Direct Answers for Search Queries in the Long Tail

Michael S. Bernstein[1], Jaime Teevan[2], Susan Dumais[2], Dan Liebling[2], Eric Horvitz[2]

[1] MIT CSAIL
32 Vassar St., Cambridge MA
msbernst@mit.edu

[2] Microsoft Research
One Microsoft Way, Redmond WA 98052
{teevan, sdumais, danl, horvitz}@microsoft.com

ABSTRACT

Web search engines now offer more than ranked results. Queries on topics like weather, definitions, and movies may return inline results called *answers* that can resolve a searcher's information need without any additional interaction. Despite the usefulness of answers, they are limited to popular needs because each answer type is manually authored. To extend the reach of answers to thousands of new information needs, we introduce *Tail Answers*: a large collection of direct answers that are unpopular individually, but together address a large proportion of search traffic. These answers cover long-tail needs such as the average body temperature for a dog, substitutes for molasses, and the keyboard shortcut for a right-click. We introduce a combination of search log mining and paid crowdsourcing techniques to create Tail Answers. A user study with 361 participants suggests that Tail Answers significantly improved users' subjective ratings of search quality and their ability to solve needs without clicking through to a result. Our findings suggest that search engines can be extended to directly respond to a large new class of queries.

Author Keywords

Search user interfaces, query log analysis, crowdsourcing

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces – Graphical user interfaces.

General Terms

Design; Human Factors.

INTRODUCTION

While search engines have long connected people to *documents*, they are now beginning to also connect people directly to *information*. The results page is no longer just a plain list of page titles and snippets. For popular topics such as weather, movies, and definitions, search engines may add custom interfaces with direct results (e.g., "77°F, partly cloudy"). These direct results, known as *answers* [7], allow searchers to satisfy their information need without clicking through to a web page. Answers have a measurable impact on user behavior in search result pages, and many users seek answer types repeatedly once they realize that they exist [7]. For common needs, answers demonstrate the value of customizing the interface to deliver information rather than documents.

Unfortunately, answers are only available for a small percentage of common needs [7,13]. Few manually curated answers exist, and those that do focus on popular query types like weather. In contrast, most searchers' information needs are unpopular: half of all queries are unique [20], and users rarely see most answers [7]. It has not been feasible for search engines to create answers to cover the long tail of less popular information needs: search engines must author the content, test which queries to trigger on, find a data source to feed the answer, and keep the answer up-to-date as the web changes [1]. As a result, searchers only receive relevant direct answers when they have extremely common information needs.

We extend search engine answers to a broad new class of queries in the long tail. By doing so, we show that *search engines can aggregate user knowledge to improve not just result rankings, but the entire search user experience*. We introduce *Tail Answers*, automatically generated search engine answers that support a large set of less common information needs. These information needs include the normal body temperature for a dog (Figure 1), substitutes for molasses, the currency in Ireland, and many more (Figure 2). Each of these needs may occur thousands of times per year, but are too far in the tail of query traffic to be worth assigning programmers, designers, testers, and product management staff to create and maintain answers.

To push answer content down into the long tail (without an exponentially-sized editorial staff), our insight is to aggregate the knowledge of thousands of everyday web users. We turn to web users in each of the three major steps of creating Tail Answers: 1) We identify answer candidates

Figure 1. Tail Answers are inline direct responses for search results. This Tail Answer addresses body temperature for dogs.

CHI'12, May 5–10, 2012, Austin, Texas, USA.

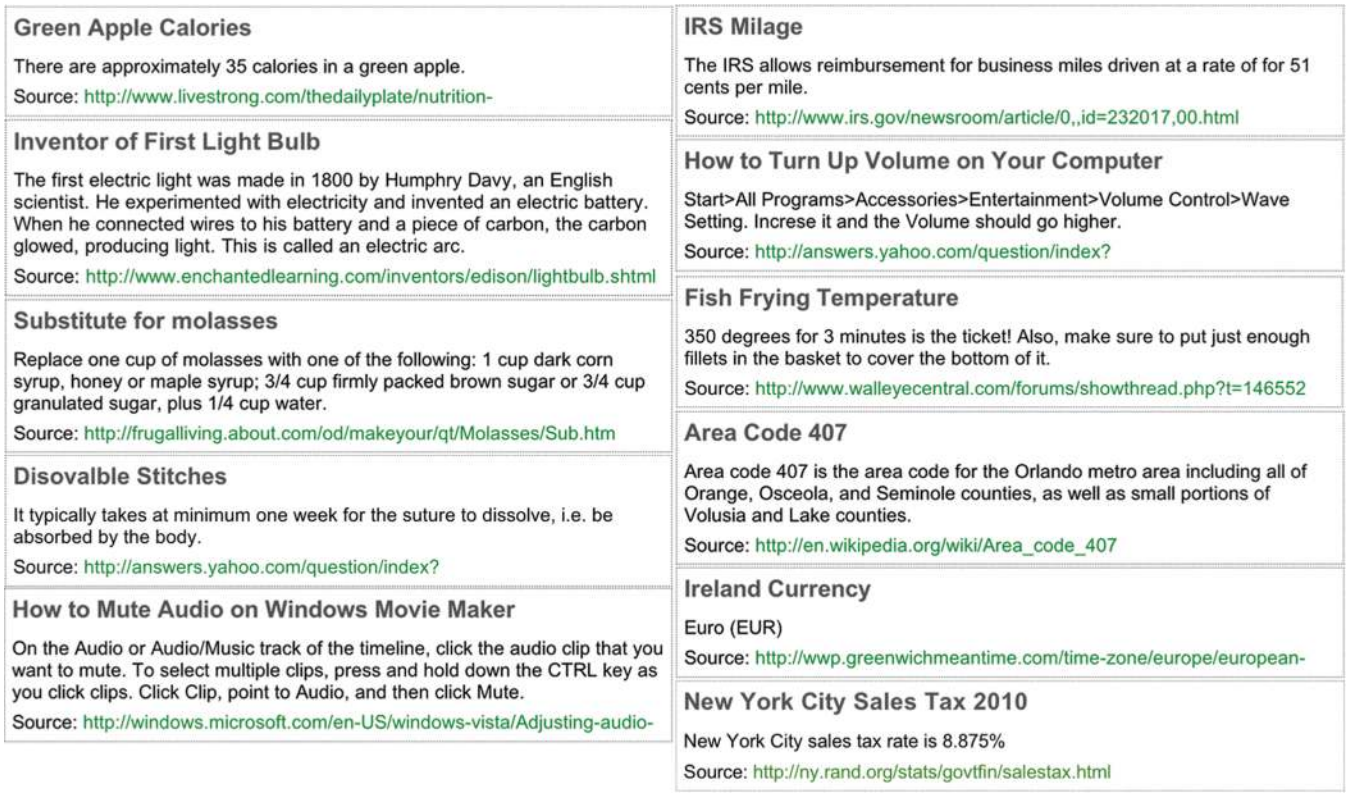

Figure 2. Tail Answers address less common information needs. These examples (including errors) were produced by the data mining and crowdsourcing processes described in the paper. They trigger on related queries, e.g., *apple calories*.

using aggregate search and browsing patterns; 2) We filter those answer candidates to ones which represent directly answerable needs, using search logs and paid crowdsourcing; 3) We extract the answer content from the web, using paid crowds to copy and paste content from the page, then author and edit the final answer text. The entire process can be effectively automated.

Following a survey of related work, we describe how we use log analysis and crowdsourcing to generate Tail Answers for information needs that search engines would not normally be able to address directly. We then present the results of an evaluation of Tail Answers that shows they significantly improved the subjective search experience, compensating for poor results and reducing perceived effort. We conclude by detailing extensions of these techniques for authoring smart snippets, integrating automatic question-answering systems, and creating new classes of answers. Our work suggests that search engines can use aggregate search patterns and crowdsourcing to improve the search experience far beyond simply better result ranking.

RELATED WORK

Search engine answers and result snippets can have a powerful influence on the web search user experience. Nearly half of the abandoned queries in a Google sample displayed a snippet that might have made any additional clicks unnecessary [13]. One quarter of all queries may already be addressed directly in the result page, especially for needs like spell checking, query monitoring, and learning about a term [17]. Successful answers will thus *cannibalize* clicks from the rest of the search results, and searchers will repeat queries to trigger an answer once they learn of it [7]. Even when no answer exists, searchers often use queries for repeated navigation, for example searching for *chi 2012* whenever they want to find the CHI papers deadline [19]. Search result snippets can also sometimes address information needs directly [8]; the snippet for a page, for example, may contain the answer in the text.

Some long-tail information needs can be addressed with automatic information extraction. Many question-answering systems are designed to address information needs with short phrases such as using search result *n*-grams to identify answers [2,5,14]. A second approach is open-domain information extraction, for example TextRunner [3]. These approaches work best when facts are repeated across multiple web pages. Finally, systems can employ curated knowledge bases such as YAGO [18] and match on them to answer some queries. However, automated approaches can make mistakes that are obvious to humans.

Question-answering systems have also recruited crowds to deliver results: for example, Aardvark [9] and Quora (www.quora.com) use members to answer questions. Rather than find domain experts, Tail Answers recruits crowd members with only basic knowledge of web search and the search context. This enables Tail Answers to cover broad content, but it raises challenges when workers do not

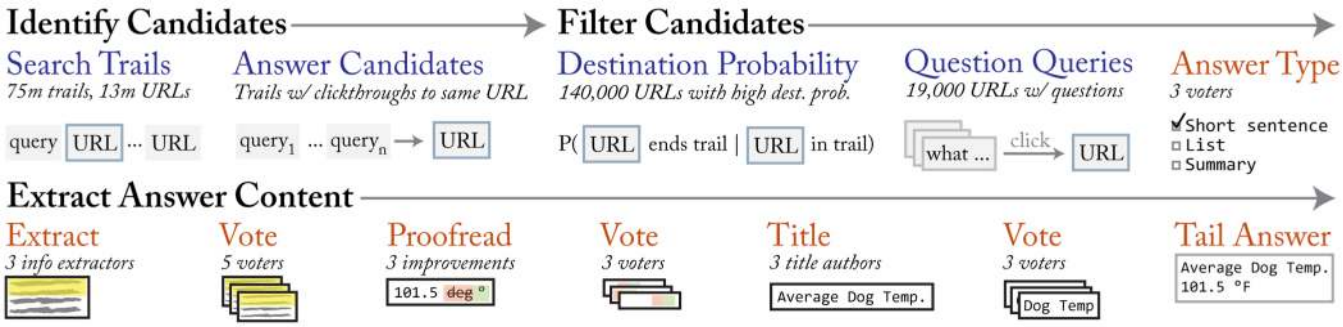

Figure 3. An overview of the three phase Tail Answers creation process, which involves 1) identifying answer candidates, 2) filtering the candidates to ones that address "answerable" needs, and 3) extracting the Tail Answer content. Steps that are implemented via data mining are indicated in blue, and those implemented via crowdsourcing are indicated in orange.

have necessary expertise. ChaCha (www.chacha.com) also uses paid, on-demand staff or crowds for question answering, but they do not vet or edit the results and do not create reusable information artifacts.

Our crowdsourcing algorithm depends on several quality control techniques from the literature. We build on the idea of gold standard tasks [12]: questions that the requester has labeled with ground truth. To cheaply and robustly control for quality, we create ground truth tasks for which the worker's answer must include at least one phrase from an inclusion list and none from an exclusion list. This inclusion/exclusion technique extends gold standard questions beyond exact matching (e.g., multiple choice) to open-ended writing or extraction tasks. Then, once the raw text is extracted, we use crowdsourcing approaches inspired by iterative text refinement [15] and proofreading [4] to improve the answer quality. Crowd planning algorithms could later be used to create high-level summaries for complex topics [11]. Search engines have also used crowds for relevance judgments; our work is the first to use crowd work to expand the search user experience.

Tail Answers improves on related work by creating the first set of direct responses for a broad set of uncommon queries. We contribute new query log mining and crowdsourcing quality control techniques to create Tail Answers in an automated fashion. We provide the first controlled, empirical evidence that these answers improve the user experience, often as much as search result quality.

TAIL ANSWERS

In this section, we describe how we create Tail Answers to extend search engine answers to tens of thousands of new information needs. Tail Answers are special results inserted inline in the search interface, as shown in Figure 1. The Tail Answer contains edited text from a webpage where other searchers found the solution to the same information need. Above the answer text is a concise title to aid skimming, and below it is a link to the source web page for attribution and further exploration. Each Tail Answer is targeted at one particular information need, although it may trigger for many different queries. When a user issues a query that matches a triggering query for a Tail Answer, that answer appears at the top of the search results.

Although answers for popular queries are currently manually authored, Tail Answers have an automated process to identify information needs that are appropriate for an answer and to author a direct result that addresses the need. In this work, we represent an information need as a set of queries with a similar intent. For example, the queries *dog temperature*, *dog fever*, and *average temp dog thermometer* represent the information need in Figure 1. In addition, we assume that Tail Answers can be associated with a web page that contains the answer (e.g., the page www.natural-dog-health-remedies.com/dog-temperature.html).

To create a Tail Answer, then, our system needs to:

1. **Identify** pages that are answer candidates,
2. **Filter** candidates that answers cannot address, and
3. **Extract** the Tail Answer content.

To accomplish these goals, we extract knowledge about answers from the activities of thousands of web users. To identify information needs, we use large-scale log analysis of web browsing patterns. To filter the needs, we augment log analysis with paid crowdsourcing. To extract answer content, we use paid crowdsourcing. Figure 3 represents this process visually. We now describe each step in detail, highlighting the technical challenges we solved to improve answer quality.

Identifying Answer Candidates

We begin by identifying information needs, which we call *answer candidates*. An answer candidate is a set of queries associated with a URL from a search result page (Table 1). A key idea is to identify browsing patterns that suggest searchers are finding a compact solution to an information need. We use query log analysis to populate our set of answer candidates. To do so, for each search session in our browser logs, we extract a *search trail* [20]: a browsing path beginning with a search query and terminating with a session timeout of thirty minutes. We then group all search trails on the first clicked URL from the result page. For each URL in our dataset, we now have a set of queries that

led to the URL and a set of trails that describe what users did after clicking through to the URL.

Filtering Answer Candidates

From these answer candidates, we must identify those that are intended for fact-finding [10] and will produce good answers. Some answer candidates have information needs that are too complex to answer; others have underspecified queries where the information need may not be clear. We developed three filters to find promising answer candidates. These filters look for particular types of 1) navigation behavior, 2) query behavior, and 3) information needs.

Filtering by Navigation Behavior: Destination Probability

Our first filter uses the search trails to identify web pages where people quickly end their search sessions. We assume that after a query, people typically end up at web pages containing information that addresses their need. If users stop browsing after they reach a page, that page likely solves the need. If users continue browsing or searching, on the other hand, the page may not succinctly satisfy their need. For example, queries such as *new york times* are often navigational [6]: searchers click on www.nytimes.com in the results, then often keep browsing and click on a link to read an article. Other information needs, like buying a new car, are complex and persist across multiple sessions [9], so searchers will typically keep browsing and returning to the search page. But, for web pages like the CHI call for papers, searchers will issue a query (e.g., *chi 2012 deadline*), click through to the page, find what they are looking for, and end their search session.

We formalize the idea of trail-ending web pages with a measurement we call *destination probability*. The destination probability for a web page is the observed probability that a searcher will end their session at that web page after clicking through to the page from the search results. In our search trails, the step immediately after a query is a click on a result web page. If a high percentage of trails end after that click (i.e., if their trail length is two), the destination probability will be high. If most trails instead include actions that return to the result page or browse to other URLs, the destination probability will be low. In other words, the destination probability for a URL is the observed probability that a click to the URL from the search result page is the last action in the search trail.

Page Title and URL	Sample Queries	Type
How to Force Quit on Mac ehow.com/how_5178032_force-quit-mac.html	*force quit mac* *force quit on macs* *how to force quit mac*	Short
Area Code 410 areacodehelp.com/where/area_code_410.shtml	*what area code is 410* *410 area code* *area code 410 location*	Short
How to bake a potato howtobakeapotato.com	*baked ptato* *how long do you cook potatoes in the oven* *best way to bake a potato*	List
Rummy 500 rules rummy.com/rummy500.html	*rules of gin rummy 500* *rummy 500* *how to play rummy 500*	Summary
Pandora Radio pandora.com	*radio* *pandora* *pandora radio log in*	—

Table 1. Pages with high destination probability, queries to them, and their crowd-voted answer category. All but the bottom row had a question query: the lack of a question signals that Pandora would not be appropriate for an answer.

Web pages with high destination probability are strong candidates for Tail Answers. We filter out any answer candidates that have destination probability of less than 0.3 or fewer than three search trails in our dataset. The 30% cutoff was tuned empirically to balance the number of possible answers (false negatives) with the number of pages with unanswerable content (false positives). Table 1 lists five web pages with high destination probabilities. For example, one contains instructions for how to bake a potato.

Filtering by Query Behavior: Question Words

Destination probability identifies pages where searchers appear to be finding immediate answers for their information needs. However, it can be very hard to infer the fact-finding intent from queries that are only two or three words long. For example, an answer for the query *dissolvable stitches* would be valuable if the searcher wanted to learn how long the stitches take to dissolve, but would not if they want to learn the stitches' history.

To avoid this problem, we make use of the minority of searchers who write queries using question words. Question-word queries are useful because they tend to be expressed in natural language, are longer than typical queries, and are more explicit (e.g., *how long do dissolvable stitches last*). These properties make the information need relatively easy to understand. Use of question words also tends to indicate fact-finding intent. We assume that question-word queries often overlap significantly with the unspecified information needs from the other queries, for example that *where is 732 area code* and *732 area code* have similar needs. When this is not the case, we rely on paid crowd members later to disambiguate the most common information need from the set of all queries.

We filter the answer candidates to remove any that had fewer than 1% of their clicks from question queries. The question words we currently look for are: *how*, *what*, *when* and *who*. The bottom row of Table 1 demonstrates the kind of error that can occur without a question word filter.

Filtering by Information Need: Answer Type

While question words are useful for identifying answer candidates, neither they nor other types of behavioral log data can help the system understand whether a concise answer could address the information need. Knowing the expected length of the answer is important because crowd workers often extract too much text in order to guarantee

that they captured the correct information and thus will be paid. However, overly verbose answers are not useful to searchers. Knowing what kind of answer to expect, for example a short phrase, can help the system perform automatic quality control using length.

To solve these problems, we use paid crowdsourcing via Crowdflower to categorize answer candidates into types. Crowdflower is built on top of Amazon Mechanical Turk and uses hidden quality-control questions known as *gold standard* questions to filter out poor-quality workers. By prototyping many answers, we developed the following three categories as useful for workers to identify:

- **Short** answers with very little text. For example: "The optimal fish frying temperature is 350°F."
- **List** answers, which typically contain a small set of directions. For example: "To change your password over Remote Desktop: 1) Click on Start > Windows Security. 2) Click the Change Password button. […]".
- **Summary or long list** answers, which synthesize large amounts of content. For example, pages requiring deep reading such as "Impact of Budget Cuts on Teachers" and "Centralized vs. Decentralized Organizations".

Workers were asked to read all of the queries that led to the web page, as well as the page itself, and then vote on the best matching category. The third column in Table 1 labels each example with its voted answer type.

Although short answers and list answers can be extracted from the web page and edited into an answer, summary answers require more synthesis. For this reason, we leave the generation of summary answers to future work. We use the data about whether an answer is a short answer or a list answer to give workers more specific instructions as they extract answer content and to enforce a maximum number of characters workers can extract from a page.

Extracting the Tail Answer

At this point, we have a set of answer candidates that can be addressed succinctly and factually by the search engine, but each candidate is only represented by a web page and a set of queries. To create an actual answer, we need to extract the information from the web page related to the unifying need, edit it for readability, and write a short answer title. Because automatic extraction algorithms are not yet reliable, we use paid crowdsourcing via Crowdflower.

The algorithm we developed to guide the crowd to create Tail Answers is as follows. Workers: 1) *extract* (i.e., copy and paste) as little text as possible from the web page using the associated queries as a guide, 2) *proofread* and edit the extracted information into an answer, and 3) *title* the answer descriptively. This information is compiled into a visually distinct search result and presented to searchers who issue the queries associated with the intent, or similar queries. Figure 3 contains a graphical representation of these steps.

Worker quality control is a major challenge for the generation of the Tail Answer title and text. Lazy Turkers

Spooning is a type of cuddling. When you spoon, you lay on your side with your back to your partner's chest and the partner behind wraps his or her arms around you and fits around you like a puzzle. The name likely came because of the way two spoons rest on each other, filling all the nooks. The "little spoon" is considered the person in front, the "big spoon" is considered the person in back. Another explanation I have read for the origin of the expression: In days of old, when a proper young man visited a proper young lady, he was supposed to do something to keep his hands occupied and away from her body. An acceptible activity was sit and carve a wooden spoon while conversing. Of a similar vintage, when the couple threw another log in the fireplace late in the evening, the neighbors would see a burst of sparks from the chimney, and know that someone was "sparking."

Figure 4. In this example workers extracted all of the text when an inclusion/exclusion lists was not used. Orange text is the same answer with inclusion/exclusion lists.

[4] will copy/paste introductory text from each page instead of the answer, and even well-intentioned, pre-qualified workers will extract entire paragraphs or large sections of the page to be sure that it contains the right answer. As a result, early prototype versions of Tail Answers were much too long and of poor quality (Figure 4).

One popular quality control technique is to generate a set of potential responses and ask workers to vote on which is the best. For example, we asked three different workers to copy and paste text from the web page and then had five other workers vote to select the best extraction. However, if there are no short extractions, the answer will be long; worse, workers tend to vote for long extractions.

So, it is necessary to add another layer of quality control to help guarantee that the extractions are short and targeted. We adapt the gold standard technique, which requires workers to demonstrate competence by agreeing with the answers to pre-authored example questions for each job [12]. Crowdflower uses gold standard testing by silently inserting gold standard questions into the worker's stream, and only keeps work from people who answer at least 70% of the gold standard questions correctly. Most gold standard tasks involve workers exactly matching the requester's input. For example, for voting we can enforce that workers agree with the authors' selection of which option is the best.

Unfortunately, requiring exact agreement fails for open-ended tasks like extraction. There are often several valid extractions for a page, and it can be just as important to specify which text workers should *not* include. To address this issue, we introduce *inclusion/exclusion lists* for gold standard testing for text generation. To use an inclusion/exclusion list for page extraction, the requester identifies sections of the page that *must* be in the extraction, as well as sections of the page that *must not* be in the extraction, in order for the work to be accepted. By doing so, we are able to tightly scope the areas of the page that are off-limits, as well as information that must be included in the answer for it to be correct. Figure 4 is a representative example of how training workers using inclusion/exclusion gold leads to shorter, more targeted answers.

We implement this technique using negative look-ahead in regular expressions. We also use inclusion/exclusion gold

in the title generation step, making sure that workers submit relevant phrases or words and that they do not copy and paste queries verbatim. Inclusion/exclusion gold standards could be useful for other open-ended crowdsourcing tasks like proofreading, replacing expensive approaches such as Find-Fix-Verify [4] as well as qualifier tasks, which cut down on the worker pool significantly.

Implementation

To generate a set of Tail Answers, we began with a one-week sample of browsing behavior from opt-in users of a widely-distributed browser toolbar starting March 22, 2011. We filtered the sample to users in the US who use English when searching. The resulting search trails represent over 2 billion browse events from over 75 million search trails for over 15 million users. We filter pages with too little data by removing ones that have been clicked fewer than three times. Filtering via destination probability and question words resulted in 19,167 answer candidates, including those in the top four rows of Table 1.

The query and web page occurrences that make up the answer candidates are distributed similar to power laws, so there are a few pages with many queries and a large number of pages with our minimum of three queries. Answer candidates had a median of three queries (μ=5.2, σ=7.4), 37% of the unique queries contained question words, and the median query had only been issued once in the dataset (μ=7.37, σ=35.0). If each answer candidate were to receive the same number of queries every week for a year as it did during our sample week, the median answer would trigger 364 times per year (μ=1992, σ=6318).

We sampled 350 answer candidates from this set for which to create Tail Answers. We combined several different sampling methods in order to get broad coverage: 100 needs were chosen randomly from the dataset in order to represent the tail more heavily, and 250 were chosen by weighted query popularity to represent query volume.

The number of workers in each stage is a tradeoff between cost and quality. Based on previous work (e.g., [4,15]), we recruited three to five workers for extraction and voting. Three workers voted on whether each of the 350 information needs should be addressed by a short answer, a list answer, or a summary answer, for 4.2¢ per need. Of the 350 needs, one hundred forty six (42%) were short phrase answers, one hundred twenty seven (36%) were short list answers, and seventy seven (22%) were summary answers. We focus here just on the short phrase answers, although the process is identical for short list answers and the results are similar. Three workers created extractions for each need (7¢), and five workers voted on the best extraction (10¢). Ten of the 146 answers were voted out by workers for having no good extractions. Of the remainder, three workers proofread the extraction (9¢), and three workers voted on the best alternative (6¢). Three workers authored potential titles (4.2¢), and three workers voted on the best title and filtered the answer if none were appropriate (4.2¢).

At the end of the process, 120 of the 146 short answer candidates became finalized Tail Answers. A number of examples are shown in Figure 2. The cost per answer was 44.6¢ plus a small extra fee for Crowdflower and the expense of the partial results for answers that got voted out. If we were to build Tail Answers for each of the roughly 20,000 candidates in our dataset, it would cost roughly $9,000. This cost can be lowered by combining extraction and title authoring into one task.

EVALUATION

In this section, we aim to better understand Tail Answers. Using manual judgments, we show they are high quality and relevant. We then present a controlled user study that shows that Tail Answers significantly improved users' ratings of search result quality and their ability to solve needs without clicking. To remove a source of variation in these evaluations, we focus on the short answers only.

Answer Quality

We first ask whether Tail Answers are high quality. This question has several dimensions: correctness, writing quality, query accuracy, and whether major search engines already have an answer to address the need. We hand-labeled each of the answers with whether the title or the content had writing errors, whether the answer was correct, whether a major search engine already had such an answer, and whether the answer addressed each query in its training set. Two authors labeled each answer; any disagreements were settled by a third rater.

We found that most Tail Answers had high-quality writing in their title and their content (Table 2). Of the titles with writing errors, workers had suggested a correct version 50% of the time, but it had been voted down. Likewise, 30% of the contents with an error had a correct version available, but the workers did not vote for it.

Correctness was more variable: some common errors are displayed in Table 3. Over two thirds of the Tail Answers were judged fully correct (Table 2). A common minor error (18.3%) occurred when the title did not match the answer: workers who wrote the answer title sometimes paid attention to the original queries rather than the content of the answer. This could be addressed through improved interfaces for the workers and more rigorous quality control in voting. (About 45% of the incorrect answers had a correct version extracted that was not the winner of the popular vote.) Other problems occurred for dead links (i.e., the data could not be extracted) and for dynamic pages (e.g., a "What's My IP?" application and YouTube videos), where workers were unable to signal that the page had no useful information. Two changes would help Tail Answers' accuracy: 1) identifying when dynamic content would make an answer impossible to build, and 2) better quality control to make sure titles are on-topic in the voting stage, since they are written after the answer content.

	High Quality	Minor Error	Major Error
Title Writing	83.3%	14.2%	2.5%
Content Writing	82.5%	14.2%	3.3%
Correctness	68.3%	18.3%	13.3%

Table 2. Hand-labeled writing and correctness ratings.

Low-Quality Tail Answer	Problem
Resume Writing A Curriculum Vitae, commonly referred to as CV, is a longer (two or more pages), more detailed synopsis. It includes a summary of your educational and academic backgrounds as well as teaching and research experience, publications, presentations, awards, honors, affiliations and other details.	Title does not match the answer
Cary Grant Cary Grant was born on January 18, 1904.	Title does not match the answer
What Reallyhappens.com Most recent WRH radio show from Rense Radio.	Dynamic page has no useful text to extract
Double Irish Tax The Double Irish method is very common at the moment, particularly with companies with intellectual property.	Extracted text is too general

Table 3. Examples of common errors in Tail Answers.

Fourteen percent of the Tail Answers we generated already had answers available on Bing, a major search engine. Unit conversions (e.g., mL in a tablespoon) were the most common, followed by weather, definitions, and dates. These answers could be filtered in a deployed system, or could be used to replace manually generated answers, which are expensive and time consuming to maintain.

We investigated how closely the answers matched the apparent intent of the queries that represented the intent. (Many queries, like *chi 2012*, may not express the searcher's full intent.) In 58% of the unique queries, it was clear that the Tail Answers addressed the query's intent. About 7% of queries were more general than the answer (e.g., the query was *az municipal court* and the answer gave the phone number to the court), so it is difficult to know whether the answer would have satisfied the information need. Likewise, 23% of queries were generally related to the answer, and the judgment would depend on the exact intent (e.g., a query for *B.C.E.* was associated with an answer for *C.E.*, the Common Era). About 12% of the unique queries were not good matches: about 9% of the queries expressed a more specific need than the answer had (e.g., the query was *fredericksburg VRE* [Virginia Railway Express] but the answer focused on the entire VRE), and about 3% of queries were unrelated to the answer. Often, pages like *C.E.* covered multiple information needs, but workers had to choose just one need for the answer. Clustering these queries into overlapping keyword sets and building separate answers for each would help.

User Evaluation

We also wanted to understand whether Tail Answers positively or negatively impact users' impressions of the search engine result page. In particular, we wanted to know whether Tail Answers improved users' subjective impressions of search results, and whether Tail Answers could compensate for poorer search rankings.

Method

We recruited 361 people (99 female, 262 male) at Microsoft to participate in our study. Most were in their 30s (30%) or 40s (42%), and used search engines hourly (58%) or daily (41%). About 30% held nontechnical jobs. Participants could complete the study from their own computers, and we raffled off $25 gift certificates in return. Participants did not know the purpose of the experiment.

We created a custom version of the Bing search engine that inserted Tail Answers at the top of the search results whenever the user issued a matching query. We gathered a sample of thirty Tail Answers from the 120 we created. Participants were shown five queries, each taken from a randomly chosen Tail Answer, and chose one they found interesting. Participants were required to invent reasons they would issue each query, which is less realistic than showing the Tail Answer when someone has the real information need. However, by giving participants a choice of queries, we hoped they would focus on more personally meaningful tasks. After choosing a query, participants were shown the result page and asked for their level of agreement on a seven point Likert scale with two statements about the search results: 1) "This is a very useful response for the query," and 2) "This page contains everything I need to know to answer the query without clicking on a link."

Our experiment used a two-by-two research design. Each query was randomly assigned either to the *Answer* condition, which displayed a Tail Answer, or to a *No Answer* condition, with no answer. It was also randomly assigned either to the *Good Ranking* condition, where the search engine displayed results ranked 1 through 10, or a *Bad Ranking* condition, which displayed results ranked 101 through 110. In the Bad Ranking condition, the search results were typically much poorer. All conditions appeared to return top-ten results, and we hid ads and other answers. Participants would see each of the conditions randomly as they rated new queries, and were required to rate at least ten queries to be entered in the lottery. At the conclusion of the study, participants filled out a final survey.

We hypothesized that Tail Answers would improve the user experience of the search engine. However, we were also interested in how users would react when Tail Answers fired on inappropriate queries or had incorrect results.

Results

Participants rated 3963 result pages. Mean ratings are reported in Table 4 and Table 5. To analyze the results, we used a linear mixed effects model, which is a generalization of ANOVA. We modeled participant, and query (nested in answer), as random effects. Ranking and answer were fixed effects. We also included an interaction term for ranking*answer. This model allowed us to control for

	Tail Answer	No Tail Answer
Good Ranking	5.81	5.54
Bad Ranking	5.12	3.73

Table 4. Mean Likert scale responses to: "This is a very useful response for the query."

	Tail Answer	No Tail Answer
Good Ranking	5.06	4.10
Bad Ranking	4.54	2.66

Table 5. Mean Likert scale responses to: "This page contains everything I need to know to answer the query without clicking on a link."

variation by answer, query, and user in our analysis. Finally, because participants were more likely to choose certain queries in our dataset, we weighted the observations so that each answer was represented equally in the data. Weighting observations is a common technique when the sample distribution does not match the population; removing the weighting produces very similar results, but we felt that weighting would be the most accurate way to represent all answers equally. We ran the model twice, once for the first Likert scale (1) overall subjective opinion of the result page, and once with the second Likert scale (2) ability to solve the information need without clicking a link.

Tail Answers and result ranking both had significant effects on overall rated result usefulness (Table 4). In the statistics to come, we note that weighting the sample leads to non-integer degrees of freedom. Tail Answer appearance, $F(1, 4307.8) = 292.0$, $p < .001$, had an estimated effect of 0.34 points on result usefulness. Good ranking, $F(1, 4306.0) = 570.6$, $p < .001$, had an estimated effect of 0.68 points on result usefulness. Result ranking, which is central to search engines, had an effect size just twice the effect size of Tail Answers: 0.34 vs. 0.68. The interaction was significant, $F(1, 4309.95) = 106.5$, with an estimated effect size of 1.03 points. The large interaction effect indicates that answers are particularly helpful when search results are poor.

Tail Answers were also useful at solving information needs without needing to click through to a result (Table 5). The addition of Tail Answers to the search results, $F(1, 4293.0) = 631.4$, $p < 0.001$, had an estimated positive effect of 1.01 points on users' rating. Good ranking, $F(1, 4291.4) = 270.3$, $p < 0.001$, had a smaller effect of 0.50 points on users' ratings, and the interaction term remained large: $F(1, 4295.8) = 60.49$, $p < 0.001$, effect size of 0.91 points. The study design removed other answers from the search results in order to control for variation. It is possible that our effect sizes would be smaller if other answers were included.

Overall, the inclusion of Tail Answers had a positive effect on users' search experience as reflected in their ratings. The impact of Tail Answers was nearly half as much as result ranking, where search engines focus much of their effort. That positive effect was more than doubled when participants were asked whether they needed to click through to a URL. Answers were able to fully compensate for poorer search results, suggesting that a single answer can be as important as good search engine ranking.

Survey Feedback

Participants filled out the survey at the completion of the experiment and provided feedback on the writing, correctness, and usefulness of Tail Answers. Participants found Tail Answers useful ($\mu=5.8$ / 7, $\sigma=1.4$), especially for directed, fact-oriented queries. For many of these queries, Tail Answers addressed the information need directly in the search results. A common theme in the responses was, "it told me exactly the right answer to my question." Participants were enthusiastic that a search engine could answer such unstructured queries. Most participants did not suspect that the Tail Answers were being human-edited.

While participants generally thought the answers were accurate ($\mu=5.3$, $\sigma=1.4$) and well-written ($\mu=5.4$, $\sigma=1.4$), relevance was a challenge. The crowd tended to create Tail Answers based on the most visible or understandable need in the query logs. When there were multiple information needs on a single URL, the answer would not cover all queries. For example, the only query with clear intent about the Phoenix Municipal Court asked about the court's phone number, so the answer was built around the phone number. However, that answer did not completely address more general queries like *phoenix municipal court*. In other cases, participants pointed out that the Tail Answer covered the high-level concept but did not have enough detail to fully satisfy their information need. In the future, we believe that it will be important to better target queries either by using the crowd to filter the set of trigger queries, or by A/B testing and measuring click cannibalization [7].

Some participants trusted Tail Answers implicitly, and others wanted more information about sources. Because Tail Answers look like they are endorsed by the search engine, we are particularly sensitive to accuracy and trust.

Generally, participants felt that Tail Answers were concise and well-written. We view this as a success, because extractions in earlier iterations on Tail Answers were much too long. The crowd-authored text had direct readability benefits: one participant remarked that Tail Answers avoided the ellipses and sentence fragments common in search result snippets. Participants occasionally requested richer structure, such as tables and images.

DISCUSSION

We have shown that search engines can cheaply and easily answer many of searchers' fact-finding queries directly. We presented evidence that Tail Answers can improve the user experience, often roughly as significantly as search result quality. Although search engines have used large-scale log data and paid judges to improve search result ranking, our findings suggest that there are new ways human effort can be applied to re-envision the search user experience.

Question Query	Algorithmic Result	
	Accepted	Rejected
What is a substitute for molasses?	brown sugar, honey	baking, recipes
What is the cost of mailing letters in the US?	44¢ to 39¢	12, 37¢, mail
Where is area code 559?	State of California	Selma CA, Clovis
How much nicotine is in a light cigarette?	Low density, 6mg	milligrams, 14mg

Table 5. Here, an automated question-answering system proposed Tail Answers and crowds filtered them.

Figure 5. The Tail Answers crowd extraction algorithm (bottom) can suggest replacements for result snippets (top).

Challenges

Because Tail Answers are presented in a way that appears authoritative, they can potentially spread incorrect or misleading information without oversight. Even simple errors like triggering a Tail Answer on the wrong query can undermine people's trust in the search engine; our evaluation suggested that trimming the query trigger list is an important step for making Tail Answers deployable.

Tail Answers may be particularly tempting targets for search engine spam because of the authority they carry. With Tail Answers, a few members of the crowd would have significant direct control over search results by including advertisements or misinformation. However, a small group of trusted individuals could check for these problems and send answers back if there are problems.

Like result snippets, Tail Answers extract information from web pages and present that content to searchers. Unlike snippets, however, the intent behind the extraction is to fully address the searcher's information need, rather than to direct the searcher to the page. In this way, Tail Answers cannibalize page views. But without the underlying web content, the answers would not exist. To incentivize content providers, one option may be for the search engine to redirect a portion of the query's advertising revenue to pages that provide valuable content. Search engines will continue walking the line between attributing sources and highlighting the most useful information from that source.

Extensions: A.I., Snippets, and More Answer Types

Despite the challenges, we believe that the insight gained through Tail Answers can deeply extend the vocabulary of search interfaces. We have prototyped several extensions and share some early results in this section.

Artificial Intelligence-Driven Information Extraction

To extract content from web pages and turn that content into an answer, we used paid crowdsourcing. As

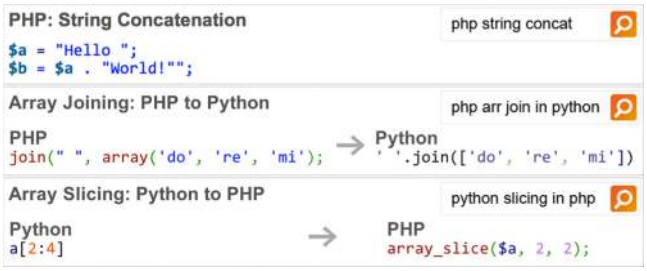

Figure 6. Code tutorial answers. Within a domain, Tail Answers like these can specialize their user interface.

technologies advance, this balance may shift: automatic systems may assume more or all of the responsibility. Our experiments with automatic systems such as AskMSR [5] and TextRunner [3] suggest that they produce too many poor guesses to be useful. However, a hybrid approach that uses the crowd to vet the answers provided by machine intelligence could be cheap and accurate. To explore this, we connected the AskMSR question-answering system to our dataset of Tail Answer queries, and asked it to generate candidate answers for the question queries. We then used the crowd to vote whether each answer was correct. Table 5 demonstrates early results, for example returning "brown sugar" as a substitute for molasses while filtering out highly-rated false positives like "baking". This vote was much cheaper than paying for extraction and proofreading.

Smart Snippets for Popular Queries

In addition to standalone answers, the crowd can help with snippets, the short page summaries that appear underneath the page title in search results. Instead of tail needs, popular queries are a good match for snippet improvement because they are seen by a large number of searchers. In particular, we focus on popular queries that have high click entropy (i.e., people click on many different results for the query). Queries like *wallpaper* have high click entropy because they have multiple meanings (e.g., computer desktop art versus home wall decoration), and searchers may not have enough information scent [16] in the snippets to make good choices. We can use the extraction routine from Tail Answers to find snippets for these sites. Figure 5 demonstrates the resulting improvements to a high-visibility search snippet for the query *wallpaper*.

New Classes of Answers

We have thus far explored short and list-style answers, but there are many more possible answer types that could be developed with our approach. For example, answers could be created to help users achieve high-level goals like creating a website or planning a vacation to Yosemite [11]. They could also summarize web content, automatically create answers for spiking queries or news stories, or even connect searchers with other users who might be able to help solve their information need [9]. To create more sophisticated answers, we expect to transition from generic crowd workers in Mechanical Turk to more expert workers like those found on oDesk. We could also give other searchers the ability to edit the answer, much like

Wikipedia. The amount of effort and cost could be applied differentially, based on potential gain, with more invested in more popular or high impact information needs.

Because Tail Answers are general-purpose, it is impossible to provide custom user interfaces. However, if we focus on a particular set of information needs, we can build special user interfaces and data extraction requirements. Figure 6 shows example answers we have built for translating commands between programming languages, for example understanding how to translate PHP's array join syntax into Python. We began with a list of programming primitives in Python, then asked workers to volunteer the mapping into PHP. With this mapping, the Tail Answers can return results for functions in either language, as well as translate between the languages, with a specially designed interface.

Destination probability can also help identify new kinds of answers. For example, pages with telephone area codes tended to have high destination probability. Armed with this information, search engines might start building answers specifically for area code queries.

CONCLUSION

Search engines increasingly aim to return information rather than links. Search companies devote significant resources to build a small number of inline answers for topics like weather and movies. Unfortunately, most information needs are unlikely to ever trigger answers. In response, we have introduced Tail Answers: succinct inline search results for less frequent and extremely varied information needs. To build Tail Answers, we draw on the aggregate knowledge of thousands of web users. We mine large-scale query logs for pages that tend to end search sessions, select candidates where searchers have used information key terms like question words, and use paid crowds to remove candidates that cannot be answered succinctly. Finally, crowds extract the information from the web page, edit it, and title it. Our evaluation of Tail Answers demonstrates that they can significantly improve the search user experience and searchers' ability to find the information they are looking for without navigating to an external web page. We demonstrate the generalizability of these techniques by prototyping ways they could be used to improve other aspects of the search engine interface.

REFERENCES

1. Adar, E., Teevan, J., Dumais, S.T., and Elsas, J.L. The web changes everything. *Proc. WSDM '09*, (2009).
2. Agichtein, E., Lawrence, S., and Gravano, L. Learning to find answers to questions on the Web. *ACM TOIS 4*, 2 (2004), 129-162.
3. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., and Etzioni, O. Open information extraction for the web. *IJCAI '072*, University of Washington (2007).
4. Bernstein, M.S., Little, G., Miller, R.C., et al. Soylent: A Word Processor with a Crowd Inside. *Proc. UIST '10*, (2010).
5. Brill, E., Dumais, S., and Banko, M. An analysis of the AskMSR question-answering system. *Proc. EMNLP '02*, (2002).
6. Broder, A. A taxonomy of web search. *ACM SIGIR Forum 36*, 2 (2002), 3.
7. Chilton, L.B. and Teevan, J. Addressing people's information needs directly in a web search result page. *Proc. WWW '11*, (2011).
8. Cutrell, E. and Guan, Z. What are you looking for?: an eye-tracking study of information usage in web search. *Proc. CHI '07*, (2007).
9. Horowitz, D. and Kamvar, S.D. The anatomy of a large-scale social search engine. *Proc. WWW '10*, (2010).
10. Kellar, M., Watters, C., and Shepherd, M. A field study characterizing Web-based information-seeking tasks. *JASIST 58*, 7 (2007), 999-1018.
11. Law, E. and Zhang, H. Towards Large-Scale Collaborative Planning: Answering High-Level Search Queries Using Human Computation. *Proc. AAAI '11*, (2011).
12. Le, J., Edmonds, A., Hester, V., and Biewald, L. Ensuring quality in crowdsourced search relevance evaluation. *Proc. SIGIR '10 Workshop on Crowdsourcing for Search Evaluation*, (2010).
13. Li, J., Huffman, S., and Tokuda, A. Good abandonment in mobile and PC internet search. *Proc. SIGIR '09*, (2009).
14. Lin, J. An exploration of the principles underlying redundancy-based factoid question answering. *ACM TOIS 25*, 2 (2007).
15. Little, G., Chilton, L., Goldman, M., and Miller, R.C. Exploring iterative and parallel human computation processes. *Proc. HCOMP '10*, (2010).
16. Pirolli, P. *Information foraging theory: adaptive interaction with information*. Oxford Press, 2007.
17. Stamou, S. and Efthimiadis, E.N. Queries without clicks: Successful or failed searches. *Proc. SIGIR '09 Wkshp on the Future of IR Evaluation*, (2009).
18. Suchanek, F.M., Kasneci, G., and Weikum, G. YAGO : A Core of Semantic Knowledge Unifying Wikipedia and WordNet. *WWW '07*, (2007).
19. Teevan, J., Liebling, D.J., and Ravichandran Geetha, G. Understanding and predicting personal navigation. *Proc. WSDM '11*, (2011).
20. White, R.W., Bilenko, M., and Cucerzan, S. Studying the use of popular destinations to enhance web search interaction. *Proc. SIGIR '07*, (2007).