



Delft University of Technology

## TaskGenie

### Crowd-Powered Task Generation for Struggling Search

Xu, Luyan; Zhou, Xuan; Gadiraju, Ujwal

#### DOI

[10.1007/978-3-030-62008-0\\_1](https://doi.org/10.1007/978-3-030-62008-0_1)

#### Publication date

2020

#### Document Version

Final published version

#### Published in

Web Information Systems Engineering

#### Citation (APA)

Xu, L., Zhou, X., & Gadiraju, U. (2020). TaskGenie: Crowd-Powered Task Generation for Struggling Search. In Z. Huang, W. Beek, H. Wang, Y. Zhang, & R. Zhou (Eds.), *Web Information Systems Engineering: WISE 2020 - Proceedings* (pp. 3-20). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 12343 ). Springer.  
[https://doi.org/10.1007/978-3-030-62008-0\\_1](https://doi.org/10.1007/978-3-030-62008-0_1)

#### Important note

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

#### Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

#### Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

***Green Open Access added to TU Delft Institutional Repository***




***'You share, we take care!' - Taverne project***

**<https://www.openaccess.nl/en/you-share-we-take-care>**

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.



# TaskGenie: Crowd-Powered Task Generation for Struggling Search

Luyan Xu<sup>1</sup>(✉) , Xuan Zhou<sup>2</sup> , and Ujwal Gadiraju<sup>3</sup> 

<sup>1</sup> Renmin University of China, Beijing 100872, China  
xuluyan@ruc.edu.cn

<sup>2</sup> East China Normal University, Shanghai 200062, China  
zhou.xuan@outlook.com

<sup>3</sup> Delft University of Technology, Delft, The Netherlands  
u.k.gadiraju@tudelft.nl

**Abstract.** Search tasks provide a medium for the evaluation of system performance and the underlying analytical aspects of IR systems. Researchers have recently developed new interfaces or mechanisms to support vague information needs and struggling search. However, little attention has been paid to the generation of a unified task set for evaluation and comparison of search engine improvements for struggling search. Generation of such tasks is inherently difficult, as each task is supposed to trigger struggling and exploring user behavior rather than simple search behavior. Moreover, the everchanging landscape of information needs would render old task sets less ideal if not unusable for system evaluation. In this paper, we propose a task generation method and develop a crowd-powered platform called *TaskGenie* to generate struggling search tasks online. Our experiments and analysis show that the generated tasks are qualified to emulate struggling search behaviors consisting of ‘repeated similar queries’ and ‘quick-back clicks’, etc. – tasks of diverse topics, high quality and difficulty can be created using this framework. For the benefit of the community, we publicly released the platform, a task set containing 80 topically diverse struggling search tasks generated and examined in this work, and the corresponding anonymized user behavior logs.

**Keywords:** Web search · User interaction · SERPs · User behavior

## 1 Introduction

Modern search engines are adequately equipped to help users in locating accurate information for well-defined queries. Nevertheless, web searchers still experience difficulty in finding relevant information when their information need is ill-defined, complex or complicated. Therefore, recent work has paid more attention to understand and support struggling search with an aim to help searchers cope with the entailing search difficulty. The notion of *struggling sessions* was first

formally introduced by Hassan et al. as “those search sessions where users experience difficulty locating required information” [12]. Task difficulty is the main factor that leads to struggling search [2, 6]. *Struggling search tasks* are defined by “topically coherent sub-sessions in which searchers cannot immediately find sought information” [22]. Researchers focusing on user behavior analysis found that whether a user is handling a struggling search task can be identified and predicted through features of his/her search activities such as queries and post-query clicks [2, 10]. Building on these outcomes, previous works proposed different support mechanisms and systems to detect and ameliorate struggling search cases [12, 13, 21, 29].

However, little attention has been paid to generating a unified set of tasks in this area. Akin to the role that TREC datasets play in typical information retrieval (IR) research, the generation of struggling search tasks is particularly important for further development and unified evaluation of new techniques in struggling search. Nevertheless, the ever-changing landscape of complex information needs would render old task sets less ideal if not unusable for system evaluation. Currently, for task generation related to struggling search, researchers tend to create struggling tasks manually by increasing task complexity, e.g. “*There are five countries whose names are also carried by chemical elements. France has two (Ga–Gallium and Fr–Francium), ... Please name the left country*” [25]. Others adhere to small-scale situated lab experiments, e.g. “*You once heard that the Dave Matthews Band owns a studio in Virginia but you don’t know the name of it. The studio is located outside of Charlottesville and it’s in the mountains. What is the name of the studio?*” [2].

These methods require extensive experience and fertile imagination of researchers and since there is no common pattern to follow, these may lead to only small-sized task sets. Though studies have shown that small task sets could work well in some experimental lab studies [2, 25], they are not sufficient for large-scale and robust system evaluation. The potential effect of participant fatigue limits laboratory experiments to a small number of topics and similar situated tasks, making the evaluation inclined to side with a subjective or biased perspective [27]. This dictates the need for a robust and cost-efficient method to generate struggling search tasks (SSTs) for evaluation. Crowdsourcing has been shown to be a powerful means for recruiting low-cost participants who are readily available around the clock [8, 9]. This provides us with an alternative source of acquiring reliable human input. We thereby propose the use of crowdsourcing to generate SSTs.

**Original Contributions.** In this paper, we focus on struggling search that manifests in fact finding or checking tasks. We propose a crowd-powered task generation framework and develop an online platform<sup>1</sup> that can be used to generate high-quality SSTs at scale. This method leverages paraphrased (redundant) information in wikis, and decompose SST task generation into several low-effort steps, suitable for crowd workflows to create questions that are difficult and can

<sup>1</sup> <http://waps.io/study/?uid=123>.

simulate struggling search. This method can easily be applied to topically dedicated wikis (e.g. **wikinews** for news, **wikivoyage** for travel, etc.)<sup>2</sup>, while in this paper we take English Wikipedia as the resource to generate a topically diverse set of SSTs. Applying crowd-workers, we generated 80 SSTs across diverse topics. Getting insights from previous studies [12, 13, 22], we evaluate the quality of these tasks by carrying out rigorous user-centric experiments, analyzing the characteristics of user behaviors elicited by these tasks. Results confirm the quality of the generated SSTs. We consolidated the tasks and publicly released the task set<sup>3</sup> containing 80 SSTs with difficulty level and success rate, which can help in developing and evaluating support mechanisms for users in struggling search. Also, we released the anonymized user logs gathered during task evaluation.

## 2 Related Literature

We discuss related work in the following areas: struggling search and task design for struggling search.

**Struggling Search.** Struggling search describes a situation whereby a searcher experience difficulty in finding the information they seek [12]. Within a single search, struggling could lead to frustrating of difficulty and dissatisfying search experiences, even if searchers ultimately meet their search objectives [11]. Characteristics of user behaviors have been used to identify whether a user was dealing with struggling search tasks – searchers dealing with a struggling search tasks can experience difficulty in locating required information, tend to issue multiple similar queries and conduct quick-back clicks as they are cycling on finding useful information [2, 11]. Struggling search has been studied using a variety of experimental methods, including log analysis [22], laboratory studies [2], and crowdsourced games [1]. Hassan et al., studied how to detect and support struggling search by extracting search sessions from real user logs [13, 22]; Aula et al. evaluated the influence of task difficulty on struggling search behaviors by setting up a small-scale lab experiment and an IR-based online study [2]. We evaluate the quality of generated tasks by analyzing the user behaviors elicited by the tasks, based on the behavior features that have been shown to be useful for identifying struggling search [11, 13].

**Task Design for Struggling Search.** Researchers in sense-making found that users will suffer difficulty when there is an information gap between what they know and what they want to know [23], as they can seldom describe their questions clearly or find a way to get close to the answer. This sheds light on task design for struggling searching tasks; key information or the task solving strategy should not be directly given by the task. Also, it has been found that task complexity can increase the task difficulty thus affect learner perceptions of struggling [24]. On the other hand, difficulty of tasks has been viewed from both objective and subjective perspectives [18]. From the subjective perspective, the

<sup>2</sup> Wikinews: <https://en.wikinews.org/>; wikivoyage: <https://www.wikivoyage.org/>.

<sup>3</sup> Anonymized URL– <https://github.com/sst20190816/WISE2020>.

same task could be difficult and complex to one without background knowledge while be easy for the other who is an expert in the related domain [2, 5]. To some extent this indicates that task design for struggling search should either try to avoid the cases that are highly influenced by domain knowledge or try to cover as many topics as possible. From the objective perspective, task difficulty can be related to task characteristics and independent of task performers, which has been supported by other works [6] – task with unknown goals, unexplored information space, accompanied by uncertainty and ambiguity would consequently mean that it could lead to a high task complexity, in turn resulting in users struggling [18]. Getting inspiration from previous work [30], we propose an online task generation framework for generating struggling search tasks at scale, covering various knowledge domains and are objectively difficult.

### 3 Task Generation Framework

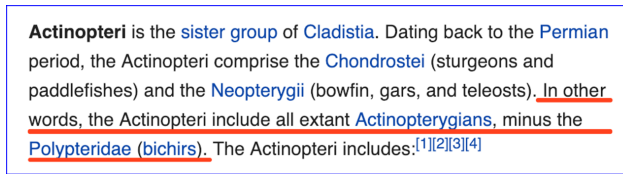
#### 3.1 Intuition and Method

We focus on a particular type of search tasks that exhibit search behavior suggestive of struggling – fact finding/checking tasks (“Looking for specific facts or pieces of information” [14]). Struggling search tasks (SST) differ from typical information retrieval tasks in that the typical informational search tasks are more like information locating problems which are well-defined, systematic and routine [28]. For example, consider the following struggling search task—*“Dave Matthews Band owns a studio in Virginia, the studio is located outside of Charlottesville and it’s in the mountains. What is the name of the studio?”* [2]. Consider that the answer to this question does exist in the document collection, but it cannot be simply matched to search queries or resolved using the state-of-the-art information retrieval techniques. Rather, it can only be described using fragmented pieces of information and obtained by searchers through navigating and comprehending content within the information space. A searcher needs to collect relevant information from the documents, comprehend it, reason about it, and very often repeat the process for several rounds, until he/she reaches a conclusion with a certain confidence. This process involves information-seeking behavior, including searching, browsing, berry-picking and sense-making [20].

**How Can We Easily Find or Frame Questions with Implicit Answers at Scale?** In this paper, we leverage paraphrased sentences, which are abundant in common writings. To create a clear and logical flow while writing, an author tends to perform reasoning narratively. This naturally results in redundancy [7]. For instance, a statement following a causative sentence connector (i.e. a conjunctive adverb) [17], such as *“in other words”* or *“that is to say”*, is likely to be a paraphrase which repeats the same meaning of the former sentence(s) in a more colloquial manner [4]. In theory, the information conveyed by the paraphrased sentences can be recovered by a searcher who has read through the preceding content. Thus, removing the paraphrased sentence will not cause information loss. The sentences following such connecting phrases are typically declarative

statements. It is therefore straightforward to turn them into questions, with the statement containing the answer.

For example, in Fig. 1, we can hide the underlined sentence and turn it into a question – “Does *Polypteridae* belong to the *Actinopteri*?” – (since ‘Polypteridae’ and ‘Actinopteri’ appear elsewhere in the article in different forms). By hiding the specific sentence that contains the answer, the answer will not be directly identifiable through information locating. A searcher may identify text fragments like ‘Polypteridae’ and ‘Actinopteri’ as their starting points. However, to understand their relation and answer the question, the searcher may need to know more and therefore be forced to explore the Web or Wikipedia further.



**Actinopteri** is the [sister group](#) of [Cladistia](#). Dating back to the [Permian](#) period, the Actinopteri comprise the [Chondrostei](#) (sturgeons and paddlefishes) and the [Neopterygii](#) (bowfin, gars, and teleosts). In other words, the Actinopteri include all extant [Actinopterygians](#), minus the [Polypteridae](#) (bichirs). The Actinopteri includes:<sup>[1][2][3][4]</sup>

**Fig. 1.** Example of a paraphrased sentence in Wikipedia.

This inspires us to generate SSTs through the following steps:

1. Identify a paraphrased sentence;
2. Hide it from the document;
3. Create an informational question based on the given paraphrased sentence.

Since the answering sentence is hidden from the document, it is hard to obtain the answer through direct information locating; the paraphrased sentence usually lacks an accurate description or explanation of the entailing information points, a task generated based on it simulates a real-life situation where people have incomplete prior knowledge or means to meet their information need. This will elicit a searcher’s struggling search behavior. The searcher may start from arbitrary documents that seem relevant, browse through parts or the whole collection, and reason about the possible answer. If the searcher is unfamiliar with the topic, he has to learn about it, since answering the question would require comprehension of related knowledge. Meanwhile, as the hidden sentence contains only redundant information, the searcher should be able to find the answer eventually.

### 3.2 TaskGenie – A Crowd-Powered Platform

Based on the task generation method, we built an online platform for task generation called *TaskGenie*, aiming to (i) generate struggling search tasks through crowdsourcing; (ii) study user behavior within the generated struggling search

tasks. To this end, this platform serves in two phases: *Task Generation*, facilitating the creation of new struggling search tasks; and *Task Completion*, facilitating search experiment on solving the tasks.

**Task Generation.** For task generation, users are first guided to choose a conjunctive phrase from a drop-down list (e.g. ‘*in other words*’, ‘*that is to say*’). They are then presented with a filtered set of articles that contain (highlighted) statements with these conjunctive phrases. Users are asked to understand the highlighted sentence in the article context and grasp the information that the sentence contains. Finally, they are asked to create a question based on the paraphrased sentence, provide the answer and source page of the question. Assuming that a task generated from a paraphrased sentence is closely related to its surrounding context, we automatically save the paraphrased sentence and its context (i.e. the two sentences ahead of the positions of the paraphrase sentence) as the supporting information for the answer to the generated question.

**Task Completion.** We present the users with a generated task in the form of a question that can be answered using a search engine. All tasks are pulled randomly from our database while the background mechanism ensures each task is finally resolved equal times. Users can choose to change the task only once; if they do not like the task assigned to them. Users are tasked with finding the answer to the question by searching using our search engine. To ensure that the users are genuinely invested in reasoning, understanding and finding the correct answer and not merely guessing, we ask users to provide a justification in an open text field that supports their answer. Users are encouraged to copy-paste excerpts that provide evidence or justify their answers. Finally, we collect the users’ opinions of the search task they completed from the following perspectives – (a) *Task Qualification* (whether or not the users found the question difficult in comparison to their usual experience of searching the Web or Wikipedia for answers); (b) *Task Difficulty Score* (how difficult/complex the users found the question to be). We divide the task difficulty scale into five equal parts using the following labels with corresponding score intervals on a sliding scale of 1 to 100 - *Easy* (1–20), *Moderate* (21–40), *Challenging* (41–60), *Demanding* (61–80), *Strenuous* (81–100). Users could select the task difficulty level and indicate an exact score using the scrollbar. Next we asked the users to indicate the reasons due to which they found the question to be difficult, and provided options (using checkboxes) that were drawn from previous work analyzing struggling search [19]. To prevent forced choices in case users did not find the task to be difficult, they could select the checkbox with the label ‘Not Difficult’.

### 3.3 System Implementation

**Pluggable Web Search Engines.** As a platform for task generation and evaluation, *TaskGenie* is designed to be compatible with main stream web search engines (e.g. Google, Bing) which provide a standardized search API. These search engines can be plugged into TaskGenie as a backend search system to



support task generation and get evaluated in task completion. In this paper, Bing Web Search API is used in the experimental study.

**Domain Controlling and User Activity Logging.** *TaskGenie*, the search domain can easily be adjusted to support searching through different domains. For example, we set Wikipedia as the domain for task generation (i.e. get all the webpages containing paraphrased sentences from Wikipedia) and we set the entire web as the domain for task completion. During task generation and completion, we also logged worker activity on the platform including queries, clicks, key presses, etc. using PHP/Javascript and the jQuery library.

**DOM Processing.** During the task generation phase, it is useful to highlight paraphrased sentences to make it more convenient for searchers to locate a target sentence. During the task completion phase on the other hand, to emulate a struggling search situation, it is essential to hide the direct answers in the retrieved documents. So that in the two phases, we need to either highlight or hide the paraphrased sentences. Drawing inspiration from previous work<sup>4</sup>, we implement this by filtering and manipulating DOM using Javascript. Given a retrieved webpage (DOM), we access all its child nodes recursively and match the regex of causative sentence connectors (*in other words* etc.) with the content of each node. The matched sentences are thereby either hidden or transformed into a different sentence according to their syntax.

## 4 Study Design

### 4.1 Task Generation

**Wikipedia – Paraphrased Sentences.** There are plenty of online archives or wikis. In this work, we choose Wikipedia as the source for our struggling search task generating framework, and *in other words* and *that is to say* as the conjunctive phrases to identify paraphrased sentences. Wikipedia is one of the richest sources of encyclopedic information on the Web, and generates a large amount of traffic. Prior work has highlighted the variety of factors that drive users to Wikipedia [26]. We explored the entire English Wikipedia (2018 version) and found 10,824 articles with on average one occurrence of the paraphrase “*in other words*”, and 2,195 articles with the paraphrase “*that is to say*”. Our findings suggest that Wikipedia is a good source for paraphrased sentences which can potentially serve in the creation of difficult search tasks across diverse topics.

**Task Generation Experiment.** We recruited 200 participants from Figure8<sup>5</sup>, a premier crowdsourcing platform. At the onset, workers willing to participate were informed that the task entailed ‘generating a task for others within the Wikipedia domain’. Workers were then redirected to the external platform, *TaskGenie*, where they completed the mission. We logged all worker activity

<sup>4</sup> <https://jilly.io/snippets/>.

<sup>5</sup> <http://figure-eight.com/>.

on the platform. During the *task generation* process, *TaskGenie* presents criteria to help a user control the quality of the generated question. We urge the users to ensure that (1) the selected sentence is a paraphrased sentence that contains enough information for creating a question; (2) they search for the answer on the Wikipedia to ensure that the generated question is challenging. This means that although the answer cannot be found easily, it can be eventually obtained through searching and exploring. We incentivize workers to strictly adhere to these criteria by rewarding workers with a post-hoc bonus payment if they successfully create a SST.

We restricted the participation to users from English-speaking countries to ensure that they understood the instructions adequately. On successfully creating a task, users received a mission completion code which they could then enter on the Figure8 platform to receive their monetary rewards. We compensated all users at an hourly rate of 7.5 USD ( $\approx 1.5$  USD and 12 mins per task).

**Task Collection.** To ensure the reliability of generated tasks, we filtered out workers in this phase using the following criteria:

- i. Workers who did not follow the required syntax in creating a question in the *task generation*. Since the aim of this phase is to generate a readable question (we described the basic syntax of a question in our instructions), those who did not meet the criteria were discarded.
- ii. Workers who create questions lacking a self-sufficient description in the way a question is phrased (for example, “*Reincarnation is possible?*”), and generate random questions ignoring the paraphrased sentence in the source page (for example, “*Is Wikipedia the best page to find anything?*”).

Using the aforementioned criteria, we filtered out 65 task generation cases, resulting 135 generated tasks. For the 135 generated tasks, we hired two students to search for the answer of each task on the web. We eliminated 55 tasks that either duplicated or for which the answer could be found within the two interactions with the search system. We finally got 80 tasks that qualified as struggling search tasks (SSTs).

## 4.2 Task Evaluation

To validate whether the generated tasks are struggling search tasks and are generally suitable for the study of struggling search, we conducted a web search experiment using the set of 80 generated tasks.

Through Figure8, we recruited 400 *Level-3 workers* (260 male and 140 female, with their age ranging from 18 to 57 years old). Workers willing to participate in the web-based task evaluation experiment were asked to “search for the answer of a given task” using our platform *Task-Genie: Task Completion*. For the web search experiment, we base the *TaskGenie* search system on top of the Bing Web Search API and extend the search domain to the entire web. We logged the user activity throughout the task completion. Using the task filtering criteria mentioned before, we filter out 31 spam workers who entered arbitrary strings

for the answer or supporting information, and those who did not finish the experiment. Thus, the evaluation is based on the 369 valid search sessions.

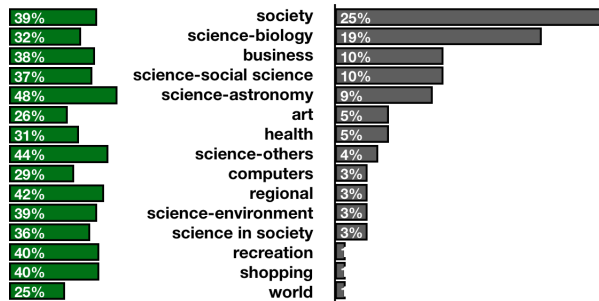
## 5 Task Evaluation Results

In task evaluation, we validate the generated tasks are struggling search tasks by examining session-level features of search behaviors shown to be useful for identifying struggling search in previous work [12, 22]: *topical characteristics*, *query characteristics*, *click characteristics* and *task difficulty*.

### 5.1 Topical Characteristics

We analyzed the topical distribution of the tasks and found that tasks in diverse topics could be generated through our task generation module. To categorized the generated tasks, we used the top-two-level categories of Curlie<sup>6</sup> (i.e. Open Directory Project; [dmoz.org](http://dmoz.org)). Assuming that the topic of tasks is consistent with the topic of its wiki source pages, we categorize the generated tasks by analyzing the topics of their source wiki pages. To this end, we used an automatic url-based classifier [3] for topic categorization. We assigned the most frequently-occurring topic for the source web page as the topic of each generated task.

Figure 2 shows the prevalence of topics in the generated tasks. We note that the task generation domain we chose, Wikipedia, contains few articles which correspond to everyday activities. Thus, only a few generated tasks were about topics spanning our daily lives such as *Shopping*, *Entertainment*, etc. However, the generated tasks cover various topics.



**Fig. 2.** Percentage of topics in generated tasks (gray color) and the corresponding success rate (green color) for each topic; category ‘Science’ is further divided into second-level categories such as ‘biology’, ‘astronomy’, etc. (Color figure online)

Corresponding to each topic, we measured the success rate of tasks. For each generated task, we regard a corresponding answer is **successful** if: a searcher’s

<sup>6</sup> <http://curlie.org/>.

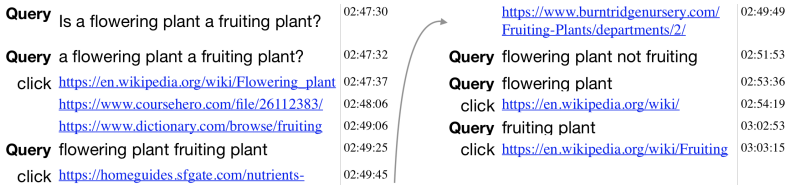
answer is correct, and the searcher provides meaningful supporting information that corroborates the answer (i.e. the supporting information is semantically similar to the that given by the task creator). We evaluated the similarity between supporting information given by searchers and that given by the task creator using an automatic text-level similarity evaluation method [15]. Of all the search sessions across different topics in our set, around 37% correspond to successful cases, which is comparable lower to that observed from real user logs (i.e. 40% in [22]). As shown in Fig. 2, the success rate varied across the different topics, ranging from 25% in *world* to 48% in *science-astronomy*.

According to the type of answer that satisfies a given task, we further analyzed the generated tasks from two standpoints: *yes/no* tasks (37 in total, the answers to 19 of them are ‘yes’, the answers to 18 of them are ‘no’), and *fact-finding* tasks (43 in total). Through a two-tailed T-test that compared the success rate across the two types of tasks, we did not find a significant difference. We also found no significant difference between tasks generated from “in other words” and those generated from “that is to say”.

## 5.2 Query Characteristics

It has been found that searchers’ struggling is reflected in their queries [2, 11]. We examine the characteristics of queries elicited by the generated tasks focusing on the following features: **query features** (i.e. number of queries, query length), **query-transition features** (i.e. query similarity, query reformulation), which have been shown to be useful for determining struggling search sessions [12, 13].

**Query Features.** Users in general issued more queries to handle a struggling search tasks [12, 22]. On average, the generated tasks comprised 5 to 6 queries ( $M = 5.55$ ) with average query length of 6 terms. Successful task solving sessions (5.48 queries, 4.76 terms per query on average) were slightly shorter than the unsuccessful counterparts (5.72 queries, 6.78 terms per query on average). We present an example to illustrate queries within a search session.



<b>Query</b> Is a flowering plant a fruiting plant?	02:47:30	<a href="https://www.burntridgenursery.com/Fruiting-Plants/departments/2/">https://www.burntridgenursery.com/ Fruiting-Plants/departments/2/</a>	02:49:49
<b>Query</b> a flowering plant a fruiting plant?	02:47:32	<b>Query</b> flowering plant not fruiting	02:51:53
click <a href="https://en.wikipedia.org/wiki/Flowering_plant">https://en.wikipedia.org/wiki/Flowering_plant</a>	02:47:37	<b>Query</b> flowering plant	02:53:36
<a href="https://www.coursehero.com/file/26112383/">https://www.coursehero.com/file/26112383/</a>	02:48:06	click <a href="https://en.wikipedia.org/wiki/">https://en.wikipedia.org/wiki/</a>	02:54:19
<a href="https://www.dictionary.com/browse/fruiting">https://www.dictionary.com/browse/fruiting</a>	02:49:06	<b>Query</b> fruiting plant	03:02:53
<b>Query</b> flowering plant fruiting plant	02:49:25	click <a href="https://en.wikipedia.org/wiki/Fruiting">https://en.wikipedia.org/wiki/Fruiting</a>	03:03:15
click <a href="https://homeguides.sfgate.com/nutrients-">https://homeguides.sfgate.com/nutrients-</a>	02:49:45		

Fig. 3. Samples of search sessions in user logs

Figure 3 shows the sample process a searcher moved through a session to solve the task “*Is a flowering plant a fruiting plant?*”. We note that to solve a task generated in this work, a searcher generally issued even more queries with longer query length than the ‘3 to 4 queries averaging around 4 terms per

query’ observed from daily-life struggling search logs in previous work [22]. This difference may also be attributed to the difference in tasks that were studied. The information inquired by the generated informational tasks are more specific and difficult to resolve than the tasks studied in previous works (e.g. find a source-page of a video).

We observed that the first query in both successful and unsuccessful search sessions are typically the task description itself or an excerpt sentence extracted from the task description (8.93 terms on average) which are longer than the intermediate queries (5.81 terms on average) and the final queries (4.18 terms on average). Existing works show that there are generally two different cases that correspond to struggling with respect to the first query of a search session: (i) the query is too common as it is general and ambiguous, or (ii) the query is quite uncommon as it might be overly specified [22]. From this we note that the long over-specified first query does not lead searchers to a target page, and might elicit struggling search consequently. However, this struggling does not determine the final success or failure of the whole search session, which is consistent with the outcomes in prior work in [22].

**Query Similarity.** It has been shown that in a struggling search session the later queries can be quite similar to the initial query. Users experiencing the struggle tend to reformulate queries that closely resemble the initial search [12, 22]. Based on prior works, we expect that in a struggling search task a user thinks of less diversified queries to explore alternatives. Thus in user logs, unique terms in the initial query persist through the future queries. To examine this, we measure the similarity between queries in the session. The similarity between any two queries  $Q_i$  and  $Q_j$  is computed using *Jaccard Index*:

$$\frac{|Q_i \cap Q_j|}{|Q_i| + |Q_j| - |Q_i \cap Q_j|} \quad (1)$$

where  $|Q_i|$  is the number of unique terms in query  $Q_i$ , and  $|Q_i \cap Q_j|$  is the number of matched terms in  $Q_i$  and  $Q_j$ .

Before measuring the similarity between queries in a session, we first normalize the queries; including lowercasing query text, deleting stop words, stemming, and unifying white space characters. For  $|Q_i \cap Q_j|$ , we consider two terms are matched if they are (i) **exact matched**: two queries match exactly; (ii) **approximate matched**: two queries match if the Jaro-Winkler distance (score) of them is larger than 0.6. In this work, we only consider the lexical-based query similarity. Assuming that for the concepts or information points in the generated tasks, users can seldom find alternative terms to search without learning through searching, we eliminate **semantic matched** cases (i.e. two queries match if semantic similarity of them over certain threshold [12]). Figure 4 shows the average similarity between queries to the first query. We found that in both successful and unsuccessful search sessions, searchers generally issue similar queries in the first three rounds. This is consistent with the outcomes in previous studies mentioned earlier [12, 22]. We found that in successful sessions, queries gradually get less similar to the initial query as the searching progresses (though the difference

was not found to be statistically significant using a two-tailed T-test at the 0.05 level). Prior work established that struggling searchers cycle through queries as they attempt to conceive a correct query to locate target information (i.e. the query similarity in struggling search sessions is generally greater than 0.4) [12]. Our findings corroborate that struggling search manifests during users’ quest to satisfy the information need, even if they finally succeed in their search missions.

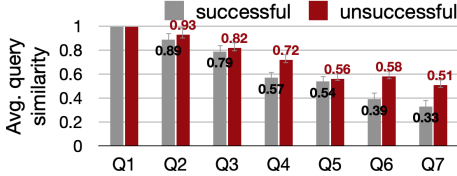


Fig. 4. Avg. query similarity in each step

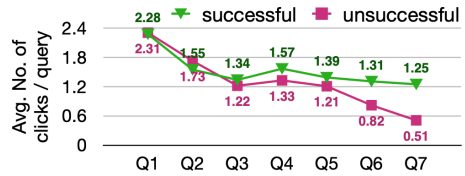


Fig. 5. Avg no. of clicks per query

**Query Reformulation.** We delve into how users employ terms from one query to another in web search. We consider the three main query transition types which have been used in previous works [12]: **Term Addition:**  $\geq 1$  word added to the first query; **Term Removal:**  $\geq 1$  word removed from the first query; **Term Substitution:**  $\geq 1$  word substituted with other lexically matched terms. Term matching is done by using lexical matching described earlier.

We found that term removal is generally the most popular strategy; almost all the search sessions contain term removal cases. This can be explained by the task description that users consumed the information prior to beginning the search session. Due to the nature of Wikipedia, most generated tasks pertain to topics which people may not encounter in their daily life. Thus, we reason that most people struggled to come up with alternative terms to describe the vague information need in the tasks. In such cases, over 2 terms were removed on average in the last query ( $M = 2.41, SD = 1.89$ ). The high standard deviation can be explained by differences between the generated tasks. For instance, a task with a long (short) information need description could elicit a long (short) initial query, finally converging to a few keywords. Term substitution occurs more frequently in successful sessions than in unsuccessful sessions (though not statistically significant,  $p = 0.052$ ) which is consistent with previous work [12].

### 5.3 Clicks Characteristics

Prior works have shown that searchers experiencing ‘struggle’ tend to exhibit no click actions or *quick-back clicks* (i.e. result clicks with a dwell time less than 10 s [16]) after certain queries [2, 12, 22]. This has been attributed to the difficulty experienced in locating target information. We examine the characteristics of users’ clicks on the SERPs in search sessions pertaining to the generated tasks.

On average, searchers exhibited 1.67 clicks after each query ( $M = 1.67, SD = 1.49$ ), and over 62% of search sessions contain quick-back clicks. We further computed the average number of clicks for a sequence of queries in a session. Figure 5 shows the average change in the number of user clicks per query. We found that within the initial 4 queries there’s no significant difference between successful and unsuccessful sessions in terms of the average number of clicks per query while the difference becomes more pronounced thereafter. Particularly, searchers in unsuccessful sessions issued less than 1 click on average after their last two queries. This is consistent with previous work, which also found that users in struggling search tasks tend to give up clicking on post-query results on the final query in an unsuccessful session [22]. From the click characteristics we find that solving the generated tasks, users are elicited with clicks in struggling, part of which could be the indicator of the eventual mission failure.

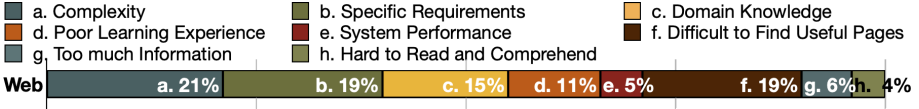
In contrast to our findings, Hassan et al. found that after several rounds of queries without locating any target information, struggling searchers tend to click on more results [12]. These contrasting findings can be explained by the difference of task types and difficulty levels. The generated tasks in our setup are generally fact-finding tasks with unambiguous final goals, while the tasks in previous works are more akin to open-ended exploratory tasks (e.g. ‘software purchase advice’, ‘career development advice’).

## 5.4 Task Difficulty Analysis

Corresponding to analysis of objective user behavior, we also investigate searchers’ subjective perception of task difficulty. In general, participants scored the task difficulty as 57 on average ( $M = 57, SD = 17$ ), which means tasks are in general *challenging* yet not *demanding*. We note that all participants agreed these tasks are much more difficult than the typical IR tasks. Among them 77% searchers thought the given tasks were more difficult in comparison to their general web search experience, rating task difficulty as 61 on average (i.e. *demanding*;  $M = 61, SD = 13$ ).

Based on the reasons collected from previous work [19], we investigated the reasons why tasks made users perceive a ‘struggling search’ experience during web search through self-reports. Figure 6 illustrates the overall impact of different reasons that contribute to users experiencing a ‘struggle’ while completing the generated tasks across the entire web. We found that the top-3 main reasons cited for task difficulty were (1) *task complexity*, wherein workers believed that there were several components of the task that needed to be addressed; (2) *difficult to find useful pages*, wherein searchers met difficulties locating proper web pages to acquire target information; (3) *specific requirements*, wherein the struggle experience was due to the information need being so specific, consequently making it more difficult to find. While the reasons spread across various aspects including task features (40%), user aspects (26%), the interaction between user and system (24%), and the readability of documents (10%).

We found that within Wikipedia domain the paraphrased sentences are generally distributed across curated articles about history, literature, physics, biology,



**Fig. 6.** Overview of the reasons why workers felt struggled in web search. Reasons are collected from 4 standpoints: task features (a, b), user aspects (c, d); user-system interaction (e, f); and document features (g, h).

etc., which people may not encounter in daily life. Thus, we observe the generated tasks correspond to subjective knowledge of users rather than more general scenarios that one may encounter in everyday life. This increases the task difficulty for most of the users; the information need of the generated tasks also requires users to process varied information from different perspectives. Moreover, self-reported difficulty reasons indicate that expanding the search domain increases the difficulty in locating useful pages to satisfy the information need (note that searchers were unaware of the fact that the source for all generated tasks was Wikipedia).

We also analyzed the influence of reasons on users’ perception of struggling. Results of the generalized linear regression indicate that there was a collective significant effect between the reasons and users’ perception of struggling in web search experiment ( $\chi^2 = 83.1, p < .01$ ). The individual predictors were examined further and indicated that *complexity* ( $t = 4.19, p < .001$ ), *specific requirements* ( $t = 1.57, p < .05$ ), *domain knowledge* ( $t = 2.03, p < .05$ ), *difficulty in finding useful pages* ( $t = 6.88, p < .001$ ) and *too much information* ( $t = 4.36, p < .001$ ) were significant predictors in the model, while searchers’ *poor learning experience*, the *system performance*, and whether the *target document is hard to read* are not the key factors that influence users’ struggling experience.

## 5.5 Publicly Released Task Set

For the benefit of the community, along with *TaskGenie* platform, we also publicly released the generated task set and user behavior logs (anonymized) gathered in our user study. We consolidated the 80 generated SSTs with different aspects including: question, answer, source page (i.e. suffixes of the sharing url “<https://en.wikipedia.org/wiki/>”), task type (i.e. “yes/no” or fact-finding), task topic (i.e. the ODP categories), task difficulty level (i.e. according to average difficulty score), and success rate. The complete task set is available online (the URL is provided in Introduction). This task set can be used to reliably simulate struggling search among users. For each task, we provide the basic success rate and task difficulty level that can be useful in the development and evaluation of methods to support users while they struggle in search tasks. Also, we provide the user behavior data collected in this work including queries, clicks, etc. Moreover, our proposed framework can be used to generate SSTs as per the topical/domain related needs at hand.



## 6 Discussion

**Why We Need ‘Humans’?** Although paraphrased sentences are a good source to create difficult questions, framing these questions automatically is far more challenging due to the variety in paraphrased sentences and their context; existing methods cannot automatically generate SST tasks in this manner. Humans on the other hand, can easily identify those paraphrased sentences which are suitable for creation of SST tasks. *TaskGenie* allows us to collect and study user behavioral logs while they solve SST tasks, and also supports the generation of SST tasks. Note that *TaskGenie* can easily be customized to execute only a single phase (task completion or task generation) if desired.

**Effects of the Document Collection.** In this work, we chose Wikipedia as the domain for generating struggling search tasks. And for simplicity, we only considered paraphrased sentences using the conjunctions “in other words” and “that is to say” as the indicators for redundant information that is summarized. However, our framework can be easily customized to include other conjunctions concomitant with paraphrased sentences. We also showed that the generated tasks correspond to a variety of topics. Moreover, our framework can be readily used to generate SSTs for specific domains by depending on the corresponding wikis<sup>7</sup>. These include **WikiTravel** about traveling and places, **tvTrope** about television and movies, **WikiNews** about the news and events, etc. All these could be a potential source for paraphrased sentences. Thus, we argue that using this framework, a comprehensive SST task set that fits domain related requirements can be realized.

**Effects of the Retrieval Model.** In this work, generated tasks are not quantitatively balanced across topics. However, through a post study analysis we found that advanced searching grammar could help in balancing topics of generated tasks in a task set by locating paraphrased sentences pertaining to specific topics. For example, by issuing a call to the Bing API with an advanced option ‘‘**in other words**’’: **recreation** targeting Wikipedia domain we could locate all the Wikipedia articles containing the phrase “in other words” and corresponding to the topic of “recreation”. We observed that in the task generation phase, despite instructions that encourage workers to select articles with highlighted paraphrased sentences more arbitrarily and neglect the ranking order, some participants still selected the top-ranked results. As a consequence we found a few duplicates in the generated tasks. Nevertheless, we collected 80 distinct tasks generated by users within the task generation framework that adequately elicited struggling search behavior of users.

**Task Pre-filtering Method.** In this paper, authors manually filtered struggling search tasks from the generated set of tasks. A manual task filtering step guarantees the high quality of SSTs, but it gets progressively more expensive with the growing size of the task set. By analyzing the generated tasks, we note that when SSTs are expressed in natural language, they are potentially

<sup>7</sup> [https://en.wikipedia.org/wiki/List\\_of\\_wikis](https://en.wikipedia.org/wiki/List_of_wikis).

more complex from a readability standpoint in comparison to typical IR tasks. Through  $K$ -means ( $K = 2$ ; Euclidean distance) for task type clustering based on the two parameters of average word complexity and readability of the generated tasks, we found that the readability of tasks could be an indicator of SSTs. Such clustering resulted in identifying SSTs with an accuracy of 80%, providing a pre-filtering method for scalable filtering of the generated tasks that can be leveraged in the future.

We note that the reading comprehension ability of a worker plays an important role in the worker's understanding of the preceding context, and the accurate generation of a SST using a paraphrased sentence. In the current setup, we recruited *Level-3* workers from Figure 8. However, we reason that to optimize the efficient generation of SSTs using our framework one can consider pre-screening crowd workers based on their proficiency in reading comprehension.

## 7 Conclusions and Future Work

By leveraging summarized (redundant) information in paraphrased sentences we proposed a task generation method and implemented it in an online crowd-powered framework. Through our task generation framework, we collected diverse questions from crowd workers with implicit task descriptions, and unambiguous answers that can be found by exploring the relevant information space. While this also results in some simple look-up tasks, these can be easily filtered out using existing criteria. We conducted a web search experiment to evaluate the task quality based on characteristics of elicited user behaviors. We showed that high quality struggling search tasks can be generated using our framework. We analyzed why searchers struggle in search sessions, and revealed insights into the independent impact of each of task characteristics that lead to users' struggle in a search session. We believe that our framework, the task set, together with our insights in this paper will help in advancing and developing methods to support users in struggling search. In the imminent future, we will test the SSTs in different search engines and explore a benchmark about how different search engines support such struggling fact finding or checking tasks.

## References

1. Ageev, M., Guo, Q., Lagun, D., Agichtein, E.: Find it if you can: a game for modeling different types of web search success using interaction data. In: SIGIR, pp. 345–354 (2011)
2. Aula, A., Khan, R.M., Guan, Z.: How does search behavior change as search becomes more difficult? In: SIGCHI, pp. 35–44 (2010)
3. Baykan, E., Henzinger, M., Marian, L., Weber, I.: Purely URL-based topic classification. In: WWW, pp. 1109–1110 (2009)
4. Bhagat, R., Hovy, E.: What is a paraphrase? *Comput. Linguist.* **39**(3), 463–472 (2013)

5. Braarud, P.Ø., Kirwan, B.: Task complexity: what challenges the crew and how do they cope. In: Skjerve, A., Bye, A. (eds.) *Simulator-based Human Factors Studies Across 25 Years*, pp. 233–251. Springer, London (2010). [https://doi.org/10.1007/978-0-85729-003-8\\_15](https://doi.org/10.1007/978-0-85729-003-8_15)
6. Capra, R., Arguello, J., O’Brien, H., Li, Y., Choi, B.: The effects of manipulating task determinability on search behaviors and outcomes. In: *SIGIR*, pp. 445–454 (2018)
7. De Beaugrande, R.A., Dressler, W.U.: *Introduction to Text Linguistics*, vol. 1. Longman, London (1981)
8. Gadiraju, U., et al.: Crowdsourcing versus the laboratory: towards human-centered experiments using the crowd. In: Archambault, D., Purchase, H., Hoffeld, T. (eds.) *Evaluation in the Crowd. Crowdsourcing and Human-Centered Experiments. LNCS*, vol. 10264, pp. 6–26. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-66435-4\\_2](https://doi.org/10.1007/978-3-319-66435-4_2)
9. Gadiraju, U., Yu, R., Dietze, S., Holtz, P.: Analyzing knowledge gain of users in informational search sessions on the web. In: *CHIIR 2018* (2018)
10. Hassan, A.: A semi-supervised approach to modeling web search satisfaction. In: *SIGIR*, pp. 275–284 (2012)
11. Hassan, A., Jones, R., Klinkner, K.L.: Beyond DCG: user behavior as a predictor of a successful search. In: *WSDM*, pp. 221–230 (2010)
12. Hassan, A., White, R.W., Dumais, S.T., Wang, Y.M.: Struggling or exploring?: disambiguating long search sessions. In: *WSDM*, pp. 53–62 (2014)
13. Hassan Awadallah, A., White, R.W., Pantel, P., Dumais, S.T., Wang, Y.M.: Supporting complex search tasks. In: *CIKM*, pp. 829–838 (2014)
14. Kellar, M., Watters, C., Shepherd, M.: A goal-based classification of web information tasks. *ASIST* **43**(1), 1–22 (2006)
15. Kenter, T., De Rijke, M.: Short text similarity with word embeddings. In: *CIKM*, pp. 1411–1420 (2015)
16. Kim, Y., Hassan, A., White, R.W., Zitouni, I.: Modeling dwell time to predict click-level satisfaction. In: *WSDM*, pp. 193–202 (2014)
17. Kolln, M., Funk, R.: *Understanding English Grammar*. Longman, London (1982)
18. Liu, C., Liu, J., Cole, M., Belkin, N.J., Zhang, X.: Task difficulty and domain knowledge effects on information search behaviors. *ASIS&T* **49**(1), 1–10 (2012)
19. Liu, J., Kim, C.S., Creel, C.: Why do users feel search task difficult? In: *The 76th ASIS&T. American Society for Information Science* (2013)
20. Mai, J.E.: *Looking for Information: A Survey of Research on Information Seeking, Needs, and Behavior*. Emerald Group Publishing, Bingley (2016)
21. Mitra, B.: Exploring session context using distributed representations of queries and reformulations. In: *SIGIR*, pp. 3–12 (2015)
22. Odijk, D., White, R.W., Hassan Awadallah, A., Dumais, S.T.: Struggling and success in web search. In: *CIKM*, pp. 1551–1560 (2015)
23. Pirolli, P., Card, S.: The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In: *IA*, vol. 5, pp. 2–4 (2005)
24. Robinson, P.: Task complexity, task difficulty, and task production: exploring interactions in a componential framework. *Appl. Linguist.* **22**(1), 27–57 (2001)
25. Singer, G., Norbistrath, U., Lewandowski, D.: Ordinary search engine users carrying out complex search tasks. *J. Inf. Sci.* **39**(3), 346–358 (2013)
26. Singer, P., et al.: Why we read Wikipedia. In: *WWW*, pp. 1591–1600 (2017)
27. White, R.W.: *Interactions with Search Systems*. Cambridge University Press, Cambridge (2016)

28. White, R.W., Roth, R.A.: Exploratory search: beyond the query-response paradigm. In: Synthesis Lectures on Information Concepts, Retrieval, and Services, vol. 1, no. 1, pp. 1–98 (2009)
29. Wilson, M.L., Kules, B., Shneiderman, B., et al.: From keyword search to exploration: designing future search interfaces for the web. *Found. Trends® Web Sci.* **2**(1), 1–97 (2010)
30. Xu, L., Zhou, X.: Generating tasks for study of struggling search. In: Proceedings of the 2019 Conference on Human Information Interaction and Retrieval, pp. 267–270 (2019)