ways of thinking about what algorithms are, what they should be, how they function and what these imaginations in turn make possible" (p. 39-40) and her use of Introna (2016)'s argument that "[t]he doing of the algorithm is constituted by the temporal flow of action and is not "in" the particular line of code, as such." [pp. 21-22]. We can identify the immediate output of the search algorithms and the design of a search engine results page (SERP). But that is not, nor does it constitute, the result of the search.

The observation that search is extended follows from the above, a straightforward consequence of using a sociotechnical lens (in my case, using the Mulligan & Nissenbaum (2020) Handoff analytic). To say that searching is *extended* (and extendable) refers to searching not being a singular or separable moment. The doing, or performance, of a web search includes the impetus to search, the generation of a query, the time and place to type, paste or speak the query into a search box and look at the search results, and the evaluation of results that continues long after the clicking, scrolling, and reading is finished. Looking at search as extended provides exploratory and explanatory advantages, as I will show.

But there are some problems. While data engineers use of web search is central to the

work and is generally successful, that does not mean that such searching has been solved[5]. The key problem is that some of the same factors that drive the successful use of search in data engineering work (namely, how search is admitted, how search failures are repaired, and privacy for searching) are perverted in some environments. In some companies, or pockets within, the acceptance of search is contorted to produce environments where data engineers are hesitant to openly ask questions and find themselves struggling and flailing about searching again and again in fear of being misjudged or mistreated for asking a question. The privacy that protects space to learn and stretch one's knowledge is expanded to block effective collaboration within the company. These negative effects particularly shape the experience of those people already marginalized within technology work. The penchant for searching and privacy can become excessive, sometimes putting newcomers and women under suspicion for both not searching enough and for having to search all the time. This is an important part of the story, and the failure here is seen more keenly when the success is clearly explained.

1.1 Background

Here I will share the origins of this research while also introducing some of the research shaping my questions and my understanding of the importance of this topic.

In a class in the fall of 2018, Professor Jenna Burrell talked about a moment of surprise that led her to consider and look for all the ways you could share a phone[6]. This led me to consider exploring how people share or share about search. I could look for the competing articulations of search, the claims of definition and legitimacy, in the "public dialogue" (Gillespie, 2014), in how they are "articulated, experienced and contested in the public domain" (Bucher, 2017, p. 40).

Consequently, for a project in that class I looked, using Twitter search, for people talking about web search on Twitter. I stumbled on numerous accounts of people in coding roles discussing their heavy reliance on web search in their work. Many people made acclamations (though sometimes nervously) of the central role that general-purpose web searching played in their work. I had been looking for examples of people talking about and sharing searching and was struck. I wondered, if they seem to so successfully incorporate general-purpose web search into their work maybe I might look closer?

Could I look at how these people in coding roles talked about and shared search to better understand what searching is and could be? Do they "just google it"? How does knowledge

[5]Kruschwitz et al. (2017) use the "solved" language in the opening line of the abstract to their book on enterprise search: "Search has become ubiquitous but that does not mean that search has been solved."

[6]This surprise led to Burrell (2010).

of the mechanisms of search inform their searching practices? How is responsibility assigned? I was initially interested in how individuals and organizations addressed the epistemic risks involved in such a reliance on web search[7]. Here was a group of people seemingly heavily reliant on web search. This same group might also have some better awareness of some of the difficulties in web search, and some of them might be able to marshal a range of responses.

By marshaling of responses I imagined that people who wrote code for their work tasks might also write code to facilitate their searching-for-work tasks. Would I find examples of innovation from "lead users" grafted to web search engines to improve some aspect, like that discussed by Hippel (1988)? Was the technology flexible enough to permit such modifications, like that discussed in Leonardi (2011)?

Starting in that final project in the winter of 2018, four years ago, I have explored the broad contours of of these questions about people who write code searching the web. All the while I was also coding myself. I have written Python code for personal or school projects since applying to the I School in 2014. I also taught the I School's Summer Python Boot Camp for incoming graduate students for three years (including to seasoned programmers learning a new language). This document itself was produced with the aid of several scripting tools I've written and many code-related web searches.

To gain tractability, I narrowed my focus to a subset of those who write code for work: data engineers.[8] I selected this site because it seemed likely to include people who were relatively technically sophisticated and it appeared to be a particularly dynamic field that would require a significant amount of learning on the fly, and so heavily reliant on search. Data engineers are also involved in constructing tools to control, replace, or surveil other people, practices, and tools. So they would likely be able to both refashion their tools and practices around search—if they saw that as beneficial—and be attentive if tools to control, replace, or surveil were directed at them.

I saw them as perhaps more likely to have sophisticated understandings of the underlying technologies and and appreciation, perhaps, for the uses and misuses of data and automation built on or around it. This lead me to think it may be a particular valuable site to consider the role of technical literacy.

The COVID-19 pandemic constrained my research approaches. I developed a plan to build

[7]The epistemic risk is meant to convey risk in relation to beliefs or ways of believing not to prescriptivist epistemology (see Dear (2001) on epistemography). The phrasing above might be better termed the risks in the ways knowing; the costs and consequences in different cognitive practices entangled in search. My initial interest was not on finding, or determining whether one found, truth. I was focused on the larger effects from the various ways in which search is used by people to come to know or to believe or to perform a knowledge or skill.

[8]This choice is discussed further in [Methods and Methodology].

on my prior work (informed by digital ethnography) with a study focused on in-depth interviewing of data engineers. I submitted my prospectus and passed my qualifying exam soon after my first son was born and took the next semester off. Then, in the summer of 2021, I started interviewing data engineers.

1.2 Overview

In **Methods and Methodologies**, I first introduce my two methodological frames for better understanding data engineer use of web search. I describe my use of the Handoff analytic (Mulligan & Nissenbaum (2020); Goldenfein et al. (2020)) and how it focused my attention on the larger and longer configurations of the sociotechnical systems—the extensions—of the data engineer web search practices, the various components and their modes of engagement, and with particular attention to how attending to the engagements between components reveal or make salient the practical achievements of the data engineers. Then I discuss the legitimate peripheral participation (LPP) analytic (Lave & Wenger (1991)) and how I use it to identify and understand the data engineering practices of situated learning. These frames, or lenses, help me recognize the role of various social and technical factors in the construction of successful uses of web search. I then discuss my methods. I describe how I developed this research site as multi-sited and networked. This methods section also covers my document analysis, interviewing, sampling and recruitment, coding and memoing, my attention to surprise and use of member checks. I close this chapter with reflections on my positionality and study limitations.

There are then four analytical chapters, discussed below, followed by a conclusion.

Admitting searching

The first chapter discusses how data engineers come to learn how to search for work, with an application of the LPP analytic (Lave & Wenger (1991)). There are two core ideas. The first is a finding, an empirical observation that there is limited explicit instruction, discussion, demonstration, or collaboration in the moments of web search in data engineering. Given the minimal instruction, I looked for "search talk", where data engineers might discuss their search queries, search result evaluation, or how they reformulate queries or follow links in pursuit of an answer. I do not find much "search talk" for new data engineers to learn from, rather, questions about it revealed how they see web search as a sensitive topic).

I then describe a key type of *talk about search* that I do find: search confessions. This is the second core idea of this chapter, an argument. Search confessions are the self-deprecating (yet proud) or hyperbolic remarks data engineers make about their extensive reliance on web search and their web searching practices. I describe how these confessions legitimate their searching, shape norms of use, and direct others to also rely on web search. The informal

nature of this legitimation, though, does not fully delineate appropriate use or the limitations of using web search for work. Search confessions also do not fully address perceptions that such reliance is a sign of weakness or is otherwise shameful. This limits the full inclusion of search into the workplace and affects organizations' abilities to build inclusive learning environments.

Extending searching

Next, I turn to look at how the two analytic frames help see how the occupational, professional, and technical components of the work practices of data engineers effectively *extend* web searching to include activity well-before and -after the time the data engineer is in the search bar or on the search results page. This claim of a sociotechnical practice be extending is not unique, but it undergirds the two core arguments of this chapter. First, it is not that the data engineers know more about the technical mechanisms of search, but that their work tools and practices and domain expertise make search work for their work purposes. This stands in contrast to claims that appear throughout literature on search engines that views individual ignorance of search mechanisms as contributing to failed searches and search literacy as a necessary, if independently insufficient, path towards mitigating search failures.

Second, the occupational, professional, and technical components, as extensions of web search, provide sites and activities for new data engineers to gradually increase participation in the search work of data engineers. Even though the search activity in the search box and on the search results page itself is not shared, the new data engineer can participate in the larger work practices that scaffold web search.

I find this extension in looking at how data engineers engage in two core aspects of web search: (1) where their queries come from and (2) how they evaluate search results. The occupational, professional, and technical components provides scaffolding, or helpful guiding structure, for their web searching in both aspects and this scaffolding provides a significant supplement to their domain knowledge. I show how data engineers find some of their queries in the tools they use (in the names of the functions they are struggling with, or the exception messages returned when there is an error). While they have learned to often quickly evaluate which links on a results page are likely authoritative or helpful, they also engage in collaborative evaluation of search results with both their systems (running a test of the changes suggested from the search, building prototypes, automated testing) and peers (meetings and code review processes). In a supplementary finding from looking at how the components of the data engineer's work interact, I identify how the data engineers larger work practices also provide ways of decoupling data engineering performance from potential issues introduced through web search, in cases where their search evaluation may fail. I also note how this success in searching is not all-encompassing and is limited to only some types of system qualities and does not include topics outside their direct remit (like ethical or legal aspects of the systems

they are designing).

Repairing searching

The data engineers sometimes are not successful in their searches. I look at how data engineers repair failed searches. For the purposes of this chapter, failed searches are where the searcher does not find workable answers to their questions. Data engineers have developed practices, similar to those found elsewhere in coding work, for addressing such search failures. Building on the two prior chapters, the first core argument of this chapter is that these practices provide both additional *talk about search*, further legitimating it within their work, and opportunity for the learning data engineers to participate in extensions of web searching.

While not focused on the search queries themselves or the moments of searching, the asking and answering of questions among colleagues provides a key opportunity for data engineers to learn about how each other search, including sometimes when, where, and for how long. The repair attempts involve carefully packaging questions, and answering them, in ways that are sensitive to respecting each other's knowledge and place as experts in their field. Here I analyze these interactions around packaging questions, performing competence, and prompting renewed searching. This repair work, in addition to filling in where web search is not working for them, also provides teaching moments — discussing problem solving & searching writ large (as well as serving as a small site for coworkers to coordinate around what they each know or not).

The second core argument of this chapter is that the search repair practices, as a whole, constitute the articulation work necessary to support such heavy reliance on web search. The search repair practices are a way to decouple from web searching itself, providing repairs necessary to retain such reliance.

Owning searching

The web search activity of data engineers remains solitary and private. In fact, the data engineers' managers and employers do not seem to manage the search activity itself, despite its importance to the data engineers work and the extensive behavior trace data available. I did not find tracking of search records or technology-enabled management of the search activity. Why? In this chapter I develop a sustained argument building on two core findings. First, individual data engineers identify themselves as responsible for their web searching. Second, management has not pursued a strategy of technocratizing search, or the intentional application of technique to influence search activities themselves.

I present data engineers describing their searching as solitary, speedy, and secret. Search is presented as done alone, apart from colleagues, and on one's own, apart from the help of others. Search is hoped to provide a faster solution to problems than alternatives. Search

activity itself is also done in secret and kept secret. I recount the lack of technocratization of search for the work of data engineers and discuss aspects of technocratization elsewhere in search. I then review a range of literature to make further sense of this, looking at the design and articulation of search engines as single-user tools and made to protect privacies, the history of "rugged individualism" in coding related fields (Ensmenger (2015)), and discuss norms of both generalized reciprocity and self-reliance in open source communities (norms and communities that overlap with many data engineers).

I build a sustained argument from the analysis of the secrecy and seeming lack of ownership by the firm in the data engineers' searching activity. To do so I appeal to both organizational theory related to learning and research on learning benefits of privacy. Organizations have distributed search responsibilities to individuals (Girard & Stark, 2002; Stark, 2009) and both "skilling up" and "keeping up" are the responsibilities of individual workers (Avnoon, 2021; Kotamraju, 2002; Neff, 2012). I analyze this as a strategic choice to pursue flexibility in the face of uncertainty (perhaps made mimetically (DiMaggio & Powell, 1983)[9]). I argue this pursuit of flexibility and associated responsibility creates a feedback of solitary and secretive searching that, while perhaps generally successful, limits the learning of the organization, particularly the sort of inclusive learning necessary to welcome newcomers that are different from those data engineers in positions of power.

1.3 So what?

The conclusion looks at the eight core findings and arguments from the preceding chapters and highlights two further arguments developed across the chapters. While the general story of this research is an examination of the factors that support successful use of web search by data engineers, around each of those factors are two risks. There are risks to organizational search performance and an inclusive learning environment. First, while generally successful for their purposes the effectiveness of data engineering search practices are limited by the ambiguity of the search confessions, the taken-for-grantedness of web search and the occupational, professional, and technical components supporting it, and firms' hands off approach to both search repair and responsibility for searching. Second, the silences and secrecies around search and the way search is framed as an individual responsibility can produce an unwelcome learning environment for new data engineers, limiting who makes effective use of web search and who learns to fully participate as data engineers.

In the conclusion I develop reflections for practice, design, and research built on key takeaways and make an argument for why this research matters. The dissertation closes with an appeal

[9] As DiMaggio & Powell (1983) argue [p. 151]: "Modeling" after, or mimicry of, other organizations "is a response to uncertainty."

for reimagining search. I suggest that if we can better see how searching the web is already shared, we might find ways to share it better.

2 Methods and Methodologies

Where does one go to see web searching happen, to see web searching at work? One could try to peer into tools used for indexing, the algorithms for ranking, the work of the humans rating the quality of the search results, or the design of the SERP. One could try to gain access to search logs, recording the inputs, SERPs, and the clicks. One could use a browser extension or similar software to monitor not only the activity on the search page but also the interactions and time spent on pages after that. One could observe, in-person or using virtual tools, as people go about work which might sometimes include searching. One could create exercises in a lab, track search and browsing activity, watch people, and probe them with questions. One could look at what people who are trying to be found on search engines say and do. One could sit a group of people together to talk about how they use search. Or send out surveys. Or use identified harms, failures, complaints, or others signs of breakdown as an opportunity to see more of search in an infrastructural inversion (Bowker, 1994).

The answer depends on how one initially imagines web searching. The above examples all provide some access to only a slice of web searching activity. Depending on what one wants to know about web searching, that may be enough. From the outset I took web searching to be a situated practice and wanted to understand how and why people search the ways they do. This is still only a slice, but the sort of web searching that I was interested involved people using web search engines. People search the web. These people have histories and contexts of action that shape how they imagine and use the search engines. People search the web within other, interconnected, practices. I took that "baseline conceptual identity" and made choices that defined (or constructed) my object of study (Marcus, 1995, pp. 105–106).

People sometimes use secondary machines to input queries or retrieve results for web searches on search engines. For instance, the WebSearcher tool built by search researcher Ronald Robertson (Robertson, 2022) (used in Lurie & Mulligan (2021)), SerpApi (a company that provides APIs (application programming interfaces) for scraping results pages from a variety of search engines, used in Zade et al. (2022). There are libraries for different programming languages[10] and various tools for searching within the command-line such as googler[11]. There are plugins for text editors[12] and integrated development environments (IDEs)[13]. Or the

[10]Try searching `[python "search results" title:google]` on Stack Overflow, `[python "search results" intitle:google site:stackoverflow.com]` on Google, or `[python "search results" duckduckgo site:stackoverflow.com]` on DuckDuckGo.

[11]See the archived repo on GitHub at https://github.com/jarun/googler.

[12]For example, Emacs is a text editor initially developed in the 1970s and actively used today. The MELPA repository of Emacs packages includes tools such as `fastdef` (to "Insert terminology from Google top search results", `google` ("Emacs interface to the Google API", and `google-this` ("A set of functions and bindings to google under point." — including "the current error in the compilation buffer").

[13]Extensions are available to make general-purpose web search engines searchable from within Microsoft's

secondary machine might be something as simple as a web browser that supports searching a highlighted portion of text through a right-click context menu.

Web search practices are not monolithic[14]. "Just google it" refers to any number of routines used in many contexts. I sought to better understand web searching, with its variations in practice and effect, by looking at the constituent components of the larger sociotechnical practice and their configurations. To do this I relied on two analytical frameworks: "Handoff" (Goldenfein et al., 2020; Mulligan & Nissenbaum, 2020) and legitimate peripheral participation (Lave & Wenger, 1991).

2.1 The Handoff analytical lens

"Handoff is a lens for political and ethical analysis of sociotechnical systems" (Goldenfein et al., 2020). I use the theoretical framework of 'Handoff' to guide my methodological approach and as an analytical lens. Handoff, or the "Handoff Model" was developed for "analyzing the political and ethical contours of performing a function with different configurations of human and technical actors" (Goldenfein et al., 2020). It "disrupts the idea that if components of a system are modular in functional terms, replacing one with another will leave ethical and political dimensions intact." (Mulligan & Nissenbaum, 2020)

What does a system do? How? So what? The Handoff analytic looks at sociotechnical systems. The functions of that system (what it does) are distributed across different actors or components[15] (how it does it). These components include people, software, physical objects, laws, and norms. These act on or engage each other in the performance of some function. These different modes of engaging may include force or perceptions of affordance[16].

Visual Studio Code IDE. These extensions are built to open search results pages from a host of search engines: Baidu, Bing, Brave, DuckDuckGo, Ecosia, Google, Yandex, and You.com.

[14]Dunbar-Hester (2020) notes the importance of remembering at the outset of her research on diversity advocacy in open technology cultures that the sites and groups and their activities she studies are not monolithic (p. 24).

[15]While it can also appear shorthanded to the broader "component actors" or subsystem, Mulligan & Nissenbaum (2020) generally use component:

> We use the generic term component to apply to both human and nonhuman parts of the sociotechnical system. While the term component does not naturally apply to human actors, for our purposes it is important to be able to refer in like manner to human and nonhuman components of a system.

[16]I take care to refer to 'perceptions' of affordance, pushing back on a tendency in some analyses to reify affordances as properties of the objects rather than accomplished in contexts and relations (see Leonardi (2011) and Vertesi (2019) for a review of the disjuncture between the development of affordances in psychology and its application in design). Leonardi (2011) draws on the terms "perceptions of constraint", "perceptions of affordance", writing that "Technologies have material properties, but those material properties afford

A Handoff is where a function of a system is shifted from one component to another.[17] The system is transformed even if it may be said to achieve the same functional goals. Some transformations may alter the achievement or protection of important values. The Handoff analytic can be used to focus attention on the components and their modes of acting on each other, for "exposing aspects of systems that change in the wake of such replacements," and illuminating those aspects and changes "that may be relevant to the configuration of values embodied in the resulting systems" (Mulligan & Nissenbaum, 2020). The Handoff analytic helps identify the redistribution of functions when sociotechnical systems are reconfigured. As Mulligan & Nissenbaum (2020) write:

> It may be that transformed systems embody more positive values, but it may be that replaced components, even performing purportedly the same task, lead to a degradation— such as, dissipated accountability, diminished responsibility, displacement of human autonomy, or acute threats to privacy.

I have been thinking with this lens since summer meetings in 2017 and 2019 as part of the Handoff team[18]. My initial search research (Mulligan & Griffin, 2018) was shaped by conversations in that first meeting. In the second meeting our conversations took on another search topic, the changed configurations of scholarly systems for search and evaluation in what became Goldenfein & Griffin (2022). While the latest formulation is in Mulligan & Nissenbaum (2020), I regularly referred also to an earlier working paper, Goldenfein et al. (2019) (presented at the WeRobot conference), which exposed the consequences of different configurations of imagined larger sociotechnical futures of autonomous vehicles.[19] I also looked to its use by Nick Doty, examining Handoffs in HTTPS and Do Not Track, and Richmond Wong, showing design workbooks as prompts for reflections on Handoffs and investigating responsibility in user experience professionals' "values work" practices, in their

different possibilities for action based on the contexts in which they are used." I do not want to stop at labeling some attribute an affordance, but locate "the networked conditions that make particular use cases possible" (Vertesi (2019)). Vertesi argues (p. 388) that:

> The notion that technologies might in and of themselves suggest, prompt, or require different ways of using them from human bodies or interlocutors neglects the richness and complexity that occurs when different groups take up technological tools to achieve local ends

I refer to standpoint theory (Hill Collins, 2002), situated knowledges (Haraway, 1988), and white ignorance (Mills, 2008) when considering how relations and contexts interact with perceptions of constraint or affordance.

[17] Goldenfein et al. (2020) provide a tight definition:

> **We define "Handoff" as the following:** given progressive, or competing, versions of a system (S1, S2) in which a particular system function (F) shifts from one type of actor (A) in S1 to another actor (B) in S2, we say that *F was handed off from A to B.* [emphases in original]

[18] Funded by the National Science Foundation under the U.S. NSF INSPIRE SES1537324.

[19] This paper was later published as Goldenfein et al. (2020).

respective dissertations (Doty, 2020; Wong, 2020). I also taught the framework in a class on Technology and Delegation (with Deirdre Mulligan) in 2019 and a Values Tools Workshop (for students in the School of Information to work through identifying implications for potential social values & ethics in their final projects) in early 2020.

Data engineers search the web at work within and across sociotechnical systems—they are also components of their firms and the field of data engineering. The various web search engines are sociotechnical systems, as is the World Wide Web they index. Data engineers use an assortment of tools within their work. Web search itself is a component of systems that people direct to, or hope might, achieve or maintain societal values such as privacy, autonomy, and responsibility. Van Couvering (2007) critiques web search engines for not advancing (and even degrading) values—or "quality goals"—such as objectivity, fairness, diversity, and representation (see also (Noble, 2018)). Tripodi (2022b) implicates web search engines as being manipulated by powerful actors spreading propaganda. Goldenfein & Griffin (2022) argues that the introduction of Google Scholar into the larger systems of scholarly search and evaluation disrupts academic autonomy. Some have suggested web search engines should or try to pursue "societal relevance", rather than system or user relevance (Haider & Sundin (2019); Sundin et al. (2021)). Ochigame (2020) argues for the pursuit of liberatory values. (This list of values and accomplishments or disruptions is not exhaustive.)

I use the Handoff analytic in my research to look at how people, and other components, perform web search, and to "scrutinize" (Goldenfein et al., 2020) alternative configurations. I narrowly look at how data engineers make web search work for them, focused on the functions that support them in accomplishing their work. There are values implicated in that, like responsibility and autonomy. I also look at dignity (how data engineers are treated, how they feel doing this work) and diversity (who web search is made to work for).

The Handoff analytic informed my research design and analysis. I do not develop a comparative study of the transformation from data engineering search work prior to web search engines to after. I use the analytic to explore searching with general-purpose web search engines, with different data engineers engaged within slightly different configurations. But I retained an appreciation for how the introduction of web search was a significant transformation. My document analysis disclosed how engineers and coders in the past searched with more emphasis on reference manuals, mailing lists, and circulars. Retired engineers would tell of their reliance on `man` pages[20] or the large volumes stored in their workplaces. While I did not focus my interviews on these reconfigurations, the shifts suggested by such changes—including

[20]These "manual pages" provide access to documentation. An engineer could access the documentation directly on their computer by typing `man [command, function, or file name]` in their terminal. You also may be able to type `man man` in your terminal or command-line application see the documentation for `man`, the "format and display the on-line manual pages". Or, for example, type `man cal` to reference documentation for the `cal` utility which "displays a calendar and the date of Easter."

the values of responsibility, community, security, and learning—stayed on my mind. I used hints of this broader shift to defamiliarize web search for me (Bell et al., 2005) and to help me look for the various components, functions, and implicated values. How might we think about the fact that employees are going to a general pool of resources and bringing code and ideas back to the firm without that errand closely monitored or managed? Wouldn't these firms want to know about information entering the organization?[21]

The analytic also shaped my findings. I will not try to present the findings anew here, but present only very high-level connections. The first analytical chapter looks at how data engineers make it clear to each other that they are allowed and expected to rely on web search. This is communicated despite the limited access people have of others searching. How are perceptions of affordance constructed? The second looks at how component actors engage with one another in revealing what they do not know. These engagements are conducted in a performance where people attempt to communicate that they have taken appropriate responsibility and that they are prepared to receive and effectively make use of information from others. These engagements often take place in forums that permit searching and publicity, attributes that shape perceptions of possibility and punishment. The finding in the third chapter—that the functional achievements of search are the product of sociotechnical accomplishments—stems from a direct application of Handoff to the descriptions of the various components and engagements provided by the research participants. The final analytical chapter deals directly with imagined or missing alternative configurations of search.

The Handoff analytic provides a complement to the situated learning framework and the legitimate peripheral participation analytic presented in Lave & Wenger (1991) (see next section), providing concepts to advance analytical applications of the latter towards learning in or about sociotechnical systems.

2.2 The LPP analytic perspective

The legitimate peripheral participation (LPP) analytic, developed by Lave and Wenger following an attempt "rescue the idea of *apprenticeship*"[emphasis in the original] and to "clarify the concept of situated learning", views "learning is an integral and inseparable aspect of social practice" (1991, p. 31).[22]

[21] A conversation with Andrew Reifers on January 1st, 2019. A later conversation with him, June 6th, 2021 helped me consider the pursuit of expertise and quality in different systems or subsystems enrolled in data engineering work. I could draw on research looking at how expertise is performed or quality is produced (or not) in open source and in recruiting, interviewing, code reviews, etc. In my notes at the time I asked: Where are quality and expertise accounted for, maintained, or responsibilized in coder use of web search? It is clear to me now that these subsystems also engage or act on data engineering web search practices.

[22] The LPP analytic is introduced more fully in the next chapter.

Lave and Wenger draw heavily on studies of apprenticeship, and pointedly not on formal schooling[23], in understanding and explicating situated learning. A chapter titled "Midwives, Tailors, Quartermasters, Butchers, Nondrinking Alcoholics" presents five ethnographic studies of apprenticeship. They use these cases to explicate and demonstrate the LPP analytic.

They argue "learning is not merely situated in practice, but is"an integral part of generative social practice in the lived-in world" and the goal of their book, "the central preoccupation" is to present "legitimate peripheral participation" as not just a descriptor but an analytic (p. 35). LPP engages with "belonging" (p. 35), "relations of power" (p. 36), and "peripherality in both negative and positive senses" (p. 37).

I had many times discussed LPP in classes (particularly with Paul Duguid), read articles making use of it, watched it presented in teaching training workshops, and referred to it in my own mentorship.[24] In a manner perfectly predictable with the analytic itself, my understanding of it immeasurably increased as I worked to apply it within this research. I looked closely for example at the applications made by others, like the books and dissertations of former PhD students from my program (Mathew & Cheshire, 2018; Takhteyev, 2012). Only once I started writing, engaging with Beane (2019)'s (referring also to his dissertation—Beane (2017)) critique of LPP and reading and listening closely to the language in Lave and Wenger's book could I say that I had learned how to apply it.

2.3 Methods

To explore the web searching of data engineers with those two lenses, I developed a multi-sited (Marcus, 1995), networked ethnographic study. I used semi-structured in-depth interviews, subject to interpretive analysis, as well as document analysis. Conducting multi-sited interviews allowed me to talk with people in data engineering or adjacent roles in a variety of companies. Multi-sited document analysis allowed me to observe people discuss or document web searches or web searching in many places online. I started collecting and analyzing documents in December 2018. I started interviewing in June 2021.

2.3.1 Multi-sited and networked

I "followed" (Marcus, 1995), or traced, data engineer web search practices. I followed data engineers as they moved through different roles and reconfigurations of coder web search. I set out to follow the things—the web search engines themselves perpetually redesigned in code and reconfigured in the practices of searches; the search queries as refined and rejected;

[23] They discuss reasons, particularly in a section titled "With legitimate peripheral participation" (pp. 39-42) for "turning away from schooling" and "school-forged theories" of learning (p. 61).

[24] Jean Lave

the search results as justified or used to justify. I followed the people as they learned of, used, and reflected on the tools they used, the search queries and the search engines. I followed the conflicts where data engineers wield web search as a weapon or wonder if its use was a sign a weakness.

I drew on tactics from those who study algorithms and who see the performance of the algorithms as involving much more than the code itself (Bucher, 2017; Christin, 2017; Introna, 2016). One difficulty with studying algorithms in the wild is that the code itself can be hard to see. My focus was not on the code or the software or the algorithms, but practices of web search. Like algorithms, web search can also be hard to see. Web search is sometimes mundane and forgotten, opaque and invisible (Haider & Sundin, 2019; Sundin et al., 2017). So I made use of what Christin (2017)[25] calls "a somewhat oblique strategy" and describes as "refraction ethnography". If we picture light refracted by the prism, we can imagine we might learn something about the prism by looking at the light. In Christin's case, we might learn something about the algorithms by looking at how organizations and people act around them. This is very similar to "scavenging ethnography" useful for studying "obscure objects" and finding where they "manifest." (Seaver, 2017, pp. 6–7). Seaver writes that "the scavenger replicates the partiality of ordinary conditions of knowing— everyone is figuring out their world by piecing together heterogeneous clues." I followed, scavenging for tracks, the "technological artifacts" of web search and the material-discursive practice of coder web search "as they circulate[d] through institutional sites and epistemic networks". Christin "focuse[d] on how algorithms are refracted through organizational forms and work practices" in web journalism and critical justice. I attended to how the various algorithms imagined and implicated in coder web search are refracted through not only the work practices of data engineers, but how they are refracted as forms that organize their labor (as structures for the work practices and aids in navigating both capital and collective configurations of coder web search).

Burrell (2009) provides steps for field construction that support such following—following the people and things through refractions. I identified "entry points" while seeking to "maintain[]

[25]While she does not use the term in some of her subsequent work, (Christin, 2018, 2020a), Christin expands on and reframes this approach in a later article that presents "studying refraction" as a "practical strateg[y] for ethnographic studies of algorithms in society", calling it "Algorithmic refraction" (Christin, 2020b). She notes that it "partly overlaps" with methodological approaches like Barley's "technology as an occasion for structuring" and Orlikowski's approach to "sociomaterial practices" (p. 10; internal citations are to Barley (1986), Bechky (2003), and Orlikowski (2007)). She closes out her section on the strategy (p. 11):

> focusing on algorithmic refraction and treating algorithmic tools as prisms that both reflect and reconfigure social dynamics can serve as a useful strategy for ethnographers to bypass algorithmic opacity and tackle the complex chains of human and non-human interventions that together make up algorithmic systems.

a concentrated engagement with the research topic" (pp. 190-191) and "consider[ed] multiple types of networks" (p. 191). Entry points included mentions of web search in relation to software engineering or coding work (and so many other mentions of web search). I'd find a popular tweet about coding work and search and read the replies, see links within to blog posts, and then find those blog posts discussed further on forums.[26] Or the replies would have stock phrases that I could search in turn. I'd search these on Twitter, Reddit, or a web search engine. One could enter into at any number of points and be sent around from one social media platform to a forum then a Q&A website, followed by a blog, and back again. This was a "traveling through" (Burrell (2012), pp. 32-33). These webs spanned decades, linking patterned jokes, questions, and articulations about search together. I had saved searched on the Twitter app on my phone. I would use idle time to search out new mentions of googling, though my Twitter feed—informed by my curation and constant engagement—also might offer the repetitive jokes and confessions of searching. Better were my friends and colleagues who would send examples they saw my way.

Another set of entry points were the questions and answers on websites like Stack Overflow, or the issues raised in GitHub repos or Slack[27] workspaces for tools used by data engineers. I saw patterns very similar to those mentioned by my interviewees, or depicted in the screenshots they shared. Then also the tutorials, blogged reflections, and formal training materials on tools and strategies for data engineers. I reviewed them for topics related to web search (and its alternatives) and to immerse myself in the language of the tools and the contexts my research participants discussed. These entry points were always partial and they "did not fully contain," even when viewed in composite, "the social phenomena under study" (Burrell, 2012, p. 32)–though scavenging and refraction anticipates that. These entry points, though, provided access to the people I would talk to, the words and tools and jokes they used, and let me observe others like them interacting. These overlapping and interlinked networks did not let me observe or question data engineers as they input a query or made sense of a SERP, but they did let me see how web search in the work of data engineers is so much more than

[26]For one cursory example, see Scott Hanselman, not a data engineer, but a developer who works in Microsoft Developer Relations. A decade ago he tweeted a link to a blogpost of his titled: "Am I really a developer or just a good googler?" (Hanselman, 2013). (This blog post is an anchor for many conversations about searching the web while working in code-related roles, this post is linked to throughout such conversations over the years.) Then, he tweeted more recently (Hanselman, 2020)

> Me: 30 years writing software for money. Also Me: Googles how to hide a div with a querySelector. Hang in there #codenewbies".

[27]Slack is a workplace messaging platform billed to replace email. Every interviewee, except for those working at or with companies that designed and sold a competitor product, mentioned using Slack in the workplace or for interacting with external collaborators. It became the "professional networking site" for the UC Berkeley School of Information during my time there.

that.

I also make space for the "[i]magined spaces" or "speculative imagination" depicted or disclosed by the words and actions of research participants and coders engaging publicly online (pp. 193-194). One such place is the imagined Google, Takhteyev (2007), in his ethnographic work on Brazilian software developers, noted that "the Google campus" "serve[d] as a source of tremendous symbolic power" and was "the single most important place in the imagination of the developers" [p .5]. With my training in scenario thinking, and following the examples in the presentations of the Handoff analytic (Goldenfein et al., 2020; Mulligan & Nissenbaum, 2020) and Noble (2018)'s comments on imaginings[28], I attended closely to suggestions of alternatives ways of searching or alternative designs of work or the search engine.

In my field construction, I "lack the "deep hanging out"" (Dunbar-Hester, 2020, p. 25) and did not pursue "total immersion." I did not gain access to proper sites of work to observe or gain employment to participant fully in the work myself. But I followed the "driving logic" of field work: "that we can gain analytic insight by inserting ourselves in the social milieu of those we seek to understand" (Coleman, 2012, p. 5). But, over the four years since first collecting and analyzing the documents on coders searching the web, I have been engaged—in an ethnographically-informed and digitally-enabled way—in the host of activities imagined of fieldwork: "participating, watching, listening, recording, data collecting, interviewing, learning different languages, and asking many questions" (Coleman, 2012, p. 5).

2.3.2 Documents

Before starting my interviewing, I collected and analyzed documents related particularly to data engineer use of web search at work, include public social media, blogs (and comments), books, YouTube videos, podcasts, instructional materials, and policy guidance. In addition to the coding approach indicated below, I conducted situated document analysis, working to include consideration of production, consumption (and use), and content (see, ex. Prior (2003)).

I read through hundreds of pages of primary material from the public web related to the use of web search by coders broadly: original posts and commentary on personal blogs, Twitter, Quora, Reddit, Hacker News, and Stack Overflow about questions like: am I a real programmer if I have to look everything up? do expert coders search all the time? how did software engineers work before Google was invented? discussions about the role of web

[28] Noble (2018, pp. 181–182):

> Indeed, we can and should imagine search with a variety of other possibilities. [. . .] Such imaginings are helpful in an effort to denaturalize and reconceptualize [. . .]

search in Stack Overflow and jokes and memes related to coder reliance on or expertise in web search (ex. referring to software engineers as professional googlers). I created Twitter lists to watch how people who coded for work talked about their work generally.

The material I gathered went back to 2003. I also looked at training materials, books (popular and academic histories, resources for coders), tutorials, and other introductions to programming to familiarize myself with the community of practice and looking specifically for elements related to coder web search (like commentary on troubleshooting or discussion of norms around "reading the fine manual"). And I've also looked at research within computer science and empirical software engineering going back decades as it relates to searching and searching the web. I have also read books and material about programming and software engineering, to familiarize myself with the work and craft and to review for connections to the subjects at hand. This includes books like Hunt & Thomas (1999) and Seibel (2009), and the more data engineering oriented Kleppmann (2017).

During the interviewing I also joined multiple Slack workspaces for open source tools for data engineers and the sub-Reddit r/dataengineering (for background, not for analysis). In the course of the interviewing portion of this research I was also provided or referred to documents by the research participants. Documents provided included training materials, screenshots of workplace messages, screenshots of code comments, tweets, blog posts, and news articles. Research participants shared these during the interviews, as well as months after.

2.3.3 Interviewing

I conducted 38 semi-structured in-depth interviews with 30 participants (see Appendix I. Research Participants). The interviews were semi-structured, some questions and topics were planned in advance (see Appendix II. Annotated Interview Guide), derived from approaches suggested by the literature and developing research. I modified the interview guide over the course of the research. The interviews were conducted over phone or Zoom video calls. I took notes during the interviews and recorded and transcribed them.

I treated "interviews as fieldwork", as "part of the world in which research subjects live and make meaning" (Seaver, 2017, p. 8). My interview guide was directed towards identifying different values, functions, different components, and the larger contexts of web search. This was to guide my analysis with the Handoff analytic. The modes of engaging between two components, particularly when one component is a human (i.e. force and perceptions of constraint or affordance), can be made somewhat accessible through interpretative interviewing. Interviewing also allowed me to approach concerns of the LPP analytic: approval, belonging, participation, and peripherality. Pugh (2013) argues interpretative interviews can provide "different levels of information about people's motivation, beliefs, meanings, feelings and

practices" (p. 50). I noted and probed "display work" (where interviewees presented their best face); metaphors and jokes; laughter, silences and non-verbal communication that show "what kind of things are uncomfortable, horrifying, emboldening, joyful"[29]; and dissonances (pp. 50-51). I attended to, questioned, encouraged, and prompted "specific examples" (p. 50).

Key changes I made to the initial versions of the interview guide were to add questions at the start and end. I asked an "initial reaction question" at the start of most of my interview, asking them how they initially reacted to hearing out this topic of research. This was a very fruitful question that revealed people noting surprise or appreciation, stating they had never thought of the topic before, or describing the extent they used web search in their work. I was then able to refer back to that initial reaction later in the conversations. I also added a question at the end to ask whether they had any final questions for me and asking for any final reflections. These worked as long as I asked those with enough time left or the interviewee granted an extension. The first was fruitful in revealing their concerns or interests. This sometimes shaped my understanding of the preceding conversation and also sometimes challenged my understanding of search at work itself. The second, asking for final reflections, was useful for that as well, but also often revealed an appreciation for the chance to talk and reflect on this work practice. I used those sorts of comments in subsequent recruitment efforts and referred to them in analyzing consequences of the often solitary and silent search practices.

2.3.4 Sampling

My sampling for research participants was driven by choices around transferability, the Handoff and LPP analytical frameworks, and pragmatic concerns. To gain tractability, when shifting to interviews, I narrowed my focus to a subset of those who write code for work: data engineers. I selected this site for four key reasons. First, it seemed likely to include people who were relatively technically sophisticated and so particularly capable of appreciating the technical mechanisms of web search. Second, it appeared to be a particularly dynamic field that requires a significant amount of learning on the fly (Avnoon, 2021; Kotamraju, 2002) and so perhaps even more heavily reliant on search than 'coding work' is generally. Data engineers are involved in building and maintaining tools sometimes used in efforts to control, replace, or surveil other people, practices, and tools. So, third, I saw them as perhaps more likely to have sophisticated understandings of the underlying technologies and an appreciation, perhaps, for the uses and misuses of data and automation built on or around it[30]. In that,

[29] Such as the moment an interviewee halted mid-sentence, appeared startled, and then urgently confirmed that I would not name their company.

[30] See the work from Passi, based on his ethnographic fieldwork with an industry data science team, and colleagues on the "problems of trust and credibility are negotiated and manage" (Passi & Jackson, 2018, p. 2), "the everyday practice of problem formulation" (Passi & Barocas, 2019, p. 3), and "what work the system

fourth, data engineers would likely also be capable of redesigning of refashioning their tools and practices in the face of perceived constraints or affordances (Bailey & Leonardi, 2015; Leonardi, 2011; Vertesi, 2019).

While following trails to find data engineers I generally passed over attempting to recruit those working in the most prominent technologies companies, sometimes grouped as FAANG (Facebook, Apple, Amazon, Netflix, and Google). It seemed likely that they may have resources and constraints very different from most data engineers and research overly focused on their web search practices may have limited transferability (or at least developing and demonstrating transferability may have been harder). Concerns about transferability also led me to look to interview only data engineers who worked in teams with other data engineers and only to people in the United States.

The Handoff analytic informed my sampling. I recruited interview participants who worked in different contexts so I might explore the configurations of components, the system functions, and emergent values in different sorts of organizations. I looked for people working in different industries, at different levels, and on different types of teams. I looked at data engineering broadly, not constraining my search to people using particular tools or platforms. Some interviewees did not formally have data engineer as their title though their work tasks included those described as data engineering. Some interviewees worked with more established technology and others were tasked to work with the newest tools. I also spoke with people in relatively adjacent roles: a developer of open source software used by data engineers, a developer advocate for open source software used by data engineers, and a site reliability engineer.

The LPP analytic drove me to sample for people new to data engineering and full members. I looked for people who had responsibility for interviewing, hiring, on-boarding, and managing. I looked for people who had transitioned to data engineering from distinct work (like from software engineering or data analysis) and whose formal training prepared them to work as data engineers directly out of college.

I interviewed people from across the United States, east coast, west coast, and in-between. These data engineers worked on internally-facing online analytical processing (OLAP) systems for use cases like business analytics and customer-facing online transaction processing (OLTP) in production systems for providing access to documents, serving advertising, or making machine learning recommendations. These people worked in apparel, computing technology, enterprise software, entertainment, financial services, fitness, healthcare, media streaming, online marketplace, open source software, retail, social media, web analytics, and web

should do, how the system should work, and how to assess whether the system works" (Passi & Sengers, 2020, p. 2).

publishing. From the principal and staff level to recent hires, these data engineers were individual contributors, managed teams, and were the technical leads for projects.

The variance, or "de facto comparison dimensions" (Marcus, 1995), was emergent. Team size and history, organizational experience with contemporary data stacks (the tools and platforms used in the production of a data service), and competitive work environments provided sources of variance that made refraction (Christin, 2017, 2020b) more visible. Large and small teams with small and long histories, companies expanding into new-to-them data spaces versus those extending prior successes, and competition were not the variables of central interest, but they seemed associated with social dynamics and workplace relationships that contributed to concrete examples shared by the interviewees. This variance, or perhaps these extremities, seemed to make workplace web searching, at least in the interviews, less mundane (Sundin et al., 2017) or invisible (Haider & Sundin, 2019).

The Handoff analytic helped illuminate how the larger systems of web search in the work of data engineering engaged with different people. I was particularly interested in identifying potential harmful patterns in these practices (including how people respond with resistance, repair, or routing around). Prior work shows that women in the coding professions are often mistreated or pushed out. This is shown in research showing how 'coding work' came to be presented, perceived, and performed as masculine work in the 20th century (Abbate, 2012; Ensmenger, 2010; Hicks, 2017), which persists (Dunbar-Hester, 2020; Misa, 2010). My document analysis, prior to starting my interviewing, suggested there was a high degree of secrecy surrounding search, so I knew it was important to hear the accounts of women to gain a fuller understanding of data engineering web search practices, and their implications. I interviewed women data engineers at nearly double their representation in the field (36.6% to 18.5%, according to one demographics report[31]). This helped me identify and document an environment within web search practices in data engineering sometimes filled with anticipations of shaming and unequal judgment, particularly visible to women and people new to the field.

I failed in my attempts to recruit black data engineers, a limitation of this study. Talking with people from multiple marginalized groups and situated within multiple may have allowed me to better understand how they both identified and responded to harmful practices. I did not focus on other demographic characteristics, unless they became salient in the course of an interview. Most of the data engineers I interviewed presented as white or South Asian.

There were also pragmatic concerns—identifying and contacting data engineers. To provide practical (as well as analytical) scoping for my research I planned to only interview people in the United States (and this was also in my IRB protocol). The multi-sited networked design

[31] Zippia. Data Engineer Demographics and Statistics in the US. https://www.zippia.com/data-engineer-jobs/demographics/

of the study also meant that sometimes I was looking for all sorts of "entry points", unable to plan or predict access (Burrell, 2009, pp. 190–191). This meant I was unable to contact for interviews many data engineers who I was able to identify contact information for or who directly reached out to me on LinkedIn.

2.3.5 Recruitment

Research participant recruitment started with snowball recruitment among with friends and classmates and my extended network. I also reached out directly to people discussing data engineering online. After six years at the School of Information and living in the Bay Area, I had many ties to individuals (friendly, professional, and collegial contacts) who worked at places where data engineers worked. While some people with connections to UC Berkeley or the School of Information noted the connection as reason to respond to my request, others I contacted stated (unprompted) an interest in helping an academic or helping researchers. Several interviewees were introduced or connected to me by colleagues or friends who I asked for recommendations or contact information.

I posted messages in channels on the UC Berkeley School of Information's internal Slack workspace[32], my Twitter account, my LinkedIn account, on a sub-Reddit for data engineering[33] (first contacting the moderators for permission), and my personal website[34]. I received multiple inquires referencing the LinkedIn post, though none specifying Reddit or Twitter. I also sent direct messages to people on Twitter where that was an option and I was not able to locate an email address. For immersive purposes and for recruitment I 'followed' publicly engaged data engineers and those in adjacent roles on Twitter, including making a Twitter list. I found data engineers to reach out to on social media websites, forums, and blogs—traversing links to find email addresses or homepages with contact forms through Hacker News, GitHub, and LinkedIn.

2.3.6 Coding

I made use of some methods informed by grounded theory in efforts to ground my observations, analysis, and representations. While not taking on all their prescriptions or descriptions (i.e. the notion of "data reduction" gained from applying metadata to other data), I used "flexible coding" (Deterding & Waters, 2018). I coded written records of my encounters with informants, transcripts of interviews, and documents collected with codes derived in part

[32] At the time of this writing there are over 4000 members of the Slack workspace, with over 200 in the `#data_engineering` channel.

[33] https://www.reddit.com/r/dataengineering/

[34] My website recruitment pitch is archived at Archive.org: https://web.archive.org/web/20211024103319/https://danielsgriffin.com/currently-recruiting-interview-participants-dissertation.html.

inductively but also according to expectations and anticipations from prior research and sense-making of the research area. (Note that while the goal was to develop an emic account of coder web search practice, the deductive codes assisted me in attending to my expectations and improved my ability to identify sources of changes in my thinking). These codes included also my "own emotional reactions to the respondent's narrative", where I recorded not only surprise but disgust and heartache (Pugh, 2013, p. 51 (Fn5)).

2.3.7 Memoing

Throughout the research I wrote memos. I wrote memos on what I was confused by, surprised by, and excited by. Rather than developing a regular rhythm or routine for memoing, I scavenged (Seaver, 2017) for memos. I regularly turned my stray remarks from others or times I caught myself going on about my research to anyone into opportunities to put my thoughts to tangible words on paper. Or I would start to type a question to a colleague and realize partway through that the question could also (or instead) be a memo. I would write memos while reflecting on meetings with advisors——trying to find the words and so do the thinking to better understand something I struggled to convey to them. In some of these memos I juxtaposed competing codes or claims from interviewees. As an example, one significant memo was from a sudden realization I had on a walk and scribbled in the Notes app on my phone: "They've black boxed everything! By-and-large the practices and the tool itself." Tweets of mine, or many deleted tweet drafts, also provided seeds for fuller memoing. Some memos were born of my own search frustrations—a constant comparison across contexts that could be "contrasted, elaborated, and qualified" when juxtaposed with data engineering web searching (Orlikowski, 1993, p. 5). when I spent an hour disassembling and reassembling a pepper grinder to add peppercorns and only later realized I could have popped the top off, which I might have known had I done a web search—the name of the grinder clearly visible. Or when I bought the wrong size screws for hanging drywall. Many of these memos started from notes taken on runs—runs where I listened to books or articles[35]. My wife put up with me sometimes stopping mid conversation to jot down the stub of something for a future memo or writing it herself if we were driving. I could not force myself to write memos[36], and I couldn't stop myself from writing them once I had a thought that was not lodged.

[35] I used the VoiceDream Reader on my phone and computer for text-to-speech for listening to articles or books (that were not prepared for audio consumption), as well as drafts of this dissertation. Listening is a form of reading (Tattersall Wallin (2021)). In future research I will attempt to get IRB approval for storing interview audio, password protected, on my phone in order to immerse myself in the material in the sort of everyday listening occasions that Tattersall Wallin describes. I will confess that I did also listen to interviews, on cordless headphones played from my laptop, sometimes while playing with our, at the time, 1-year old.

[36] Early in this research I wrote a small script that would create a new blank memo file for me to fill at the start of every work day. I thought to set aside time at the start of the day to ensure I did my memoing. My morning memo. Eventually I went back and deleted hundreds of files that I had left empty.

A significant breakthrough in my sense of my study occurred when one of my interviewees asked if I would talk to their organization about my initial findings. This led to a flurry of activity as I hurriedly repackaged my initial memos and summaries of interviews into tentative findings. I then sat with the raw materials of my presentation, the starts of different framings and outlines. I reflected on how I felt in the presentation, claiming or hinting at different tentative findings. This forced me back to the interviews, to make codes I had felt but had not yet recorded. Soon thereafter, I had conducted 25 interviews at this point, I reached "meaning saturation" (Burrell, 2009, p. 194), at least within the scope of work possible for this research project.

2.3.8 Surprises

I attended carefully to surprises, noting them through in interview notes, memos and scavenged memos from off-the-cuff response to questions from advisors and colleagues. Surprises noted include:

- lack of criticisms of the search engine, minimal sharing about search practices or search stories
- no bespoke code to aid searching ("gap-bridging" (Bailey et al., 2010)), missing strong norms of reciprocity
- no hints of surveillance of web search, even reference to one's own search history.
- near ubiquity of the use of Slack workspaces
- hyperbole in confessional statements of heavy reliance on web search (mirroring material found on the web, but surprised in the moment by the repeated strength of exaggeration in interviews)
- vast variety in revealed knowledge of the mechanisms of web search
- data engineers saying they had never thought about search or had taken it for granted—search as invisible or mundane (Haider & Sundin, 2019; Sundin et al., 2017)—the data engineers were so reliant on search that I had imagined they would see it as their "topic, or difficulty" (Star, 1999) rather than taken for taken-for-granted infrastructure

I created a document to track when I changed my perspective on what my data meant. These were my shifts or pivots, a change in direction due to a realization or recognition of something previously unseen, something like a surprise. As I combed through prior memos I began to write mini memos of those shifts I had not explicitly recorded. Memos began to be a way for me to understand my prior shifts in emphases or framing. I could see where sometimes my understanding of a theme or chapter completely changed. These shifts were not small. For example, at first I did not see the knowledge embedded in the practices. I did not see where legitimate peripheral participation might exist in these solitary and secretive practices. And just a month before filing, my thoughts about the privacy of searching shifted

from a focus on the benefits to recognizing a need to be clear about potential harm. Some of these shifts or pivots were from doing the work of writing the drafts—the "activity of writing", the "physical act of writing" itself, was "a process of discovery [. . .] and surprise" (Lofland & Lofland, 1995). Some shifts and pivots were from challenges raised by peers, advisors, or research participants. The intentional processes of memoing—including reviewing, synthesizing and building on my memos—and working to explain my findings to others helped me to re-examine the data and see where it was showing me something different from what I had originally seen.

2.3.9 Member checks

Starting in the summer of 2022, I reached out to some of the data engineers I had previously interviewed and new participants for member checks. Also called "participant or respondent validation" (Birt et al., 2016), these are for "communicative validation of data and interpretations with members of the fields under study" (Flick, 2009). For both groups I conducted a form of "synthesized member checking". Birt et al. (2016) writes that this "addresses the co-constructed nature of knowledge by providing participants with the opportunity to engage with, and add to, interview and interpreted data, several months after their semi-structured interview." I shared my analysis and findings with the interviewee, including referencing the prior interview and particular places I was planning on using their words, as applicable. For new participants I provided an overview of my research like I did with initial interviews, asked my initial reaction question, and only asked a few orientation questions before presenting my findings.

I conducted member checks with seven prior participants (and exchanged only emails with another) and five new participants. I emailed the participants and used a scheduling software[37] to allow them to select to talk with me for different lengths of time, from 15 minutes to one hour. I did two 30-minute member checks with one participant. One of the new participants was someone I had reached out to in the previous round but never received a reply. Another participant I reached out to because they had tweeted about something that another member check had just mentioned. Only one other new participant was not on my original list of potential interviewees.

My emails inviting participants included a brief description of what I was writing in my dissertation drafts and a single tagline for the four analytical chapters (as developed at the time of the emails).[38] I opened the interviews with a high-level discussion of where I was in

[37] Calendly.

[38] I copied my early member check invitation emails to my website, linked to on my homepage and from my prior interview recruitment page. It is archived at Archive.org: https://web.archive.org/web/20221125080137/https://danielsgriffin.com/currently-conducting-member-checks.html

my research process (conducted initial interviews, coding and memoing, following by iterating on drafts of chapters, and now turning to member checks) and a description of the goals for the member checks. I told them it was similar to an interview, requesting consent to take notes and record. New participants were also given the same consent for research forms. For some of the participants I mentioned specific things they had said in the initial interviews that I was quoting and discussing in my text, sometimes at the front but mostly interspersed with our conversation.

I wrote a guide for myself on how to introduce the member checks. While I often paraphrased in the actual conversation, I had written the following: "I want to get feedback (including pushback & resistance), ideally feedback in your own words (rather than yes/no) that might refine, extend, or challenge what I'm saying. This is co-constructed research, so I'll be upfront: my dream is sorta like in improv: I say something and you say "yes, and", "yes, but" OR "no, actually"." I also suggested high-level questions for them to consider:

- Is the analysis believable/credible?
- Anything feel off? Or exciting?
- Is the analysis useful or meaningful to you?

After the first two member checks I made sure to add this as well: "Please do challenge me, in my last member check someone raised a question that helped me identify a key connection I hadn't seen at all before." I presented the member check in this manner with the goal of the participants seeing themselves fit to provide "commentary, correction, and elaboration" (Orlikowski, 1993, p. 10).

I generally discussed my tentative findings in the order I'd arranged the chapters in my drafts at the time of the interview (for most of the interviewees that was STRUCTURING, LEARNING, SHARED, GAPS).[39]

As I told the member check interviewees I would, I pulled material from our conversations into the dissertation. Sometimes they provided descriptions of experiences or reflections that filled gaps or reinforced my findings. I also make note and comment in the chapters that follow where they provided pushback. Sometimes that pushback identified for me things that I may be better off avoiding discussing or needed to take more care in discussing. I also highlight spots where comments from participants shifted my understanding of my research.

[39] Here is a mapping from those drafts chapters to the this final version:

- STRUCTURING => Extending searching
- LEARNING => Admitting searching
- SHARED => Owning searching
- GAPS => Repairing searching

2.3.10 Positionality

Before presenting my data and findings, I will describe some of my positionality. With this I "seek to acknowledge [my] situatedness and make it an explicit component of the research process" (Christin, 2020b, p. 13). This provides an overview of my background as related to my questions and sites of research, as well as a place for me and the reader to prepare to qualify or critique my standing for different claims I will make in the subsequent chapters. I had early exposure to computers, but I could not name them, unlike the developers that Coleman (2012) spoke with, who "would almost without fail volunteer the name and model number of the specific device" (p. 28). I was very attentive to approaches to teaching and systems of education. My father's parting words were almost always, "have fun." I learned that "creativity and play are socially motivated and socially learned" (Ames, 2019, p. 32) and of their role in learning, collaboration, and teamwork. I participated in a wide range of tutoring, mentoring, apprenticeship, and immersive training in social environments that ranged from caustic to liberatory. Experience with searching and teaching search—trying to search alone, managing teams of intelligence analysis reliant on search, searching as a data analyst, and teaching new programmers—shaped my attention and the comparative analysis across domains and situations for searching. I will focus on experiences related to searching, learning to use software systems or programming languages, and training (particularly that outside of formal schooling).

I remember being excited when my oldest sister mentioned HotBot, a search engine launched in 1996. It seemed to me so much better at returning relevant results than the mismatch of search engines and search aggregators that I used at the time. But those were better, for my perspective as a child exploring, than the confines of AOL search, itself better than Microsoft Encarta. I was fortunate to have access to the family computer in our dining room starting sometime in elementary school. The first web search that I remember was searching `[marines]`, and then being whisked away by my mother when the search engine, perhaps Juno, returned gay porn. We also made regular use of the local library, so much so that I had my library card memorized for placing holds on books from home.[40] I vividly recall two older boys, at a birthday party sometime before middle school, trying to one-up each other while talking about how they knew how to set up a website. I was curious but didn't really understand. I was homeschooled until fourth grade, then in a co-op classroom for two years, before joining an "innovative" public middle school (where everyone had to take band). A close friend showed me how he'd learned to display scrolling text in his computer class in middle school, but there was not one offered at my school. I do not recall any formal lessons on searching the web, though I did take a keyboarding class in high school. After two years

[40]When I applied to work at the library while in high school, the librarian conducting the job interview noted my name and commented, "oh, you're the one checking out all the science books." I regularly walked out of the library with a stack of books much larger than I could read.

at the local public high school I switched schools to a private k-12 school, before spending my entire senior year taking classes at the community college.

In undergrad I slowly learned more about the internet, the World Wide Web, and the tools supporting their use. I set up a short-lived blog. After my first year I made a small static website as a parting gift to my dormmates—just a grid of names and photos. I thought I would create a website where students could share and collaborate on notes. I even pitched it to my undergrad advisor—a history professor. I got so far as setting up a website with MediaWiki and spamming classmates anonymously with links to my class notes. I installed a Linux operating system on my laptop but could not figure out how to connect to campus WiFi. My best friend and roommate was a math and computer science major but I was focusing on history, political science, and philosophy. I also studied New Testament Greek, and thought of possibly becoming a pastor. I'd signed up for a programming class one semester but changed my mind before classes started. I worked as a "junior office manager" in a small architecture firm my first summer after starting college. I made coffee, checked the mail, answered the phone, organized the office bookshelf, and convinced my boss I could learn enough by searching the web to make updates to the firm's website. My second summer I did an immersion program in Arabic. Both summers, and for a couple months after graduating from college, I also worked at the local McDonalds.

I joined the U.S. Army after college. I enlisted to be an all-source intelligence analyst, under the impression that I could take my philosophy degree overseas and make a difference.[41] "All-source" meant my role was to contribute to planning and synthesis of intelligence collected through a variety of intelligence sources, including imagery, geospatial, signals, and human intelligence debriefing and interrogation.[42] That involved a lot of searching. Beyond training in intelligence analysis—including in searching various systems—, I also learned how to operate weapons, jump out of airplanes, rappel out of helicopters, drive a 2 1/2 ton truck, and run in formation while singing cadences.

While deployed in Iraq my lieutenant required that at the end of every day I share the list of searches that I had done. These, on a secure network, were generally searches in Query Tree, an application on the U.S. Army's multi-billion dollar Distributed Common Ground System-Army (DCGS-A), but also searches of reports and briefings stored on Microsoft SharePoint and OneNote. One reason for monitoring searches was that, despite boolean search operators, searching Arabic names transliterated into English was fraught. While I could reference a gazetteer to identify preferred spellings, names were not controlled. During

[41] This is the same job that Chelsea Manning had while in the U.S. Army. Her unit replaced mine in Iraq. I talked and walked her, and other soldiers in her unit, through the workflows we had developed, including the search tools we used.

[42] In a fashion visible also in jokes about programmers just copy-and-paste code from the internet, people in more specialized roles would regularly joke that the all-source analyst's job was just to copy-and-paste.

that deployment I learned the basics of Microsoft Access and how to edit VBA scripts in Microsoft Excel, creating systems to store the results of my searches because the servers running DCGS-A regularly ran slow or had to be restarted.

Over the years I was promoted and became responsible for the work of a team of analysts. I taught them rudimentary lessons in searching the web and the multiple secure networks and databases we used. My teaching responsibility here mostly involved one-on-one and small team instruction and mentorship. As an intelligence analyst I taught (or trained) my soldiers geography, history, software systems, analysis techniques, and public speaking. They joined the army with an array of educational backgrounds, from a GED to a college degree. On the side I thought I should learn to program. I started basic introductory tutorials for Rails, Lisp, and Python but did not get far. My unit tested out Palantir, which definitely felt fancy and fast, even if it made excessive assumptions about the searcher's goal and the underlying data. I later deployed to Afghanistan, endlessly searching while generally safe in an air conditioned tent. I realized we'd been misinterpreting aggregated location data that Palantir provided (it functioned differently than our standard systems) and convinced our shop to stop using them, though got nowhere in my attempts to convince their engineers over email to make changes. I used other tools as well. I could read enough of Python to adapt some automation behavior in ArcGIS, a geographic information system that we used. I recorded macros in Microsoft Word to help me prepare documents in particular formats.

I finished my enlistment and applied to graduate schools to study collaborative learning, perhaps with new tools. I'd had fellow analysts search themselves into a frenzy of overblown fear or search just enough to be able to overclaim some copy-and-paste expertise. I'd had superiors think we could just pick up and use our systems without regular training. Even with training our tools could be frustrating to use and unreliable. My philosophy degree only got me so far.

After applying to graduate schools, I took a job as a data analyst at a charter school in New York. I helped prepare testing materials, process test scores, develop a file naming convention, review data analysis software for potential adoption, and fumbled around using jQuery to adapt Microsoft SharePoint. During that time I found out I had been accepted to the UC Berkeley School of Information. I received an email from the admissions coordinator, saying "the admissions committee does have concerns about your preparation in programming [. . .] Before the beginning of the Fall semester you are expected to take at least one introductory programming course [. . .]." So I took an "Introduction to Programming with C++" course at Borough of Manhattan Community College.

I took a short "Python Boot Camp" intensive before my first semester started. Then jumped into programming, taking an applied natural language processing class my first semester.

I've used Python regularly since[43], writing class projects or utilities for personal use. This document itself was produced with the aid of several scripting tools I've written and many code-related web searches. I then taught the "Python Boot Camp" for incoming graduate students for three years. This was a three-week intensive summer class for incoming masters and PhD students who were either completely new to programming or new to Python. The use of "boot camp" was amusing to me. I made sure I did not replicate the sort of embarrassment, harassment, or other punishments that came with some of my military training. (Though I had to be pushed by students to stop using a phrase from the military. I would often—intentionally—use "Too easy" to refer to things that were actually quite hard. It was supposed to motivate people, but I suppose only in the right contexts and in relation with people well-situated within those contexts. Part of the reason it didn't land in the class was because I had students who had never programmed and who weren't members of groups who were generally expected or welcomed in coding communities.) In classes I also worked in R and JavaScript. I was a professional masters student before joining the PhD program. My master's final project team built an automatic question generation system and I contributed to the Python codebase. I wrote Python code to pull tweets from the Twitter API for Burrell et al. (2019) and Griffin & Lurie (2022).

[43] I briefly used the Emacs text editor extensively, including writing small List snippets to change the behavior to the tool.

3 Admitting searching

> Now scientists everywhere use the air pump, say, or the electrophoresis gel without thinking about it. They look through the instrument the way one looks through a telescope, without getting caught up in battles already won over whether and how it does the job. The instrument and all of its supporting protocols (norms about how and where one uses it, but also standards like units of measure) have become self-evident as the result of social processes that attend both laboratory practice and scientific publication. (Gitelman, 2006, p. 5)

> Sometimes I just wonder, like, who taught them how to search? (Victor)

"To be frank I've like really never thought about it myself even though it's kind of like 90% of my job to just like look up things." That is what Amar told me at the start of our interview. To a certain extent the data engineers use web search without thinking about it. "It is kinda like breathing" (Phillip), "something that people maybe take for granted" (Lauren). So how then do they learn to search at work?

There is very little explicit instruction on web search practices in the data engineering workplace. Despite it constituting a significant portion of their work, not only are data engineers not taught how to search the web, they are also not evaluated directly on their search performance. While there are a range of onboarding processing and mentorship models, generally only in the earliest stages of their careers are new data engineers offered any direct advice about how to search or told how more experienced data engineers do so. Even at this early stage, advice and insight is sparse. Furthermore, data engineers rarely discuss their search queries, search result evaluation processes, or how they reformulate queries or follow threads in pursuit of an answer, what I call "search talk".

There is little opportunity for data engineers to directly observe or participate in other data engineers' searching. Its form–a small box on a terminal designed for individual use–affirms search as a solo act. In the absence of formal training, limited professional discussion, and a form factor that limits observation, one might predict that web search for data engineering, like learning to program a VCR, may be difficult to learn, as compared to "a fundamentally social practice" like learning to drive a car (Brown & Duguid, 1996, p. 51).

The analytic of legitimate peripheral participation (LPP), however, helps identify where data engineers are provided opportunities to participate and learn what it means to effectively use web search as a data engineer. Modifying Beane (2019)'s concept of shadow learning, "a set of practices in the shadows outside the legitimate peripheral participation typical of the literature on communities of practice" (p. 91) I locate participation and legitimacy in how search is admitted: "search confessions"—the self-deprecating or hyperbolic remarks

data engineers make about their extensive reliance on web search and their web searching practices (the topic of this chapter) and the occupational, professional, and technical forces that explicitly and implicitly structure search practices (discussed in chapters 4 and 5).

In the absence of formal training or apprenticeship, "search talk", or even visibility into the successful search practices of other data engineers, data engineers collectively wrestle with and affirm the appropriateness of their reliance on web search through "search confessions". At face value "search confessions" appear to be jocular, off-the-cuff jabs at the profession's reliance on search. However, in practice they affirm reliance on search—acting as informal search approbations. In conjunction with "search confessions", the absence of "search talk" further affirms the implicit acceptance of such heavy reliance on web search while also marking searching practices as private and generally and appropriately free from remark and appropriately protected from direct scrutiny. Search confessions are a site of legitimate peripheral participation, by exposing new data engineers to the constant process through which reliance on search and norms about its use are constantly negotiated, re-made and affirmed. While data engineers do not directly engage each other in the moment of searching, their web searching is informed by this confessional talk about search. Rather than being directly taught how to form and reform queries or how to evaluate and course correct, I find that through confessions about and around search and silences about exactly how to do it ("search talk") data engineers learn how to search.

The reliance on search confessions to normalize the use of search and, to some extent, train and educate data engineers about effective and legitimate use of web search in work practice comes at a cost. It presents barriers to those marginalized in technology work today (discussed in chapter 6).

The next section looks closer at the LPP literature, focusing on learning in the shadows. This is followed by a presentation of my empirical findings and analysis. Then I discuss implications for our understanding of LPP and re-situate this chapter within the dissertation.

3.1 The LPP analytic

Lave & Wenger (1991) claim that: "Learning viewed as situated activity has as its central defining characteristic a process that we call *legitimate peripheral participation*"[44] (p. 29). This concept is the outgrowth of their desire to write about apprenticeship and their phrasing highlights that learners "inevitably participate in communities of practitioners" and success in learning requires learners to "move toward full participation" (p. 29).

[44]I cleave to the label legitimate peripheral participation rather than shifting to language of "communities of practice" or "situated learning" in order to retain analytical purchase. This—"Taking into account the learner's perspective"—is the central focus of the theory and "has often been ignored" (Duguid, 2008, p. 3).

They propose the concept of legitimate peripheral participation to describe "engagement in social practice that entails learning as an integral constituent" (p. 35) The "central preoccupation" of their book "is to translate this into a specific analytic approach to learning" (p. 35). Cautioning against decomposing the concept into three components, they write that it is "to be taken as a whole." (p. 35) That is, there is not an illegitimate peripheral participant that learns and so challenges the theory, but rather the sort of legitimacy of participation will shape what is learned. Similarly, peripherality is about the "ways of being located in the fields of participation defined by a community" (p. 36). They suggest these "ambiguous potentialities" provide "access to a nexus of relations otherwise not perceived as connected" (p. 36) offering a new and distinct "analytical perspective".

Over the last 30 years LPP has been widely applied. John Seely Brown and Paul Duguid popularized it, also in 1991 (see Contu & Willmott (2003), p. 283). Other work on LPP includes Brown & Duguid (1991), Orr (1996), Brown & Duguid (1996), Brown & Duguid (2001), Contu & Willmott (2003), Bechky (2006b) (referring to both Bechky (2003) and Bechky (2006a)), Duguid (2008), Takhteyev (2012), and Gasson & Purcelle (2018).

3.2 Learning to be a googling data engineer

The LPP analytic lens helps direct attention to interactions between data engineers, to learning opportunities and participation. While direct participation in the moment of searching is rare (the data engineers rarely "pair search", even though some may pair program), participation in data engineering work practices provides opportunities for participation in the larger search practices. Data engineers do not have formal training in search. They do not collaborate at the search box or on the SERP. So I considered talk.

Talk is a key element of participation. Partly through talk, stories and jokes, people construct shared understandings of the work and their identity. This is seen in Orr (1996). Orr finds that "[n]arrative forms a primary element" of the practice of photocopy repair technicians [p. 2]. Talk is "instrumental", stories and conversations circulate knowledge of machines, customers, and the task of diagnosing and fixing problems. It also shapes identity, the technicians "tell tales to establish their membership in the community" [p. 142].

3.2.1 Search talk?

First, data engineers say they don't talk about it.

3.2.1.1 Don't talk about it There is limited explicit instruction. In the interview with Ross, after we talked about the various sorts of places he would search at work, he said the

following in talking about on-boarding a new hire:

> I probably had a brief conversation with them. That was, you know, five sentences that summarized what we've already talked about. 'You go to the web for this kind of stuff. Go to the wiki for this kind of stuff, and Slack for this kind of stuff.

Amar had recently started on-boarding for a new job after several years at a previous company. He had been successful there, rising to a technical leadership role within his team.

Midway through the interview I asked Amar: "Are you talking with your team about the searches you're doing? When they join your team are you saying: 'Here's my process for searching. This is what you should do.' Or?"

> Um, I think, that's an interesting question. [pausing and proceeding slowly at first] I don't believe I've ever done that except for... except with one engineer and the reason why I did that with that engineer is that was an intern and they were not very— That person was an intern who joined [the company] full-time but they didn't have a lot of professional work experience outside of internships.
>
> So they were a fresh grad, fresh out of undergrad. For them, because they didn't have a dedicated process—and it wasn't me going out of the way, because I'd never prescribe that this is how you should do it—But they kind of were 'Hey, every time I have a problem I have to like, like do a couple of Google searches and if I can't find anything I have to come to you and then you, even if you don't have an answer immediately you pretty much find it, find resources pretty quickly, how? So what do you do?'
>
> And then that's when, that's the only time I kinda said, hey this is my process and this *works for me* [emphasis from interviewee] but outside of that engineers don't really—at least in my experience or at least within my team—will not explicitly discuss their process.

Following up, saying:

> I don't think I've actually talked with the team, but maybe I should, as like a personal note. [. . .] it is just part of the job that usually not very apparent unless its like very very inexperienced engineers.

Pair programming (working together on the same code at the same time either next to each other or remotely) might be a place where searching is talked about. While the data engineers I talked to generally did not practice pair programming[45], those that did generally reported

[45]Several research participants did some pair programming, only a few indicated it was a consistent part of their work.

searches being hidden and not discussed.

Christina said there is less pair programming in data engineering than elsewhere, but said, "If I were pair programming with someone and I was sharing my screen I would have a tendencies to pull up my search on a separate screen."

Likewise, Megan said:

> I'll notice people turn off sharing when they switch to searching, and then they'll find something and turn sharing back on. . . . a lot of stuff people are really interested in collaborating on, but search is very private. It is something you go do and then you come back and share the results of your search. . .

Ross said that if he were doing a screenshare with a colleague on one of his screens and had to look something up he'd open a new tab in the other screen, not shared, and do his search, saying "I don't think I'm the only one like that."

3.2.1.2 Submerging of web search Second, there is a *submerging* of web search itself in talk nominally about web search. Even interactions that interviewees would describe as being about searching the web were not directly so.

Here is a response from Sameer, as we were talking about mentoring interns or new college graduates, when I asked him for an example of "politely suggesting googling":

> I've realized that schools, depending on which program you go to, computer science majors, have a lot of theoretical knowledge. Graduates will have knowledge about distributed systems, algorithms, data structures, but then actually coming to a company and writing code is different. So there's a lot of guidance and mentorship around that. And obviously if the intern or new college graduate does not have experience in industry then sometimes I do think we need to, politely, point them to search—'Hey, have you tried googling it? Because it seems like a very simple thing you can find yourself.'
>
> But I think sometimes people, when they are stuck in that rabbit hole it is a very thick forest. When you are googling things you can hit one web page and be like 'oh, I don't know what this means' and then go to a second line and "oh I don't know this is either!" and suddenly you're learning about quantum physics, right? [laughter] So, so, very far away from what you started out with. So you kinda need to understand what to Google, where to stop, and where to just ask someone for help.

I asked a follow-up: Can you recall any of those conversations or times when you politely suggested googling?

> So the way I do this is by, one of the easiest ways to do this, is just to send someone a Stack Overflow link and go, 'oh hey, someone already answered this question. Here you go.' And I hope they read between the lines, 'I should be googling this.'
>
> If I don't send a Stack Overflow link and I'm just solving the problem for them then I will definitely have a one-on-one and have a conversation, 'OK, this is how you should be solving it.' I totally get that this [answer to the question asked] is not common knowledge. But I hope that the intern or new college graduate can read between the lines. I don't want to have a conversation with anyone saying 'did you try googling this yet?' It's not very polite, I feel.
>
> And most people pick it up. It is very rare that someone would bother me with something easily queryable again and again.

Sameer presented this story of sharing a link (that he had found by searching) as an example of politely suggesting googling. He made no mention of googling or web search in his description of what he explicitly communicated to his colleague. The use of the web search tool itself is kept below the surface.

The coworkers in this example avoid mention of web search. This sort "tactful inattention" (Goffman, 1956, p. 147) reproduces boundaries or norms around whether or how the tool is mentioned itself.

3.2.1.3 Speculating on how junior engineers learned to search Third, there is speculation about how junior engineers learned to search. This demonstrates further the lack of formal instruction on search.

Victor, one of the data engineers who reported their team regularly pair programmed, described working with a junior colleague:

> One thing that I find like when I pair program with more junior engineers is the way they do the query searches is very different than I would approach it. So, they'll just ask me, they're like: 'Um, how do I run this command in docker.' It's like, I don't know. What— Like let's— I don't know. Like do you think I just memorize it? No. Let's, let's Google it. And their search: [docker [command]]
>
> Do you think that search is going to get the answer you want? How does that even happen? 'Well, what would you search?' Let's just repeat the question you asked me and type that into Google. Sometimes I just wonder, like, who taught them how to search? I remember being in like 5th or 6th grade and having a class about how to do web searches. I guess that's something not everyone does

> anymore

This speculation highlight the lack of visible formal search education in the workplace—made visible here in a collision of generational perspectives, perhaps. Sundin (2020) studied how older youth in Sweden use general-purpose search engines. He found that "search engines almost never seem to have been a visible information infrastructure for the current generation of teenagers" (Sundin, 2020, p. 378).

3.2.2 Talk about search

More broadly, these comments and stories from interviewees point to how search is multiply and complexly hidden. Search talk is absent partially because search has, for many interviewees, become infrastructure—habit and routine, "like breathing." It is also avoided because of the sensitivity developed from what it might reveal or suggest about one's own or another's knowledge, or lack thereof (the secrecy covered in the final analytical chapter, Owning searching). But, it may also be a tacit "action-centered skill" (Zuboff, 1988, p. 186), knowledge or knowing that "cannot be put into words" (Polanyi, 1967, p. 4).[46] Zuboff (1988) writes of how a richly textured tacit skill, is "deeply embedded in crisscrossing relationships, and too continuous to be captured in a verbal description" (p. 187). It is likely, Zuboff writes, that "attempts at explication of such tacit knowledge must always be incomplete. The knowledge is too layered and subtle to be fully articulated. That is why action-centered skill has always been learned through experience (on-the-job-training, apprenticeships, sports practice, and so forth)" (p. 188). Even if it were that explication of tacit knowledge about search activity is always incomplete and could not be fully articulated, that does not mean explication couldn't be useful and wouldn't be tried. But the difficulty of communicating tacit knowledge may multiply the sensitivity in talking about searching. If attempts to discuss it themselves seem to indicate some lack of self-awareness or inability to communicate. But in this case it isn't just that the data engineers don't *verbally* share their search activity, they don't *materially* or *experientially* share their search activity either. The data engineers do not join together in the action in the search bar or on the search results page.

The search activity itself seems relegated to (or reserved for) the backstage. As Goffman (1956) writes (p. 69):

> A back region or backstage may be defined as a place, relative to a given performance, where the impression fostered by the performance is knowingly contradicted as a matter of course. There are, of course, many characteristic functions of such places. It is here that the capacity of a performance to express something beyond

[46] See Cambrosio & Keating (1995, pp. 49–50) for a discussion of how "the unsaid" tacit knowledge can be "formally transmitted" and "articulated".

> itself may be painstakingly fabricated; it is here that illusions and impressions are openly constructed.

Certain impressions of an individual's knowledge or knowing, impressions necessary for data engineers to engage as experts in workplaces that conceive of knowledge as something possessed by and the responsibility of individuals, are developed in a protected backstage.

But they do talk *about search*—spotting examples of such talk led to my interest in studying this site. Even though they do not engage in "search talk"—discussions about what they input into a search box or how to parse the results pages—they talk about search. They discuss searching and the motivations behind searches although the searches themselves go unmentioned or are only subtly implied. This talk about search shapes their understanding of acceptable use of search. Here I will focus on confessions data engineers make about their use of web search. Repairing searching, the fifth chapter, discusses another space where data engineers talk about search, sharing about and fixing failed searches.

3.2.2.1 Search confessions Data engineers profess to making extensive use of web search at work. I expected such professing given my initial experience leading to this research. I did not expect, though, the forcefulness and apparent hyperbole and overgeneralization in these statements from the data engineers.

At the start of our conversation Amar laughed as he said, as though revealing some embarrassing secret, "it's kind of like 90% of my job to just look things up." Christina chuckled, saying "probably 90% of my job is Googling things." Ross, likewise, said, "it's a large part of my job." Over email while scheduling the interview, Vivek wrote: "Web Search is a part of everything I do." Noah said, "I consider it a core of doing my job."

I call these search confessions. Search confessions are, often self-deprecating or hyperbolic, statements about one's reliance on web search. Many of these confessions are delivered as though admitting something somewhat shameful, of something that others may find wrong or weak. Sometimes the confessions accompany a statement that there is nothing to be shameful about. Search confessions are statements that individually admit of a reliance on web searching and collectively admit the practice of searching for work into the work practices of data engineering.

These statements mirrored those that started my research, both the initial tweets and the subsequent blog, forum posts, tweets, and TikToks I later found. A key difference being that the above examples were directed to me and not to a general audience of peers or fellow community members.

Megan, talking of how people will admit to searching but not share the searching itself, shared:

> People who are all on board for 'do your work in public', 'show your mistakes', will still keep hidden the specific process of searching. That is something that they're not as eager to share. There are a number of people who say searching isn't bad, we all do it all the time. But it still feels somewhat shameful. I don't know what that is about. [. . .] Um, um, and they're not, they're not even like hiding it necessarily. They're like, they'll say like, oh, I'm just going to look this up real quick and pop it in, but they still do it in a separate window. See even if they're not trying to like, deceive about the fact that they're searching, they don't want the process of searching to be visible. Which I think is interesting. I don't know.

Christina is a data engineer working as a consultant helping external clients with her company's enterprise software tools. I asked about her initial reaction to hearing about the research topic and she said, "That probably 90% of my job is Googling things," and chuckled. She laughed when I told her that I wanted to know why everyone says 90%. She said, "because it's the majority, and it's not just the majority. I wouldn't go 50%, 60%, its like I can barely think of other things I do." Then, through laughter: "Meetings. And meetings. But I multitask so even while I'm in a meeting."

At the close of the interview I asked for any final reflections. Christina said:

> It was interesting because I hadn't connected the way I search day-to-day with our whole company initiative for developer experience. [. . .] So it makes a lot of sense, that while I am not often searching, googling, company-specific things, I wish I could but the answers aren't there.
>
> I guess my 90% I said at the beginning is actually pretty wrong because I probably spend a lot more time asking people questions than asking the internet questions.

In a follow-up member check interview I told Christina how I had identified and described search confessions. She said, "Yeah, yeah. This is completely coincidental," and went on to share that her IP address (working from home) had recently been blocked by Stack Overflow.

> I screenshotted the other day that Stack Overflow blocked my IP address because I sent too many requests. I screenshotted it and sent it to our team Slack channel. So it was kind of an acknowledgement of 'oh, look, this is kind of embarrassing I've asked too many questions.' My team is three people. We all know we ask a lot of questions, so it's not shameful.[47]

At the close of the follow-up with Christina, I apologized for running so long over our planned

[47]The "feels somewhat shameful" mentioned by Megan is regarding openly sharing the search process, distinct from admitting reliance on web search to others in a team of three.

meeting time and she replied, laughing: "No, that's OK. I've needed a break from searching."

Comments and jokes about Stack Overflow offer another means of confession. Recall Noah, who said he wasn't "the greatest googler of all time", and that if he found out that some of his coworkers were searching the web a lot less than him, he would "actually wonder if they weren't doing their job as effectively as they could." When I asked him about Stack Overflow memes he shared that he has a laptop sticker that says "Copying & Pasting from Stack Overflow". He mentioned the sticker again a year later in our follow-up member check.

Jane, an analyst working with data engineers at a prominent social media company, shared that the IP address at her company was blocked intermittently by Stack Overflow for a couple months. When it was blocked, engineers would gather on a page explicitly for memes and jokes, asking facetiously who was using it too much or joking that it must have been their individual fault for searching too much while trying to fix a bug. She mentioned Stack Overflow being blocked at her company to her friends, telling them not to have imposter syndrome because nobody knows what they are doing.

Jillian shared a story of joking among colleagues about surveillance productivity software:

> One time we were joking about these different productivity surveillance tools that some companies use, for working from home environments specifically. They might take a screenshot of what you're looking at on your monitor.
>
> And I was like, "oh, I would hate that because I'd be working but it would show that I'm like googling 'what is a computer' or like something rudimentary."
>
> And then, but we, everyone on my team was kind of joking about things like that, you know, just like talking about looking up, you know, this page for 'explain it to a kindergartner', whatever.

These search confessions serve multiple learning purposes. The confessions ritualistically provide a space for community members to affirm their commitment or conviction to this way of being an expert. Search confessions at once open up web search practices to challenge and create iterative openings for the community to foreclose potential threats to professional identity used by this general purpose tool through collective affirmation. Posted on blogs, forums, or Twitter, search confessions like the "Copying and Pasting from Stack Overflow" sticker on Noah's laptop or the "I HAVE NO IDEA WHAT I'M DOING" dog meme[48], both

[48]This meme was the center of a flurry of introspection in the online engineering community in November 2021. Comments on search confessions sometimes punctuate the everyday routine and elicit considerable discussion. I will only highlight a response from Lorin Hochstein (2021), a software engineer at Netflix, expert on resilience engineering, and regular commentator on the field. He wrote a blog post reflecting on discussion around a post from David Heinemeier Hansson, creator of Ruby on Rails and sometime tech influencer, which centered on the claim that "In the valiant effort to combat imposter syndrome and gatekeeping, the

mock and validate the data engineering community's reliance and dependence on web search. Confessions acknowledge and celebrate the difficulty of their work, constantly at the edge of the field or juggling far more (constantly changing) information than might seem possible.

Participating in these confessions is part of the ritual of these search-reliant fields. The rituals introduce and admit new members to the community and facilitate interactions between members. Borrowing from (Goffman, 1956, p. 121), we can identify search confession is "unofficial communication" that "provides a way in which one [data engineer] can extend a definite but compromising invitation to the other", through this sort of " 'putting out feelers' ", a "guarded disclosure." The confessions legitimate web searching and allow data engineers recognize each other's shared orientation towards search and drop pretense. Goffman goes on to write:

> By means of statements that are carefully ambiguous or that have a secret meaning to the initiate, a performer is able to discover, without dropping his defensive stand, whether or not it is safe to dispense with the current definition of the situation. [. . .] it is common for colleagues to develop secret signs which seem innocuous to non-colleagues while at the same time they convey to the initiate that he is among his own and can relax the pose he maintains toward the public.

These search confessions are a sort of secret sign. LPP identifies learning not as absorbing facts, but "deploying through practice the resources [. . .] available to you to participate in society, a process [. . .] inseparable from the development of a social identity" (Duguid, 2008, p. 3).

Confessions are not admissions of deviance, though they acknowledge a felt-deviance. They

programming world has taken a bad turn down a blind alley by celebrating incompetence." Hansson wrote, "You can't become the I HAVE NO IDEA WHAT I'M DOING dog as a professional identity. Don't embrace being a copy-pasta programmer whose chief skill is looking up shit on the internet." While many saw this as a critique of a reliance on web search, and provided apologias for searching, Hochstein focuses at a slightly higher frame, arguing (with citation to Bucciarelli (1996)) that the meme isn't focused on search so much as the conditions of the work that necessitate solutions such as search and that it joins other jokes and stories that shape affective orientations towards search. He writes, the dog meme:

> uses humor to help us deal with the fact that, no matter how skilled we become in our profession as software engineers, we will always encounter problems that extend beyond our area of expertise to understand.
>
> To put it another way: the dog meme is a coping mechanism for professionals in dealing with a domain that will always throw problems at them that push them beyond their local knowledge. It doesn't indicate a lack of professionalism. Instead, it calls attention to the ironies of professionalism in software engineering. Even the best software engineers still get relegated to Googling incomprehensible error messages.

are ritual acts designed to elicit assurance and renew the shared conviction to the norm of reliance on web search. It affirms that this search work—'just' turning to a general-purpose search engine, which seemingly anyone could do—is the work of the field. These confessions thus legitimize search work and the status of the searching worker as a data engineer. They are little openings for the field to reiterate and maybe rearticulate not only what their work is, but also jurisdictional claims.

3.3 Discussion: Opportunities and challenges in confessions

In the place of search talk we find search confessions. The LPP analytic makes search confessions visible, making it possible to see constituent elements of learning, shared and accepted practice. Though recurrently achieved & reproduced, these confessions are informal. The informality with which this community approaches learning to search presents opportunities and poses challenges.

3.3.1 Searching for opportunities

First, the informality and ambiguity of admitting search through search confessions keeps open space for maneuverability—the uses of web search engines are kept open to be adjusted as changing circumstances may dictate. As a general strategy, web search is used in data engineering in order to manage uncertainty. This is discussed at length in the fourth analytical chapter, Owning searching, but the fundamental point is that firms have delegated responsibility to individuals to "keep up" (Kotamraju, 2002) and engage in intensive self-learning (Avnoon, 2021). These confessions are not moves of "rhetorical closure," their confessional frame is *probative*, serving to test or try, not *dispositive*, serving to close or finish. The form of legitimation, a confession, keeps open some "debate and controversy" (*The Social Construction of Technological Systems*, 1993, p. 111) about the use of the tool itself.

Pulling in the language of Handoff, the widespread and ritualized confessions *engage* individual data engineers to perceive that relying on web search is acceptable and encouraged. The perceptions of affordance are not guaranteed. We could consider alternative modes of admitting search into the professional work of data engineers that may provide more closure, but that would also do more to stabilize the search practices and make them potentially brittle as technologies and problems in the firm's context change. Firms could record and rate searches, building in explicit incentive structures or technologies to manage or motivate web searching in the workplace. Imagining how these modes might engage data engineers, with force or potentially leading to exaggerated perceptions of constraint and affordance, highlights the flexibility of the search confessions.

Second, the confessions bring attention to norms and remind data engineers, especially those

full-fledged members, that here searching is appropriate. Even though searching is solitary and secretive it is admitted as appropriate for data engineers in the confessions. Rather than norm-constrained or norm-defying, the norms of search use are reproduced and improvised as the occupational community makes jurisdictional moves (and distinguishes their norms from the faulty reliance on search decried by critics in school settings or elsewhere in society). The confessions acknowledge a felt-deviance of searching in the shadows but assert their searching is different and necessary. The search confessions perform part of what Orr (1996) discussed of stories and narrative. The confessions, sometimes humble-like, can also be hedging practices, a way to protect themselves from being judged for not knowing something on the spot. And confessions can be used, much like Orr's technicians' stories, for pointing to the sheer difficulty and variety of things they must think about and understand, even if the data engineers don't hold all this knowledge immediately in their head.

To see the benefit of the confessions, compare "shadow learning" that is not confessed. In his doctoral research at the MIT Sloan School of Management, Matt Beane studied the learning of and use of surgical robots ((2017)) under the supervision of Wanda Orlikowski (with Katherine Kellogg and John Van Maanen on his committee)[49]. He explored "productive deviance", where "norm- and policy-challenging practices that are tolerated because they produce superior outcomes in the work processes governed by those norms and policies". He was particularly focused on the deviance amidst "significant technical reconfiguration of surgical work".

Beane's dissertation reviewed deviance in organizations, the history of productive deviance in the surgical profession, and then turned to two empirical studies on robotic surgery. The first will be discussed here. The study looks at how few surgical residents became confident and competent in the robotic surgery methods. He focuses on the barriers to such learning and shows how those who did learn did so through norm-challenging practices that he calls "shadow learning."[50]

[49] I make note of these influences because it may helpfully explain his trajectory in his engagement with LPP, not focalized, as I do here, through Lave & Wenger (1991). While he cites to Lave & Wenger (1991), he does not lean on their language or note on their engagements with some of the research he mentions (which I will note below). Takhteyev (2012) remarks on "substantial currency" of the notion of "communities of practice" in the "organizational studies and business literature" (p. 25), citing to Duguid (2008)'s review of community of practice noted it was "rapidly domesticated" (p. 7). (I learned of Yuri Takhteyev's research through a personal conversation with Paul Duguid.)

[50] Beane (2019, p. 91):

> Successful trainees engaged extensively in three practices: "premature specialization" in robotic surgical technique at the expense of generalist training; "abstract rehearsal" before and during their surgical rotations when concrete, empirically faithful rehearsal was prized; and "under-supervised struggle," in which they performed robotic surgical work close to the edge of their capacity with little expert supervision—when norms and policy dictated such supervision.

Beane (2019) presented shadow learning as norm-challenging practices that work around constraints of efficiency and liability pressures. The work of learning to be a web searching data engineer, often done in the shadows, is a distinct type of shadow learning. Efficiency pressures in data engineering have the effect of increasing reliance on the use of the web search tools, in the shadows and seemingly by individuals isolated from others. Liability pressures in the data engineering organizations appear to produce a distinctly different effect than that found by Beane. Beane found hospital concerns about liability keeping trainees from realistic training. Rather than liability pressures removing training opportunities, liability pressures in data engineering encourage various responses that surround and manage the contributions of new trainees.

The liability pressures, the regulatory or contractual constraints and incentives engaging other components within the systems of the data engineering organization, ground the structuring of the data engineering work practices generally, and so also the web search activity. When veteran data engineers write code it is generally reviewed in some manner and tested before production. Code from new entrants goes through the same processes of review and controlled deployment. Systems established for addressing liability in data engineering allow junior data engineers to participate deeply in many aspects of the work. The data engineering work is organized in such a way that many errors made by individuals are caught and repaired in the normal functioning of the system. This includes errors potentially introduced from web searches.

3.3.2 Finding challenges

First, the search confessions are not discriminating. They do not identify the sorts of web searching put to successful use within data engineering. The web search practices of the data engineers are refined or adapted for searches related to the data engineers' core work tasks. There is an operating envelope for such searching, "a range of adaptive behavior" (Woods, 2018, p. 435). Data engineer's use of web search for other sorts of tasks may not work so well, even for work-relevant searches if they fall outside the operating envelope. As the next chapter shows, occupational, professional, and technical components structure the selection of inputs and the evaluation of search results. This structuring is particularly well-directed towards concerns central to the responsibilities of the data engineer and aligned with the core interests of their firm or profession.

Situated in and shaped by larger and longer data engineering work practices, these practices for web search generally recede from view and avoid explicit attention. Web search, as an information infrastructure, is often transparent or invisible (Haider & Sundin, 2019).[51] The

[51]Throughout, but see particularly the section titled "Search as information infrastructure" (pp. 54-55).

invisibility or 'taken-for-grantedness' of search reflected by my research participants is a key feature of infrastructures.[52] The occupational, professional, and technical components of the work of data engineers come together to form an infrastructure for search that generally escapes their notice. The data engineers acknowledged they had "never really thought about it" or that searching was "kinda like breathing." This role of the structuring and liability pressures are themselves taken-for-granted. They are absorbed within the larger infrastructure of data engineering work, becoming transparent. Search confessions do not distinguish those searches that are likely supported or not, thus there is a risk in encouraging uncritical searching outside the firm's "sensing routines" (Carlo et al., 2012, p. 870).

Second, with the appropriateness of search constantly questioned and affirmed only informally, those more on the periphery—marginalized within technology work or newcomers—are left with few signals about the appropriateness of search. These confessions are honest, if not a full accounting of data engineer web searching. But they are also humorous and so by design could be misread by those not fully included. In her ethnography of Debian developers, Coleman (2012) writes that humor "gets us closer to the most palpable tension in the hacker world—that between individualism and collectivism" (p. 92). She discusses humor in-depth in one chapter[53], defining it as "a play with form whose social force lies in its ability to accentuate the performer, and which at times can work to delineate in-group membership" (pp. 103-104).

Misreadings of the search confessions may lead to a misaligned under-reliance on search or over-reliance on search alone (rather than searching facilitated by conversation with others about questions and failures). Those who are already fully participating may actually find it easier to search, and may search more and more effectively because they do have more domain knowledge and are more fully situated within organizations and the field so as to better inform their search queries and evaluation. This suggests that those most peripheral from the

[52]The invisibility or 'taken-for-grantedness' of infrastructure is widely remarked on in infrastructure studies (Bowker et al., 2010). You can, for instance, follow citations from Haider & Sundin (2019) through Star (1999) ("The taken-for-grantedness of artifacts and organizational arrangements is a *sine qua non* of membership in a community of practice" (p. 381)) and Star & Ruhleder (1996) ("Strangers and outsiders encounter infrastructure as a target object to be learned about. New participants acquire a naturalized familiarity with its objects as they become members" (p. 113)) to Bowker & Star (2000) and Lave & Wenger (1991), all discussing the taken-for-grantedness of infrastructures and particularly how infrastructures are visible to newcomers to a practice or community, but shift out of view as they become full members. (Whether web search is generally visible to newcomers is questioned, though, in Sundin (2020), who finds "search engines almost never seem to have been a visible information infrastructure for the current generation of teenagers" (p. 378).) The references to Lave & Wenger (1991) are to the book as a whole, but a direct discussion can be found on pp. 101-102. Bowker & Star (2000) cite to Cambrosio & Keating (1995), who discuss both infrastructures taken for granted and how "[w]idely distributed know-how," or tacit knowledge, can be taken for granted.

[53]Ch. 3: The Craft and Craftiness of Hacking

data engineer practices are those most in need of developing search skills from alternative practices (and trying to re-apply them in the data engineering context) or reflexive practices in their data engineering work. Newcomers and those marginalized within technology work are those most likely to treat web search as a "topic, or difficulty" and not an infrastructure (Star (1999), p. 380). This may slow or reduce learning opportunities for those already on the periphery and, at the extreme, exclude them.

I'll expand on these challenges from "searching in the shadows" by looking at the "consequences" Beane identified as resulting from "shadow learning" in robotic surgery. Beane (2019) noted that barriers to traditional modes of LPP had "problematic implications" (p. 102). "The routine enactment of shadow learning [. . .] led to [. . .] outcomes that were quite problematic for shadow learners, their cohort, and their profession: hyperspecialization, fewer learning opportunities for less-skilled residents, and limited learning." (p. 111).

The implications, or consequences, that Beane identified were (1) a reliance on robotic surgery in cases without clear benefit, (2) a "Matthew effect" where only the most skilled received more opportunities to practice (reducing the supply of qualified surgeons), and (3) that the silence of the shadow learning stopgaps kept attention away from how little was learned in the few opportunities for participation. On that third implication, Beane wrote (p. 113):

> a lack of broader, more open discourse on the failures of [providing] legitimate peripheral participation and the effectiveness of shadow learning for robotic surgical technique essentially prevented the profession from learning

These implications are related to the challenges from "searching in the shadows": the potential reliance on searching out of scope and questions about whose learning is best supported. Beane found that trainees that engaged in shadow learning became "hyperspecialized", and "faced strong pressures to perform robotic surgery on their patients, even when it was unclear whether robotic surgery was the best course of treatment" (p. 112). The contextual factors driving over-reliance on robotic surgery where not appropriate do not mirror the factors surrounding potential over-reliance on web search in data engineering. My argument above was that the risk of over-reliance on web search results results from its informal legitimation making the operating envelope, the ability to search with support from the broader data engineering work practices serving as search infrastructure, even harder to see. This can contribute to a sort of hyperspecialization in two senses: underdevelopment of other sensemaking or discovery techniques and web search skills practiced principally within the operating envelope of data engineering-supported searching (rather than *general-purpose*). The other two implications are more directly related to the second challenge. The silence around the searching activity and the informality of the search confessions contribute to a Matthew Effect for data engineers searching, those who are most adept at searching like a data engineer are more supported in searching more, with little attention giving to improving

opportunities for participation for others. These challenges, or concerns, are raised in the following chapters.

3.4 Conclusion: Condoning or celebrating searching?

Confessions, filling in for the absence of search talk, present opportunities and challenges. The discursive and humorous mode of the search confessions, as a way of engaging with other actors, constructs searching as a flexible tool. The confessions assuage the felt-deviance. But the search confessions do not help the data engineers identify if a search is within the operating envelope of their work. The search confessions do not describe the limits of such searching, to be discussed more in the next chapter. And due to its informality, the legitimating purpose of search confessions may not be clear enough to newcomers and those kept on the outside to encourage successful searching. Even for practiced data engineers, it is not clear if the search confessions are merely condoning or fully celebrating searching.

This chapter examined part of how data engineers learn to search as data engineers. Legitimate peripheral participation in data engineering web search work occurs in (1) search confessions, the focus of this chapter, and (2) the structuring of search practices through occupational, professional, and technical forces, the focus of the next two chapters. Search confessions acknowledge and legitimate searching the web for work. Engaging with others or taking on the role of confessor oneself, is a form of participation in web searching for the data engineer. The confessions do not generally include the search inputs themselves or the search results, but are part of the social practice and social construction of web searching. This chapter describes the work of affirming that it is common, acceptable, and necessary for data engineers to rely heavily on web search. The search confessions do not only normalize, they reproduce the data engineer web search practices. The legitimacy granted the data engineers, partially through search confessions, give them access to the resources structuring search, discussed in the next chapter. Their peripherality is shaped by the material design of search and the larger culture's identification of search as a solitary and intimate performance (see the Repairing searching and Owning searching chapters for a larger exploration of this).

4 Extending searching

The cover of Daniel M. Russell's, *The Joy of Search: A Google Insider's Guide to Going Beyond the Basics*, book describes him as the "Senior Research Scientist for Search Quality and User Happiness at Google"(Russell, 2019). The marketing materials from MIT Press say that "readers will discover *essential tools* for effective online searches [emphasis added]" (The MIT Press, 2020). In the book he writes:

> Most of the searches that Google sees in a typical day are fairly straightforward. The goal is clear, and the results are pretty obvious and unambiguous.
>
> But a significant number of searches are not. Searchers might have a goal in mind, but *they can't figure out how to express it* in a way that will give them what they want. Sometimes their query is precise, but *they don't know how to read and interpret the results*. It drives me to distraction as a researcher because *I know that the searcher is missing just one small but critical piece of information*. We try to build as much as we can into the search algorithm, but people still need to know a bit about *what the web is, and how search engines crawl, index, and respond to their queries*. [emphases added]

Data engineers are not explicitly taught much about doing web search as data engineers. Maybe extra lessons are unnecessary. Danny Sullivan, Google's Search Liaison[54] has said, quoted in a Google blog post (Kutz, 2022):

> Today, it feels like people are born knowing how to search. You just type what you want into a magic box, and poof! It delivers results — no classes needed.[55]

Are the search confessions and experience searching in school and everyday life enough for them to use search successfully as data engineers? Does engaging in search confessions provide enough *participation* to support the learning of these practices? Or do they have the *essential tools* or a *critical piece of information* that helps them? How is their technical knowledge and knowledge of their craft applied to make search work? Where is that knowledge?

I interviewed people with hard-won technical knowledge about the design of data pipelines, the characteristics of databases, deterministic and probabilistic algorithms, and insight into the cost and performance tradeoffs in production systems. My interviewees possess deep insight into how information is moved around and processed by computers and who might be said to "highly value information handling" (Teevan et al. (2004)). And they use web search extensively. They rely on web search and it seems to perform well for the purpose to which they have put it. I thought I would talk with them and look with them at how they

[54]The functions of the Google Search Liaison is examined in Griffin & Lurie (2022).

[55]This quote is also in speculating on how junior engineers learned to search

searched to see what lessons we might learn. Particularly, I wanted to look at the role of their knowledge of the mechanisms[56] of web search in their search practices.

But the data engineers didn't talk about their knowledge of search technology. Moreover, in response to my questions and prompting they did not suggest that technical understanding of search itself played a key role in their searching successes. The practices described by my interviewees did not present as, or appear to be, grounded in or driven by individual knowledge of search mechanisms. There were generally no references to crawling, indexing, ranking, advertising, nor parsing of queries. Search experiences were addressed with the search engine black boxed and the results reified as the results.

Instead, the data engineers talked about their web searches heavily structured by practices and artifacts. They described various work practices and artifacts (occupational, professional, and technical components of their work) that I argue produce and maintain scaffolding for web searching. I found that the knowledge brought to bear to make their searching work is embedded in their work practices and their work practices are extensions of web search. Here I will use the analytical frames from Handoff and LPP to show web search as extended beyond the moment of searching and into the broader work practices and artifacts. I will show how the knowledge of successful searching is embedded in those practices and artifacts. I argue those practices and artifacts provide the participation necessary for the situated learning of web search.

I next discuss the sort of knowledge that might be anticipated, before showing what knowledge is in the work practices and artifacts of the data engineers.

4.1 Knowledge to make search work

> How will you look for it, Socrates, when you do not know at all what it is? How will you aim to search for something you do not know at all? If you should meet with it, how will you know that this is the thing that you did not know?[57]

Calls for search platform transparency or explainability and digital or search literacy are rarely accompanied by cases demonstrating the benefits for the searcher. The assumption that knowledge is the problem is accepted and left untested.[58]

[56] The next section will address this more fully, but the language of 'mechanisms' is drawn from Introna & Nissenbaum (2000) & Tripodi (2018). By the end of this chapter I will discuss 'mechanisms' expansively, but at the start I'll use it in a largely technical sense, as a placeholder for the subject of various transparency and literacy concerns.

[57] Plato (2002)

[58] Compare arguments about the governance benefit from transparency or observability (Ananny & Crawford, 2018; Rieder & Hofmann, 2020), and the limits of transparency (Ananny & Crawford, 2018; Burrell, 2016).

It would be beneficial to examine closely a site, such as that of data engineers, where people search extensively, seemingly effectively, and with technical knowledge of mechanisms similar to those in web search.

It is important to look at this question because of its place in research and policy debates about the use of new technologies in society and the appropriate role for regulation or design in facilitating effective use of knowledge of those new technologies. Better understanding the role of knowledge of the mechanisms of web search may feed back into we shape our practices and technologies to better achieve our goals.

To set the stage for the material in this chapter I will first discuss the mechanisms of web search and the role that some researchers have given knowledge of the mechanisms. Then I will provide an overview of work on the use of web search by coders.

4.1.1 Mechanisms of web search

So what about the mechanisms of web search must be understood for ordinary successful use? Introna & Nissenbaum (2000) describe three kinds of mechanisms of web search: market, regulatory, and technical mechanisms. Introna & Nissenbaum (2000)'s core focus is on the "market mechanism", yet they broadly write of the "systematic mechanisms that drive search engines" and argue it would be a "bad idea" to "leave the shaping of search mechanisms to the marketplace" (p. 177). They mention regulatory mechanism in a quote from McChesney (1997) on "the notion that the market is the only 'democratic' regulatory mechanism". They also use "mechanisms" in quoting from Raboy (1998). Raboy's quote presents the internet as new media and a role for public policy to promote a model that with "new mechanisms" might be "aimed at maximizing equitable access to services and the means of communication". The technical mechanisms in Introna & Nissenbaum's "brief and selective technical overview" are the "technical mechanisms" of crawling, indexing, ranking, and "human-mediated trading of prominence for a fee" (p. 181).

In presenting these mechanisms, Introna & Nissenbaum (2000) were not explicitly discussing knowledge required for practical success in searching. Rather, their conclusions focused on policy and values in design directed towards addressing "the evident tendency of many of the leading search engines to give prominence to popular, wealthy, and powerful sites at the expense of others" and how that "undermines [. . .] the substantive vision of the Web as an inclusive democratic space" (p. 181). They did seem to indicate, though, a beneficial role for individual awareness of search mechanisms. They noted unfamiliarity and lack of awareness of "the systematic mechanisms that drive search engines", and wrote "[s]uch awareness, we believe, would make a difference" (p. 177).[59] Though they noted it was only a partial

[59]Here is the containing text for the quoted material, from Introna & Nissenbaum (2000, p. 177):

response[60], they demanded transparency (p. 181):

> full and truthful disclosure of the underlying rules (or algorithms) governing indexing, searching, and prioritizing, *stated in a way that is meaningful to the majority of Web users*. [. . .] We believe, on the whole, that informing users will be better than the status quo. [. . .] Disclosure is a step in the right direction because it would lead to a clearer grasp of what is at stake in selecting among the various search engines, which in turn should *help seekers* to make informed decisions about which search engines to use and trust. [emphasis added]

Researchers from web search studies, critical algorithm studies (Mart, 2017), new media studies (Dijck, 2010), epistemology (Miller & Record, 2013), and misinformation studies (Tripodi, 2018) have suggested that searchers' knowledge of the mechanisms of web search may help searchers better achieve their search goals. Distinct from that claim, some of those researchers additionally argue, alongside others, that such knowledge can first be put to use by researchers, regulators, journalists, and librarians and other educators, who may then act in their expert capacities to shape and advance successful web searching. For instance, Pasquale (2015) argues for "qualified transparency", disclosure and audits by the FTC or another agency (p. 160-164).

Some of these studies also point to information literacy as an ideal and an intervention in pursuit not of individual search goals evaluated by accuracy or relevance but of societal goals that embrace values such as neutrality or privacy (Dijck, 2010). Noble (2018)'s work, by contrast, is focused on algorithmic literacy for building alternative search engines (p. 25-26) and argues that reconceptualized (p. 133) or reimagined (p. 180) search engines might advance transparency at their core.[61]

> Given the vastness of the Web, the close guarding of algorithms, and the abstruseness of the technology to most users, it should come as no surprise that seekers are unfamiliar, even unaware, of the systematic mechanisms that drive search engines. Such awareness, we believe, would make a difference.

[60] Additional policies (2000) suggested considering were "public support for developing more egalitarian and inclusive search mechanisms and for research into search and meta-search technologies that would increase transparency and access", noting that the market, even with disclosure requirements, was not sufficient on its own. They also called for search technology design and research that was directed by "an explicit commitment to values" [p. 182].

[61] Noble does not present transparency as a solution for the search problems she details (except insofar as algorithmic literacy informs the development of alternative search engines), saying instead (p. 179):

> What is needed is a decoupling of advertising and commercial interests from the ability to access high-quality information on the Internet[.]

Some research appears to bundle individual search success with other goals or responsibilities. Sundin (2020), for instance, calls for "critical awareness" that supports the user of search to be "an informed and critical citizen," (pp. 365-376):

> How do you learn to use an information infrastructure such as search engines? Just as with electricity, which you can use by just switching on the lamp, many times you just have to type a word to get a result that is at least *good enough*. At the same time, the workings of search engines for a large number of sectors have dramatic consequences in society, such as for business, tourism, politics and schools. Again, you do not have to know very much in order to find out *something* about what you are looking for. But just as with electricity, if you want to be an informed and critical citizen this is not enough. [emphases added]

Bhatt & MacKenzie (2019) looked at digital literacy, including search engine use, with "a social practice approach to literacy" (p. 304). They start from the position that literacy is "always associated with, and realised through, 'social practices' rather than a purely formally-schooled understanding of correct language" (p. 303) and that literacy is "always embedded within social activities, is socially situated, and mediated by material artefacts and networks" (p. 303). Their claim regarding the consequences arising from the lack of knowledge of mechanisms also bundles multiple effects (p. 305):

> Without knowing just how such platforms work, how to make sense of complex algorithms, or that data discrimination is a real social problem, students may not be the autonomous and agential learners and pursuers of knowledge they believe themselves to be."

Two aspects of the mechanisms of web search that researchers have identified as important are the data collected and the inferences made by the search engine about the purpose of the search and the searchers. Warshaw et al. (2016), in work performed at Google, find "a substantial gap between what people believe companies are doing with their data, and the current reality of pervasive, automatic algorithms." Successful use of web search, drawn most broadly, may require an awareness of the data collected and the inferences made about the searchers.

Some researchers attribute individuals' acceptance of and persistent belief in propaganda to searchers' knowledge deficit. This concern is implicit in Tripodi's writings. Tripodi (2018) describes a pattern in her interviewees that indicated "users do not have a consistent or accurate understanding of the mechanisms by which the [search engine] returns search results." (p. 28) Tripodi (2018) referred particularly to the role of the keywords in the search query (including how others act to spread and write content to match those keywords) and the ranking of results (and the interpretations of 'top content' by searchers). While the focus of

Tripodi's book building on her prior research (2022b) is on how propagandists "wield the power of search" (p. 18) to pursue their ends, her description of her respondents' knowledge implies a key role for knowledge deficits in the process: "few seems to understand how much keywords drive returns" (p. 103), "believed that top returns were more legitimate" (p. 109), and not realizing "returns are rooted in the search engine's monetary interests" (p. 111).[62]

I originally read Tripodi (2018) within the context of a surge of research (from, in part, critical algorithm studies (Seaver, 2019)) about political demands for transparency, explainability, and contestability in algorithms (Ananny & Crawford, 2018; Burrell, 2016). There was also considerable alarm about, and interest in, "fake news" or propaganda (Jack, 2017)—with "a resurgence of mis/disinfo studies" (Caplan & Bauer, 2022)—alongside arguments about the absence or failure (boyd, 2018) of media/digital/search literacy. Alongside Tripodi's work there were others looking at the role of search engines in misinformation (Metaxa-Kakavouli & Torres-Echeverry, 2017) and problematic search results (Golebiewski & boyd, 2018). It was an open question whether platform transparency about the mechanisms of search might be needed to help users search more effectively.

Google's myth making complicates attempts to understand the mechanisms (Gillespie, 2014). Like many companies (Burrell, 2016), Google keeps the design and performance of its algorithms opaque or black-boxed (Noble, 2018). As Bilić (2016) argues, "Google employs powerful ideological engineering of neutrality and objectivity in order to keep the full context of its search engine hidden from everyday users."

In our paper on Google's Holocaust problem (2018), Deirdre Mulligan and I repeat the same general theme in our analysis of a disconnect between how Google engineers and managers imagine and see their tool and how it is perceived and used by others. Our contention, though, was not that search users should adopt the perceptions of the search producer or that knowledge was the answer, rather that the Google engineers and managers should recognize and remediate the consequences from those conceptions. The conceptions of the regular users were produced in part, we argued, by Google's own myth making.

Researchers have pointed out that not all search users can identify the distinction between paid for results and so-called organic search results (Commission, 2013; Daly & Scardamaglia, 2017; Ofcom, 2022). Some research suggests some people are unaware of the commercial nature of Google Search. Safiya Noble has shared about talking with a librarian who had been under the impression that Google was a nonprofit organization(House of Lords, Select

[62]While Tripodi weaves those observations into her broader analysis, her contribution is centered on the activities of propagandists and how they leverage cultural frames to manipulate searchers and media. The final sentences of her book are "By exposing the schemes behind the propaganda, my hope is that coders and information seekers alike will see the light. For we all need to advocate for greater transparency and verification in the sources we use to learn about our culture, our political candidates, and our world." (p. 215)

Committee on Democracy and Digital Technologies, 2020).

The opacity or obfuscation from the company only means that the users may only know a part of the algorithm, though that may be the more important part for some uses. Users develop practical knowledge of algorithms, "knowing how to accomplish X, Y, or Z within algorithmically mediated spaces" (Cotter, 2022), and may understand algorithms in ways inaccessible to the designers of the systems (Cotter, 2021) as "[n]ot even people on the "inside" know everything that is going on" (Seaver, 2019).

4.1.2 use of web search by coders

Early in my exploration of search and then coders use of web search, even before I narrowed to data engineers, it became clear to me that some of the concerns around getting 'bad information' from web searches may be obviated by the nature of their work. Prior work on the use of web search by coders shaped my attention to these concerns.

Unlike searching in many domains, the coders would often, as a part of their work practice and an affordance of their work, quickly validate what they found in the search results and see for themselves. This validation step, in a limited way, provides some friction that may help coders be reliant on web search while not overly reliant on the search engine rankings.[63] They could make a change to their code and then test it by trying to run their code to see if it 'worked'. This wasn't some special skeptical bent, but part of their work to get things working. Searches were only complete if they found some workable results.

This (sometimes and relative) capacity for and practice of testing the results (in the standard use of the search tool) is described in the literature.

Brandt et al. (2009) describes some of this where participants in their lab study—tasked with building an online chat room— would search for a tutorial and, finding one, "would often immediately begin experimenting with its code samples".

> One participant explained, "there's some stuff in [this code] that I don't really know what it's doing, but I'll just try it and see what happens." He copied four lines into his project, immediately removed two of the four, changed variable names and values, and tested. (p. 1592)

But they also described such testing as not always happening immediately. Saying "Participants typically trusted code found on the Web" and that errors or misapplication of the

[63] Such over-reliance on the ranking of results by search engines may be analyzed as a form of "automation bias": "the use of automation as a heuristic replacement for vigilant information seeking and processing" (Skitka et al., 2000, p. 86). This is addressed in a few sections in Decoupling performance from search.

code found on the web was then not noticed immediately, which complicated the coders' remediation efforts. (p. 1593)[64]

In other domains the testing or trying of search results is done at first by the searcher in interaction with the various results and their presentation. If one wasn't sure or wanted to know more the quickest path is a further web search (or reading and browsing further on in the search results pages or on the SERP—there are tools built into major search engines, like Google's three-dots, that might provide some transparency about a page).

But workable in the moment isn't enough. There is other information that isn't immediately testable by running it through a compiler or interpreter and seeing it is "works". This is particularly the case for *non-functional properties* (normally including things like security, reliability, and scalability but could also include considerations of harm both in the implementation and use of the code (Widder et al., 2022, p. 8)). There is research looking at insecure or `unsafe` code on Stack Overflow. Fischer et al. (2017) showed that of the 1.3 million Android applications on Google Play 15% "contain vulnerable code snippets that were very likely copied from Stack Overflow" (p. 135). Subsequent work found that, on Stack Overflow questions related to Java security, "On average, insecure answers received more votes, comments, favorites, and views than secure answers" (Chen et al., 2019, p. 545). Firouzi et al. (2020) examined the use of possible unsafe `unsafe` keyword in C# code snippets on Stack Overflow.

4.2 Seeds for queries and spaces for evaluation

There are many mechanisms of web search at various layers and levels of interaction with the work of data engineers. Here I focus on the role of knowledge of such mechanisms as the mechanisms relate to two aspects of data engineering web search activity: how they choose search queries and how they evaluate search results.

I asked Shawn, a Developer Advocate in open source data software and former data engineer, to talk about the thinking that went into writing error codes. He talked about gradient descent and the error message's role.

> You can definitely search yourself into a—gradient descent, you've heard of that algorithm, in machine learning?— You can always find yourself at a very local minimum or maximum and actually you're not even close to the solution. It is very much a gradient descent type of problem. 'Oh, this looks like the right

[64] Contra the Fischer et al. (2017) findings, in their lab study Brandt et al. (2009) wrote: "Participants typically trusted code found on the Web, and indeed, it was typically correct." The *correctness* of the code, though, was limited to it being workable—it would run. Brandt et al. (2009) made no mention of testing the code for security, reliability, scalability of other factors of code quality.

> solution.' This also comes with experience too, right? Having that context of knowing that this is not the correct solution even by just a weird gut feeling. So it is interesting how Google's algorithms can only take an input and that input is very much shaped by the experience of the engineer on the other side. And the way that they take in the outputs of the result of that search will also take them in one direction or the other. To some extent being able to have the right context, the right understanding it, always comes down to context.

The error message provides some context for the unknown engineer who may find an exception when running your code. This context shapes the situation, partially situates, even, the data engineer as they work to search for solutions to their problem. The error message is an example of a search seed and a space for evaluation, the next two sections.

4.2.1 Query formulation

Jillian was a new data engineer on a small data engineering team at a fitness app. Talking about search struggles, she shared the following:

> Every so often I have such a hard time phrasing what I'm trying to look for. I continually am searching something new and I just aimlessly click through all of the search results. Whether or not I'm even being intentional about reading what the link even says. I'll click on it, look through and it doesn't have the answer. It's always in this, not panicked state, but frustrated state. Where I'm like I have such a simple question.
>
> And I also feel that its the sort of question where If I could ask someone a question and phrase it, then it could probably be addressed in 15 seconds. But instead, there have probably been times where I have spent two hours trying to look through, because I don't know how to phrase what I'm trying to do.
>
> when I can't find my answer then I'm just like throwing darts. Putting random things in the search line trying to see if I can find what I'm trying to do. [. . .] the darts rarely stick. These are the scenarios where I'm really kinda desperate at this point. I'll sometimes even look at what I've searched and I'll be like 'like why on earth would that give me the response I'm looking for, I didn't even include the coding language. What have you done!?' You're brain is just like shutting off at this point and you're in this habit of like click, scan, click, scan... And you just put in random keywords.... that aren't even the keywords that you necessarily need to get your answer.

Upon realizing she's fallen down this hole, Jillian said that she will tell herself "Let's just pause and think about it." And write out by hand what she is looking fork, using other

knowledge of her craft to remember or reconstruct different search queries.

> Say for example I was trying to understand some transformation I might have to write down a table and kind of look at it. Am I trying to join? Am I trying to merge? Am I trying to union these things? What is really happening with my data? Maybe I have to understand the 'actions at play' so that I can then translate those kind-of-like visual transformations into words that I can Google search, since you can't just Google search a concept floating [chuckling]

Generally, though, the data engineers are not throwing darts or attempting to search floating concepts. Their search queries are often given to them, or seeded, through their work practices.

I will discuss two aspects of query formulation arising from production and socialization. Data engineers find queries as they are *prompted by the code* and *immersed in linked conversations*. This resembles the literal searching described by Tripodi (2018), where she shows people typing exact phrases they'd seen in the news or overheard and demonstrates that "*the phrase someone Googles dramatically affects the information they receive*." [italics in original].

4.2.1.1 Prompted by the code and "trail indicators"

Noah, the data engineer at a media streaming company:

> I copy the part of the error message that seems most relevant. There will be a whole stack trace and a bunch of stuff and usually there is one line that its like, 'here is what went wrong'. So I'll copy the generalized portion of that. So it might say error in file x.txt and then some error message. So you copy the generalized portion... and just paste that into Google and see what comes up. Often the first result will be a Stack Overflow question or occasionally a GitHub issue.
>
> And you can go in and see, alright. Was this person trying to do something roughly similar to what I was doing. Which is a little bit error message dependent. Some error messages apply to a million different situations so you have to further find the one that is more similar to your situation. Other error messages only come up when you're trying to do that specific thing. But, yeah, that's the process. Copy-paste-search.

After I prompted Noah with Teevan et al. (2004)'s distinction between "orienteering" and "teleporting" searches ("directed situated navigation" versus "jump[ing] directly to their information target using keyword search"), he said

> With the error messages, I think a lot of times there is a hope that it is a teleportation search and a fear that it's not. I can search this error message and

> if I'm lucky the first result is going to tell me exactly what I need to know. And if not I might have some orienteering to do. Where it is narrowing in on the problem, narrowing in on what the cause is, figuring out of five people getting the same error message which one is doing something resembling what I'm doing.

While the error messages may be generic, the stack trace or some aspect of the message itself or the top search result might still guide the searcher.

Raha, a senior data engineer at a media entertainment company, in response to seeing the "Googling the Error Message" book cover (a parody of the O'Reilly books), observed that "sometimes they give you a hint that this is a generic error."

Shawn, a former data engineer who is now developer advocate at an enterprise software company that develops tooling for data engineers, referred to the stack trace as usually providing 'trail indicators' even if final error message wasn't itself particularly useful:

> The typical keywords that really resonate in searches are error messages. In particular, it you get some sort of particular string. There's types of errors that come out. There will be an error message that is a—
>
> —we'll go with the most, the least descriptive possible exception that you could think of in Java, in Java land, which is the NullPointerException in Java.
>
> If I just put [NullPointerException Java] I'm going to basically get searches that take me to how stupid NullPointerException is because it pretty much the thing you run into when you don't know what is the problem. It is basically when basically you've misprogrammed something. It is so undescriptive in the language of Java. It is more of a Java problem, not necessarily a problem that you're dealing with directly at the cause of your specific issue. You've misprogrammed something and you forgot to account for a null and unfortunately the Java language doesn't account for nulls super well. So like, you've come to realize that is not a good thing. You can't just use this NullPointerException as a way to find the answer to your question. Right? So what you have to do at your next step is, I need to find some other sentence or some other piece on this stack that might...
>
> So usually you'll get a stack of different things that have called other things, right. All the way from the root process of whatever started this call on a particular thread. You'll be able to track and see what called what up until that point. And so usually along the way there are trail indicators of, 'OK, this is where I was at in this code and this called this piece of this code'. And you can actually look at the source code you know and start trying to understand that. Maybe you don't even need that. Sometimes when it comes to just, you know,"cheating",

> you could just basically find the right amount of keywords or the right amount of word exceptions, exceptions with sentences within a word that when you put it in there, 9 times out of 10 you're actually going to get directed directly to the solution that you are looking for.
>
> So that's typically I would say–and that is not data engineering specific at this point—that's generic to any developer that has to work with any kind of framework.
>
> Those frameworks will have certain error messages that are phrased in just the right way that have just the right exception and that is enough.

Shreyan, a data engineer at an enterprise software company, shared this as his initial response to my asking if he could talk a little bit about he uses search:

> Whenever you have say an error. You copy the error. You either try to copy pieces of it or the complete error. ... copy it and paste it into the web

Patrick, likewise a data engineer at an enterprise software company, said:

> if I get an error message I've never seen before
>
> just copy and paste it straight in

Arjun, a principal data engineer at an enterprise software company said:

> So one technique I always say is copy the first line of the error, of the stack track, and post that in the google. You'll get a much better understanding then. So that's an easy way for you to debug.

Nisha, the director of Director Of Data Services at an enterprise software company also shared about searching an error:

> Sometimes you are stuck with small things that you need to understand... What's going wrong? Why is something not working? And then you want to quickly look up an error that you're seeing on Google and you can get more ideas as to what the possible issues could be.

Charles, a data scientist on a data engineering team at large online marketplace:

> I think the most common thing, which I know a lot of people do, is I would just copy and paste the error message straight into Google

Finally, also Phillip, another data engineer working in enterprise software:

> when we encounter certain errors, especially as the tools that we use, a lot of errors that can come up kinda just copy and paste into Google and kinda justhopewe find some results there

It isn't only errors that data engineers search. Noah talked about searching the web to find a subsequent term to then search in internal code searches.

> I think it's kinda a spectrum of how much do you know what you want to do. I was building a pipeline in a technology I wasn't that familiar with and I needed to find out how to write tests for it. I was using Beam. So the pattern was: I searched Google for [testing in beam]. And found, OK you've gotta this class called TestPipeline. So I get a little bit out of the public documentation. But then I wanted to see how people in my company were using it in practice and crib off them. So I used internal code search and searched that class name. Maybe I got fifty results. So I open up the first ten in new tabs and I just looked through their code, and go, 'ah does this seem similar to what I want to do.' Then try to adapt it to. Sorta a hybrid internal-external search. I didn't really know what I wanted to do. I used Google to figure out what I wanted to do. Then used internal search to find out how other people might have done the same thing.

Rather than searching phrases "strategically signaled" in talking points from politicians or found in propaganda (Tripodi (2018)), the data engineers are searching phrases drawn directly from their work material. While they do reflect on their query choices, they do not have to bring considerable knowledge of the mechanisms of the search engines to bear. The code socializes the potential search terms for querying and so scaffolds the search process.

4.2.1.2 Immersed in linked conversation Ajit, the staff data engineer at a major retailer, shared about an informal weekly meeting where he and his team talk about their work:

> In our team, we do weekly catchups. Like stand up meetings. Those are basically around: What are you doing on a day-to-day basis? So mostly people talk about the work that they are doing
>
> If they have any questions on that, if they just want to explain, 'hey, this is the approach that I am taking.'... And then it is just an open forum. So we just go around the table. A person just tells about what they are trying to do. Anyone on the team who has sorta worked in it before. Or, anyone who sorta has even no idea about it, can sorta ask questions. 'This might be able to help you.' Or: 'Can you explain how this thing is going to achieve what we are trying to do here as part of this project?'

The data engineers talked about being immersed in the languages, tools, and problems of their work—engaged in varied conversations with colleagues. Hiring or team-onboarding processes provide a base-level for internal communication, a minimum shared vocabulary: "It is useful to kind of level-set on minimums so that we can all talk, with the same at least vocabulary" (Aditya) "You start with having a conversation with the actual product team." (Amar) "So, if you find, if you find good resources you share it with the team" (Amar) "Like a lot of times we'll just share random blog posts or articles for things that we think are interesting." (Victor)

Practices designed to coordinate the engineers also provide introductions or additional exposure to the language of their work. Here is Phillip:

> In a typical demo session, usually demos are before the work is complete. It is pretty much to do some knowledge sharing. Each person on my team is working on something well. Pretty much I would not know anything about what someone on my team is working on... So these demo sessions are pretty much just to share what the team is up to as a whole.

These conversations, distinct from direct question-and-answer interactions, are linked to their work and to the larger occupational community. Various participants in these conversations previously worked elsewhere and also learned some of what they converse about through searching. There are conversing in a shared language that has been made searchable. Regularly referred to in these conversations are the topics of their work: the code, the business logics, and the work processes and tooling.

These conversations take different forms. Sometimes it is over lunch or informal meeting, and other times it is through Slack, email, of through the more asynchronous sort left in the code or internal wikis or other knowledge management systems.

The data engineers often have the exposure to their peers using language for potential use as search queries. While they can look elsewhere for such material, they are not solely responsible, or alone, in identifying terms for their searches. They are regularly, presented with potentially queriable terms through the work talk in their occupational field.

4.2.1.3 Search seeds

I use "search seeds" to refer to the keywords or terms presented to the searcher that the searcher in turn uses in a search query.[65] I want to be able to refer to the suggestions or opportunities to search. These are the strings of text someone comes

[65] "Seed" in some web search research is used to refer to the terms used by the researchers as they collect the suggested or alternative queries provided by the search engine (Mustafaraj et al., 2020). Seed in that case is used more akin to seeding the algorithm perhaps as one provides a seed for a random number generator.

to think to send to the web search engine[66] as a search term. Or, these are components in the larger environment (spoken words or printed strings of text) that a potential searcher, situated with appropriate resources, may perceive to afford a successful search. This text may be in a code comment, function or method name, the error message, overheard in workplace conversation, or found on the web.

I'm using 'seeds' to draw attention to stages of web search activity that occur before a query is entered into the search box, the germ of an idea to search. While there are always stages before the activity in the search box, sources of a query are often indefinable, much more diffuse and distant than a single seed. A search seed is not an inkling in the mind of the searcher, but a material sign—a spoken or written phrase—of a possible search[67]. But much like affordances are situated and relational (Leonardi, 2011; Vertesi, 2019), so too are search seeds. In that, a search seed may suggest a query to the prepared searcher. A search seed is not the same as the fully grown formulated search query. But the seed may suggests a full query, or some of the terms. Search seeds are not guarantees of search success for the searcher, success will depend on other components of the search system being configured in a way such that the content for the seed is made, the content is indexed, the query is parsed, and the connection is found by the searcher high enough in the search rankings. Seeds are inscriptions that can be mobilized for search (Latour, 1986).

Search seeds may or may not be explicitly referred to as things to be searched by the people sharing them or artifacts conveying them. In a piece in Wired (Tripodi, 2019a), building on her 2018 Data & Society research, Tripodi uses the phrase "strategic keyword signaling" to refer to the practice of distributing soundbites or catchy phrases for audiences to search. I first used 'seed' to refer to this activity broadly while describing that article (2019). I propose search seed, and the dissemination of search seeds, to encompass such strategic signalling.[68]

[66]These strings of text are often sent to the web search engine through a search bar, but they can also be sent directly, on many web search engines, through the URL. Some of my interviewees did mention using links to directly go to the search results page for a particular query.

[67]While search engines are designed around digitized text, there are other modalities available for search, all materially bound. Voice-based search transformed audio into text. Some people may recognize digital images as search seeds, with reverse image search. Some search engines and other search tools also support searching from a photograph. There is also some support for searching with music or even humming. Chen et al. (2022) demonstrate search queries from electroencephalogram (EEG) signals. As web search expands in these directions it may be necessary pursue new approaches to showing which pictures, sounds, smells, or thoughts might effectively link questions and answers.

[68]It has also been used by Sam Wineburg, an education professor at Stanford, in a 2021 tweet referencing Tripodi's 2019 Wired article: "Typing "the claim into a search engine" will often lead you to exactly where the rouges want you to go. Bad actors "seed" the words they want you to search for & then populate the Web with content supporting their view" (Wineburg, 2021). Tripodi, rather, has used "seed" in a tweet to refer to the seeding of content: "Propagandists seed the internet with problematic content and manipulate Search Engine Optimization to ensure their content dominates top returns" (Tripodi, 2022a) In her book she

There is other work that touches on search seeds. Mike Caulfield, a misinformation researcher at the University of Washington's Center for an Informed Public, has referred to search suggestions or directives from others that drive searchers to conspiratorial content as the "Google This Ploy" (2019a). Ronald Robertson, a search researcher at the Stanford Internet Observatory, has referred to search suggestions on social media as "search directives" in unpublished work. Seeds may be identified in the search results themselves (as queries are reformulated with reference to found content), the SERP itself (with the "People also ask" and "Related searches" rich-content features on Google or the titles and snippets for results), or in the autocomplete search suggestions when typing a query. The seeds from the autocomplete, like the problematic suggestions identified by Cadwalladr in 2016 (2016a, 2016b).[69] Seeds are also shared in conversations about "do[ing] more research", such as in online groups discussing the vitamin k shot given to newborns, providing [vitamin k shot] as a seed as documented by Renee DiResta, now a misinformation researcher at the Stanford Internet Observatory (2018). President Biden provided an explicit search seed in his tweet telling people to "Google "COVID test near me" to find the nearest site where you can get a test" (2022). These are all examples of search seeds. (Note that those all discuss where search seeds are present, not where they are absent.)

The search terms used matter. Tripodi (2018, 2019a, 2019b, 2022b), along with Gillespie (2017), Golebiewski and boyd (2018, 2019), Caulfield (2019a), and others, have identified the importance of the interactions before the selection of the search query. Tripodi discusses concerns about how the people providing the search seeds and search engines act and what role greater searcher knowledge of the mechanism may have. Gillespie (2017) presents a case study of strategic competition over the results returned by Google for the term "santorum" and the mediating role that the search engine plays in such contests. After the 2003 campaign to redefine the term, to critique homophobic remarks from former politician Rick Santorum, the search results returned for the query [santorum] on the major US search engines are still

sources "seeding" to digital marketing, writing (Tripodi, 2022b, pp. 127–128):

> prominent personalities within the right-wing information ecosystem understand the media technology du jour, and use that medium to cross-promote their ideas and serve as guests on one another's shows. Conservative thought leaders also signal-boost specific keywords and phrases in their ideological dialect to ensure that their message dominates users' search results. Digital marketers call this process "seeding"—distributing content across the web to increase brand awareness and turn viewers into customers.

She uses a variation of the term (seed, seeded, or seeding) six additional times, each time in reference to content being seeded (p. 134, 140, 180, 183, 207, and 215).

This "content seeding" related to the "information seeding" done to encourage the development of online communities (Nagaraj, 2021).

[69] See Karapapa & Borghi (2015) for a discussion of search engine liability (in Europe) for search autocomplete suggestions.

markedly different from those for [rick santorum]. Golebiewski and boyd introduce "data void" to describe search terms for which there are problematic content due to the the limited amount of total relevant content. These search terms are sometimes targeted by people acting strategically or they may distribute both the search seeds and the content. Caulfield (2019a) presents an example of a search seed for a data void in a "Google This Ploy" with the suggested searches written on beach balls at a rally, discusses the example as a "data void", and suggests what educators can teach students to do when searching.[70]

I find the knowledge of the mechanisms of search seeds embedded within the work practices of data engineers. The occupational, professional, and technical components in the work of data engineering around the production and socialization of search queries (the terms or keywords to search, or the search seeds) incorporate and embed, or hold, knowledge of the mechanisms of web search.

4.2.2 Spaces for evaluation

There are two key spaces for evaluation in the work practices of the data engineers.

- space for running workable code
- space for gathering feedback – inc. meetings, prototypes, testing, code review, CI/CD

4.2.2.1 Running workable code The data engineers do not depend solely on their knowledge of the problem domain and of the mechanisms of web search to evaluate search results. They often will take a portion of code from a web search, or an idea of a possible solution to their problem, and test it quickly on their system.

Here is Shreyan:

> Iteratively figure out if you can actually get something close to what you require. And trying things out, and once you try one thing out, you try another thing out and you try a third thing out. An iterative process of a solution based upon multiple layers of search.

[70] Caulfield does not suggest educators teach students the mechanisms of search engines, but to do three things which he expands on: 1. Choose your search terms well ("First, let students know that all search terms should be carefully chosen, and ideally formed using terms associated with the quality and objectivity of the information you want back."). 2. Search for yourself ("There's nothing more ridiculous than a person talking about thinking for themselves while searching on terms they were told to search on by random white supremacists or flat-earthers on the internet." He also suggests that educators tell students to "avoid auto-complete in searches unless it truly is what you were just about to type".) 3. Anticipate what sorts of sources might be in a good search — and notice if they don't show up ("Before going down the search result rabbit hole, ask yourself what sort of sources would be the most authoritative to you on those issues and look to see if those sorts of pages show up in the results.")

> [. . .] If this is in the ballpark of what I want, I would try it out, of course. Or if I don't try it out. Can I get a more clearer solution from that? Let's say I search, OK this is the error. 'SQL code 350-some random shit'. Then that gives you a Stack Exchange page where somebody is talking about something but not exactly the problem. I would try to see, OK, is this the problem I want? Or I would try out the solution.
>
> So searching is sometimes faster than trying the solution out. So it would dependent upon which one it is. If I can see, oh, this makes sense, let just try. If it won't take me too long to run the code, so I'll just run it like this. Otherwise, my habit is to give it a couple more searches to actually figure things out.

This is seen in how Shawn, the developer advocate, discussed searching for resolutions of bugs or errors:

> And you can try it out, run a couple quick tests on it and make sure it actually works. And then you can move on with your day.

Running workable code, or proofs-of-concept, can also be done as a shorthand test of not just whether the found-answer supports quality code, but of the quality of the whole search experience when searching about a particular tool. That is, the findability of answers can gauged by seeing how difficult it is to get a simple proof-of-concept running in that language or software. Raha discussed searching for technology solutions, in a more strategic exploratory search phase, and how she was attuned to the content and community around the tool. It would be concerning, in this exploratory phase, if it wasn't quickly apparent how to run a tutorial or whether a particular bug was addressed in the tool:

> I think its not the search engine,its just the lack of community at the time. [. . .]
>
> It is pretty much very easy to do some sort of like POC and test it out and see it's not working.

I will present more from Amar regarding proof of concepts and getting feedback from others in the organization in the next subsection, but an initial feedback is whether the code is running: "proof of concepts up and running". Feedback from a snippet of code running is limited. Shawn:

What context you're dealing with and with every search you do is only adding to your context of what you know and understand. It is kinda like an experiment every single time. [. . .] [. . .] Sometimes things will make the test pass or give you output that is seemingly good but it is actually performing at scale very poorly. That actually gets you the correct output at small scale but whenever you put it into production.... that's where the rubber

hits the road and you realize that your search solution was actually the incorrect solution even though it gave you the green light at a smaller scale.

Getting some code running can, and in some workplaces is expected to, include immediate test cases. As Vivek noted:

> If I copy something from Stack Overflow and link that, a lot of times people ask "what are the test cases that it is passing?"" They don't let it go just because it was from Stack Overflow. So you do code review that part.

While for Ross, workable meant it wasn't breaking something (and with some eye on the future, which is addressed further in decoupling performance from search):

> when you use something new I think, it's expected here at least that you're going to do enough testing that it works in the way we are going to use it.
>
> [. . .]
>
> You try to foresee [tech debt and complexity] as you can, but you really you just want to make sure that it wont break something. It works in what we are doing now, that is the testing that you do. With an eye on future use, and then that's enough.

This pattern reveals that data engineers rely on the ability to quickly and iteratively test and validate search results or the results-of-search. Their work practices, organizational setting and the technologies of code itself provide distinct means of evaluating search results. These situated resources, rather than data engineer's individual knowledge of search mechanisms, bolster and inform the internal expertise of the data engineer.

4.2.2.2 Tend to gather feedback

Amar discussed feedback around proofs of concepts as an opportunity to address how even senior engineers overlook resources and benefit from feedback.

> I think earlier in the quarter my work is mostly about architecting systems, coming up with ideas, coming up with designs, breaking down a complex problem into smaller achievable pieces building proof of concepts to make sure the idea works at a small scale and then coming up with what the actual thing is going to be. And that's like, that's a very research heavy phase. Validating ideas and solution.

Amar said they would "propose a solution with a proper list of pros and cons" and "get external feedback". In his case this sometimes included subject matter experts, so database administrators within the company with decades of experience to check the nuances of a planned use of a particular tool.

Later in the interview he said:

> Again, even experienced engineers can overlook resources, so that's why you have these things. Where anytime you work on certain things you tend to gather feedback. Say I work on aproof of concept. I want to make sure Igather feedback before really executing it because I may have overlooked a resource that someone else might be aware of. So they might point out "hey did you check this, maybe you could find better solutions there" or something like that.

Devin also talked about feedback earlier in the work cycle:

> For those big projects, currently at [our company] is we do a 20% review. 20% of the work has been done. This how my planning process has been going. This is the discovery I've done. And this is what I'm planning on the next step which is going to be development. Let's bring the whole team together of ten-ish engineers and let's talk about what I've done to get this far, let's talk about the decisions that I've made to determine what I'm going to be doing next and let's have everybody collectively weigh in on my decisions and potentially agree that things are going well and this is the path we are going to go down or iterate on things we should be doing differently. Usually by the time you get to the 20% review you've already done the legwork. You've already reached out to people specifically to say 'this is what I've been doing, this is how we are going to solve it' and so usually you're not going to find many surprises in those meetings. But it is sorta a gut check so that we can all sit down and agree that we don't go down this path that we are going to regret.

And then there are also code reviews, which (earlier in the interview) Devin distinguished from feedback:

> I think feedback is more generally like this is what went well, this is what didn't go well and this is what we can do differently down the road.
>
> I think code reviews are very specifically more 'like this has passed the litmus test, let's go ahead and accept that and move on'. Or 'this code hasn't passed the litmus test, let's correct it.' Its very specific.

And there are also automated tests or checks sometimes built into the code review workflow. So Ross would push code to a fork, "check off that you double checked some things" and then, prior to another engineer reviewing the code, integration, syntax, and styling tests are run. He said, "sometimes you just push it up and see if it passes".

Others also talked about presenting prototypes or minimum viable product (MVPs) for feedback. Phillip talked about feedback from the users (in his case, internal customers who

the engineers interact with directly) before the formal code review:

> Typically the process is we usually build a prototype, or an MVP, and try to test out the feasibility. If the data is accurate or if it fits the requirements. Then we'll just re-iterate based on the feedback from customers and stakeholders, which we call user acceptance testing. So it is a lot of re-iteration. And so we finally get to that final result.

The work practices around acceptance testing, feedback, and code review collectivize or distribute the evaluation of the search results.

4.2.3 Decoupling performance from search

Current data engineering work practices incorporate elements of "decoupling" between provisional evaluations of web search results and key actions, such as deploying code into production. This decoupling is a result of the lack of automation, with the gap[71] between written and deployed code ideally providing slack, alternative methods to evaluate, and "buffers and redundancies" that are "designed-in, deliberate" (Perrow, 1984, p. 96).[72] This ideally provides time and space for re-evaluation and recovery, if necessary. Data engineers' work components are configured to support this decoupling by handling exceptions, errors, and other faults by routine. This margin limits the immediate effects that potential issues introduced through web search may have on the products or pipelines built by data engineers. Their systems for mitigating risk include code and processes that envelope deployment and fallback-and-recovery systems.[73] The decoupling is inclusive of the above spaces for evaluation, and extends beyond them.

[71] While I rely on the term "decoupling", it is related to the "wide gaps" between different technologies discussed in Bailey & Leonardi (2015). While they found efforts to automate across gaps between technologies in hardware engineering, they did not find that in structural engineers. Instead they argue (p. 123):

> Because senior engineers in particular viewed the navigation of wide gaps as beneficial for the cultivation of testing acumen and prudent in the face of liability concerns and government regulations, there was little impetus to hasten automation by limiting the number of technologies that lined each gap in structural engineering

[72] Though I am drawing on similar imagery and so their use is somewhat related, my sense of decoupling is distinct from the "decoupling" in Meyer & Rowan (1977) (which is actually much more related to the absence of technocratization of web search, discussed in Owning searching). My use is also related to the discussion of tight and loose coupling in Perrow (1984), although not directly drawn from that and describes instead a situation at the point of deployment where the data engineering organization can draw on tendencies of both tight and loose coupling. Perrow does cite to Meyer & Rowan (1978) (a later piece), building on their description of loose coupling. See also decoupling in Christin (2017), drawing on Meyer & Rowan (1977).

[73] See also the discussion above regarding liability pressures in Admitting searching: Searching for opportunities.

But even with seeds and running code and feedback there are still ways that poor quality code found or shaped by a date engineer's web searches can be added to the organizational code base. Structures in the work practices of data engineers designed to address a variety of concerns can be used to address this as well. The work practices of data engineers are directed to anticipate (effectively if not explicitly) failed search evaluations. Ajit, the staff data engineer at a major retailer, talked about introducing version control to his team:

> These are the different changes that happened on this specific piece of code over time to understand why did we make those changes and to also track, hey something happened, say, 1st of October. That's why version control comes into play. And, oh, we made a code change and that code change sorta messed this up.
>
> I knew about GitHub. I knew about the basics, like you could fork someone's repository. But then like pull requests and submitting stuff for review. Comparing... commenting... challenging people: 'these five lines of code make no sense.' Now they are a day-to-day practice on my team because it makes it greatly easy for you to do a lot of things.
>
> Especially when you are onboarding new team members, you don't have to worry about this person has built something or changed something which just messed everything up. Now you have version control and you have reviews so all of your core production level stuff is never touched without proper reviews.

While his description of the processes include feedback and review just discussed, the ability to identify mishaps in the code is a place where knowledge of the mechanism is externalized and embedded.

Handling exceptions by routine is also seen in how the core work aim of some of the data engineers was directed towards measuring this dimensions of quality in the code or platform. Charles discussed his work:

> The main project I was working on was doing a lot of metrics and data pipelines for this one massive experiment experiment we were running. We were making a ton of changes to the product listing page and testing...

This is the case for Kari's work as well. Working with the data scientists in her company and stepping in once they have a model they want to put into production:

> My team kinda comes in to figure out: OK, how do we turn this model into an API that is ready for production? How do we integrate that into the business based on the current engineering and product structures that we have setup for that feature? How do we start to scale this and monitor this properly considering

> the fact that we have client facing traffic now?

And also with Victor. He discusses "productionalizing":

> Then we build out that MVP and my team focuses on productionalizing it, making sure it will work without failing, that it is reliable, that we can count on the results, and that we've setup the after-the-fact monitoring to make sure it is still up and running and not falling over and it is doing what we said it would do.

The web search practices are also presented as ways to repair faults or limitations that do make their way into production code. Here is Shawn, again talking about writing the messages in exceptions that engineers will see if that exception is raised:

> The more expressive and descriptive you can be when you're dealing with tests, when you are throwing exceptions, real-time exceptions, especially ones that are at runtime. You know something is going to go wrong real-time.
>
> You never know where your exceptions are going to be thrown. And you want to assume it is always going to be in some production state. And that there will be something that will help somebody find this information online or quickly just by reading it understand what the hell's going on and that they need to react in some way shape or form.
>
> So having that context given to you so clearly, without having sometimes even having to search it. The ideal state for an exception is not having to be searched, but that is rarely– that is something that is almost—you know depending on the expertise and what the person knows, the framework, and the jargon, and all the other stuff— You can't make, "OK, to understand this exception, let's start with 'what is a variable' " [chuckling] You can't express everything at the starting point for every person that is going to be possibly be running into that error code. You're always going to be missing some context and that is where search comes in at some point.

Amar also shared about a role for search, after architecting the system plan and getting feedback on prototypes, in the designers of the systems learning how to handle exception by routine:

> And towards the other half of the quarter it is mostly about doing things, following best practices and trying to figure out, doing everything by the book, perfecting all aspects of running a production system, that includes: How do you easily deploy things? How do you deploy changes? How do you automate a lot of things? How do you, I guess, plan for failures?

> It's mostly about, once you have an idea that this will work, how do you make sure this will work 99.9999% of the time?
>
> So that's where you'll spend a lot of time building a system up, but a different side of the system, more on the operation side of things again you need a lot of research in that because there are times where you already have best practices in play but a lot of times things are new, things are something that you haven't worked with before. Learning how to do certain things, its always something that you have to look it up.
>
> Overall, pretty involved, you need a lot of web search for that.

As "things are new", doing things "by the book", meant, in part, doing your web searches. He highlights the role of search while also presenting the work involved in implementing a system in such a way that exceptions can be handled by routine.

The work practices of data engineers are structured or channeled in ways able to avoid or recover from many anticipated and unexpected errors. Together, the socialization of seeds, the feedback running workable code and gathered from various collaborative work practices, structures in the storage and handling of the code and data serve as risk-limiting mechanisms. At least insofar as code quality is concerned, search failures or failures in searching are handled by routine.[74]

This capacity to handle such exceptions steps in where knowledge of the mechanism and associated skill in use of it are missing or found wanting.

These risk-limiting mechanisms, including elements of friction, decouple effective performance in key actions from search automation bias.

Skitka et al. (2000), looking at human-machine interactions in aviation, defined automation bias as "the use of automation as a heuristic replacement for vigilant information seeking and processing" [p. 86]. Decoupling limits the automation bias widely found in research on the uses of web search. Some such research focuses on the automaticity in what search results are clicked and attended to. Vaidhyanathan (2011) writes that "our habits (trust, inertia, impatience) keep us from clicking past the first page of search results" [p. 15]. Haider & Sundin (2019) discuss a range of this work, pulling together White (2016) on "position bias" (which he notes is "also referred to as 'trust' bias or 'presentation bias' ") (p. 65); Pan et al. (2007), Schultheiß et al. (2018); and Höchstötter & Lewandowski (2009), writing (p. 33-34):

> line of research investigates what people choose from the search engine results

[74] There are other effects from search failures or failures in searching that are not handled within these structures. There are related to interpersonal relations, status, and the associated affective experiences of search and search failures (to be discussed in subsequent chapters).

> page – often referred to as SERP – and why they choose as they do. This work convincingly shows that how people choose links is primarily based on where these links are located on the search engine results page.

Narayanan & De Cremer (2022) review this and other research, writing, "as these various studies show, the average user seems to treat Google with "a default, *prima facie* trust" [emphasis in Narayanan and de Cremer]" (p. 21).[75] I use the term "search automation bias" to describe this. The harm from such bias depends not only on the credence granted higher-ranked results, but also the subject of the results and whether the results, or simply the page titles and snippets, are "likely to mislead" (Lurie & Mulligan, 2021). The risks of harmful effects from search automation bias are likely higher in other areas of searching, particularly when the automation bias is regarding searches for which the search engine returns problematic results, such as search results reproducing representational harms (Noble, 2018).

Despite their significant domain knowledge data engineers may at times rely on the ranking of search results "as a heuristic replacement for vigilant information seeking and processing" (Skitka et al., 2000, p. 86). The data engineering web search activity is part of the larger configuration of information seeking and processing of the organization, with configurations of organizational components creating separation, or decoupling, between the web searches and the performance of post-search activity. This decoupling is achieved through occupational, professional, and technical components that include and extend past the provisional evaluation in the spaces discussed above, and through the various steps involved in handling exceptions by routine. The components of interest are not linked to web search in a way that forces any particular action, it is not actually automated. Errors introduced in web searching, the effects of which for data engineers are often externalized into code, configuration files, or data infrastructures can perhaps be repaired in data engineering work in a manner that is distinct from wrong or less useful beliefs that individuals might develop from inadequate evaluation of search results in other situations, such as in a medical emergency, while filing out a ballot, or when making a major purchase.[76]

[75] The internal citation is to Gunn & Lynch (2018), who, perhaps more accurately, write of "most people" rather than "the average user" and use the word "googling" rather than "Google": "most people treat googling with default, prima facie trust" [p. 42].

[76] Google's Search Quality Evaluator Guidelines for instance, notes that some search topics have "a high risk of harm because content about these topics could significantly impact the health, financial stability, or safety of people, or the welfare or well-being of society" and that pages on such topics "require the most scrutiny for Page Quality rating" (Google, 2022). They refer to these topics as "Your Money or Your Life" topics, using also the acronym YMYL. For more on the guidelines and Google's contractor evaluators, see Bilić (2016) and Meisner et al. (2022), respectively.

4.3 Discussion: Missing search knowledge

The preceding shows the knowledge and practices used to make search work distributed and embedded in the occupational, professional, and technical components of their work practices, as an expert field.[77]. In this we can see the searching as extended throughout those more distant practices. It follows that participation in those shared practices may be where data engineers learn the practice of searching as a data engineer and make use of the embedded expertise.

The finding to be shown in this section is that the occupational, professional, and technical components supply a significant amount of context, or structure, as scaffolding for query selection and search evaluation. I do not examine here the emergence or evolution of these work practices, sociomaterial practices are constantly reproduced and maintained, and there is a "constitutive entanglement" between people and the material (Orlikowski (2007), building on Giddens (1991)'s work on social reproduction, "repetitive activities" and "regularized social practices across time and space").

The knowledge used to make search work is distributed in the work practices around the production and socialization of search queries (the terms or keywords to search, or the search seeds) and search evaluation (the manner of evaluating and making-use-of search results). This knowledge is importantly distinct from knowledge of the mechanisms of web search. This knowledge is of other ways of validating search results for their purposes, knowledge very situated in the material resources and expertise of this community. The data engineers did not, in interviews, present themselves as having sophisticated knowledge about the mechanisms of web search in their heads. Or other sophisticated knowledge for searching in their heads. But they do make use of such knowledge through their work and practices.

We can look at the artifacts and the practices to find the knowledge that seems to be missing (Latour, 1986, 1990, 1992). Existing work identifies knowledge embedded in relationships (Badaracco, 1991; Blackler, 1995)), distributed throughout social and technical arrangements (Hutchins, 1995) and within "supporting protocols (norms about how and where one uses it[])]" (Gitelman, 2006, p. 5). Knowledge, not directly of the mechanisms but effectively so through referred knowledge of the occupational use of the mechanism, is also embedded in social norms (Feldman & March, 1981) or "genre rules" (Yates & Orlikowski, 1992). Cambrosio et al. (2013) argues that knowledge and know-how "cannot function as"expertise" unless they become part of a network." Expertise isn't possessed or held by any one searcher, but there is a "a network of expertise composed of other actors, devices and instruments, concepts, and institutional and spatial arrangements, distributed in multiple loci yet assembled into a

[77]Christin (2017) uses the phrases "expert fields" to describe "configurations of actors and institutions sharing a belief in the legitimacy of specific forms of knowledge as a basis for intervention in public affairs", building on Collins & Evans (2007) & Fourcade (2010). See also Kluttz & Mulligan (2019, pp. 860–861Fn29).

coherent collective agency." (Eyal, 2019)

Lave & Wenger (1991) looked at early work from Hutchins on apprenticeship in shipboard navigation, before publication of Hutchins (1995), as one of their case studies, writing about the participation available from how the components of the navigation deck were configured [p. 102]:

> Apprentice quartermasters not only have access to the physical activities going on around them and to the tools of the trade; they participate in information flows and conversations, in a context in which they can make sense of what they observe and hear. In focusing on the epistemological role of artifacts in the context of the social organization of knowledge, this notion of transparency constitutes, as it were, the cultural organization of access. As such, it does not apply to technology only, but to all forms of access to practice.

Knowledge of how to search the web like a data engineer is provided in the extensions of search in the work of data engineers. These extensions span individuals and organizations. The embedded knowledge is in the wide work of data engineering itself. Rahman et al. (2018) open their paper evaluating developers use of general-purpose web search for retrieving code saying, "Search is an integral part of a software development process." It is that. But I also turn that around, arguing that that data engineering development processes are integral to the data engineers practice of search. Search and the knowledge for it is extended in the various tools the data engineers use, in the tools they build, and in their relations with each other. For the fully participating data engineer, the occupational, professional, and technical components of their work interact to force, incentivize, and constrain work practices to align with with successful uses of web search.

4.4 Conclusions: Search extended and knowledge embedded

Seeing search as extended and knowledge for effective search embedded in this way of work, rather than in individuals[78], may reorient calls for education, design, or regulation that are premised on transparency or explainability and web search literacy designed to fill individual knowledge deficits. We could develop habits and practices that do not require peering into

[78]Though the individuals do have considerable domain expertise—knowledge of terms to search, of the qualities of the various resources on their topics of work, and of how to navigate the wide work practices of fellow data engineers so as to maintain such knowledge. Individuals may take, or translate, their knowledge of their way of work to other places of work with similar arrangements. But their knowledge is not directly about searching the web. They may not so effectively search about some topics if they do not have knowledge (or access to communities to learn it) of how that topic area structures its knowledge of web search (here, how search queries are seeded and search results evaluated) or in places where such seeds and evaluation are not embedded in the way of work.

the opaque and ever-changing systems of web search. In some settings we may be able to make effective use of search tools without transparency into its data sources and algorithms. We may shift focus to understanding how to mobilize and recognize effective search seeds in different domains. Rather than encouraging every searcher to understand the mechanisms, we could focus on developing and calibrating our ability to evaluate, individually and collectively. It may help us work towards identifying in which contexts the suggestion that someone turn to web search may lead to results-of-search of differing quality. Where and with whom is 'just google it' helpful or not? Finally, this framing may be used to guide the identification of aspects of the web search infrastructures, topics (or places)[79], or situations of searching that may be reconfigured for more effective web search practices.

[79]From the Greek, *topos* for a place.

5 Repairing searching

Successful practitioners establish ways of working around the limits of their tools. What happens when web searches fail? Descriptions of successful practices need to include how practitioners navigate failure.

The sort of search failures I'm discussing in this chapter is that of someone confronting a problem, turning to web search, and not finding resolution. I am not here concerned with situations where a searcher leaves a search with a wrong answer (their methods of search evaluation are addressed in the Extending searching chapter), though sometimes the identification of a wrong answer is the impetus to seek help from peers. The failed searches I am concerned with are where the searcher does not find useful answers to their questions. I am also not particularly concerned with identifying root causes of the search failures here. The causes may seem to stem from any number of interacting individual, organizational, and other factors that affect:

- the material available to be searched,
- the design and performance of the search engine's index, algorithms, and interface,
- the knowledge, memory, and attention of the searcher

How do data engineers navigate failed searches?[80] Particularly, how do they do that within this environment where (1) professional credibility might be at risk (despite all the search confessions) and where (2) organizational and professional strategies have rendered the work searchable. In this chapter I talk about how the data engineers bridge web search gaps and demonstrate and share their skills and expertise in the process.

This chapter involves a Handoff analysis of the configurations of components and their modes of engagement in web search practices that take place well-beyond the search box and the SERP (see Mulligan & Nissenbaum (2020) and Goldenfein et al. (2020) in The Handoff analytical lens).

A key observation of data engineering work is that they are operating at the edge, working with the new, or new to them, tools, systems, or other ways of handling data. Navigating

[80] An earlier draft of this chapter focused on 'ignorance', a polysemous concept I was using to refer to knowledge gaps or the limits of knowledge of the data engineer, their organization, the search engine, and the Internet. I was relying on concepts developed in 'ignorance studies', partially around how identifying, naming, and navigating ignorance can create or spread knowledge and how ignorance is socially constructed (Bhatt & MacKenzie, 2019; Gross & McGoey, 2015, 2022; Natarajan, 2016; Peels & Pritchard, 2021; Proctor & Schiebinger, 2008; Smithson, 1985). My initial use of 'ignorance', a term often viewed and felt, viscerally, as a pejorative, unsteadily straddled both mixed causes and effects. My focus now on "failed searches" highlights a particular sort of situation and how data engineers address it without using 'ignorance' in multiple confusing registers and applied to different types of actors.

around these edges, or learning around the edge, necessitates constantly working to "keep up" (Kotamraju, 2002) and engage in "intensive self-learning" (Avnoon, 2021) in a fluid, and loosely defined, field. They use web search to keep up. What do data engineers do when their searches fail?

I answer that by looking at how they talk about asking and answering questions of or for colleagues. I show how data engineers package questions for colleagues, carefully explaining what they do or don't know and have or haven't tried. That includes a due diligence. Both how they do and communicate due diligence, helps them learn across their search failures, to access or understand something new, different, or unexpected. I consider the valuation and legitimation work that is part of both asking and answering and the impetus these interactions with others have on workplace web searching.

5.1 Asking and answering

Successful web search practices include articulation work—"work that gets things back 'on track' in the face of the unexpected, and modifies action to accommodate unanticipated contingencies" (Star & Strauss, 2004, p. 10)—that repairs failed searches. Web search itself may often be a sort of articulation work for the data engineers, and in the repairs of failed web searches the articulation work is turned back on itself. Jackson (2014) writes that repair work is "itself a facet or form of articulation work (and vice versa)" [p. 223]. Jackson (2014) also provides an appropriate warning for the findings of this chapter, to keep in mind that "repair is not always heroic or directed toward noble ends, and may function as much in defense as in resistance to antidemocratic and antihumanist project" [p. 233].

What we know comes from how we navigate not knowing. Data engineers are able to operate at the edge through the ways they navigate not knowing, uncertainty, and change. I describe here the practices around the asking and answering of questions. The core of this chapter focuses on the empirical material introduced around asking and answering. This articulation work is practical and networked.[81] It facilitates communication and coordination around learning at the edge.

Data engineer practices for asking and answering questions of and for colleagues allow them to navigate search failures as they package questions with due diligence, find renewed search impetus from (anticipated) conversation, and jockey around valuations and legitimations. This section presents these three patterns with examples from interviews, interweaving analysis and references to related prior literature. Data engineers employ these practices to

[81] This work is part of the construction and maintenance of practical networked knowledge from Extending searching. "Practical", recall, is following Cotter (2022), "networked" is drawn from Cambrosio et al. (2013)'s reference to "networked skills".

navigate discrete search failures and also justify the nature of their search-reliant expertise. The practices allow for demonstrations of individual and occupational legitimacy in the face of search failures. The sites of supposed search failures become sites of shared learning that provide coordination for other work interactions. This cooperative problem solving is intrinsically valuable to the participants, as well as a site for constructing and presenting themselves as data engineers. As Sabel (1984) wrote discussing "collaboration between labor and capital", a craftsperson will put up with even assembly line work "or as feeders of automatic machines" if "at least occasionally they can test their craft knowledge against unforeseen problems". (14)

Conversations in the interviews ranged from web search to the complements, substitutes, and complications of web search. Most interviewees discussed searching internal systems (various enterprise search systems, including searching on their workplace chat platform, an enterprise version of Stack Overflow for Teams, or otherwise searching internal documentation) and asking colleagues questions. I also explicitly asked what people did when search (seemed to have) failed.

First I will provide a high-level intro to the conversations I had about asking and answering. Many discussions of how to ask colleagues questions (whether a Slack direct message or on a dedicated Slack support or troubleshooting channel; rarely via email) delve into how one is expected to demonstrate prior searches or search attempts. The reasons for this have been presented as ranging from demonstrating one deserves to ask a question to simply what is necessary to get the fastest response. Some people discussed this from the perspective of having to field questions and discussed things from wanting people to show they are invested in the answer to techniques to reduce the rate of questions (from not answering immediately to requiring 'trouble tickets'[82] to be completed).

Some interviewees discussed heightened emotions or explicit concerns about how they may be judged (by colleagues or management) or treated that kept them from asking questions of colleagues until they'd exhausted their other resources (principally internal and web search). They don't want a colleague to ask if they've searched yet or to suggest they 'turn-it-off-and-back-on-again'. One interviewee provided some insight on an internal discussion around sharing guidelines for when and how to ask questions with those requesting help of their team, which I will expand on below. Individuals and teams seem to largely manage questions in informal routines, though generally in somewhat explicit channels[83]. This produced, to

[82]Trouble tickets are formalized and explicit requests for assistance. Ostensibly used to facilitate the repair or resolution of some problem or issue, they are also used for tracking issues and quantifying work performance of those tasked with their completion. They can be used also to increase the barrier to request help, at least formally.

[83]I use the word 'channel' here to refer to various 'communication channels' though it is also the name in Slack and Microsoft Teams to refer to what internet chat relay services might call 'rooms', as in a 'chat room'.

some participants, a way of subtly or crudely exercising power (Freeman, 2013).

Highlighting some of that informality (and repeating some of what was discussed in Admitting searching), here is an overview from Noah on what he tells new colleagues:

> When I have those onboarding meetings with new joiners, I point them to some of the search tools we have informally and I always point them to the support channels that we have.
>
> And I tell them — when I started I was very hesitant to ask questions in these support channels kinda for some of the reasons that I've been talking about, oh I didn't want someone to be a dick to me. Or like think that I hadn't done enough leg work before asking the question and what I tell them is, you know, 'just get over that, especially as a new employee, don't sit around wasting your time and wondering if you phrased the question right, just ask it. Over time you will learn, from paying attention and watching other people ask, what the right way to phrase things is and the expectation around questions, but when you're new just ask them, and people will realize that you're new.
>
> No one is going to be mean to anyone for violating these norms on question asking on one instance. They are going to notice it over time, 'oh that person always has bad questions', 'that person never really does the leg work and wants people to do it for them'. And that's when people will say something to you. But on a one-off instance people might say, oh, you could have searched here or they might just give you the answer.

I will now present the three patterns.

5.1.1 Due diligence and packaging questions

Michael:

> 'hey I did my own due diligence on this'

Ross:

> . . . show that I tried and that I have some grasp of the problem space

Christina:

> This is what I tried. This is what I tried to find a solution. And I'm now stuck.

Lauren:

> '**I've done all these other things, I'm aware of all these other things**, and I'm coming to you as a last resort.'

This first pattern is focused on packaging questions, this includes doing and demonstrating due diligence. I asked data engineers about question-asking outside of search, initially hoping to surface examples of how people talked about search. Gradually I recognized the question-asking practices as repairing or completing searches, rather than simply a substitute for web searching.

After Noah shared the comments above, I started to ask a follow-up question and he jumped in to correct my framing of "asking too many questions" as being rude or what is of concern. Rather than "too many" being the concern, in his eyes, it is how they are presented. He said:

> I don't think it's "asking too many questions". If you **show that you have done, that you have put some effort into this**, even if it's just a little bit, "here's what I tried". I think that that tends to make people (and I don't even think its just a tech thing) much more receptive to helping you. **Showing that you have invested some of your own time** and now you are asking someone else to invest a little bit of their time to help. It is much easier to, as the second person, to feel like you're making a good investment if you feel like **the first person has bought in**.

I said, "Yeah, bought-in" and he quickly continued:

> And, obviously, this is all a very fine line. That is the exact same that allows some people who are real jerks on Stack Overflow to justify what a jerk they are being. Right? Like, 'oh that person just asked a stupid question so I was being a jerk to them because **they didn't make enough of a time investment before bothering me**.' Right? It's a spectrum, it's a complicated thing.

I asked Michael when he would decide to ask colleagues a question. He noted that different engineers would approach things differently and then provided an extended answer (below are excerpts book-ending ten minutes of our conversation). He described the packaging of the questions similarly to Noah:

> try it out on your own, try to solve it the best you can, and keep searching, and spend all your resources first, and then go to your team if you're actually truly stuck and you can't figure out something. As an engineer: don't depend too much on hand holding. But if I am stuck on something then I go to more senior technical leaders.
>
> [. . .]

> OK, I'm going to try my best to solve something and when I go to this lead or this senior engineer or someone above me to help me answer it. I want to have available the findings that I found from my own research so that **I can present to them and kinda validate, 'hey I did my own due diligence on this**. I'm not quite sure what the problem is but maybe you can see something I can't or have an idea of something I can try. Or, altogether just scratch that because altogether this is an impediment and we don't want to waste time on that.'

Noah and Michael present the packaging of the question as intended, or read, as indicating investment or effort prior to the asking. Ross also mentioned due diligence, though added that its presentation also serves to show that he has "some grasp of the problem space".

Early in our conversation Ross said he'd do due diligence before he would "ping" someone with a question, saying "I try not to bother people until I've done my **due diligence**." I probed on what he did when was not finding an answer.

> Yeah, it's funny.You think that might be a time I post a question. I tend not to. I tend to loosen my search criteria. I tend to go more for something similar. This is a personal reflection of me.
>
> I don't like bothering people. I kinda don't like asking people to take time to respond to my questions until I've done **a real true diligence search**. Like, you know. 'I tried. Here's what I found and this encompasses what's out there and its not me, my thing. So, here's my question.'

Soon after I asked directly about the presentation of "due diligence"[84]. He summarized here:

> I present some information. I try to present some summary of what I've done to show that I tried and that I have some grasp of the problem space. And then I present my question.

He followed with, "I don't really need to see due diligence but I feel like I owe it to others." He also shared how he recollected coming to recognize the role of the presentation of due diligence:

[84] Direct 'due diligence question' of Ross:

> Q: Could we go to that a little bit? You had said 'your due diligence' before you ask someone. So you've described in part some of the due diligence that you do. But you're also presenting to someone that you did your due diligence when you ask them a question?
>
> Yeah
>
> Q: Could you talk about that a little bit? How you frame different questions?

> When you first get hired——oh, people are so helpful and everything——and then you're expected to tow your own load. Then you ask a question and you get a response back:
>
> "Well, what have you tried?" Oooh. Oooh. OK. 'Sorry. Here's what I tried.'
>
> Maybe it doesn't bother some people. But if I got that response, it's like, OK. I didn't do enough, I didn't show I needed it. I think early, at some point when I started, I got that from someone somewhere. And you're like, 'OK'.

Lauren also mentioned both prongs of the packaging of due diligence, the investment and the understanding. Her first mention of due diligence was what she would do before asking a question in an internal Slack support channel.

> Unless I was absolutely sure that my question was unanswerable some other way I wouldn't just blast it out for a million people to see.

(I checked, asking whether it was actually a million people and she clarified the channels she would ask questions in would have up to 100 people.) I asked her whether the context of being "absolutely sure" was provided when asking a question.[85]

> If you're trying to get the quickest answer from somebody you're like: **I've done all these other things, I'm aware of all these other things**, and I'm coming to you as a last resort.
>
> I would be very clear about that because especially when you're talking to developers or people on the super back-end the first thing they'll do is ask you, "oh, did you turn it off and turn it back on again, yes, I turned it off and back on again and I also did all these other things and that's why I'm coming to you as a last resort.
>
> Or it's an emergency. This page is down and I've refreshed it a million times.

Charles talked about how people would get "pretty annoyed" at the lack of due diligence, or its presentation.

[85] Direct 'due diligence question' of Lauren:

> Q: When you reach out to an individual to ask this question that your initial searches were not fruitful with, do you make sure provided this context? 'I tried XYZ, now I don't know.'
>
> Yeah.
>
> Q: Were there strict rules or policies or—
>
> No. [continued above]

> I'm really conscious about making sure that I don't ask a question that has already been asked. Because I feel like people on the forums, either general or even on our company's [internal Stack Overflow] page would get pretty annoyed if somebody was asking a question that had already been asked and answered before. Um, and so would really go to pretty good lengths to look through the previous ones and make that sure my question hadn't already been asked.
>
> Versus, you know, I've seen cases where somebody the second that they get an error from their IDE, they just take it and they type it into Stack Overflow and ask a new question. And I think people get pretty annoyed by that.
>
> [. . .]
>
> I would definitely try to convey to the person that I had been struggling with this for awhile and that is wasn't just some error message that I just came across while I was messing around.

He also spoke of people complaining about question-asking:

> Sometimes the ways that you learn is by learning what not to do. And so, I was good friends with a lot of my coworkers, most of whom where more experienced. And so, if we were at lunch or whatever, someone might say 'this person on our team, he keeps pinging asking me questions for this thing even though he hasn't tried XYZ.' A lot of times I would heart things like that so I would take mental notes, "don't do that". It was more of my own kind-of observations.

This pattern allows for learning around search failures. This includes search failures that result in part from the occupational and organizational approach to learning at the edge. It also engages with concerns of "keeping up" as individuals likewise navigate the edges of their own knowledge (Avnoon, 2021; Kotamraju, 2002). There are multiple functions of due diligence and packaging questions. Demonstration of investment encourages reciprocation. Communicating known context allows for quicker resolution. This enables people with different skills and knowledge to communicate effectively without the rudeness of assuming the contours of the other's understanding. The pattern also integrates with, or functionally supports, the two patterns discussed next. Packaging questions in particular ways works toward legitimating the question-asker as a responsible participant, a valid member of the community. Expectations of due diligence play a role in the impetus provided in conversation (though the expectations, and their backing in humiliation or other judgement, are not the whole impetus).

5.1.2 Valuations and legitimations

I focus here on the evaluations and legitimation work of *jockeying*. Due diligence and packaging questions both affect how others evaluate you as an engineer and are opportunities for the engineer to legitimate themselves, and not just their question. The asking & answering of questions broadly are a place for evaluation and legitimation.

At the time of our interview Nisha was a data engineer at a contractor working at a large tech company. She provides a clear introduction to the presence of evaluation/valuation around question-asking. I had slipped concerns other interviewees had mentioned about people perhaps being mean into my question around due diligence.[86] She said:

> I've seen that as well.
>
> That's another reason why people try to go to search groups or Google search rather than posting it internally to a group of developers that or might not know you and judge you.
>
> It's a competitive world[87], right? I mean, your peers, they probably want to help themselves before they help you, right? So—[pause] it also depends on the team dynamics. Yeah, I've seen that as well.

I then asked about the packaging of questions in these large groups.

> That's how you are expected to ask questions in these groups...
>
> You have a problem. You've tried searching for it. But you've not had any success. You've tried a few things. You want to list them down because, for example: Your speaker is not working. Someone will suggest restarting the machine. You've already restarted your machine. You don't them delivering the same information.
>
> So, its always helpful when you're asking in a group, you ask very specific questions.
>
> When you're doing a Google search it's OK to see the same replies because you can always scroll down and see more.

[86] a 'due diligence question' of Nisha:

> Q: Is there— I'm curious. Some of the people I've talked to have talked about all the work they do to prepare or package their question that they submit. Because they're nervous people are going to be mean or mean in the past. Do you see that or do you experience that?

[87] Nisha had worked in multiple organizations as a data engineer and noted her current organization was particularly competitive.

> But when you're interacting with people you want to give them very specific point to point information. So you give them exactly whatever you need to get the information out of them.
>
> [. . .]
>
> Q: Do you ever answer questions in these groups?
>
> Sometimes. If I've faced an issue, yes I try to answer. Because I know how it is to ask questions. It takes a lot of courage. People usually ask questions after they've done what they can. So I try to be helpful and mindful.
>
> I don't want to be that person who does not respond when there is a problem and I know the solution.
>
> Q: You said it takes a lot of courage. Could you just say again why it takes a lot of courage.
>
> Because your managers would be on the chat group. They would be figuring out that you are not able to solve this problem. And your peers. You can also be judged by your peers.
>
> So, only after a person has searched, done what they can, tried things, is when they would normally post questions.

The data engineers are not narrowly focused on demonstrating a need to ask a question, or their prior investment in trying to answer it and their knowledge. They are also looking to avoid negative judgement and present themselves as experts.

This judgment is not evenly applied, elsewhere in the interview Nisha shared about being a woman in her organization:

> Like for me, I know that for whatever reason, women are minority in an engineering group. And they get picked on more. So we constantly have to watch our backs ensure that, even if you're putting a comment in code, which will not be checked into the code base, but it's only going to be run as a temporary fix, you still have to ensure that your comment, even the comments, are perfect.

Jamie said she would generally try to answer her questions on her own because she enjoys it but that she doesn't hesitate to ask her coworkers questions. Then she paused, and provided a fuller picture:

> Let me rephrase that. My natural state is not to hesitate. There have been times in the past where I have hesitated to do that. And that is largely due to my minority status in the tech industry and what people's assumptions are

> about a women in the tech industry who is asking questions, right? Those same assumptions are not made of men who are asking questions.
>
> So definitely in those companies where that male-centric tech view has been more present, I have been more resistant to ask questions. Because, culturally, there has been this implication that if you're asking questions then you don't know what you are doing, which is ridiculous, totally absurd. Right?
>
> So it is not my natural state to behave that way. But in some companies, for people to think of me as the expert that I am I have had to change the way that I've done thing.

She went on to distinguish her current workplace from past experiences. Megan also distinguished different workplace experiences, though her experience was somewhat reversed. I talked to her in a new participant member check, describing my initial findings and asking for her reactions. She said:

> All the women I've ever worked with in technical roles have been at times either effectively sidelined or people have tried to dismiss them as non-technical. That fear of appearing non-technical is a real thing.
>
> I just started a new job. Once I've been at an organization for a while, I have an established a reputation. I feel like I know people, people know me, and I'm very comfortable asking questions out in the open. And especially when I was an engineering manager, I tried to model that behavior and asking questions out in the open. But I can tell you right now, I just started the job and 100% I'm sticking my questions under private channels, because you don't want to be perceived as struggling.
>
> ... My horror of doing my job badly is worse than my horror of asking questions. But I will say it does push me into private channels as opposed to public ones.

Fear of the consequences that may come from being judged stupid or lazy drives data engineers to search first and furiously before seeking advice from colleagues. Even if a question does not interrupt someone's work, it might be seen as indicating a lack of knowledge or responsibility, or as wasting your colleagues time (which is undesirable given data engineers' interest in speedy searching, next chapter). As Kari said, you want to "err on the side of: don't waste people's time."

The practice and place of search repair provides opportunities for data engineers to jockey for legitimation. Not just an opportunity, search repair demands such performances. As Goffman (1956) writes (p. 156):

> Audiences also accept the individual's particular performance as evidence of his capacity to perform the routine and even as evidence of his capacity to perform any routine.

In their answers and the quality of their due diligence the data engineers can present themselves to colleagues as capable, though some are required to demonstrate more to achieve equal esteem.

5.1.3 Renewed impetus

The web search practices of data engineers are shaped by the searcher being situated around others and the anticipations of those others' perceptions of the searcher's expertise or diligence. They are not normally anticipating interacting with others about a search, instead often expecting it will be quick. But when their searches start to show signs it might fail, conversations with their colleagues and the search repair practices come to mind. Search repair practices start before the question is asked. The expectation that a question for colleagues must include packaged due diligence and that it is a site for demonstrating legitimacy provides a renewed or refined impetus to search.

Searching amidst or in anticipation of (even potential or speculated) conversation is very different from searching alone. Formulating a question to ask a colleague, or even a search strategy you might be willing to admit to a colleague, is distinct from formulating a query for a search engine.

For help with working through problems, some even suggest formulating a question or talking through a problem with an inanimate object, like a rubber duck. "Rubber duck debugging" is a common reference point within coding work, even if it isn't explicitly practiced.[88] Though they call it "Rubber Ducking", here it is described in Hunt & Thomas (1999)'s book for practitioners, on the craft of programming, in a chapter on debugging [p. 95]:

> A very simple but particularly useful technique for finding the cause of a problem is simply to explain it to someone else. The other person should look over your shoulder at the screen, and nod his or her head constantly (like a rubber duck bobbing up and down in a bathtub). They do not need to say a word; the simple act of explaining, step by step, what the code is supposed to do often causes the problem to leap off the screen and announce itself.
>
> It sounds simple, but in explaining the problem to another person you must explicitly state things that you may take for granted when going through the code

[88] I introduced my Python students to rubber duck debugging when I taught the summer 'Python Boot Camp' to incoming graduate students at the School of Information (bringing in enough miniature rubber ducks for each of them).

> yourself. By having to verbalize some of these assumptions, you may suddenly gain new insight into the problem. [internal footnote omitted]

Rubber duck debugging achieves some of what anticipated, and rehearsed, conversations do for data engineers. Kari is a data platform engineer. Part of her responsibilities, besides systems design and implementation, include responding to support request from others in her organization. She said she, and her colleagues, will be "pretty busy" and get frustrated with the volume of requests for answers. Unless the question-asker shares enough of their due diligence she finds she has to respond first asking for more information.

> I always suggest: 'Hey, if you're going to ask me a question, give me a bunch of context on it beforehand, so that I can actually answer your question. So don't just send me a stack trace. Tell me what you were doing. What is the stack trace? Link to the code. All this stuff.'
>
> [. . .] It is totally fair to just send the question back and say, 'hey we need more information'.

I asked her if perhaps sometimes pushing people to provide more information will help them answer it on their own.

> Yeah, lot's of times. . . [chuckling]

I repeated: 'Lot's of times', and she said:

> Part of the exercise in the first place, of getting people to give us context is that as soon as you have to pose it and ask the questions and give us information, half the time you are going to realize or come up with an idea towards answering your question in the first place. Part of that [the asking for context] is actually to just get people to answer their own questions.

While data engineers do not generally talk with each other directly about their searching practices[89], their conversations with each other and other colleagues shape and scaffold, or structure, their searching.[90] Mokros & Aakhus (2002) proposes conceptualizing information-seeking behavior as "socially grounded" *meaning engagement practice*. They write how an information need, here an impetus to search, is "generated through social connection and the circumstances that arise from engagement with others and efforts to realize or avoid such engagement" (p. 309). A search can be renewed or refined by the prompt of an interaction. This may be the partially drafted question in a Slack chat interface or glancing at the clock or Zoom icon and anticipating your next standup or one-one-one is a socially grounded impetus to search.

[89] See Admitting searching: [don't talk about it]
[90] See Extending searching: immersed in linked conversation.

This is particularly the case for those who have received, perceived, or fear negative attention for their questions. These individuals may search excessively, rather than risk censure. Interviewees shared that even *starting to ask* a question helps them see their problem anew or see a new search to do. Orr (1996) describes technicians' work of diagnosis. Orr, himself referencing Suchman (1987) regarding when "transparent activity becomes in some way problematic" (p. 50), writes:

> Becoming problematic" may mean that the activity has been disrupted by failure, that one is perplexed about how to proceed, *or merely that someone else has inquired about the activity, requiring an explanation.* [emphasis added]

This 'starting to ask' introduces another person with the potential to inquire about the search. Interactions with others, including the 'merely' anticipated and imagined indirect relations, prompt activity to become problematic.

I can only speculate on the internal mechanisms involved.[91] Perhaps by becoming problematic, the problem is shifted out of the automatic search solution space. Rather than jumping to search, to try incantations to teleport to the solution, imagining or mimicking interacting with others calls upon conversational repertoires that may be well suited to aid reflection. This also may be as simple as starting to ask a question suggesting a framework for due diligence. The reliance on search may induce a sort of automation bias, a bias-to-search that frames problems in delimited ways. Alternative framings, or repertoires, includes preparing to address likely replies like 'what are you trying to do?'[92], what have you tried¿, or 'what do you know?'.

5.2 Discussion: Visibility, friction, and valuing privacies

5.2.1 Visible

Search repair practices provide a stage for performances that do four key things: legitimate the question asker and the answerer, legitimate web search reliance, enable participatory learning, share knowledge and who-knows-what.

First, the data engineers openly jockey within search repair to demonstrate to each other

[91] Being part of a transactive memory system may be involved. For instance, Ferguson et al. (2015)'s research showed people with access to the internet when asked questions reported a lower "feeling-of-knowing" than people without access. This might suggest that the moment someone decides to stop searching and start to ask a colleague they may shift into a different way of assessing their knowledge and gain confidence enabling a renewed search. Smith & Rieh (2019) provides an overview of several studies showing how web search is used as a transactive memory.

[92] Some in coding-related work refer to this as the XY Problem (*What Is the Xy Problem? - Meta Stack Exchange*, 2010).

what they know. This is one way they present, construct or perform, their legitimacy and expertise. This jockeying secures or shifts status claims. How they ask & answer questions about search provides another place for narratives where data engineers show and shape their selves (and judgements of competence and expertise) and their work. Feldman & March (1981) write on "decision making", but search repair functions similarly (pp. 177-178):

> decision making in organizations is more important than the outcomes it produces. It is an arena for exercising social values, for displaying authority, and for exhibiting proper behavior and attitudes with respect to a central ideological construct of modern western civilization: the concept of intelligent choice.

Navigating failed searches in these public ways (and the work of search confessions) "provides a ritualistic assurance that appropriate attitudes about [searching the web] exist" and provides opportunity for "a representation of competence and a reaffirmation of social virtue" (p. 177). Following Feldman and March, Leonardi (2007) writes that "visible aspects of information [. . .] practices are used as implicit measures of one's ability to make an informed decision" and "certain individuals come to have more power in the decision-making process due to the perceptions of their information-acquiring practices" (p. 814). The repair practices provide this opportunity.

Second, the reliance on web search is also reaffirmed, further legitimated. Though sometimes a question may be answered by directing the searcher to an internal search tool or other resource not available on the open web, the use of web search itself is not made a cause for concern. The search repair practice joins search confessions in providing informal and implicit approval for searching the web for work.

Third, Search repair provides a place for participatory learning. At first the data engineers may only lurk and watch, gradually seeing how others ask and answer questions. Then they try asking questions and getting feedback. In the process, they may learn how their peers frame questions and problems, identify resources and search strategies, and position themselves to evaluate. These are not facts and verbalized procedures that they learn, so much as how to be a data engineer who searches to do their work. As McDermott & Lave (2006) wrote (p. 108):

> The point: the *product* of laboring to learn is more than the school lessons learned. Over time, laboring to learn produces both what counts as learning and learners who know how to do it, learners who know how to ask questions, give answers, take tests, and get the best grades. Making what counts and making those who seek to be counted, these together compose the product of learning-labor. [emphasis in original]

The visibility of the work involved in these repairs, the stories, and the visible judgments

(though the judgments are not always visible) create learning opportunities for those providing help as well, providing space for practice, and get implicit feedback on, problem solving and communicating (Perlow & Weeks, 2002).[93]

Fourth, the repair practices allow the engineers (including those asking questions) to make knowledge and who-knows-what visible. This is beneficial for coordination and collaboration within and across teams—coordination a core requirement for this engineering work (Aranda, 2010). It also addresses a concern about the solitariness of web search. One might view web search practices as wasteful.[94] People repeatedly search for the same thing that someone else in their team had searched. But the speed and ease of most web searches makes it possible to work without very much interruption (of oneself or others) and reduces the requirement for sharing knowledge. Then the difficult searches that are visibly repaired become opportunities for sharing how to search (mentioned just above, perhaps making more searches faster and easier), sharing some of that difficult to find information, and sharing who knows what across a team or organization.

5.2.2 Friction

Even when questions are successfully received, and the data engineer isn't found to be lacking, asking questions produces what Tsing (2005) calls "friction". Tsing writes: "Cultures are continually co-produced in the interactions I call "friction": the awkward, unequal, unstable, and creative qualities of interconnection across difference" (p. 4). The mismatches in conceptions and communicated understandings, produces friction that both further reveals the edges of knowledge and creates opportunities to see and shift or shape how those edges are engaged with.

These sites for repair are components in the larger system of the work of data engineers. Within these sites the data engineers engage with each other in various ways and the work of repair as a whole engages engineers, prompting the visible jockeying as well as perceptions of how much more there is to know. This space for friction challenges the data engineers move beyond rigid evaluations of what they know as they work at the edge. Brown & Duguid (1998) write of "productive tension" from Hirschhorn (1997), and "creative abrasion", from Leonard-Barton (1995). Brown & Duguid (1998) write that such tension or friction can generate knowledge. For the purposes of the data engineers, it generates new search seeds (or identifies them) and evaluations of search results—supporting the search performance of the

[93]There is a stream of research on 'advice networks' (see Leonardi (2007) for an overview, he remarks that "the research on technology and advice networks is sparse") that also discusses the role of status in asking and answering questions. See also Shorey et al. (2021) for a discussion of the "participatory debugging"—or "collaborative technical troubleshooting"—of young people in online environments.

[94]A data engineer, Christina, raised this concern in a member check.

organization. Girard & Stark (2002) write of how "new media projects" at the start of the 21st century constantly reinterpreted their provisional "project settlements" [p. 1947]:

> the unsettling activity of ongoing disputation makes it possible to adapt to the changing topography of the web across projects in time. Friction promotes reflection, exposing variation from multiple perspectives.

A similar space for reflexive re-engagement around problem solving and search strategies appears around the search repair practices.

A rigidly technocratized approach to search might redistribute friction to places less visible or with slower feedback loops, some friction in the repair work, from the different ways of problem solving and searching, may improve the search performance of the organization. Passi & Jackson (2018) cites Stark (2009) in describing functional benefits of friction, writing "actors use diverse 'evaluative and calculative practices' to accomplish practical actions in the face of uncertainty" (p. 4). The friction comes from multiple forms of valuation—of searching, problems, and solutions—, developing contextually in different situations. Conflict, Passi writes, is integral to organizational diversity in which the "productive friction" between multiple ecologies of valuation helps accomplish justification and trust—(dis)trusting specific things, actions, and worlds" (p. 4). Stark (2009) also calls this "generative friction" (p. 16, p. 19), explicitly building on Leonard-Barton (1995) and Brown & Duguid (2001).

5.2.3 Valuing privacies

Successfully navigating failed searches is entangled with privacy[95]. The data engineers do not, generally, ask for details of the question-asker's search history. They do not force the question asker to show their recent browsing history. Both of those things, in the right setting, may greatly facilitate question answering. But the data engineers value privacy in searching and search repair. The valuing of privacy is an active result of visible practices. New data engineers see and participate in such valuing. This furthers their learning how to be, and search as, a date engineer.

This respect for privacy includes a range of orientations or conceptions of privacy. I will not here analyze the privacy dimensions (Mulligan et al., 2016) involved in the search repair practices, but only suggest three broad areas relevant to search repair:

- avoiding costly knowledge[96]

[95] A desire for secrecy in search practices is discussed in the next chapter, under Secretive searching.

[96] There is much to learn that only be learned at an unwelcome cost of the learner's time and attention. Bhatt & MacKenzie (2019) say it can be good practice to be "highly selective in the things we know or seek to know in order to remain epistemically functional, particularly now that most of us are almost exclusively immersed in information-dense digital environments" (p. 306).

- privacy allows each individual some autonomy in navigating the failed searches, rather than a surveillance that dictates the values of various knowledge from the top.
- there is also confidentiality at more of the occupational group level, privacy conceals the workings of their expertise

The data engineers I interviewed valued the lack of direct scrutiny of search practices. This was most visible in the strong objections and resistance to imagining sharing their search history. This privacy over the search activity is not only a matter of respecting or protecting web search as a place where they perform their expertise and professional judgement. Data engineers recognize that web search is where they regularly assemble and construct their expertise and professional judgement.

Data engineers value privacy here because, in addition to the individualizing mythos of search and also of coding work (to be discussed further in the next chapter), this is a way of confronting what they do not know and navigating around failed searches—their inability to know. In that, privacy in the search practices protects data engineers from likely judgements of their knowledge, expertise, or skill, and affords the space to experiment and grow. Judgments of visible or perceived ignorance or lack of skill already lead to mistreatment and poor judgment of performance in evaluations. Revealing more details of searching practices, baring other changes in the conceptions of search and expertise, may lead to harsher penalties to some. The sites of search repair already involve a significant amount of openness and sharing, a place sometimes safe but often including conversation. There would likely be significant unknown tradeoffs from mandating more openness, as discussed by Turco (2016) and Star & Strauss (2004).

The sociomaterial privacy of the web search practices of the data engineers makes space for "the play of everyday practice" at the edge of their knowledge, as "the motivating force behind creative practice, subject formation, and material practice" and a capability that promotes, with due consideration of other workplace factors, human flourishing (Cohen, 2012). Building on Cohen, Ohm & Frankle (2018) analyze a design principle, to make space for or to protect human values, that they call *desirable inefficiency*.
Certain aspects of the existing configurations of data engineering work highlight apparent inefficiencies or frictions that provide opportunities for the data engineers. This include not just the broad values of some sense of privacy and autonomy, but to learn more deeply than they might if principally learned through directives rather than searching and experimenting or to serendipitously discover in their web search navigation.

As discussed above, asking and answering questions promotes partially shared understandings and provides opportunities for jockeying (as well as the larger reproduction of workplace norms and values). The visibility mentioned above is fairly distinct from "forced representation." Star & Strauss (2004) write that "Many studies over the years have cautioned that forced

representation of work (especially that which results in computer support) may kill the very processes which are the target of support, by destroying naturally-occurring information exchange, stories, and networks" and "it should be clear that we are not recommending "more visibility"" [p. 24]. Cohen highlights the value of gaps[97] in the space for play and that "maintaining those gaps requires interventions designed to counterbalance the forces that seek to close them". Ohm & Frankle following her, present a gap imposed "in pursuit of fairness" (p. 826). This play and gaps are also relevant to the discussion of technocratization, as data engineers' work and web search practices challenge forces such as the informating affordance of information technology and an apparent data imperative.

5.3 Conclusion: Learning from search repair

The search repair practices provide visibility and friction useful for the learning organization. They respect various privacies. These three all support the learning of data engineers to be data engineers reliant on search, providing examples and spaces to participate.

The search repair practices also do the articulation or repair work that sustains reliance on web search. The organization can push data engineers to search, and rely on that despite not training them in search itself because they are immersed in search repair's collaborative problem solving. Search repair is practiced across multiple components within the data engineers' configuration of work and is itself a key factor in their successful reliance on web searching.

Search repair takes various shapes but could be performed differently. The packaging of a portion of due diligence could be automated, or have a required form. Questioning and answering could be logged by management for performance evaluations. Other incentives or constraints could be introduced with the goal of improving some facets of search repair. Some changes like these might improve the rate of search repair, but perhaps at the cost of limiting the opportunities for data engineers to demonstrate and test their knowledge, share their knowledge across the organization, or challenge accepted knowledge. Or at the cost of data engineers doing less searching on the fly or being less willing to reveal what they do not know. Much more than search repair happens in these interactions. If some aspects of the search repair were compelled or the perceptions of constraints and affordances were distorted these multiple values for the learning organization may no longer be achieved.

As it is, each data engineer within the various organizations engages differently with the sites of search repair. Some may not perceive the benefits of the visibility and friction because of where they are situated. Some may have different tradeoffs for the risks of revealing

[97] Cohen uses "semantic discontinuity" to refer to "gaps and inconsistencies within systems of meaning, and to a resulting interstitial complexity that leaves room for the play of everyday practice".

something they do not know, because they are already under heightened scrutiny, perhaps because of their minority status within the field. While others are able to use their experience and vantage of seniority to use the sites of search repair to demonstrate what they see as their expertise.

6 Owning searching

Outside the search confessions, the broader practices that support web search, and the shared repairing, the data engineers report experiencing and discursively defend web search as a solitary and private professional endeavor. Why does search remain a private and solitary practice given the rich trace data generated by search, the rapacious appetite for data collection and analysis among the data engineering community and the companies in which they practice? And what allows this community to avoid the scrutiny generally deemed essential to self-reflection, optimization, efficiency, and innovation?

I will show data engineers describing their web searching as being solitary, speedy, and secret. I show two aspects of how data engineers talk about searching solitary, solo, or alone. First, describing search as though the activity of web searching is wholly their responsibility, to be completed by the individual without or apart from the support of colleagues—by oneself. Second, describing search as apart from others entirely, suggestive of an apparently autonomous individual—on one's own. Then I discuss two further descriptions: data engineers' interest in speed, in search being fast, and how data engineers talk about wanting to keep their searches private, that they would be embarrassed to share them.

Several frames from related literature suggest different interpretations of the solitariness and privacy of the data engineers' web search activity. I look at the design of search, constructed but now default expectations of (perceived) privacy, both "rugged individualism" (Ensmenger, 2015) and norms around generalized reciprocity (Coleman, 2012; Weber, 2004) in the coding professions. These all contribute to an understanding of the solitary and secretive searching. Then I ground my argument in work on the value of privacy for learning and on the learning strategies of the organization.

I show how the firm delegates the task and practice of search to such a degree that it has foregone ownership of searching. As a consequences the already marginalized suffer, from the current informality workplace web searching in data engineering and what that hides, organization has poorer learning at the expense of preserving the status quo.

6.1 Solitary, speedy, and secretive searching

6.1.1 Solitary searching

The data engineers offered different motives for solitary searching. Some reported being expected to search alone, while others reported needing, or wanting to search alone. The examples in the two subsections below detail two aspects of a standing repertoire for referring to web searching. The repertoire suggests particular orientations towards or understandings of web search. In [by oneself] I'll share examples of how data engineers talk about searching

as a responsibility to be performed apart from the immediate support or involvement of their colleagues. In on one's own, I'll show how they talk of search as apart from others entirely, where the standard references to the search activity do not bring to mind a recognition of any relation to others who might be preparing material to be found or to those who searched before, shaping the results they will see.

6.1.1.1 [by oneself] Data engineers spoke of searching, finding, teaching, or figuring by themselves, yourself, or oneself. Sameer said he'd carefully suggest searching to new engineers[98]: " 'hey, have you tried googling it because it seems like a very simple thing you can find yourself.' "

Devin discussed self-reliant searching as a key strategy for dealing with the diversity and complexity of the tooling environment for data engineering:

> you really have to . . . figure out those tools yourself.
>
> When somebody says to go and google it, to get that answer, they also expect you to know what is a good answer, what is a bad answer. [. . .] I think as a data engineer you are expected to know what's good and bad. So I don't think there is a problem with saying 'google it'.

While Aditya was hesitant to place much emphasis on being able to search, what he described fits well within this narrative that individual engineers are expected to be able to independently search for (or 'find') information:

> I wouldn't say that the expected skill is that you know how to search. Like, I think that's almost like not something someone focuses on. But that you can—teach yourself or find yourself the information to learn some skill and, like, how you do it could differ.

and learn independently and on the fly: "I expect everyone to have gaps and as the problem arises, we, basically say:"Teach yourself what you need to solve that problem."

These comments convey the role and perceived value of autonomy and individualization of search work in data engineering practice. It isn't just that the engineers are expected or allowed to search the web on their own, they need to. Moreover, they also must be able to choose means besides searching the web necessary to do their work. These descriptions of why search is solo and the value of it, are distinct from the practical and spatiotemporal factors that shape search as a solitary exercise described in Admitting searching). This language frames and justifies solitary, or more specifically autonomous searching, in terms of values—efficiency, flexibility, adaptiveness.

[98] This interview excerpt is also in Admitting searching.

6.1.1.2 on one's own Just above, I introduced a sense that the activity of web searching is responsibly completed by the individual, without or apart from colleagues. Now note here *how* that ostensibly solitary work is described. The *on [your/their/my] own* phrasings are suggestive of an apparently autonomous individual (absent relations to others)[99].

I asked Michael how he thinks through the process of considering whether to ask a colleague or manager a question when feeling unable to find some bit of information. I was asking here particularly not about internal tooling (as interviewees commonly spoke of that as something they'd ask questions about internally). He said most of them time he'd ask questions about internal tooling or project-specific components he was interacting with. But, regarding "general development" work (his language), he discussed a sense of individual responsibility he learned from his mentors, managers, and peers (what he also considered "a common practice within the industry").

> For more general development, development where it doesn't matter what the component is, its more of something that you as an engineer should be able to solve on your own.
>
> It is definitely different for other engineers, but from from mentorship and my managers, and other peers, it's more of try it out on your own, try to solve it the best you can, and keep searching, and spend all your resources first, and then go to your team if you're actually truly stuck and you can't figure out something. As an engineer don't depend too much on hand holding. But if I am stuck on something then I go to more senior technical leaders.
>
> And I say, 'OK, I can't figure this out can you help me debug this' and sometimes that works very well because they can see things that I didn't see or I skipped over, et cetera.
>
> I do think that is more of a common practice within the industry and just engineers in generally. Is sort of: Be a 'go getter' and try to solve it on your own
>
> And also discern whether or not you'll be able to do it quickly and then, if not, seek help, et cetera.
>
> But then that's obviously a problem because if you are trying to solve it on your own through web search, etc., you're also depending on verified, validated, quick

[99] This search work *on one's own* is related to the 'due diligence' that research participants described doing before asking colleagues for help (discussed in due diligence and packaging questions). You can turn to colleagues for help, generally after assuming responsibility for searching, and they may provide help but they are unlikely to assist in the search box or on the SERP. The web search activity consisting of interacting directly with the search engine is generally the responsibility of the solitary data engineer.

> results to come up where it would help you in a way where you could actually solve it on your own.

I'll flag here some of the language he used:

- "solve on your own"
- "try it out on your own"
- "try to solve it on your own"
- "trying to solve it on your own through web search".

He's describing an expectation set by others that he has adopted for himself. This is an expectation, or an evaluative criteria, I found throughout my interviews. Yes, data engineers are physically remote from each other. However, the language of the interviewees describe search as performed alone in a distinct way. They position themselves in dialogue with an inanimate corpus of material or perhaps the search engine, while in reality searches connect data engineers with searchers across the World Wide Web, and across time and space. Ironically, while they describe search being performed alone they often include many references to actual and anticipated interactions with other people who variously constrain, compel, or coach searching. These people may also be distant in time and space, but they are closer to the heart and mind then the searchers hidden behind the screen.

Sometimes the solitary phrasings were only used to indicate the lack of search talk.[100] I asked Shreyan: "Do you talk about searches with your co-workers?" He replied: "they would search on their own". There may be this larger solitary assumption underlying that remark, but it was principally addressing search talk.

Jamie used 'on my own' phrases to refer simply to not asking colleagues for help:

- "I will try to get as far as I can on my own"
- "I wanted to attempt to solve the problem on my own"

Similarly, when Jillian shared asking her colleagues questions, she said that while they are very forgiving and nice and would want her to ask questions, she generally does what she says they probably would *not* want her to feel obligated to do:

> find the answer first, try and figure it out on your own. And then ask if you're having an issue figuring it out. Or if you know you're not going to be able to figure it out.

There is a general obligation that data engineers will take responsibility to search first. Some of what Jillian was discussing was a generally noted tendency to be so hesitant to ask for help, or being so used to turning first to the search engine, that data engineers will get lost

[100] See in Admitting searching: search talk?.

in rabbit holes, searching repeatedly without forward progress. It is those long searches that Jillian's colleagues express an interest in avoiding. The "obligation to know", to have searched and to search, "exists in tension with the expectation of asking questions" (Reagle, 2016, p. 698). Some of this solitary searching is driven by anticipated negative interactions providing a renewed impetus to search (discussed in the last chapter). Again, while the *on [your/their/my] own* phrasing above may at the first level refer to the data engineer searching apart from their colleagues active and in-the-moment engagement or presence, they of course reveal deeper interconnection because web search engages with a *web*, a network of people and other actors.

Phillip, discussing not ever having asked a question on Stack Overflow, said he didn't know how long it would take[101], and: "I just try to figure it out on my own." (thought still making use of, among other resources, Stack Overflow and questions and answers from others).

Kari described the progression from a novitiate heavily reliant on others to being "fully functioning on your own". Kari uses "figure things out yourself" and "on your own" language in reference to searching the web. I asked her about talking about web search with newcomers to her organization. She described how she would underemphasize searching the web when onboarding new engineers onto her team (she's also referring back to a comparison when she said she encourages the data scientists asking her questions to search first, and contrasting with how she described her own search work practices).

> One of the first things that I'll tell people is, for search specifically, I usually say the opposite for the people I work with. Ask a ton of questions and don't try to figure things out yourself. I wouldn't say that to a data scientist or every person at the company. But I think it is good at the beginning to really support new coworkers. Make sure that they don't feel stupid for asking questions because it is expected that they don't know. We don't want people to get stuck in a hole, alone, and stuck on something... trying to figure it out by themselves when they can rely on people. In your first year at a company it is important to have support from everyone else and then you are fully functioning on your own and you're not really going to need that as much anymore.

These excerpts affirm (and, if they faithfully represent comments within the workplace, reproduce or reinforce) the individualizing mythos of autonomous searching and an explicit individual rather than interdependent assignment of responsibility. (This is despite the actual interdependence in web searching and the many references to others influence and shaping of searching.)

[101] An interest in speedy searching, next section.

6.1.2 Speedy searching

Data engineers value speed. Their descriptions of the performance and articulation of web search highlights this interest in speed. The data engineers spoke of web search as providing the quickest or fastest way to answer their question.

At the very start of the interview, Ross noted that in his field "there are so many technologies, languages, frameworks, software packages, that you can't know it all". This research was "really interesting" because, he said, "being able to quickly find information is key to my job."

Noah:

> There could be a, some docs, I know exactly what I want but the fastest way to get it is just searching and- and try to get the first result so I use it to get to- to navigate documentation.

Mentioned previously[102], Shawn spoke of quickly resolving problems by searching the exception message:

> And that will cover you very quickly in about 90% of the time. You usually find the answer. And you can try it out, run a couple quick tests on it and make sure it actually works. And then you can move on with your day.

John said that often times using web search he's looking for the "quickest way": > I'm trying to find the quickest waywhere I can see an example of someone else's code. [. . .] I just need to—quick—glance at something to find a typo or identify an error.

Later he said (discussing whether his simple queries were embarrassing):

> That's what I do. I'd rather search it than try the [SQL] query, see that it erred, read the error, debug it.I'd rather quickly look it up often times.

Phillip said he'd never asked a question on Stack Overflow because he's "not sure fast the response would be."

> So searching is sometimes faster than trying the solution out.

Data engineers value speed (or local efficiency) in their use of web search.

A comment from Nisha underscores this. She's responding to a question from me about whether she'll read through the documentation for a tool or library when she's stuck. She says instead she'd do web searches or reach out to the internal customer support team because that's "the fastest way". Reaching out to support is 'ask[ing] someone for help', but web search is not described that way:

[102]This interview excerpt is also in Running workable code.

> I knew that reading documentation, if I got stuck on a specific area, was not going to add much. So I would either do a Google search, or if that didn't help I would reach out to customer support. Scan through the documentation, obviously, do a search on the keywords in the documentation, orreach out to support because that's the fastest way of getting my information. Rather than, you know the general framework, right, when you're cycling, if a keyword search will not help, if you do not find anything in the documentation, reading through the whole documentation will just waste your time. It is better to just ask someone for help.

Searching speed may not only be about clock-time. John said he wasn't sure if the quick searches were actually saving time, that he could perhaps alternately try a couple syntax formulations and debug from this computer and it might be faster, but that searching "feels more straightforward".

Sometimes the speed of searching is linked explicitly with the ease. Here's Ajit:

> I do remember, oh, I did something similar. But instead of actually going back and looking into the code where this happened, it's much easier for me to just do a quick web search.It's "more straightforward" and "much easier", to do a quick web search.

The speed is partially necessary because of how they are expected to learn on the fly, or just in time—as Aditya said:

> I expect everyone to have gaps. And as the problem arises, we, basically—Teach yourself what you need to kinda solve that problem. Rather than let's proactively try to just fill gaps across the place. there are going to be gaps

6.1.3 Secretive searching

Searching is also secretive, or kept secret. There is a deep intimacy to searching the web that carries over to workplace web searching. Data engineers indicate a strong desire to keep their searching secret. This is also a motivation for searching alone.

As discussed throughout, there is limited talk about search. I asked Nisha if people ever talked about search. She just shook her head no, while laughing and smiling.

Recall Amar, who said, "it's kind of like 90% of my job to just look things up." He said:

> engineers—at least in my experience or at least within my team—will not explicitly discuss their process

I asked Shreyan if he talked about searches with his co-workers and he said it happens very

little, saying "I don't want to restrict their pattern of thinking[103]."

Jillian was a new data engineer when I first talked with her, only a few months on the job. In talking about embarrassment about what she thought was an excessive reliance on web search, she said: "I would assume that I am searching things far more frequently than my peers." About talking about searches, Jillian said:

> I don't think I necessarily talk about it with them. I feel like I try and hide. I feel like I know very little and try and hide that from my peers. I don't want them to know how little I feel like I know. Let's just say I wouldn't want them to see my search history of my coding related things.

In a member check a year later, she no longer thought she was searching more than others:

> I am realizing that whenever I ask questions to people who I deem as smart or intelligent, I'm now really realizing that the skills that they have are the ability to quickly search.
>
> They don't just sit and think through it to give me an answer. They immediately go to their computer, maybe it's searching code if it's clearly not something that's going to be on the internet, but if it's going to be on the internet, they're really googling it.
>
> It'd be a big task to turn the whole narrative of googling things being an embarrassing thing, maybe to being a very admirable task, but I do start to recognize that, okay, people that I want to be like are definitely just constantly looking things up.

But the collective narrative hadn't shifted.

> One of the worst things I could have imagined happening, for my job, would be if people could see my search history, because then it exposed all the things I didn't know.
>
> And I think the flip side of that is it would probably actually just expose maybe the things I'm going out and trying to learn and understand better, but you don't necessarily know.

[103]This last line from Shreyan may be read to suggest an explanation or justification for the secrecy around search and its distributed nature. There may be a desire to avoid 'resonance', a term to describe something akin to groupthink perhaps, that Beunza & Stark (2012) use to describe "a dangerous form of cognitive interdependence" where productive dissonance is disrupted by the lack of diversity in approaches. Distinctly different, when I clarified with Shreyan later in the interview, he referred to research on brainstorming that suggested it was most effective when people were able to generate ideas on their own rather than in a group, before critiquing them.

The secrecy around search shapes people's understanding of its use, driving misperceptions that lead to shame (Jillian definitely was not searching more than others) and hiding the value of searching the web and what it can support.

With the description of these three findings in mind, and before proceeding to the discussion of them, I will next describe what I looked for but did not find: technocratization of search.

6.1.4 Technocratization of search

While doing this research there was an ever-present question on my mind: are or might companies monitor and manage the web searching practices of their workers in order to improve performance? Would data engineers turn their skill at data analysis upon their own workflows, to improve their own work and potentially competitive standing, or to collectively optimize data engineering work? On the other end, might companies or collectives of professionals develop tools to share search learnings or regularize search strategies through more structured and automated means?

I imagined that I might find, or research such as mine might unintentionally encourage, surveillance or control of web searching. I was concerned with attempts to surveil or supervise search in ways that reduce the autonomy of workers. Surveillance or control might harm people, violate rights, limit learning and even undermine efficient use of search.

I use technocratization of search to group various practices that I anticipated I might find. By technocratization of search, I mean the intentional application of techniques to influence search practices. By its similarity to 'technocracy', I hoped to convey some notion of "rule by experts" through the development of surveillance or quantification, processes or routines, or built artifacts that might shape searching practices. I imagined this might include the automation of portions of the search process (perhaps the gap-bridging studied by Bailey and Leonardi, see below) or modification of browsers to constrain or encourage particular behaviors. I imagined technocratization of search as consisting of tools for logging searches, tools for facilitating searches from the editor, tools for warning someone about the length of a search session, or tools for removing some websites from search results.

While it is distinct from the sort of technocratization of web search I was looking for, some have speculated that code generation tools my serve as a substitute for web search in coding work. It is easy to find claims on social media of people saying that their use of a code generation tool has or will replace their use of web search for their coding work. The development to new code generation tools expanded rapidly during the course of my writing. While this was not the focus of the research, its relevance is undeniable. I have prepared an appendix reflecting on such tools: Appendix III. Code Generation Tools and Search.

The technocratization question comes amidst hopes and fears in regard to the political and

economic power of companies wielding data and computation (ex. futures of work oriented popular press books like 'Second Machine Age', 'Fourth Industrial Revolution') and research examining the underlying mechanisms possibly explaining company strategies (here, Fourcade & Healy (2017) re "data imperative" and Zuboff (2015) re "logic of accumulation") and looking at the introduction of new technologies into work processes (ex., the buffering and resistance detailed in Christin (2017)).

I initially drew on the language of *informating* from Zuboff (1988). Zuboff juxtaposed machines and automation (to automate) with information technology and informate. She argued that information technology "both accomplishes tasks and translates them into information."

> Information technology not only produces action but also produces a voice that symbolically renders events, objects, and processes so that they become visible, knowable, and shareable in a new way. [. . .] The word that I have coined to describe this unique capacity is informate. Activities, events, and objects are translated into and made visible by information when a technology informates as well as automates."

Zuboff argued that:

> [. . .] [W]hen the technology also informates the processes to which it is applied, it increases the explicit information content of tasks and sets into motion a series of dynamics that will ultimately reconfigure the nature of work and the social relationships that organize productive activity

In addition to attending to the potential collection of data on searching or application of that data to influence searching, I also looked for automation of portions of the web search activity.[104] A thread of research on bridging gaps between two technologies (Bailey et al., 2010; Bailey & Leonardi, 2015) and the imbrication of routines and flexible technology (Leonardi, 2011) describe situations where tools are sometimes created or adapted to improve coordination between people and their tools. Would I find gap-bridging in the data engineer web search practices?

The question can also be formulated as: Where is the data gathering (or the surveillance and informating, the logic of accumulation and the data imperative) and the gap-bridging to improve the labor process in the work of data engineers? I did not find such informating of

[104] A fuller treatment would engage more deeply with writings on the future of work, surveillance studies, and perhaps the quantified self-movement. Lianos (2010), for instance, could be read to suggest that the logic of what to do with accumulated data is not clear, writing "accumulated data do not necessarily amount to a plan and even less so to a totalitarian plan" (p. 72).

web search or accumulating of search logs. Why do I seem to find so little technocratization of web search?

It may be suggested that the search work of the data engineers is a tacit skill (see also the discussion of searching as tacit-knowing in Admitting searching: Talk about search), not easily explainable nor reducible to programmed instruction. My questions about the lack of technocratization, though, don't suggest or expect automation of the search work. Shestakofsky (2017)'s research identifies how "[i]n some instances, workers' tacit skills [give] them an advantage over computer code in performing *nonroutine tasks* [. . .] because people possessed competencies grounded in tacit knowledge that could not easily be programmed [emphasis in original]" (p. 387). A significant amount of the search work may be nonroutine, but I was not looking for the tacit knowledge of the data engineers to be automated by machine, rather for any intentional application of technique to influence the searching of the data engineers.

While it may be the case that the search activity of the data engineers is intuitive and inexplicable even if noted or remembered (Dreyfus & Dreyfus, 2005), the question here is not about the development of a rule-based expert system, but why responsibility has been completely handed down to the engineers, who in turn do not built tools to scaffold or reflect on their search activity. Hodgson (2001) writes that "Workers have always possessed some tacit and other skills beyond the reach of managerial comprehension" [p. 193]. But technocratization of search is also not developed by the workers themselves, workers well capable of developing logs of searches for their own reference or building tools to further scaffold their interactions with the search engines.

6.1.4.1 Informating and imperative accumulation by web search Web search engines are no stranger to data collection and application of such data to further their goals. Early researchers looked at how to use encoded-links to aid in the automation of the process of finding and accessing distributed content, leading to hyperlinks and the web itself. Then web search engines informated from that structure. At the first level, we can imagine three primary parties involved in web search. The searcher, the search engine, and those producing content to be searched for. Organizations of the latter two types have heavily sought control and profit through informating points where they interface with searching (with the lattermost using search analytics to change their behavior to increase the quantity or quality of web site visits). In the coding work under examination here there is a fourth party, the coding firm, that intentionally or not exhibits control around the searching practice (from the reasons to search to the reception of the results-of-search) of the searching coders. These lenses provide a way to explore workplace web search, to explore the pressures shaping the context of searching, the search engines, and the searched for content.

The data generated, or informated, as byproducts of the web search of data engineers can largely be seen put to use by providers of search engines and websites, with limited tangential use by individuals, their organizations, or shared in larger communities. Various researchers have identified that technology or routines may be shaped by organizational pressures such that information is produced to allow for better control of the work processes (Beniger (1986), Zuboff (1988) and (2015)) or in the belief that such information will prove valuable (the "data imperative" in Fourcade & Healy (2017)).

The data generated as byproducts of the web search of data engineers can largely be seen, besides in the continued development and maintenance of the systems that support web search, in the research and activities of the major search engines (i.e. research from Microsoft query log analysis which may perhaps "provides a pulse of what software engineers are searching for and what problems they face" (Bansal et al. (2019)) and Google's "Foobar"). It is also used by builders of websites (learning from the searchers, clickers, and lurkers Antin & Cheshire (2010)). (Three interviewees noted the use of such website analytics. Two interviewees described examples of using analytics to guide marketing or documentation—one a former data engineer-turned-evangelist for an open source tool for data engineers, the other also involved in developing tools for use by data engineers. One data engineer I interviewed formerly worked in search engine optimization and also attested to the value to websites in the search logs.) But I did not find any examples of the data engineers or their employers using search data for informating.[105].

6.1.4.2 Gaps I looked for bridging between technology gaps (Bailey et al., 2010; Bailey & Leonardi, 2015) or gaps between two technologies. Gap-bridging is where a new technology is introduced to connect a gap between two other technologies that previously the knowledge worker had to traverse manually. A simple example is if the result of a calculation from one piece of software has to be manually typed into another because they were not designed to interoperate. An engineer may then consider whether it is cost-effective to write new code to traverse that gap automatically, or at least without manually clicking and typing.

I imagined I might find examples of people who had written tools to transform an error message directly into a Google search. While such tools, or prototypes, exist, I did not find anyone building or using them. Or perhaps a tool to directly move from notes to a search[106],

[105]Except insofar as tooling or structure is added to the process of asking questions of colleagues, discussed in Repairing searching.

[106]For instance, I wrote in Python a simple a clunky plugin for myself for the Sublime text editor that takes a simple notation (the search tool code followed by the query in square brackets) and with a hotkey opens a browser tab with the search. So I can write g[search this] or bmail[search this] in my notes and my hotkey (i.e. Cmd+l+o) will open a tab in my browser to the Google or Berkeley Gmail, respectively, search for [search this]. This sacrifices some features of searching in the address bar or in the search box. For

or organizing queries from a search session. While many reported taking notes, they did not report integrated notes with the inputs or outputs of their search activity.

Interviewees also did not discuss monitoring or scaffolding, for example guardrails to structure or hints or timers, efforts to improve search.[107] The scaffolding discussed in Extending searching was developed to support the data engineering work as a whole, and it only incidentally embeds knowledge for successful searching. Those scaffolds were not an intentional application to influence search practices and I found no intentional supports for web searching.

One interviewee talked about how integrated development environments (IDEs), with the ability to provide reference information within the display, replaced some web searching. But the ability to reference documentation in the worker's coding environment without web search has been around since before web search. The use of man pages for documentation reference goes back to the 1970s (Dzonsons, 2011).

People have long been used to fill in gaps between technologies, and perhaps different people than those who would have been previously running a routine. This is discussed in work on heteromation (Ekbia & Nardi, 2014, 2017) and computational labor (Shestakofsky, 2017). I did not find individuals used to fill gaps in searching outside of that discussed in Repairing searching.

One interviewee[108] invited me to speak at a small workshop at their company. They shared with me that they'd spoken with their manager about our conversation and that it had already provided an opportunity to talking explicitly about expectations in searching and asking questions. A month later I presented initial findings in a small teleconference with members of the company. Following up with this individual I learned the earlier conversations and the presentation had spurred the use of a Slack channel for sharing about things different team members had recently learned or looked up.

> we fired up a channel that was a "no stupid question" sort of thing, and I think people have used that a lot more, like that's been clear that people use them, and that was kind of one of the like action items we proposed after that was post questions here, or even if you search something and you find out something dumb, we've also done like a lot more like today I learned sort of reflections, which I think has been helpful, we didn't used to really do that so much.
>
> [. . .]

example, I do not see suggested searches or auto-completed phrases from the search engine.

[107] Such monitoring or scaffolding does not fall under the rubric of gap-bridging discussed by Bailey & Leonardi (2015), but could provide support for navigating gaps.

[108] I am not using the pseudonym of the interviewee to reduce the likelihood of identification.

> I don't know how everyone's individual searches and what not have gone, but I can see that one of the things that we definitely did was the channels and those are way more active. There were five people in it before but no one was using them, and now it's pretty much our entire data group will post things, or when they like just do things wrong, they pretty much posted in there too, so it kind of morphed out of like a, 'finding answers' to also just like a more general like humility, I don't know like, this is a learning environment, I guess, but that was helpful, like that was great, I mean even if that was not like one of the intended aspects.

This is not the sort of technocratization of web search that involves logging search queries or technical tools to shape the searching itself, but it was an intentional application of technique to influence the wider search practices.

Despite the heavy-reliance on web search in data engineering, the heavy demands on the data engineers to learn, quickly and constantly, neither the engineers nor their employers seem to have intentionally applied technique to influence search practices. Remarkably, at least to me, data engineering workplaces are free from the technocratization overtaking many other workplaces.

6.2 Discussion: Delegating or foregoing ownership

What is going on? Why is search, so heavily used, so very much admitted into the work practices, still so solitary and secretive? Why does it appear as though technocratization of search is absent?

6.2.1 Initial explanations

Single-user design of web search

Prior work has discussed the single-user design of web search. Morris & Horvitz (2007) developed a prototype for collaborative web search, SearchTogether. They based its design on a survey suggesting that people (Microsoft employees were surveyed) want to collaborate "with friends, relatives, and colleagues when searching the Web" and many already engaged in collaborative searching behaviors (also called "joint searching" and "multi-user searching" by the authors). They noted that—in 2007 at least—"current Web search tools are designed for a single user, working alone." Morris & Teevan (2009), reviewing that prototype and two others in a review of collaborative web search, opened by remarking: "Today, Web search is treated as a solitary experience. Web browsers and search engines are typically designed to support a single user, working alone." Three years later, Morris (2013) wrote, "the features of the primary tools for digital information seeking (web browsers and search engines) continue

to reflect a presumption that search is a single-user activity." But the design of the tool—the "technical functionality"[109]—cannot be definitive, the people I interviewed are sophisticated builders and users of technology. The "latent structural constraints" (Surden, 2007), from the design of web search, creating some of the privacy in searching, are probably not the strongest determinants keeping data engineers from informating or programmatically interacting with their own searching activity. They recognize technologies as flexible (Leonardi, 2011).

Articulation of web searching as private

The expectations of privacy in web search may play a role. Despite competition and critique, Google has been very successful in articulating the use of its search engine as private. People recognize that their searches are sensitive and should be secured. Major media attention in the wake of the release of the AOL search logs in 2006 likely contributed to such a recognition.

Rugged individualism

A history of "rugged individualism" in software work likely plays a role (Ensmenger, 2015). Or, rather, the still active legacy of the manufactured perceptions of the role and responsibility of the individual during the 1960s and 1970s likely shapes some of the solitary and secretive searching. Ensmenger describes how the work was seen as involving "individual skill", "individual expertise", "individual programmers", and "individual ability". But it also was not wholly solitary work. Ensmenger also describes masculine competition and comparison as the stage for "display" (p. 59) or presentations of such skill. In their interactions, programmers would engage in "ritualized forms of competition" (p. 62) or otherwise find ways of "establishing dominance within the community hierarchy" (p. 57). This conceptualizing of skill as something owned by an individual and rituals that reinforce associated epistemologies and responsibilities persists in computing work. (Reagle, 2016) discusses the "obligation to know" and the performances of and for status or stature of knowers in his exploration of "geek knowing".

It may be that desires to perform appropriate individualism and an awareness of competitive comparisons to others, with both shaping norms around shaming, drives some of the dynamics around solitary and secretive searching. But this also is not determinative. It doesn't explain what exactly activity, knowledge, or skill might be concealed and what would be proudly and ritually revealed. Transparency in searching could potentially be the sort of artificial challenge that Ensmenger suggests provide opportunity for showmanship. Masculine competition could

[109] Nissenbaum (2011b) wrote of Adnostic, a system for privacy preserving targeted advertising:

> In the effort to gain a toehold for Adnostic, technical functionality is not the greatest barrier. We have found ourselves up against a cultural mythology of innovation, incredibly powerful in the context of the Internet and web.

be performed with displays of superior querying or tooling for searching or memorization in lieu of searching. The research from Ensmenger may actual increase the tenor of the question, as he also shows how despite the impressions, computing work is very social (p. 58):

> despite the stereotype of the computer person as individualistic and "disinterested in people," the computer center was a profoundly social space. [. . .] they were more than simply working alone, together. In practice, computer centers were abuzz with conversation and other forms of social interaction.

Generalized reciprocity and self-reliance

We can also look at another seemingly core tenet of computing professions: generalized reciprocity. Weber (2004), discussing open source and quoting from Constant et al. (1996), wrote (p. 140):

> Generalized reciprocity is a firmly established norm of organizational citizenship within this community. Contributing code and helping others is a sign of "positive regard for the social system in which requests for help are embedded," a manifestation of pro-social behavior observed in other technically oriented settings as well.

The solitary and secretive behavior might seem to stand in great contrast to notions of generalized reciprocity presented as a hallmark of open source and other technical communities. But reciprocity needn't imply radical openness (Turco, 2016). Search activity may simply not be included in the community norms around what could or should he shared. The gift giving might flourish while never encroaching on cherished conceptions of individual knowledge and backstage privacy for search activity.

The reciprocity in these communities exists in coordination with the rugged individualism. As Weber goes on to write (p. 145):

> The popular image of an open source hacker as a lone ranger emphasizes the self-reliant attitude that is certainly present but misses the deep way in which that self-reliance is known to be made possible through its embedding in a community. The belief is that the community empowers the individual to help himself.

We see the same in the Debian open source community chronicled by Coleman (2012). She observes that "hacker sociality alternates between communal populism and individual elitism" (p. 105). Coleman writes (p. 107):

> On the other hand, hackers often express a commitment to self-reliance, which can be at times displayed in a quite abrasive and elitist tone. The most famous token of this stance is the short quip "Read the Fucking Manual" (RTFM). It is

> worth noting that accusations or RTFM replies are rarer than instances of copious sharing. [. . .] These two poles of value reflect pervasive features of hacker social and technical production as it unfolds in everyday life. It only takes a few days of following hacker technical discussion to realize that many of their conversations, whether virtual or in person, are astonishingly long question-and-answer sessions. To manage the complexity of the technological landscape, hackers turn to fellow hackers (along with manuals, books, mailing lists, documentation, and search engines) for constant information, guidance, and help.

Coleman writes of the credo or values of openness, transparency, and access within the open source community. Providing content to be found by searches, including answering Stack Overflow questions, is part of this openness.[110] But while some of those answers do discuss how to search, the transparency generally seems to not include the search activity itself. Though there is significant reciprocity in Repairing searching and community work in Extending searching, the ostensible moments of searching seem to be treated differently. The activity at the search box and on the SERP seemed to be that portion of the work that people are expected to do on their own (and that people may interpret as protected—through mutually enacted secrecy (Seaver, 2017., p.5)—to do on their own, in the backstage (Goffman, 1956).[111]

6.2.2 Learning in organizations and in privacy

We can pull these all together—and address solitary and secretive searching and the absence of technocratization of search—with support from literature from organizational learning and privacy for learning. A body of research shows that learning (including organizational learning) may be facilitated by privacy (Bernstein, 2012). If you are expected to do a lot of learning, you might pursue a zone of privacy to avoid exposing what you do not know or to avoid being disrupted in your learning efforts.

Data engineers rely on web search because neither they nor their larger organization knows all they need to know. Web search is a tool for pursuing answers to definite questions and reliance on it is a strategy that affords significant flexibility—in so far as the extensions of search are well-aligned with the search problems in question. Thompson (1967) writes that "Uncertainty appears as the fundamental problem for complex organizations, and coping with

[110] Very few of my research participants made mention of providing content on forums such at Stack Overflow, whether asking questions or answering. I do not venture an explanation for this lack of reciprocity, though note that research like that from Antin & Cheshire (2010) indicate readers and lurkers still provide significant value to online communities. So these individuals can still be seen to contribute to their broader community through their website visits.

[111] Goffman (1956, p. 70): "the back region will be the place where the performer can reliably expect that no member of the audience will intrude."

uncertainty, as the essence of the administrative process" (p. 159). Part of that coping is through bounded rationality through their structures, "the bounding of rationality requires structural decentralization, the creation of semiautonomous subsystems." (p. 54; p. 161). Organizations pursue "dual searches for certainty and flexibility" (p. 150). At the higher levels of the organization, at the time of his writing, flexibility and slack was the focus. At the lower decentralized levels at the technical core, it was certainty.

Subsequent work, looking at the constant change and competition in 2000s era web design and new media, suggests the decentralization extends to the level of individuals—with flexibility also being pushed lower. Girard & Stark (2002) write of search as "generalized and distributed". Kotamraju (2002) writes of the expectation that the "flexible reinvented worker" will be constantly "keeping up". Neff (2012) writes how "work itself has been largely individualized" (p. 14) and how "individuals now bear most of the costs of flexibility and are responsible for activities previously thought of as within the purview of companies" (p. 14). Neff writes (p. 18):

> Considering the quick turnaround times on the development of computer applications, employees are expected, in the words of one programmer, to "hit the ground running" with continually updated skills, including new programming languages and familiarity with new technologies.

The organizational approaches to (and shaping) the dynamism of 2000s era web design and new media may be compared to the organization of work in data science, machine learning, and data engineering today. Avnoon (2021) writes of data scientists "intensive self-learning" to "keep[] abreast" and "the emotional stress caused by continuously attempting to keep pace with technological and theoretical developments." Data engineers I interviewed also remark on there being too much to know, constant change in tools and practices being stressful, and flailing around frustrated and desperate in the search bar.

My argument is that the solitariness and secrecy of search is a consequence of this organizational—and professional—response to uncertainty made within the context of the preceding norms of (1) searches as private, of (2) knowledge as something 'owned' and performed for status, and of (3) the boundaries of reciprocity. Perhaps data engineers do not grant that their searching might be technocratized, informated or automated, because of the challenge is would pose to their identity and ways of knowing. Perhaps they do not publicize and compete in, or share and share in, their searching because they need their *solitary* searching to remain private so they might present the appearance of rugged individualism. Transparency (and concomitant competition in a masculine culture) might shatter the illusion of individualism, meritocracy, genius[112], and challenge the systems that favor them.

[112]Perhaps the solitary and secretive searching preserves the appearance of genius, the esoteric. Ensmenger

Technocratized or collaborative and transparent searching might risk removing artificial barriers to the fields. As it is, the extensive solitary searching and demand for self-reliance may be an artificial barrier. Ensmenger (2015) argued that the "whiz kid" or "computer boy" identity (p. 65):

> provided programmers with many of the perceived benefits of professionalization: the establishment of barriers to entry to the discipline, the possession of a "monopoly of competence," and mastery over an esoteric body of knowledge. [internal footnotes omitted]

Various approaches to change may require a shift to explicitly viewing requests for help as a chance to connect and learn (Perlow & Weeks, 2002) rather than an interruption[113] and a sign of individual weakness. Technocratized or collaborative and transparent searching may also require addressing cultures of shame (or they could risk slowing learning as people hide or avoid searching). Turco (2016) demonstrates how surveillance can be imagined by the surveilled to be a form of access to managers a benefit (see also Stark & Levy (2018)).

Changing the technology as a way of changing the culture may be a "punctuating force" and could "disrupt an established social structure" (Leonardi, 2007) and "forced representation of work" (Star & Strauss, 2004) may have unanticipated consequences. The decision makers within the field may resist changes because they benefit from the flexibility that structurelessness provides (Freeman, 2013).

6.3 Conclusion: Uncertainty sink

Individuals are identified as responsible for their searching. This contributes to solitary and secretive searching. People still participate in the larger shared work practices, shown in Extending searching, and to a lesser degree newcomers may gradually participate to varying degrees in fixing failed searches, shown in Extending searching. This participation introduces

(2015) tells of John Backus, developer of FORTRAN, critically calling programming of the 1950s "a black art, a private arcane matter":

> While Backus did not intend this description to be complimentary—as an aspiring computer scientist he saw this reliance on individual ability and local knowledge to be demeaning—many other programmers saw this emphasis on personal creativity and esoteric skill as the source of their professional authority. To be a devotee of a dark art, a high priest, or a sorcerer (all popular metaphors used to describe programming in this period) was to be privileged, elite, master of one's own domain. It was certainly preferable to being characterized as a glorified clerical worker or a "mere" technician. [internal footnote omitted]

[113] Orr (1996) writes to the systemic approaches of the copy repair technicians making their work "interruptible": "Being systematic has the advantage of being interruptible" (p. 145).

new data engineers to the ways of working and searching as a data engineer. For many data engineers that is enough, they are able to jockey and save face as they ask questions to repair failed searches, and make friends or otherwise develop strong rapport with more experienced coworkers. Those relationships allow them to ask questions outside of the search repair channels and not suffer as much when their public questions are out of place. But others stay on the periphery, already marginalized in the larger community, they are judged more harshly for their lack of knowledge and made more responsible for searching on their own. How responsibility for searching is positioned creates this cyclic loop that keeps penalizing those on the outside and the responsibility for is distributed to individual searchers,

The status quo for the data engineers is devoid of technocratization (intentional application of technique to influence search activity). While searching is a collaborative endeavor, the data engineers' searches are solitary and secret. While some privacy for searches facilitates individual learning, the current balance of secrecy may limit organizational learning and limit effective inclusion in data engineering. Data engineers search "on their own" and conceal their learning work. Individual searchers act as an "uncertainty sink"[114], allowing the organization to act nimbly, assigning to the individual "flexible reinvented worker" (Kotamraju, 2002) the responsibility to maintain and improve their skills. But that responsibility is not governed or managed by the organization.

[114]Like a heat sink (or heatsink), which moves heat from a heat generating component to a medium that dissipates heat, some business uncertainty is addressed by the delegation of searching to individuals.

7 Conclusion

I close by taking a step back, shifting my lens away from discussing data engineers searching at work, and consider what lessons and provocations my findings offer for search more broadly.

Web search engines have been *granted too much power*[115] and searchers both claim and are assigned too much responsibility. The search box, SERP, and the results navigated are a small slice of web searching. Web search is made to work, where it is, not only by the technical mechanisms (the indexes, algorithms, and designs) and social construction (articulations or imaginaries) of the search engine—and not just by the searcher themself or the websites seeking to be found—but by interactions among the various components in the larger systems and the contexts in which search is situated.

Success in using search matters greatly, both for data engineers and for society broadly. The four analytical chapters point to how data engineers' practices stabilize norms about acceptable use and reliance on search; context contains knowledge that constrains and shapes search–in material (technical, language), organizational, and professional forms, including methods for addressing search failures; and their current practices are solitary and secretive as data engineers are made to bear responsibility for their searching practices. The partial success of search for data engineers is a product of its situatedness (or enrollment or embeddedness) within broader data engineering work practices. The relatively tacit modes of sharing knowledge about productive use of search in data engineering work have disparate implications for historically marginalized people. The do-it-yourself image portrayed by data engineers hides a deep web of taken-for-granted dependence on the knowledge of others.

Below I will present two further arguments developed across the chapters, then briefly review my core findings and arguments with attention to applications beyond data engineering, highlight the importance of my findings to our understanding of search and their broader implications, and end with an appeal for imagining search differently.

7.1 Two further arguments: search envelopes and gatekeeping in data engineering

While web searching for data engineering is generally put to effective use, I also make two arguments about limitations. Here I will organize concerns introduced throughout the chapters. First, the effective use of web search by data engineers cannot be expected to be as stable for searches outside the structuring provided by the data engineering work practices. Second, the maintenance of search as solitary and backstage reproduces uneven access to community norms and knowledge, limiting people's ability to make effective use of search

[115] Borrowing from Barad (2003)'s opening line (p. 801), "Language has been granted too much power."

(and gatekeep in a way that is at odds with efforts at changing the demographics of technical professions).

First, the limits to someone's or some community's success in searching depends on how resources around them are aligned or marshaled towards particular sorts of learning and doing. The effective search envelope is dependent on the knowledge of other people and artifacts. "Search envelope" refers to the operating envelope or performance envelope of the data engineer's model of search and the surrounding system that supports the searching.[116] Without the right search inputs, relevant evaluation, and environment for acting on search, searchers will struggle (or leave search sessions misinformed). While generally successful for their purposes, the effectiveness of data engineering search practices is limited by the ambiguity of the search confessions, the taken-for-grantedness of web search and the occupational, professional, and technical components supporting it, and firms' hands off approach to both search repair and responsibility for searching.

Efforts to improve search exclusively or primarily through improving search literacy of individuals, by building greater technical expertise in the mechanisms of search, are missing important ways in which search is made to work in contexts. My findings reveal that web search in data engineering is constrained by the imprecision of the confessions, situated searching supported and scaffolded by the extensions of search (in the query generation, space for evaluation, and decoupling), the repair work, and a possessive and shame-inflected approach to knowledge and ignorance. The limitation on successful searching is akin to an individual attempting to search or learn outside their "zone of proximal development" and without scaffolding from experts and the environment. This is related to notions of a firms' "absorptive capacity" (Cohen & Levinthal, 1990; Roberts et al., 2012), their "sensing routines" (Carlo et al., 2012, p. 870), or "range of adaptive behavior" (Woods, 2018, p. 435).

Data engineers and their organizations can effectively rely on searching because it is supported by occupational, professional, and technical components of their work practices. Their success is not the result of individuals' sophisticated technical knowledge of the mechanisms of search. Data engineers apply their domain expertise and draw from code, exception messages, and interactions with colleagues in developing search queries. A single data engineer learning an error-prone code pattern from a web search is likely to learn of the problem in their attempts to run code, in code review, or in testing prototypes. The findings from this research help identify the limits of those supports and pathways towards improving search (in data engineering and beyond). In a near irony, I propose data engineers might improve the effectiveness of their searching by learning not of the technical mechanisms of the search engines, but of the sociotechnical mechanisms supporting their own situated use of web search.

[116] This is introduced in a challenge identified in Admitting searching: Discussion: Opportunities and challenges in confessions, with reference to Woods (2018).

I hope that this might shift the language of confessions and other *talk about search* to better describe the networked and configured search successes, to perhaps keep attention on the role and limits of the search supports, shift assignments of blame in search articulation-and-repair work, and identify how to align judgment of search performance with an appreciation for even individual searches as a collaborative performance.

Second, not all of the data engineers are fully brought into success of the searching practices. As researchers look at factors that push people out of technology work, this analysis of the situated searching experiences of data engineers presents an interesting case study of interactions concerning people's status and inclusion within a workplace. I discuss how people describe the effects of a fear of being mistreated and misjudged because of how their peripheral position within the system shapes their performance and perceptions of their knowledge and ability, their skill and thus their responsibility. The general hands-off approach of management towards searching, informality around talk about web search, and lack of technocratization of search (intentional application of technique to influence search activity) produces "a way of masking power" (Freeman, 2013, p. 232) and maintains the "fiction of technological meritocracy" (Hicks, 2017, p. 16), providing illegitimate cover for reproductions of hierarchy.

A data engineer's expertise is a shared accomplishment. Data engineers' ability to function as knowing experts—to know and to be seen as knowing is a product of their situated searching with the support of occupational, professional, and technical components of their work practices. Power relations are maintained, in part, by the invisibility of the searching practices—the engagements of the system taken-for-granted—and the maintenance of expertise as an individual possession (rather than composed in networks (Cambrosio et al., 2013; Eyal, 2019)). These web search practices can be a barrier to new, and particularly already marginalized, data engineers' full participation in their workplace. These findings provide a partial "account[ing] for the normalization and production of systematic advantage" (Hoffmann, 2019, p. 910) and disrupt notions of "exclusive forms of technical expertise" (Hoffmann, 2021, p. 11).

7.2 Lessons for searching and further research

This research shows how data engineers have made web search work for them, with lessons for research and the teaching and design of search more broadly, and revealing in the process limitations to their organizational learning and inclusion. My study of a discrete group of people, data engineers working in companies across the United States, provides lessons for our understanding of search that are applicable to searching in other domains and communities. I conducted interviews and performed document analysis with the two analytical lenses of Handoff and LPP to see how search is learned and used and how that *how* impinges on core societal values—namely responsibility, privacy, and fairness. Iterative comparison to

literature and examples of searching in other domains and cases shaped my attention and analysis. This site—with data engineers technically sophisticated, well-resourced, and reliant on search—was selected in order to develop broadly applicable lessons. While it could be framed as a special case of broader interactional and organizational processes, looking back to work from Goffman (1956) and Thompson (1967), I focus here on the lessons and inspirations most closely related to the design and use of web search.

I will quickly review the core ideas, the findings and arguments, and then reflect on key takeaways.

This research tells a straightforward story of data engineers' success in making use of web search for work (though with a couple clear caveats and concerns raised along the way). It starts with the observation that data engineers are reliant on web search for their work, seemingly successfully, and present a potential 'best case' study for exploring the role of technical knowledge of web search mechanisms. Interviews reveal there is limited explicit instruction, discussion, demonstration, or collaboration in the moments of web search in data engineering. But search confessions legitimate their searching, shape norms of use, and direct others to also rely on web search. Data engineers' success in search is not because they know more about the technical mechanisms of search, but because their work tools and practices (and domain expertise) make search work for their work purposes (supporting query formulation, evaluation of results, and decoupling performance from search automation bias). The occupational, professional, and technical components, as extensions of web search, provide sites and activities for new data engineers to gradually increase participation in the search work of data engineers. Search repair practices provide data engineers both additional *talk about search*, further legitimating it within their work, and opportunity for the learning data engineers to participate in extensions of web searching. The search repair practices constitute articulation work necessary to support such heavy reliance on web search. With firms delegating responsibility for searching down to individuals as a strategic approach to uncertainty, individual data engineers identify themselves as responsible for their web searching. The firm (pursuing only the delegation and no further ownership of search) nor the data engineers have intentionally applied technique to influence the search practices themselves. Successful data engineering use of web search is constrained to the imprecision of the confessions, situated searching supported and scaffolded by the extensions of search, the repair work, and a possessive and shame-inflected approach to knowledge and ignorance. Not all of the data engineers are fully brought in to participate in the success of the searching practices.

Below I identify five takeaways, and for each a key observation and provocations for future research.

1. Web search in data engineering is continually re-legitimated, *in this case*, through talk

about search—search confessions and search repair work.

Identifying similar legitimation work, or resistance and denial, to web search in different situations will provide entry points for understanding relations of power and constructions of learning, knowledge, and expertise within a situated practice—beyond the questions of web search practices themselves.

Such examinations may include looking at participant interventions to shift the legitimation as well as the modes of engagement (targeting, for instance, perceptions of constraint, affordance, shame, or celebration). How are search confessions and search memes ("just google it", "google is your friend", "LMGTFY", "google knows everything until you have an assignment") enrolled in other settings? Search directives? Are there settings where web search has reached something closer to closure (Bijker et al., 1993)? Such legitimation work is more formal in some settings (such as in proscriptions of searching the web in classrooms (Haider, 2017)), but may appear in jokes, memes, or something similar to confessions elsewhere. In healthcare related contexts there is a meme that patients find printed on coffee mugs and posters: "Please Don't Confuse Your Google Search With My Medical Degree". The legitimation (or not) of web searching may not be discursive, but dictated in relations of access (Burrell, 2018; Haider, 2017; Robinson, 2009; Sundin et al., 2017). What are the calls for memorization rather than reliance on web search? Who is using their own search engine? Legitimation of googling in school work is heterogeneous (interviews expressed differing experiences). Does legitimation of web search reliance in schooling in some fields map onto the use of web search in the workplace? How is the legitimation in these various settings taken up by or against different people within the setting?

2. Web search is extended beyond the search box and the SERP.

This is not a new sort of claim, but following it provided visibility into the role of knowledge and space for participatory learning in this case. Identifying the extensions of search in different search situations may point to potential reconfigurations to reduce the dependence on individual search performance.

Are there search practices that distribute more aspects of searching (than query formulation and results evaluation) to other systems and people? How are seeds disseminated and found in other settings? Are there other search settings where evaluation of results can be done so decoupled from the search and the performance effects? What types of searches, in what settings provide the least space for evaluation and decoupling? How are searches for the various "Your Money Your Life" topics—topics that have "a high risk of harm because content about these topics could significantly impact the health, financial stability, or safety of people, or the welfare or well-being of society" (Google, 2022)—extended in different settings? How do different settings provide impetuses for search? Can seeds be disseminated

as an intervention?

3. Successful use of web search did not hinge on personal knowledge of the technical mechanisms of web search, *in this case*.

Identifying the extent to which the successful use of web search in different situations and search purposes depends on varying degrees of domain knowledge or personal knowledge of the mechanisms of search will provide entry points for understanding the embeddedness of expertise and decoupling.

How does this finding change if the non-work related searches of this group of people are also scrutinized? Under what conditions does personal knowledge of the technical mechanisms matter significantly in other cases of web search, search, or the use of other information technologies? How do we identify the boundary conditions? What specific advantages might knowledge of the technical mechanisms of web search provide in different situations? Do very high levels of domain or search mechanism expertise counteract search automation bias? How does frequency, variety, or urgency of searching interact with these questions? Social media is sometimes used as a substitute or complement for web search (Oeldorf-Hirsch et al., 2014; Shah, 2017) and the use of question-and-answer sites like Stack Overflow, Quora, and some subreddits (Gilbert, 2018) have been characterized as social search or "asymmetric collaborative information seeking" (Morris, 2013; Morris & Teevan, 2009). How is this used, not as a substitute or comparison, but to formulate or evaluate web searches?

4. Web search is entangled with notions of responsibility, both credit and blame, for (and possession of) knowledge, *in this case*.

The hiddenness of searching to protect people's status as knowing individuals, concealing their ignorance or the resources they employ to learn is found in many domains of searching. How could we encourage more sharing of searches while retaining the value of intimacy for learning, retaining the safety to search within cultures that still penalize the learner?

Might we draw on contextual integrity (Nissenbaum, 2011a)? Can we find examples where such sharing is practiced and effective? Is web search treated differently by practitioners who adopt lessons from human factors and safety science (blameless postmortems, no single cause of failure)? How does the TIL (Today I Learned) movement treat responsibility in search? How does the use of school or workplace "no stupid questions" channels treat responsibility in search? Are there elements of open source coding movements, with the tension between generalized reciprocity and individualized responsibility, like those studied by Weber (2004), Coleman (2012), Dunbar-Hester (2020) that have varying treatments of responsibility for knowledge? Koonin (2019)'s "Everything I googled in a week as a professional software engineer" is widely shared, has her professional experience, and that of those who also publicly shared (or share) their searching, shifted? How do teachers or livestreamers who code and

search live discuss the "obligation to know" (Reagle, 2016) and the assignment of credit and blame for knowledge? Tripodi (2022b) describes a "do-it-yourself" approach to search where propagandists appear to successfully convince people they need to think for themselves and that web searching fact-checking is the legitimate approach to that, all while feeding them search queries that constrain what they might find.[117] Does the mantle for searching that these people have taken on still maintain a backstage for the actual searching activity, or is it proudly shared?

5. Technocratization of web search, the intentional application of technique to influence search activity, did not make an appearance, *in this case*, but it could be pursued in multiple forms with varying effects.

Web search conceptions, activity, and the tools themselves are plastic and different occupational or organizational factors may produce very different findings and futures.

While the search activity itself of the data engineers I interviewed were neither surveilled nor intentionally scaffolded, there are tools for this, as well as search engines and tools that may be substitutes for general-purpose search engines. This includes tools to facilitate memorization rather than searching, to substitute for a subset of searches. While it may be easy to identify examples of workers who do have their searching activity formally surveilled, restricted, or nudged by management, how do they resist or adopt these constraints/affordances? How are people using large language models to influence different aspects of search activity?[118] Search engine optimization experts use a variety of tools to better understand searching behavior of potential customers, do they adapt those tools to their own everyday or YMYL searching activity? How can searchers, and designers supporting them, make or modify tools to better see, distribute, and accomplish the always unfinished and context-dependent job of the search user interface?[119]

> to aid users in the expression of their information needs, in the formulation of their queries, in the understanding of their search results, and in keeping track of the progress of their information seeking efforts.

7.3 Why search matters

How we search and how we think about it shapes not only what we learn but how we learn with and from others. My research findings suggest ways forward in a hypercompetitive (Weber, 2019) and information saturated world. Though "people are *supposed* to parse

[117] Compare the "Search for yourself" approach presented by Caulfield (2019a), mentioned above in Extending searching:Search seeds.

[118] See Shah & Bender (2022) for an overview of concerns and directions.

[119] Hearst (2009, p. 1) opens by outlining the job of the search user interface:

through" the "sheer abundance of information [emphasis added]" on their own (Burrell & Fourcade, 2021, p. 18), this research shows how we are often not alone and draws attention to where we might act to move towards greater solidarity. Questions about relying on web search in the workplace are not about deskilling or even reskilling so much as about what skills are, who is allowed or seen to have skills, and "the political disenfranchisement and dehumanization of those people who are categorized as unskilled" (Iskander, 2021, p. 256).

People will attempt to turn, in part, to web search—or some transformed variant—to understand their changing world, including climate emergencies and resettlement, wars and their rumors, future pandemics and new vaccines, changes in schooling and healthcare, increasing inequality, and myriad new technologies from robotics and gene-editing to artificial intelligence and green technologies. Web search will continue to be a battleground where democracy is defined and practiced. These and other "looming disequilibria" (Weber, 2019, pp. 15–17)[120] prompt potentially life-altering queries that people will navigate within their systems of searching, for better or worse. Who will make successful use of search? How?

7.4 Imagining searching

In this research I have pursued "provocative generalizability", I "attempt[] to move [my]] findings toward that which is not yet imagined, not yet in practice, not yet in sight. [. . .] rather than only understanding (or naturalizing) what is" (Fine, 2006, p. 100). This is "a normative orientation" (Liboiron, 2021, p. 154). The lessons from this dissertation can be put towards defamiliarizing (Bell et al., 2005) web search and joining others as they "make, unmake and remake the search engine" (Sundin et al., 2017), "imagin[ing] search with a variety of other possibilities" (Noble, 2018, p. 180).

Vaidhyanathan (2011) concludes his exploration of *Googlization* with a section titled "Imagining a Better Way", writing that "[t]he question is not whether Google treats us well but whether this is best we can do." We can make new search engines and more people can make better use of search through their practices and reconfigured contexts.

We can find ways to clearly legitimate effective web search practices, celebrating searchers rather than stigmatizing them. We can learn to distinguish effective search practices from those that are manipulated or poorly modeled and likely to misinform or fail to inform. While

[120] Weber (2019) defining "looming disequilibria" [pp. 15-16]:

> build-up of entropy and sometimes destructive energy conditions that are out of balance and that create dynamic tensions and frictions that won't remain in the state in which they are without significant compensatory counterforce [. . .] Disequilibria don't necessarily signal an imminent break of some kind, but they do represent a build-up of entropy and sometimes destructive energy beneath the surface.

we must seek knowledge of the mechanisms of web search engines in order to reshape or replace them, we can find places to search around the opacity. We could share habits and practices that are not constrained by lack of transparency on the part of the decisions of commercial web search engine companies or inherent in the systems they build. We can focus on building the knowledge for effective searching into our practices, tools, and environments. We can work on mobilizing and recognizing effective search seeds in different domains. We could focus on developing and calibrating our individual and collective ability to evaluate search results and our results-of-search. We can look for configurations of components that let us decouple from search automation bias. We might see more of our interactions as spaces where we participate in formulating and evaluating searches with others. We can spread the practices for search repair that connect and encourage people rather than cut them apart and tear them down. We can see the extensions of search and find or fashion our own techniques to scaffold our searching and refashion our search practices. We can address search gaps in ways apart from turning to automation. We can determine how to share our search activities in ways that are appropriately sensitive to the relations between people and their goals in different contexts. We can make the extensions and effects of search more visible. We can make talking about search less shameful. We can find more ways to search together. We can recognize how search is a shared performance and can be a shared responsibility.

8 Appendices

8.1 Appendix I. Research Participants

Table 1: Participant Table

Participant	role	industry	gender identity
Shreyan	Data Software Engineer	enterprise software	man
Shawn	Developer Advocate, former data engineer	open source data software	man
Noah*	Senior Data Engineer	media streaming	man
Sameer	Senior Software Engineer	computing technology	man
Raha*	Senior Data Engineer	media, entertainment	woman
Aditya	Senior Engineering Manager	enterprise software	man
John*	Data Scientist	apparel	man
Amar**	Software Engineering Manager	enterprise software	man
Ajit	Senior Data Engineer	retail	man
Kari	Data Platform Engineer	apparel	woman
Vivek*	Senior Data Scientist	social media	man
Jillian	Data Engineer	fitness software	woman
Charles***	Data Scientist	online marketplace	man
Phillip*	Data Engineer	enterprise software	man
Devin	Data Platform Software Engineer	healthcare	man
Arjun	Principal Software Engineer	enterprise software	man
Christina	Senior Technical Consultant	enterprise software	woman
Michael	Data Engineer	financial services	man
Jamie	Senior Data Engineer	web publishing	woman
Ross	Senior Site Reliability Engineer	web analytics	man
Victor	Senior Data Engineer	web analytics	man
Nicole	Executive	data analysis software	woman
Patrick*	Data Engineer	enterprise software	man
Nisha***	Director Of Data Services	enterprise software	woman
Lauren	Machine Learning Engineer	online marketplace	woman
Jane	Analyst	social media	woman
Logan	Analyst	nonprofit	man

Participant	role	industry	gender identity
Amy	Data Platform Engineer	financial services	woman
Megan	Senior Data Engineer	business intelligence	woman
Zayn	Data Engineer	real estate	man

Note: Currently listed are the roles and title at the time of interview. * indicates role/industry change since initial interview; ** changed companies around time of interview and was onboarding at a new company, we discussed their prior role, which is listed; *** two interviewees who spoke of experiences prior to transitions away from that work.). The last five individuals were new participant member checks.

8.2 Appendix II. Annotated Interview Guide

The questions in my interview guide were initially built around these five research questions developed for my prospectus:

1. how and why is search used
2. how is it imagined as useful for the purpose its enlisted
3. what limitations are identified and addressed (or not)
4. how do conceptions & practices of web search reconfigure work practices
5. how are reconfigurations both shaped by and reshaping responsibility/accountability for work processes across individual professionals and organizations

Most of my interviews were conducted with the principal guide on-hand being a list of topics (pasted at top of a document with extensive annotations though rarely referenced during interviews):

- initial reaction question (IRQ)
  - After finding it quite useful when interviewees shared their initial reactions, I added this as my starting question.
- role
- team
- search
- talk about search?
  - LMGTFY
- search fails/struggles
- ask people?
- support channels?
- documentation/intranet/enterprise search?
- feedback & code review
- notes & cheatsheets
- mentorship
- on-boarding
- any questions for me & reflections on the interview
  - these two questions were also quite useful in eliciting open-ended and unanticipated responses.

I also had a list of interviewing reminders (though not hard and fast rules) for myself that grew throughout the first several interviews:

- CONCRETE examples
- pauses are good
- don't talk too much

- "say more" & push
- one question at a time
- do not interrupt
- note/probe laughter & annoyance/frustration

8.3 Appendix III. Code Generation Tools and Search

The latest generation of plugins for IDEs that some people[121] suggest might replace web search are those that support code generation directly from comments written within the code. This is a variant of a larger class of tools, called code generation tools.[122] One such plugin is GitHub's Copilot, based on the OpenAI Codex model, itself based on the Generative Pre-trained Transformer (GPT) models from OpenAI. My interview research did not generally directly address GitHub's Copilot.[123] The data engineers I asked about it had not used it. I will not go into the technical mechanisms of these systems, other than note that they are designed to take a prompt and predict the most likely text strings to follow. If I type "Mary had a" into the OpenAI GPT-3 Playground[124], the system completes the nursery rhyme.

GitHub Copilot, free to GitHub verified students, teachers, and maintainers of popular open source projects[125], is trained on portions of the significant amount of code uploaded to GitHub, which raises legal, ethical, and security concerns.[126] can It perhaps be imagined as an advanced autocomplete. Rather than suggesting the completion of a function name or command in your code, the plugin will suggest an entire block of code, perhaps the entire function. When the user types a comment or a line of code, the plugin will suggest a completion. Some users appear to have been satisfied with these suggestions. GitHub wrote a blog post in July 2022 reporting on a survey of Copilot users combined with data on their

[121] I am referring to popular commentary on Twitter from software developers and data engineers.

[122] I have used an older tool, TabNine, based on an earlier generation of OpenAI's GPT, since the summer of 2019. It is installed in my text editor, Sublime Text, which I use for all of my writing and python coding (until early 2022 when I started using PyCharm from JetBrains to gain familiarity with the sorts of integrated development environments available to my interviewees). I have used it in my Python coding and in any writing that I've done. It runs locally on my machine and provides a several predictions to suggest an autocomplete for most any string that I type. In my prose writing it is particularly helpful for spelling suggestions and remembering the shorthand for inserting a citation. It has not replaced a significant part of my web searching.

[123] The first technical preview of Copilot was available in the summer of 2021, with the subscription service starting in the summer of 2022.

[124] https://beta.openai.com/playground

[125] I signed up for a free trial and started paying $100.00/year in September 2022. (While I am notionally a student, the GitHUb verification process does not recognize UC Berkeley's filing fee status.) I have found it at times both very frustrating and very useful. The coding I do is very different from that of data engineers coding within their company codebase and systems. My use gives me only limited insight and I rely mostly on the findings from my interviews on the larger practices of the data engineers and external commentary and research on Copilot.

[126] I will not address the legal concerns and ethical concerns here except to note the broad sort of concerns and that they have some (but not total) similarity to early conversations and legal suits regarding web search. The concerns include questions about the copyright protection for and licenses of the uploaded code as well as GitHub's terms of service, the power imbalance between GitHub, now owned my Microsoft, and many developers who have uploaded code, and the absence of profit sharing.

shown and accepted Copilot suggestions (Ziegler, 2022). They claim "[u]sing GitHub Copilot correlates with improved developer productivity". GitHub continues to publish reports along this line (Kalliamvakou, 2022), while there is also intense interest from external researching on the use and effects of GitHub Copilot.

External researchers are particularly examining security vulnerabilities (some in a manner similar to that of Fischer et al. (2017) and others mentioned in Extending searching). A team from NYU and the University of Calgary examined code suggested by GitHub copilot scenarios developed relevant to MITRE's "Top 25" Common Weakness Enumeration (a regularly updated list of significant software vulnerabilities). Across the 89 scenarios they had Copilot produce over 1,500 programs of which they found approximately 40% to be vulnerable (Pearce et al., 2021). They recommend that Copilot "should be paired with appropriate security-aware tooling during both training and generation to minimize the risk of introducing security vulnerabilities."

Security concerns, and such precautions, are also acknowledged by GitHub[127]:

> You should take the same precautions as you would with any code you write that uses material you did not independently originate. These include rigorous testing, IP scanning, and checking for security vulnerabilities. You should make sure your IDE or editor does not automatically compile or run generated code before you review it.

These suggested responses to vulnerabilities in Copilot mirror some of what I discussed in Extending searching around the evaluation of search results and the decoupling from search.

The plugins for IDEs, and the voice-search, free-text, as in the OpenAI GPT-3 Playground[128] and chat-based, ex. ChatGPT[129], language interfaces are user interface components that, in the language of Handoff, provide distinct engagements for interactions. One of the benefits

[127] https://github.com/features/copilot

[128] https://beta.openai.com/playground

[129] OpenAI released their GPT-driven ChatGPT for public beta testing on November 30th 2022. One of my interviewees sent me a screenshot of them using it, asking the chatbot "How do data engineers use google search for their job?"

> Data engineers can use Google Search to find information and resources related to data engineering. This can include finding tutorials, documentation, and other resources that can help them with their work. For example, a data engineer might use Google Search to look up the syntax for a particular SQL query, find a tutorial on how to use a certain data analysis tool, or search for information on best practices for data engineering.
>
> Additionally, data engineers can use Google Search to stay-to-date on the latest developments and trends in the...

of using a general-purpose search engine is the contestability and interrogatability, perhaps not of the ranking of the websites on the SERP, but of the results. Data engineer searchers can look at the websites where they find information to gain clues as to its provenance and trustworthiness. GitHub Copilot and ChatGPT are black boxed and do not currently provide access for that, though GitHub has announced future product changes that will allow some interrogation[130], beyond directly engaging with the system for alternatives.

The tool designers will likely continue to improve the tool, and IDEs and companies may adapt practices to pull in such untested code in way primed for effective testing. OpenAI and others continue to do research looking into the hazards posed by such tools (Khlaaf et al., 2022). Copilot does and will likely continue to replace some subset of searching done by some data engineers. And also just like promises of automatic programming in the past (Ensmenger, 2010), if it does lower the cost of programming it will likely only increase the demand for more programmers. The programming languages used by my research participants are far simpler to use and understand than even the "automatic programming" languages of the past, like FORTRAN and COBOL. The hard problems remain, how to use a tool to do something you or someone else wants.

A web search engine provides some means to find what others, not only the search engine, say about top ranking search results, in addition to viewing the source website. This is a capacity of the configuration of web search that is leveraged in misinformation research. Some search engines provide interface options to learn general information about a website, to, for instance, see if the website presenting itself as a news organization is identified by Google as one. Mike Caulfield's SIFT model for basic fact-checking practices has four moves: Stop, Investigate the source, find better coverage, trace the original context (Caulfield, 2019b). Those moves are not supported by the Copilot configuration itself. Users of Copilot and other such tools will still find it helpful to refer to web search, and turn to sources of search repair, for learning things they do not already know.[131] GitHub's investments in Copilot come alongside a significant redesign of their search platform, for searching for code within

[130]GitHub announced plans in November, 2022 to identify public repositories of code that contain matching code, of a particular character count (Salva, 2022).

[131]Google and other search engines already use large language models, such as BERT, as components in their search engine. These components generally are used to support the same configurations in the user interface of the SERP, a list of links with some rich features. Some search engines, like metaphor.systems a large language model trained on Hacker News posts(Metaphor, 2021), have adapted the search box to be a longer freetext field, but still provide a list of results. Another, Andi, operates as a chatbot that also provides a list of results. Other general-purpose search engines and search tools have also introduced distinct plugins on the SERP making use of generative AI. You.com, a general-purpose web search engine, has applications that provide generative AI tooling for code writing, short prose, and image generation from prompts directly within the search bar.

GitHub (GitHub, 2022)[132], suggesting a recognition from the creators of the tool that search will not be fully replaced.

[132]Significant investments into GitHub's code search were announced in December 2021 (Avgustinov, 2021). While my research participants reported searching GitHub, they were generally searching for particular issues posted to repositories for debugging.

Bibliography

Abbate, J. (2012). *Recoding gender: Women's changing participation in computing (history of computing)*. The MIT Press. https://mitpress.mit.edu/9780262304535/

Ames, M. (2019). *The charisma machine : The life, death, and legacy of one laptop per child*. The MIT Press.

Ananny, M., & Crawford, K. (2018). Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. *New Media & Society*, *20*(3), 973–989. https://doi.org/10.1177/1461444816676645

Andalibi, N., & Bowen, K. (2022). Internet-based information behavior after pregnancy loss: Interview study. *JMIR Form Res*, *6*(3), e32640. https://doi.org/10.2196/32640

Andalibi, N., & Garcia, P. (2021). Sensemaking and coping after pregnancy loss. *Proc. ACM Hum.-Comput. Interact.*, *5*(CSCW1), 1–32. https://doi.org/10.1145/3449201

Antin, J., & Cheshire, C. (2010). Readers are not free-riders: Reading as a form of participation on wikipedia. *Proceedings of the 2010 Acm Conference on Computer Supported Cooperative Work*, 127–130. https://doi.org/10.1145/1718918.1718942

Aranda, J. (2010). *A theory of shared understanding for software organizations* [PhD thesis, University of Toronto]. https://tspace.library.utoronto.ca/bitstream/1807/26150/6/ArandaGarcia_Jorge_201011_PhD_thesis.pdf

Avgustinov, P. (2021). *Improving github code search | the github blog*. https://github.blog/2021-12-08-improving-github-code-search/.

Avnoon, N. (2021). Data scientists' identity work: Omnivorous symbolic boundaries in skills acquisition. *Work, Employment and Society*, *0*(0), 0950017020977306. https://doi.org/10.1177/0950017020977306

Badaracco, J. (1991). *The knowledge link: How firms compete through strategic alliances*. Harvard Business Press. https://archive.org/details/knowledgelinkhow0000bada

Bailey, D. E., & Leonardi, P. M. (2015). *Technology choices: Why occupations differ in their embrace of new technology*. MIT Press. http://www.jstor.org/stable/j.ctt17kk9d4

Bailey, D. E., Leonardi, P. M., & Chong, J. (2010). Minding the gaps: Understanding technology interdependence and coordination in knowledge work. *Organization Science*, *21*(3), 713–730. https://doi.org/10.1287/orsc.1090.0473

Bansal, C., Zimmermann, T., Awadallah, A. H., & Nagappan, N. (2019). The usage of web search for software engineering. *arXiv Preprint arXiv:1912.09519*.

Barad, K. (2003). Posthumanist performativity: Toward an understanding of how matter comes to matter. *Signs: Journal of Women in Culture and Society*, *28*(3), 801–831.

Barley, S. R. (1986). Technology as an occasion for structuring: Evidence from observations of ct scanners and the social order of radiology departments. *Administrative Science Quarterly*, *31*(1), 78–108. http://www.jstor.org/stable/2392767

Beane, M. (2017). *Operating in the shadows: The productive deviance needed to make robotic surgery work* [PhD thesis]. MIT.

Beane, M. (2019). Shadow learning: Building robotic surgical skill when approved means fail. *Administrative Science Quarterly*, *64*(1), 87–123. https://doi.org/10.1177/0001839217751692

Bechky, B. A. (2003). Object lessons: Workplace artifacts as representations of occupational jurisdiction. *American Journal of Sociology*, *109*(3), 720–752.

Bechky, B. A. (2006a). Gaffers, gofers, and grips: Role-based coordination in temporary organizations. *Organization Science*, *17*(1), 3–21.

Bechky, B. A. (2006b). Talking about machines, thick description, and knowledge work. *Organization Studies*, *27*(12), 1757–1768. https://doi.org/10.1177/0170840606071894

Bell, G., Blythe, M., & Sengers, P. (2005). Making by making strange: Defamiliarization and the design of domestic technologies. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *12*(2), 149–173. https://doi.org/10.1145/1067860.1067862

Beniger, J. (1986). *The control revolution: Technological and economic origins of the information society*. Harvard university press.

Bernstein, E. S. (2012). The transparency paradox: A role for privacy in organizational learning and operational control. *Administrative Science Quarterly*, *57*(2), 181–216. https://doi.org/10.1177/0001839212453028

Beunza, D., & Stark, D. (2012). From dissonance to resonance: Cognitive interdependence in quantitative finance. *Economy and Society*, *41*(3), 383–417. https://doi.org/10.1080/03085147.2011.638155

Bhatt, I., & MacKenzie, A. (2019). Just google it! Digital literacy and the epistemology of ignorance. *Teaching in Higher Education*, *24*(3), 302–317. https://doi.org/10.1080/13562517.2018.1547276

Biden, J. (2022). *I know covid testing remains frustrating, but we are making improvements.* https://twitter.com/POTUS/status/1478761964327297026.

Bijker, W. E., Hughes, T. P., & Pinch, T. (1993). Introduction. In *The social construction of technological systems* (pp. 9–50). MIT Press.

Bilić, P. (2016). Search algorithms, hidden labour and information control. *Big Data & Society*, *3*(1), 205395171665215. https://doi.org/10.1177/2053951716652159

Birt, L., Scott, S., Cavers, D., Campbell, C., & Walter, F. (2016). Member checking: A tool to enhance trustworthiness or merely a nod to validation? *Qualitative Health Research*, *26*(13), 1802–1811. https://doi.org/10.1177/1049732316654870

Blackler, F. (1995). Knowledge, knowledge work and organizations: An overview and interpretation. *Organization Studies*, *16*(6), 1021–1046. https://doi.org/10.1177/017084069501600605

Bowker, G. C. (1994). *Science on the run: Information management and industrial geophysics at schlumberger, 1920-1940 (inside technology)* (Hardcover, p. 199). The MIT Press. https://sicm.mitpress.mit.edu/books/science-run

Bowker, G. C., Baker, K., Millerand, F., & Ribes, D. (2010). Toward information infrastructure studies: Ways of knowing in a networked environment. In J. Hunsinger, L. Klastrup, & M. Allen (Eds.), *International handbook of internet research* (pp. 97–117). Springer Netherlands. https://doi.org/10.1007/978-1-4020-9789-8_5

Bowker, G. C., & Star, S. L. (2000). *Sorting things out: Classification and its consequences*. MIT press. https://mitpress.mit.edu/books/sorting-things-out

boyd, danah. (2018). You think you want media literacy... do you? In *Medium*. https://points.datasociety.net/you-think-you-want-media-literacy-do-you-7cad6af18ec2

Brandt, J., Guo, P. J., Lewenstein, J., Dontcheva, M., & Klemmer, S. R. (2009). Two studies of opportunistic programming: Interleaving web foraging, learning, and writing code. *Proceedings of the Sigchi Conference on Human Factors in Computing Systems*, 1589–1598. https://doi.org/10.1145/1518701.1518944

Brown, J., & Duguid, P. (2001). Knowledge and organization: A social-practice perspective. *Organization Science*, *12*, 198–213.

Brown, J. S., & Duguid, P. (1991). Organizational learning and communities-of-practice: Toward a unified view of working, learning, and innovation. *Organization Science*, *2*(1), 40–57. http://www.jstor.org/stable/2634938

Brown, J. S., & Duguid, P. (1996). *Situated learning perspectives* (H. McLellen, Ed.). Educational Technology Publications. https://www.johnseelybrown.com/StolenKnowledge.pdf

Brown, J. S., & Duguid, P. (1998). Organizing knowledge. *California Management Review*, *40*(3), 90–111.

Bucciarelli, L. L. L. (1996). *Designing engineers* (Paperback, p. 230). MIT Press. https://mitpress.mit.edu/books/designing-engineers

Bucher, T. (2017). The algorithmic imaginary: Exploring the ordinary affects of facebook algorithms. *Information, Communication & Society*, *20*(1), 30–44. https://doi.org/10.1080/1369118X.2016.1154086

Burrell, J. (2009). The field site as a network: A strategy for locating ethnographic research. *Field Methods*, *21*(2), 181–199. https://doi.org/10.1177/1525822X08329699

Burrell, J. (2012). *Invisible users: Youth in the internet cafés of urban ghana (acting with technology)*. The MIT Press.

Burrell, J. (2016). How the machine "thinks": Understanding opacity in machine learning algorithms. *Big Data & Society*, *3*(1), 2053951715622512. https://doi.org/10.1177/2053951715622512

Burrell, J. (2018). Thinking relationally about digital inequality in rural regions of the u.s. *First Monday*, *23*(6). https://doi.org/10.5210/fm.v23i6.8376

Burrell, J. (2010). Evaluating Shared Access: Social equality and the circulation of mobile phones in rural Uganda. *Journal of Computer-Mediated Communication*, *15*(2), 230–250. https://doi.org/10.1111/j.1083-6101.2010.01518.x

Burrell, J., & Fourcade, M. (2021). The society of algorithms. *Annual Review of Sociology*, *47*(1), null. https://doi.org/10.1146/annurev-soc-090820-020800

Burrell, J., Kahn, Z., Jonas, A., & Griffin, D. (2019). When users control the algorithms: Values expressed in practices on twitter. *Proc. ACM Hum.-Comput. Interact.*, *3*(CSCW). https://doi.org/10.1145/3359240

Cadwalladr, C. (2016a). Google, democracy and the truth about internet search. *The Guardian*, *4*(12). https://www.theguardian.com/technology/2016/dec/04/google-democracy-truth-internet-search-facebook

Cadwalladr, C. (2016b). Google is not "just" a platform. It frames, shapes and distorts how we see the world. *The Guardian*, *11*(12). https://www.theguardian.com/commentisfree/2016/dec/11/google-frames-shapes-and-distorts-how-we-see-world

Cambrosio, A., & Keating, P. (1995). *Exquisite specificity: The monoclonal antibody revolution*. Oxford University Press.

Cambrosio, A., Limoges, C., & Hoffman, E. (2013). Expertise as a network: A case study of the controversies over the environmental release of genetically engineered organisms. In N. Stehr & R. V. Ericson (Eds.), *The culture and power of knowledge* (pp. 341–362). De Gruyter. https://doi.org/doi:10.1515/9783110847765.341

Caplan, R., & Bauer, A. J. (2022). *How would one write/call the period immediately after everyone-started-caring-all-at-once-about-disinfo era? [Conversation]*. https://twitter.com/robyncaplan/status/1522643551527579648.

Carlo, J. L., Lyytinen, K., & Rose, G. M. (2012). A knowledge-based model of radical innovation in small software firms. *MIS Quarterly*, *36*(3), 865–895. http://www.jstor.org/stable/41703484

Caulfield, M. (2019a). *Data voids and the google this ploy: Kalergi plan*. https://hapgood.us/2019/04/12/data-voids-and-the-google-this-ploy-kalergi-plan/.

Caulfield, M. (2019b). *SIFT (the four moves) | hapgood*. https://hapgood.us/2019/06/19/sift-the-four-moves/.

Chen, M., Fischer, F., Meng, N., Wang, X., & Grossklags, J. (2019). How reliable is the crowdsourced knowledge of security implementation? *2019 Ieee/Acm 41st International Conference on Software Engineering (Icse)*, 536–547. https://doi.org/10.1109/ICSE.2019.00065

Chen, X., Ye, Z., Xie, X., Liu, Y., Gao, X., Su, W., Zhu, S., Sun, Y., Zhang, M., & Ma, S. (2022). Web search via an efficient and effective brain-machine interface. *Proceedings of the Fifteenth Acm International Conference on Web Search and Data Mining*, 1569–1572. https://doi.org/10.1145/3488560.3502185

Christin, A. (2017). Algorithms in practice: Comparing web journalism and criminal justice. *Big Data & Society*, *4*(2), 1–12. https://doi.org/10.1177/2053951717718855

Christin, A. (2018). Counting clicks: Quantification and variation in web journalism in the united states and france. *American Journal of Sociology*, *123*(5), 1382–1415.

Christin, A. (2020a). *Metrics at work: Journalism and the contested meaning of algorithms*. Princeton University Press.

Christin, A. (2020b). The ethnographer and the algorithm: Beyond the black box. *Theory and Society*. https://doi.org/10.1007/s11186-020-09411-3

Cohen, J. E. (2012). *Configuring the networked self*. Yale University Press. https://doi.org/10.12987/9780300177930

Cohen, W. M., & Levinthal, D. A. (1990). Absorptive capacity: A new perspective on

learning and innovation. *Administrative Science Quarterly*, *35*(1), 128–152. http://www.jstor.org/stable/2393553

Coleman, E. G. (2012). *Coding freedom: The ethics and aesthetics of hacking*. Princeton University Press. https://gabriellacoleman.org/Coleman-Coding-Freedom.pdf

Collins, H. M., & Evans, R. (2007). *Rethinking expertise*. University of Chicago Press. https://press.uchicago.edu/ucp/books/book/chicago/R/bo5485769.html

Commission, F. T. (2013). *Sample letter to general purpose search engines*. https://www.ftc.gov/sites/default/files/attachments/press-releases/ftc-consumer-protection-staff-updates-agencys-guidance-search-engine-industryon-need-distinguish/130625searchenginegeneralletter.pdf.

Constant, D., Sproull, L., & Kiesler, S. (1996). The kindness of strangers: The usefulness of electronic weak ties for technical advice. *Organization Science*, *7*(2), 119–135.

Contu, A., & Willmott, H. (2003). Re-embedding situatedness: The importance of power relations in learning theory. *Organization Science*, *14*(3), 283–296.

Cotter, K. (2021). "Shadowbanning is not a thing": Black box gaslighting and the power to independently know and credibly critique algorithms. *Information, Communication & Society*, *0*(0), 1–18. https://doi.org/10.1080/1369118X.2021.1994624

Cotter, K. (2022). Practical knowledge of algorithms: The case of BreadTube. *New Media & Society*, 1–20. https://doi.org/10.1177/14614448221081802

Daly, A., & Scardamaglia, A. (2017). Profiling the australian google consumer: Implications of search engine practices for consumer law and policy. *J Consum Policy*, *40*(3), 299–320. https://doi.org/10.1007/s10603-017-9349-9

Dave, P. (2022). *Google cuts racy results by 30*. Reuters. https://www.reuters.com/technology/google-cuts-racy-results-by-30-searches-like-latina-teenager-2022-03-30

Dear, P. (2001). Science studies as epistemography. *The One Culture*, 128–141.

de Laet, M., & Mol, A. (2000). The zimbabwe bush pump: Mechanics of a fluid technology. *Social Studies of Science*, *30*(2), 225–263. https://doi.org/10.1177/030631200030002002

Deterding, N. M., & Waters, M. C. (2018). Flexible coding of in-depth interviews: A twenty-first-century approach. *Sociological Methods & Research*, 0049124118799377. https://doi.org/10.1177/0049124118799377

Dijck, J. van. (2010). Search engines and the production of academic knowledge. *International Journal of Cultural Studies*, *13*(6), 574–592. https://doi.org/10.1177/1367877910376582

DiMaggio, P. J., & Powell, W. W. (1983). The iron cage revisited: Institutional isomorphism and collective rationality in organizational fields. *American Sociological Review*, 147–160.

DiResta, R. (2018). *The complexity of simply searching for medical advice*. https://www.wired.com/story/the-complexity-of-simply-searching-for-medical-advice/

Doty, N. (2020). *Enacting Privacy in Internet Standards* [Ph.D. dissertation, University of California, Berkeley]. https://npdoty.name/enacting-privacy/

Dreyfus, H. L., & Dreyfus, S. E. (2005). Peripheral vision: Expertise in real world contexts. *Organization Studies*, *26*(5), 779–792.

Duguid, P. (2008). *Community, economic creativity, and organization* (A. Amin & J. Roberts, Eds.). Oxford University Press. https://oxford.universitypressscholarship.com/view/10.1093/acprof:oso/9780199545490.001.0001/acprof-9780199545490-chapter-1

Dunbar-Hester, C. (2020). *Hacking diversity: The politics of inclusion in open technology cultures* (1st Edition). Princeton University Press. https://press.princeton.edu/books/hardcover/9780691182070/hacking-diversity

Dzonsons, K. (2011). *History of unix manpages*. https://manpages.bsd.lv/history.html.

Ekbia, H., & Nardi, B. (2014). Heteromation and its (dis) contents: The invisible division of labor between humans and machines. *First Monday*. https://doi.org/https://doi.org/10.5210/FM.V19I6.5331

Ekbia, H. R., & Nardi, B. A. (2017). *Heteromation, and other stories of computing and capitalism*. MIT Press.

Elish, M. C., & Watkins, E. A. (2019). *Algorithms on the shop floor: Data-driven technologies in organizational context*. https://datasociety.net/library/algorithms-on-the-shop-floor/

Ensmenger, N. (2010). *The computer boys take over: Computers, programmers, and the politics of technical expertise*. The MIT Press.

Ensmenger, N. (2015). "Beards, sandals, and other signs of rugged individualism": Masculine culture within the computing professions. *Osiris*, *30*(1), 38–65. http://www.jstor.org/stable/10.1086/682955

Eyal, G. (2019). *The crisis of expertise*. Polity Press. https://www.wiley.com/en-us/The+Crisis+of+Expertise-p-9780745665771

Feldman, M. S., & March, J. G. (1981). Information in organizations as signal and symbol. *Administrative Science Quarterly*, *26*(2), 171–186. http://www.jstor.org/stable/2392467

Ferguson, A. M., McLean, D., & Risko, E. F. (2015). Answers at your fingertips: Access to the internet influences willingness to answer questions. *Consciousness and Cognition*, *37*, 91–102. https://doi.org/10.1016/j.concog.2015.08.008

Fine, M. (2006). Bearing witness: Methods for researching oppression and resistance—a textbook for critical research. *Social Justice Research*, *19*, 83–108.

Firouzi, E., Sami, A., Khomh, F., & Uddin, G. (2020). On the use of c# unsafe code context: An empirical study of stack overflow. *Proceedings of the 14th Acm / Ieee International Symposium on Empirical Software Engineering and Measurement (Esem)*. https://doi.org/10.1145/3382494.3422165

Fischer, F., Böttinger, K., Xiao, H., Stransky, C., Acar, Y., Backes, M., & Fahl, S. (2017). Stack overflow considered harmful? The impact of copy&paste on android application security. *2017 Ieee Symposium on Security and Privacy (Sp)*, 121–136.

Flick, U. (2009). *An introduction to qualitative research*. SAGE Publications Ltd.

Fourcade, M. (2010). *Economists and societies*. Princeton.

Fourcade, M., & Healy, K. (2017). Seeing like a market. *Socio-Economic Review*, *15*(1), 9–29. https://doi.org/https://doi.org/10.1093/ser/mww033

Freeman, J. (2013). The tyranny of structurelessness. *WSQ*, *41*(3-4), 231–246. https://doi.org/10.1353/wsq.2013.0072

Gasson, S., & Purcelle, M. (2018). A participation architecture to support user peripheral participation in a hybrid foss community. *Trans. Soc. Comput.*, *1*(4). https://doi.org/10.1145/3290837

Giddens, A. (1991). *The consequences of modernity*. Polity Press in association with Basil Blackwell, Oxford, UK.

Gilbert, S. A. (2018). *Motivations for participating in online initiatives: Exploring motivations across initiative types* [PhD thesis, University of British Columbia]. https://doi.org/http://dx.doi.org/10.14288/1.0372890

Gillespie, T. (2014). *The relevance of algorithms* (T. Gillespie, P. J. Boczkowski, & K. A. Foot, Eds.; pp. 167–193). The MIT Press. https://doi.org/10.7551/mitpress%2F9780262525374.003.0009

Gillespie, T. (2017). Algorithmically recognizable: Santorum's google problem, and google's santorum problem. *Information, Communication & Society*, *20*(1), 63–80. https://doi.org/10.1080/1369118X.2016.1199721

Girard, M., & Stark, D. (2002). Distributing intelligence and organizing diversity in new-media projects. *Environment and Planning A*, *34*(11), 1927–1949.

Gitelman, L. (2006). *Always already new: Media, history, and the data of culture*. MIT Press. https://direct.mit.edu/books/book/4377/Always-Already-NewMedia-History-and-the-Data-of

GitHub. (2022). *Introducing an all-new code search and code browsing experience / github changelog*. https://github.blog/changelog/2022-11-09-introducing-an-all-new-code-search-and-code-browsing-experience/.

Goffman, E. (1956). *The presentation of self in everyday life*. University of Edinburgh.

Goldenfein, J., & Griffin, D. (2022). Google scholar – platforming the scholarly economy. *Internet Policy Review*, *11*(3), 117. https://doi.org/10.14763/2022.3.1671

Goldenfein, J., Mulligan, D. K., Nissenbaum, H., & Ju, W. (2020). Through the handoff lens: Competing visions of autonomous futures. *Berkeley Tech. L.J.. Berkeley Technology Law Journal*, *35*(IR), 835. https://doi.org/10.15779/Z38CR5ND0J

Goldenfein, J., Mulligan, D., & Nissenbaum, H. (2019). *Through the handoff lens: Are autonomous vehicles no-win for users*. https://pdfs.semanticscholar.org/341d/a18649eb9627fe29d4baf28fb4ee7d3eafa3.pdf

Golebiewski, M., & boyd, danah. (2018). Data voids: Where missing data can easily be exploited. *Data & Society*. https://datasociety.net/library/data-voids-where-missing-data-can-easily-be-exploited/

Google. (2022). *Searchqualityevaluatorguidelines.pdf*. https://static.googleusercontent.com/media/guidelines.raterhub.com/en//searchqualityevaluatorguidelines.pdf.

Griffin, D. (2019). *When searching we sometimes use keywords that direct us...* https://twitter.com/danielsgriffin/status/1183785841732120576.

Griffin, D., & Lurie, E. (2022). Search quality complaints and imaginary repair: Control in articulations of google search. *New Media & Society*, *0*(0), 14614448221136505. https://doi.org/10.1177/14614448221136505

Gross, M., & McGoey, L. (2015). *Routledge international handbook of ignorance studies*. Routledge. https://doi.org/10.4324/9781315867762

Gross, M., & McGoey, L. (2022). *Routledge international handbook of ignorance studies*. Routledge. https://doi.org/10.4324/9781003100607

Guendelman, S., Pleasants, E., Cheshire, C., & Kong, A. (2022). Exploring google searches for

out-of-clinic medication abortion in the united states during 2020: Infodemiology approach using multiple samples. *JMIR Infodemiology*, *2*(1), e33184. https://doi.org/10.2196/33184

Gunn, H. K., & Lynch, M. P. (2018). Googling. In *The routledge handbook of applied epistemology* (pp. 41–53). Routledge. https://doi.org/10.4324/9781315679099-4

Haider, J. (2016). The structuring of information through search: Sorting waste with google. *Aslib Journal of Information Management*, *68*(4), 390–406. https://doi.org/10.1108/AJIM-12-2015-0189

Haider, J. (2017). *Controlling the urge to search. Studying the informational texture of practices by exploring the missing element*. *10*(1). https://doi.org/http://dx.doi.org/10.14288/1.0372890

Haider, J., Rödl, M., & Joosse, S. (2022). Algorithmically embodied emissions: The environmental harm of everyday life information in digital culture. *IR*, *27*. https://doi.org/10.47989/colis2224

Haider, J., & Sundin, O. (2019). *Invisible search and online search engines: The ubiquity of search in everyday life*. Routledge. https://doi.org/https://doi.org/10.4324/9780429448546

Hanselman, S. (2013). *Am i really a developer or just a good googler?* https://www.hanselman.com/blog/am-i-really-a-developer-or-just-a-good-googler.

Hanselman, S. (2020). *Me: 30 years writing software for money*. https://twitter.com/shanselman/status/1289804628310110209.

Haraway, D. J. (1988). Situated knowledges: The science question in feminism and the privilege of partial perspective. *Feminist Studies*, *14*, 205–224.

Hearst, M. A. (2009). *Search user interfaces*. Cambridge University Press.

Hicks, M. (2017). *Programmed inequality: How britain discarded women technologists and lost its edge in computing*. MIT Press.

Hill Collins, P. (2002). *Black feminist thought: Knowledge, consciousness, and the politics of empowerment* (Rev. 10th anniversary ed). Routledge.

Hippel, E. (1988). *The sources of innovation*. Oxford University Press.

Hirschhorn, L. (1997). *Reworking authority: Leading and following in the post-modem organization*. MIT Press.

Hochstein, L. (2021). *I have no idea what i'm doing*. https://surfingcomplexity.blog/2021/11/28/i-have-no-idea-what-im-doing/

Hodgson, G. M. (2001). *How economics forgot history: The problem of historical specificity in social science*. Routledge.

Hoffmann, A. L. (2019). Where fairness fails: Data, algorithms, and the limits of antidiscrimination discourse. *Information, Communication & Society*, *22*(7), 900–915. https://doi.org/10.1080/1369118X.2019.1573912

Hoffmann, A. L. (2021). Terms of inclusion: Data, discourse, violence. *New Media & Society*, *23*(12), 3539–3556. https://doi.org/10.1177/1461444820958725

House of Lords, Select Committee on Democracy and Digital Technologies. (2020). *Corrected oral evidence: Democracy and digital technologies*. https://committees.parliament.uk/oralevidence/360/html/.

Höchstötter, N., & Lewandowski, D. (2009). What users see – structures in search engine results pages. *Information Sciences*, *179*(12), 1796–1812. https://doi.org/10.1016/j.ins.2009.01.028

Hunt, A., & Thomas, D. (1999). *The pragmatic programmer: From journeyman to master*. Addison-Wesley Professional.

Hutchins, E. (1995). *Cognition in the wild*. MIT Press.

Introna, L. D. (2016). Algorithms, governance, and governmentality. *Science, Technology, & Human Values*, *41*(1), 17–49. https://doi.org/10.1177/0162243915587360

Introna, L. D., & Nissenbaum, H. (2000). Shaping the web: Why the politics of search engines matters. *The Information Society*, *16*(3), 169–185. https://doi.org/10.1080/01972240050133634

Iskander, N. (2021). *Does skill make us human?: Migrant workers in 21st-century qatar and beyond*. Princeton University Press.

Jack, C. (2017). Lexicon of lies: Terms for problematic information. *Data & Society*, *3*, 22. https://datasociety.net/output/lexicon-of-lies/

Jackson, S. J. (2014). *Rethinking repair* (T. Gillespie, P. J. Boczkowski, & K. A. Foot, Eds.; pp. 221–240). The MIT Press. https://doi.org/10.7551/mitpress/9780262525374.003.0011

Kalliamvakou, E. (2022). *Research: Quantifying github copilot's impact on developer productivity and happiness | the github blog*. https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/.

Karapapa, S., & Borghi, M. (2015). Search engine liability for autocomplete suggestions: Personality, privacy and the power of the algorithm. *Int J Law Info Tech*, *23*(3), 261–289.

https://doi.org/10.1093/ijlit/eav009

Katznelson, I. (2005). *When affirmative action was white: An untold history of racial inequality in twentieth-century america*. W.W. Norton & Company.

Khlaaf, H., Mishkin, P., Achiam, J., Krueger, G., & Brundage, M. (2022). *A hazard analysis framework for code synthesis large language models*. arXiv. https://doi.org/10.48550/ARXIV.2207.14157

Kleppmann, M. (2017). *Designing data-intensive applications : The big ideas behind reliable, scalable, and maintainable systems*. O'Reilly Media.

Kluttz, D. N., & Mulligan, D. K. (2019). *Automated decision support technologies and the legal profession*. https://doi.org/10.15779/Z38154DP7K

Koonin, S. (2019). *Everything i googled in a week as a professional software engineer - localghost*. https://localghost.dev/2019/09/everything-i-googled-in-a-week-as-a-professional-software-engineer/.

Kotamraju, N. P. (2002). Keeping up: Web design skill and the reinvented worker. *Information, Communication & Society*, *5*(1), 1–26. https://doi.org/10.1080/13691180110117631

Kraschnewski, J. L., Chuang, C. H., Poole, E. S., Peyton, T., Blubaugh, I., Pauli, J., Feher, A., & Reddy, M. (2014). Paging "dr. Google": Does technology fill the gap created by the prenatal care visit structure? Qualitative focus group study with pregnant women. *J Med Internet Res*, *16*(6), e147. https://doi.org/10.2196/jmir.3385

Kruschwitz, U., Hull, C., & others. (2017). *Searching the enterprise* (Vol. 11). Now Publishers.

Kutz, J. (2022). *Our search liaison on 25 years of keeping up with search*. https://blog.google/products/search/danny-25-years-of-search/; Google.

Latour, B. (1986). Visualization and cognition. *Knowledge and Society*, *6*(6), 1–40.

Latour, B. (1990). Drawing things together. In M. Lynch & S. Woolgar (Eds.), *Representation in scientific practice*. MIT Press.

Latour, B. (1992). Where are the missing masses? The sociology of a few mundane artifacts. In W. E. Bijker & J. Law (Eds.), *Shaping technology/building society: Studies in sociotechnical change* (pp. 225–228). MIT Press.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge university press. https://www.cambridge.org/highereducation/books/situated-learning/6915ABD21C8E4619F750A4D4ACA616CD#overview

Leonard-Barton, D. (1995). *Wellsprings of knowledge: Building and sustaining the sources of innovation*. Harvard Business School Press.

Leonardi, P. M. (2007). Activating the informational capabilities of information technology for organizational change. *Organization Science*, *18*(5), 813–831. https://doi.org/10.1287/orsc.1070.0284

Leonardi, P. M. (2011). When flexible routines meet flexible technologies: Affordance, constraint, and the imbrication of human and material agencies. *MIS Quarterly*, *35*(1), 147–167. http://www.jstor.org/stable/23043493

Lianos, M. (2010). Periopticon: Control beyond freedom and coercion and two possible advancements in the social sciences. In K. D. Haggerty & M. Samatas (Eds.), *Surveillance and democracy* (pp. 69–88). https://www.taylorfrancis.com/chapters/edit/10.4324/9780203852156-11/periopticon-control-beyond-freedom-coercion-two-possible-advancements-social-sciences-michalis-lianos

Liboiron, M. (2021). *Pollution is colonialism*. Duke University Press. https://www.dukeupress.edu/pollution-is-colonialism

Lofland, J., & Lofland, L. H. (1995). *Analyzing social settings*. Wadsworth Publishing Company.

Lurie, E., & Mulligan, D. K. (2021). *Searching for representation: A sociotechnical audit of googling for members of U.S. Congress (Working Paper)*. https://emmalurie.github.io/docs/preprint-searching.pdf

Mager, A. (2009). Mediated health: Sociotechnical practices of providing and using online health information. *New Media & Society*, *11*(7), 1123–1142. https://doi.org/10.1177/1461444809341700

Mager, A. (2012). ALGORITHMIC ideology. *Information, Communication & Society*, *15*(5), 769–787. https://doi.org/10.1080/1369118X.2012.676056

Marcus, G. E. (1995). Ethnography in/of the world system: The emergence of multi-sited ethnography. *Annual Review of Anthropology*, *24*, 95–117. http://www.jstor.org/stable/2155931

Mart, S. N. (2017). The algorithm as a human artifact: Implications for legal [re] search. *Law Libr. J.*, *109*, 387. https://scholar.law.colorado.edu/articles/755/%0A%20%20%20%20

Mathew, A. J., & Cheshire, C. (2018). A fragmented whole: Cooperation and learning in the practice of information security. *Center for Long-Term Cybersecurity, UC Berkeley*.

McChesney, R. W. (1997). *Corporate media and the threat to democracy*. Seven Stories Press.

McDermott, R., & Lave, J. (2006). Estranged labor learning. In P. Sawchuk, N. Duarte, & M. Elhammoumi (Eds.), *Critical perspectives on activity: Explorations across education, work, and everyday life* (pp. 89–122). Cambridge University Press. https://doi.org/10.1017/CBO9780511509568.007

Meisner, C., Duffy, B. E., & Ziewitz, M. (2022). The labor of search engine evaluation: Making algorithms more human or humans more algorithmic? *New Media & Society*, *0*(0), 14614448211063860. https://doi.org/10.1177/14614448211063860

Mejova, Y., Gracyk, T., & Robertson, R. (2022). Googling for abortion: Search engine mediation of abortion accessibility in the united states. *JQD*, *2*. https://doi.org/10.51685/jqd.2022.007

Metaphor. (2021). *Today we're releasing wanderer 2Metaphor*. https://twitter.com/metaphorsystems/status/1428793313663111170.

Metaxa-Kakavouli, D., & Torres-Echeverry. (2017). *Google's role in spreading fake news and misinformation*. Stanford University Law & Policy Lab. https://www-cdn.law.stanford.edu/wp-content/uploads/2017/10/Fake-News-Misinformation-FINAL-PDF.pdf

Meyer, J. W., & Rowan, B. (1977). Institutionalized organizations: Formal structure as myth and ceremony. *American Journal of Sociology*, *83*(2), 340–363.

Meyer, J. W., & Rowan, B. (1978). The structure of educational organizations. In M. W. Meyer & Associates (Eds.), *Organizations and environments* (pp. 78–109). Jossey Bass.

Miller, B., & Record, I. (2013). JUSTIFIED belief in a digital age: ON the epistemic implications of secret internet technologies. *Episteme*, *10*(2), 117–134. https://doi.org/10.1017/epi.2013.11

Mills, C. (2008). White ignorance. In L. S. Robert Proctor (Ed.), *Agnotology: The making and unmaking of ignorance*. Stanford University Press.

Misa, editor, Thomas J. (2010). *Gender codes: Why women are leaving computing* (1st ed.). Wiley-IEEE Computer Society Press. libgen.li/file.php?md5=e4a6849b080f4a078c79777d347726c2

Mokros, H. B., & Aakhus, M. (2002). From information-seeking behavior to meaning engagement practice.. *Human Comm Res*, *28*(2), 298–312. https://doi.org/10.1111/j.1468-2958.2002.tb00810.x

Morris, M. R. (2013). Collaborative search revisited. *Proceedings of Cscw 2013*. https://doi.org/10.1145/2441776.2441910

Morris, M. R., & Horvitz, E. (2007). SearchTogether: An interface for collaborative web

search. *Proceedings of the 20th Annual Acm Symposium on User Interface Software and Technology*, 3–12.

Morris, M. R., & Teevan, J. (2009). Collaborative web search: Who, what, where, when, and why. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, *1*(1), 1–99. https://doi.org/10.2200/S00230ED1V01Y200912ICR014

Mulligan, D. K., & Griffin, D. (2018). Rescripting search to respect the right to truth. *The Georgetown Law Technology Review*, *2*(2), 557–584. https://georgetownlawtechreview.org/rescripting-search-to-respect-the-right-to-truth/GLTR-07-2018/

Mulligan, D. K., Koopman, C., & Doty, N. (2016). Privacy is an essentially contested concept: A multi-dimensional analytic for mapping privacy. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2083), 20160118.

Mulligan, D. K., & Nissenbaum, H. (2020). *The concept of handoff as a model for ethical analysis and design*. Oxford University Press. https://doi.org/10.1093/oxfordhb/9780190067397.013.15

Mustafaraj, E., Lurie, E., & Devine, C. (2020). The case for voter-centered audits of search engines during political elections. *FAT* '20*.

Nagaraj, A. (2021). Information seeding and knowledge production in online communities: Evidence from openstreetmap. *Management Science*, *67*(8), 4908–4934. https://doi.org/10.1287/mnsc.2020.3764

Narayanan, D., & De Cremer, D. (2022). "Google told me so!" On the bent testimony of search engine algorithms. *Philos. Technol.*, *35*(2), E4512. https://doi.org/10.1007/s13347-022-00521-7

Natarajan, M. (2016). *The making of ignorance: Epistemic design in self-tracking health* [PhD thesis, University of California, Berkeley]. https://escholarship.org/uc/item/5pn009mw

Neff, G. (2012). *Venture labor: Work and the burden of risk in innovative industries*. MIT press. https://mitpress.mit.edu/books/venture-labor

Nissenbaum, H. (2011a). A contextual approach to privacy online. *Daedalus*, *140*(4), 32–48.

Nissenbaum, H. (2011b). From preemption to circumvention: If technology regulates, why do we need regulation (and vice versa). *Berkeley Tech. LJ*, *26*, 1367.

Noble, S. U. (2018). *Algorithms of oppression how search engines reinforce racism*. New York University Press. https://nyupress.org/9781479837243/algorithms-of-oppression/

Ochigame, R. (2020). Informatics of the oppressed. In *Logic*. https://logicmag.io/care/informatics-of-the-oppressed/

Oeldorf-Hirsch, A., Hecht, B., Morris, M. R., Teevan, J., & Gergle, D. (2014, February). To search or to ask. *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*. https://doi.org/10.1145/2531602.2531706

Ofcom. (2022). *Children and parents: Media use and attitudes report 2022*. https://www.ofcom.org.uk/___data/assets/pdf_file/0024/234609/childrens-media-use-and-attitudes-report-2022.pdf.

Ohm, P., & Frankle, J. (2018). Desirable inefficiency. *Florida Law Review*, *70*(4), 777. https://scholarship.law.ufl.edu/flr/vol70/iss4/2

Orlikowski, W. J. (2007). Sociomaterial practices: Exploring technology at work. *Organization Studies*, *28*(9), 1435–1448. https://doi.org/10.1177/0170840607081138

Orlikowski, W. J. (1993). CASE tools as organizational change: Investigating incremental and radical changes in systems development. *MIS Q.*, *17*(3), 309–340. https://doi.org/10.2307/249774

Orr, J. E. (1996). *Talking about machines: An ethnography of a modern job*. ILR Press.

Pan, B., Hembrooke, H., Joachims, T., Lorigo, L., Gay, G., & Granka, L. (2007). In google we trust: Users' decisions on rank, position, and relevance. *Journal of Computer-Mediated Communication*, *12*(3), 801–823. https://doi.org/10.1111/j.1083-6101.2007.00351.x

Pasquale, F. (2015). *The black box society*. Harvard University Press.

Passi, S., & Barocas, S. (2019). Problem formulation and fairness. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 39–48.

Passi, S., & Jackson, S. J. (2018). Trust in data science: Collaboration, translation, and accountability in corporate data science projects. *Proceedings of the ACM on Human-Computer Interaction*, *2*(CSCW), 1–28.

Passi, S., & Sengers, P. (2020). Making data science systems work. *Big Data & Society*, *7*(2), 2053951720939605.

Pearce, H. A., Ahmad, B., Tan, B., Dolan-Gavitt, B., & Karri, R. (2021). An empirical cybersecurity evaluation of github copilot's code contributions. *ArXiv*, *abs/2108.09293*.

Peels, R., & Pritchard, D. (2021). Educating for ignorance. *Synthese*, *198*(8), 7949–7963. https://doi.org/10.1007/s11229-020-02544-z

Perlow, L., & Weeks, J. (2002). Who's helping whom? Layers of culture and workplace behavior. *Journal of Organizational Behavior*, *23*(4), 345–361. https://doi.org/https://doi.org/10.1002/job.150

Perrow, C. (1984). *Normal accidents: Living with high-risk technologies*. Basic Books.

Plato. (2002). *Plato: Five dialogues: Euthyphro, Apology, Crito, Meno, Phaedo* (J. M. Cooper, Ed.; G. M. A. Grube, Trans.; 2nd ed.). Hackett.

Polanyi, M. (1967). *The tacit dimension*. Doubleday & Co.

Prior, L. (2003). *Using documents in social research*. SAGE Publications Ltd. https://doi.org/10.4135/9780857020222

Proctor, R., & Schiebinger, L. (Eds.). (2008). *Agnotology: The making and unmaking of ignorance*. Stanford University Press. https://archive.org/details/agnotologymaking0000unse

Pugh, A. J. (2013). What good are interviews for thinking about culture? Demystifying interpretive analysis. *Am J Cult Sociol*, *1*(1), 42–68. https://doi.org/10.1057/ajcs.2012.4

Raboy, M. (1998). Global communication policy and human rights. In *A communications cornucopia: Markle foundation essays on information policy* (pp. 218–242). Brookings Institution Press.

Rahman, M. M., Barson, J., Paul, S., Kayani, J., Lois, F. A., Quezada, S. F., Parnin, C., Stolee, K. T., & Ray, B. (2018). Evaluating how developers use general-purpose web-search for code retrieval. *Proceedings of the 15th International Conference on Mining Software Repositories*, 465–475. https://doi.org/10.1145/3196398.3196425

Reagle, J. (2016). The obligation to know: From faq to feminism 101. *New Media &Amp; Society*, *18*(5), 691–707. https://doi.org/10.1177/1461444814545840

Rieder, B., & Hofmann, J. (2020). Towards platform observability. *Internet Policy Review*, *9*(4), 1–28. https://doi.org/https://doi.org/10.14763/2020.4.1535

Roberts, N., Galluch, P. S., Dinger, M., & Grover, V. (2012). Absorptive capacity and information systems research: Review, synthesis, and directions for future research. *MIS Quarterly*, *36*(2), 625–648. http://www.jstor.org/stable/41703470

Robertson, R. (2022). *WebSearcher*. https://github.com/gitronald/WebSearcher

Robinson, L. (2009). A taste for the necessary. *Information, Communication & Society*, *12*(4), 488–507. https://doi.org/10.1080/13691180902857678

Russell, D. M. (2019). *The joy of search: A google insider's guide to going beyond the basics*. The MIT Press.

Sabel, C. F. (1984). *Work and politics: The division of labor in industry*.

Salva, R. J. (2022). *Preview: Referencing public code in github copilot | the github blog*. https://github.blog/2022-11-01-preview-referencing-public-code-in-github-copilot/.

Schultheiß, S., Sünkler, S., & Lewandowski, D. (2018). We still trust google, but less than 10 years ago: An eye-tracking study. *Information Research*, *23*(3). http://informationr.net/ir/23-3/paper799.html

Seaver, N. (2017). Algorithms as culture: Some tactics for the ethnography of algorithmic systems. *Big Data & Society*, *4*(2), 1–12. https://doi.org/10.1177/2053951717738104

Seaver, N. (2019). Knowing algorithms. *Digital STS*, 412–422.

Seibel, P. (2009). *Coders at work: Reflections on the craft of programming*. Apress.

Shah, C. (2017). *Social information seeking*. Springer International Publishing. https://doi.org/10.1007/978-3-319-56756-3

Shah, C., & Bender, E. M. (2022, March). Situating search. *ACM SIGIR Conference on Human Information Interaction and Retrieval*. https://doi.org/10.1145/3498366.3505816

Shestakofsky, B. (2017). Working algorithms: Software automation and the future of work. *Work and Occupations*, *44*(4), 376–423. https://doi.org/10.1177/0730888417726119

Shorey, S., Hill, B. M., & Woolley, S. (2021). From hanging out to figuring it out: Socializing online as a pathway to computational thinking. *New Media & Society*, *23*(8), 2327–2344. https://doi.org/10.1177/1461444820923674

Skitka, L. J., Mosier, K. L., Burdick, M., & Rosenblatt, B. (2000). Automation bias and errors: Are crews better than individuals? *The International Journal of Aviation Psychology*, *10*, 85–97.

Smith, C. L., & Rieh, S. Y. (2019, March). Knowledge-context in search systems. *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*. https://doi.org/10.1145/3295750.3298940

Smithson, M. (1985). Toward a social theory of ignorance. *Journal for the Theory of Social Behaviour*, *15*(2), 151–172. https://doi.org/https://doi.org/10.1111/j.1468-5914.1985.tb00049.x

Star, S. L. (1999). The ethnography of infrastructure. *American Behavioral Scientist*, *43*(3), 377–391.

Star, S. L., & Ruhleder, K. (1996). Steps toward an ecology of infrastructure: Design and access for large information spaces. *Information Systems Research*, *7*(1), 111–134. https://doi.org/10.1287/isre.7.1.111

Star, S. L., & Strauss, A. (2004). Layers of silence, arenas of voice: The ecology of visible and invisible work. *Computer Supported Cooperative Work (CSCW)*, *8*, 9–30.

Stark, D. (2009). *The sense of dissonance: Accounts of worth in economic life*. Princeton University Press.

Stark, L., & Levy, K. (2018). The surveillant consumer. *Media, Culture & Society*, *40*(8), 1202–1220. https://doi.org/10.1177/0163443718781985

Suchman, L. A. (1987). *Plans and situated actions: The problem of human-machine communication*. Cambridge university press.

Sundin, O. (2020). Where is search in information literacy? A theoretical note on infrastructure and community of practice. In *Sustainable digital communities* (pp. 373–379). Springer International Publishing. https://doi.org/10.1007/978-3-030-43687-2_29

Sundin, O., Haider, J., Andersson, C., Carlsson, H., & Kjellberg, S. (2017). The search-ification of everyday life and the mundane-ification of search. *Journal of Documentation*. https://doi.org/10.1108/JD-06-2016-0081

Sundin, O., Lewandowski, D., & Haider, J. (2021). Whose relevance? Web search engines as multisided relevance machines. *Journal of the Association for Information Science and Technology*. https://doi.org/10.1002/asi.24570

Surden, H. (2007). Structural rights in privacy. *SMU Law Review*, *60*(4), 1605. https://scholar.law.colorado.edu/articles/346/

Takhteyev, Y. (2007). Jeeks: Developers at the periphery of the software world. *Annual Meeting of the American Sociological Association, New York, Ny*. https://pdfs.semanticscholar.org/1a98/c87ffd0b07344c5e0aae6e7e498a5c69da00.pdf

Takhteyev, Y. (2012). *Coding places: Software practice in a south american city*. The MIT Press.

Tattersall Wallin, E. (2021). Audiobook routines: Identifying everyday reading by listening practices amongst young adults. *JD*, *78*(7), 266–281. https://doi.org/10.1108/JD-06-2021-0116

Teevan, J., Alvarado, C., Ackerman, M. S., & Karger, D. R. (2004). The perfect search engine is not enough: A study of orienteering behavior in directed search. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 415–422.

The MIT Press. (2020). *Author talk: The Joy of Search by Daniel M. Russell*. https://mitpress.mit.edu/blog/author-talk-the-joy-of-search-by-daniel-m-russell/.

The social construction of technological systems. (1993). MIT Press.

Thompson, J. D. (1967). *Organizations in action: Social science bases of administrative theory* (1st ed.). McGraw-Hill.

Toff, B., & Nielsen, R. K. (2018). "I Just Google It": Folk Theories of Distributed Discovery. *Journal of Communication*, *68*(3), 636–657. https://doi.org/10.1093/joc/jqy009

Tripodi, F. (2018). Searching for alternative facts: Analyzing scriptural inference in conservative news practices. *Data & Society*. https://datasociety.net/output/searching-for-alternative-facts/

Tripodi, F. (2019a). Devin nunes and the power of keyword signaling. In *Wired*. https://www.wired.com/story/devin-nunes-and-the-dark-power-of-keyword-signaling/.

Tripodi, F. (2019b). *SenateHearing + written testimony*. https://www.judiciary.senate.gov/imo/media/doc/Tripodi%20Testimony.pdf; Senate Judiciary Committee Subcommittee On The Constitution.

Tripodi, F. (2022a). *Step five: Set the traps - misinformation that fails to conform to dominant search engine rules is functionally invisible*. https://twitter.com/ftripodi/status/1520078674417967105.

Tripodi, F. (2022b). *The propagandists' playbook: How conservative elites manipulate search and threaten democracy* (Hardcover, p. 288). Yale University Press. https://yalebooks.yale.edu/book/9780300248944/the-propagandists-playbook/

Tsing, A. (2005). *Friction : An ethnography of global connection*. Princeton University Press. https://press.princeton.edu/books/paperback/9780691120652/friction

Turco, C. J. (2016). *The conversational firm: Rethinking bureaucracy in the age of social media*. Columbia University Press.

Urman, A., & Makhortykh, M. (2022). "Foreign beauties want to meet you": The sexualization of women in google's organic and sponsored text search results. *New Media & Society*, *0*(0), 14614448221099536. https://doi.org/10.1177/14614448221099536

Vaidhyanathan, S. (2011). *The googlization of everything:(And why we should worry)*. Univ of California Press. https://doi.org/10.1525/9780520948693

Van Couvering, E. J. (2007). Is relevance relevant? Market, science, and war: Discourses of search engine quality. *Journal of Computer-Mediated Communication*, *12*(3), 866–887.

http://doi.org/10.1111/j.1083-6101.2007.00354.x

Vertesi, J. (2019). From affordances to accomplishments: PowerPoint and excel at NASA. In *digitalSTS* (pp. 369–392). Princeton University Press. https://doi.org/10.1515/9780691190600-026

Warshaw, J., Taft, N., & Woodruff, A. (2016). Intuitions, analytics, and killing ants: Inference literacy of high school-educated adults in the US. *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, 271–285. https://www.usenix.org/conference/soups2016/technical-sessions/presentation/warshaw

Weber, S. (2004). *The success of open source*. Harvard University Press. https://www.hup.harvard.edu/catalog.php?isbn=9780674018587

Weber, S. (2019). *Bloc by bloc: How to build a global enterprise for the new regional order*. Harvard University Press.

What is the xy problem? - meta stack exchange. (2010). https://meta.stackexchange.com/questions/66377/what-is-the-xy-problem.

White, R. W. (2016). *Interactions with search systems*. Cambridge University Press. https://doi.org/10.1017/CBO9781139525305

Widder, D. G., Nafus, D., Dabbish, L., & Herbsleb, J. D. (2022, June). Limits and possibilities for "ethical AI" in open source: A study of deepfakes. *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. https://davidwidder.me/files/widder-ossdeepfakes-facct22.pdf

Wineburg, S. (2021). *"Problem 2: Typing "the claim into a search engine" will"*. https://twitter.com/samwineburg/status/1465542166764081157.

Wong, R. (2020). *Values by design imaginaries: Exploring values work in ux practice* [Ph.D. dissertation, University of California, Berkeley]. https://escholarship.org/uc/item/3vt3b1xf

Woods, D. D. (2018). The theory of graceful extensibility: Basic rules that govern adaptive systems. *Environ Syst Decis*, *38*(4), 433–457. https://doi.org/10.1007/s10669-018-9708-3

Yates, J., & Orlikowski, W. J. (1992). Genres of organizational communication: A structurational approach to studying communication and media. *The Academy of Management Review*, *17*(2), 299–326. http://www.jstor.org/stable/258774

Zade, H., Wack, M., Zhang, Y., Starbird, K., Calo, R., Young, J., & West, J. D. (2022). Auditing google's search headlines as a potential gateway to misleading content. *Jots*, *1*(4). https://doi.org/10.54501/jots.v1i4.72

Ziegler, A. (2022). *Research: How github copilot helps improve developer productivity | the github blog*. https://github.blog/2022-07-14-research-how-github-copilot-helps-improve-developer-productivity/.

Zuboff, S. (1988). *In the age of the smart machine*. Basic books.

Zuboff, S. (2015). Big other: Surveillance capitalism and the prospects of an information civilization. *Journal of Information Technology*, *30*(1), 75–89. https://doi.org/10.1057/jit.2015.5