

June 4, 2020
DRAFT

Supporting Global Context under Evolving User Intents during Data Exploration

JOSEPH CHEE CHANG

CMU-LTI-20-005

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Aniket Kittur, Carnegie Mellon University, Chair
Jeffrey Bigham, Carnegie Mellon University
Adam Perer, Carnegie Mellon University
David Karger, Massachusetts Institute of Technology

*Submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy.*

June 4, 2020
DRAFT

Abstract

Whether its consumers comparing all the available options on Amazon, novice learners synthesizing information scattered across many online tutorials and discussion boards, or data scientists analyzing datasets to find patterns and themes, users often need to explore large quantities of unstructured information beyond an individual's capacity to process them fully. Typically, users reduce task uncertainty by learning the unknown unknowns as they process individual pieces of information to gain deep qualitative insights. However, the cost of evaluating learned insights (known unknowns) under the global context can be high, prohibiting users to evaluate their generalizability and whether they lead to high-yield information patches [171]. For example, a consumer who encountered a product recommendation on one webpage may need to search across the Web and consider many other sources to figure out if it is worth adding it to their shortlist for deeper comparisons. As they read online reviews they often discover new criteria that fits their personal context and interests, but it can be difficult for them to figure out how well a new criteria can differentiate all the different products on their shortlist. Similarly, a team of scientists who observed an interesting phenomenon on a subset of data also needed to spend a lot of effort figuring out whether it generalizes to the rest of the dataset [49]. Most existing approaches either focused on aggregation techniques of unstructured data (e.g., topic modeling, review summarization and aspect extraction) or interaction techniques for exploring structured data (e.g., faceted navigation and multivariate visualizations), and do not support this process of bottom-up exploration and interpretation of unstructured online data.

This thesis explores systems and interaction techniques that support users in exploring large and unstructured data by allowing them to both examine each piece of information to gain local insights and at the same time evaluate them under the global context. I identify and focus on two domains where addressing this issue can lead to high impact. The first half of the thesis focuses on the domain of crowdsourced sensemaking, in which an individual's capacity for understanding large datasets is scaled up by segmenting data into microtasks to be processed by a group of crowdworkers. I describe two approaches that allowed crowdworkers whom each saw a small subset of data to generate categories that were more globally coherent compared to existing crowd-based and computation-based approaches (Chapters 3 and 4). The second part of the thesis focuses on supporting individual sensemaking, in which an individual explores and synthesizes online information scattered across different webpages for their own personal tasks, such as product comparison or trip planning. I describe three systems that allow users to discover important options and criteria from one source, and evaluate them across information sources and different options to gain a deeper global understanding with lowered interaction costs (Chapters 5 to 7). Through lab and field deployment user studies, I investigated the costs and benefits of the systems for supporting personal online sensemaking.

Contents

1	Introduction and Thesis Statement	1
1.1	Global Context and Crowdsourced Sensemaking	1
1.2	Global Context and Individual Online Sensemaking	3
1.3	Thesis Statement and Overview	4
2	Background	6
2.1	Evolving User Intention when Exploring Large Data	6
2.2	Structuring Information with Crowdsourcing	7
2.3	Exploratory Search Interfaces	7
3	Alloy: Coherent Categorization with Crowds and Computation	9
3.1	Introduction	9
3.2	System Design	12
3.3	Evaluation	17
3.4	Application: Knowledge Accelerator	24
3.5	Discussion	29
4	Revolt: Exploiting Disagreements For Concept Evolution in Crowd Labeling	32
4.1	Introduction	32
4.2	Related Work	34
4.3	System Design	37
4.4	Evaluation	41
4.5	Discussion	49
5	SearchLens: Capturing and Composing Complex User Interests	51
5.1	Introduction	51
5.2	Related Work	54
5.3	System Design	55
5.4	Evaluation	61
5.5	Discussion and Limitations	68
6	Weaver : Entity-Centric Foraging across Webpages in the Browser	70
6.1	Introduction	70
6.2	Related Work	73
6.3	System Design	75
6.4	Evaluation	81
6.5	Discussion	86
7	Mesh: Scaffolding Comparison Tables for Online Decision Making	89

7.1	Introduction	89
7.2	Related Work	91
7.3	Exploratory Interviews and Design Goals	93
7.4	System Design	95
7.5	Evaluation	99
7.6	Discussion and Limitations	107
8	Conclusion	109
8.1	Take-Aways	109
8.2	Design Pattern	113
8.3	Conclusion	114
	Bibliography	116

List of Figures

3.1 A conceptual overview of the Alloy system.	10
3.2 The interface and steps of the Head Cast HIT.	13
3.3 Example clips from two datasets with crowd keywords.	14
3.4 HIT interface for the Merge Cast and the Tail Cast.	15
3.5 Categories comparison for Q1	22
3.6 Performance comparison of using different number of crowdworkers.	23
3.7 Alloy clusters synthesized into a report articles using the Knowledge Accelerator.	25
3.8 The Knowledge Accelerator (KA) with Alloy as the Clustering Stage.	25
3.9 Results across questions and websites.	27
3.10 Categories induced from different stages of KA.	28
4.1 A high level view of the Revolt system.	33
4.2 Overview of Revolt stages.	37
4.3 HIT interface for the Vote Stage of Revolt	38
4.4 HIT interface for the Explain Stage of Revolt	39
4.5 HIT interface for the Categorize Stage of Revolt	40
4.6 Accuracy of different approaches as a function of post-hoc requester effort.	44
4.7 Accuracy improvement as a function of requester effort.	46
4.8 Work duration of each crowdworker under different conditions.	47
5.1 An overview of the SearchLens system.	52
5.2 SearchLens provides keywords suggestions based on currently Lenses.	58
5.3 The visual explanation and exploration feature of SearchLens.	59
5.4 Baseline system interface for the SearchLens lab study.	62
5.5 Number of Lenses and keywords saved the participants.	64
5.6 Participants frequently interactive with their Lenses instead of sift through reviews	65
5.7 Number of Lenses and keywords under different conditions.	66
6.1 An overview of the Weaver browser add-on.	71
6.2 Expanded view for an entity card.	76
6.3 An project overview page created by one participant.	77
6.4 Linking entities from webpages to open and commercial knowledge bases.	79
6.5 Creating manual entity cards.	80
6.6 One participant's project screenshot in the baseline variant of Weaver	82
7.1 The Table View for Mesh	90
7.2 Importing tabs into Mesh as product options	93
7.3 The Evidence View for Mesh.	98
7.4 Mean statistic for the objective criteria study.	101

7.5 The initial spreadsheet for the baseline condition.	102
7.6 Mean statistics for the subjective criteria study.	103
8.1 A general system design pattern.	113

List of Tables

3.1	Datasets used for evaluation	17
3.2	Evaluation results for Alloy	20
3.3	Comparing KA output with top websites for the eleven questions.	26
3.4	Average number of worker tasks and cost of running KA.	27
4.1	Evaluation for Revolt — Labeling accuracy under different conditions.	43
4.2	Revolt categories for the Car datasets.	44
5.1	Number of Lens editing actions performed under different conditions.	63
6.1	Evaluation results showed participants who used Weaver collected more evidence from more sources.	83
7.1	List of participants and usage pattern for the deployment study.	104

Chapter 1: Introduction and Thesis Statement

Whether it is a consumer reading reviews to compare products, a learner reading different tutorial and forum posts, or a data scientist analyzing a large dataset, users are often faced with large quantities of unstructured information beyond an individual's capacity to process them fully. Typically, users reduce task uncertainty (unknown unknowns) by processing individual pieces of information in order to learn deep qualitative insights. This process of learning the unknown unknowns in the dataset allows users to iteratively refine their goals and interests, which can both potentially invalidate prior decisions and opens new research directions [149]. However, the cost of evaluating learned insights (known unknowns) under global context can be high, prohibiting users to evaluate their generalizability and whether they lead to high-yield information patches [171]. For example, a consumer who encountered a product recommendation on one webpage may need to search across the Web and consider many other sources to figure out if it is worth adding it to their shortlist for deeper comparisons. As they read online reviews they often discover new criteria that fits their personal interests, but it can be difficult for them to figure out how well a new criteria can differentiate all the different products on their shortlist. Similarly, a data scientist who observed an interesting phenomenon on a subset of data also needed to spend a lot of effort in order to figure out whether it generalizes to the rest of the dataset [49]. Most existing approaches either focused on aggregation techniques of unstructured data (e.g., topic modeling, review summarization and aspect extraction) or interaction techniques for structured data (e.g., faceted navigation and multivariate visualizations), and do not support this process of bottom-up exploration and interpretation of unstructured online data. This thesis explores systems and interaction techniques that support exploring unstructured datasets by allowing users to both gain deep insights from each piece of information and at the same time evaluate such local phenomena under global context. I investigated this approach under the following two domains of crowdsourced sensemaking and individual online sensemaking.

1.1 Global Context and Crowdsourced Sensemaking

Crowdsourcing markets, such as Amazon Mechanical Turk, offer a new paradigm for on-demand data processing at scale, allowing researchers and machine learning practitioners who do not have the capacity or resources to process a dataset themselves and request a group of crowdworkers to process them. This typically involves the task requester who examines a small part of the data to design the microtask instructions, and divides the dataset into smaller chunks to be distributed across different crowdworkers. Traditionally, crowdsourcing had focused on collecting simple human judgements such as extracting contact information from a webpage [82] or recognizing characters on an image that automatic optical character recognition (OCR) algorithms were unable to process [216]. However, many real-world sensemaking tasks are often more complex and interdependent, requiring each crowdworker who only received a small subset of data to evaluate them with a better understanding of the global context.

Considering the task of organizing a collection of text snippets by their common themes. Since

each crowdworker only saw an arbitrary subset of the entire dataset, it can be difficult for them to come up with categories that are coherent and comprehensive under the global context. For example, if all the items in the sample were closely related, a crowdworker might generate overly fine-grained categories; conversely, if all the items in the sample were dissimilar, the crowdworker might not be able to identify common patterns to generate useful categories.

Further, even in classification tasks where a requester provided predefined categories to the crowdworkers, it can be prohibitively effortful for requesters to explore enough data in order to generate clear and comprehensive guidelines (i.e., label definitions) that can eliminate ambiguity in data that allowed for subjective interpretations by the crowdworkers. This is due to the fact that prior work has shown even in expert labeling scenarios where machine learning practitioners labeled all items in a dataset using predefined categories, their own mental definition of the categories will typically evolve throughout the process as they examine more items [133]. For example, a requester developing labeling guidelines for a seemingly simple task of labeling images as either about “cats” or not might not be aware of the long tail of edge cases in the dataset. These edge cases can then be interpreted differently by different crowdworkers, leading to inconsistent labels for images about cartoon cats or tigers. Fundamentally, the main challenges here are firstly each crowdworker can only see a small subset of data due to the scope of microtasks that typically range from a few seconds to a few minutes. Secondly, the requesters who relied on crowdsourcing to scale up their capability to process larger quantities of information also may not have enough global context to generate comprehensive guidelines for the crowdworkers. Failure to address these challenges can lead to incoherent structures and inconsistent labels.

In the first part of this dissertation, I explore novel crowdsourcing approaches for generating globally coherent structures. For this, I built two systems that introduced new framework for crowd categorization (clustering) and classification, respectively, that can provide better support for global context :

- **Alloy:** A novel crowdsourcing workflow that focuses on organizing a collection of textual snippets into globally coherent categories by combining crowdsourcing and computation (Chapter 3). Instead of showing a fixed subset of data to each crowdworkers, Alloy introduced a novel interaction technique that allowed each crowdworkers to repeatedly sample from the entire dataset until they build up a better understanding of the global context to generate coherent categories.
- **Revolt:** A novel paradigm for collecting classification labels for training machine learning models (Chapter 4). Instead of requiring requesters to generate comprehensive guidelines beforehand, which can potentially require them to explore a large portion of the datasets, Revolt uses crowdworkers to identify ambiguous items in data and generate categories for post-hoc decisions made by the requesters.

The two systems were evaluated against state-of-the-art crowdsourcing and machine learning approaches on a wide variety types of data including web snippets extracted from Google search results, research paper abstracts, Web images that were a subset of ImageNet [68] and Webpage classification datasets from a prior work [133]. We found evidence that the proposed techniques can provide better global context, either to crowdworkers or the requesters, leading to more coherent structures and consistent labels.

The main focus of the Alloy system was to categorize web snippets extracted from webpages in a Google search results using queries such as how do I grow better tomatoes or What does a planet need to support life. To further investigate the usefulness of the Alloy structures for end-users, I built a third crowdsourcing system, **Knowledge Accelerator** (Section 3.4), that synthesized the categories of web snippets into sections of overview articles. We were surprised to find that the Alloy structure led to crowd-generated articles that outperform top Google search results published by experts. These results revealed that end-users valued information that were synthesized from across many different online sources, especially when an authoritative information source was not available. This led to the second part of this dissertation that explores novel systems that support individuals when foraging and learning from online information, and evaluating discoverings from reading information pieces of information under the global context of many information sources.

1.2 Global Context and Individual Online Sensemaking

Whether planning a trip to a new city or deciding which product to purchase, consuming and foraging information online through exploratory search has become how people make sense of the world. People now have instant access to an enormous online bazaar of information produced by experts and novices with different personal preferences, backgrounds, and assumptions. Consider purchasing a desk lamp at an office supply store versus online, and the differences in scale of available options and evidence. Amazon lists over 4,000 different options and up to thousands of reviews for each option; Google returns hundreds of “best desk lamps” listicles; and Reddit¹ lists thousands of discussions on desk lamps. While this rich repository of diverse perspectives has the potentials to empower consumers and learners to explore and understand available options thoroughly and make better decisions [64], the seemingly infinite number of options and evidence scattered across numerous information sources are often well beyond an individual’s capacity to process them [196]. While existing research largely focused top-down approaches to support this process, such as presenting average review ratings, summarizing reviews [99, 140] or making recommendations directly [30], prior studies have instead shown that consumers often take a bottom-up approach of deeply examining each pieces of information to gain insights and gradually build up a personal understanding of the information space [84, 165].

One explanation for the bottom-up approach is that online evidence, such as reviews, can be messy, subjective, potentially biased and scattered across online sources [50, 96, 177, 232]. This required users to both interpret each piece of evidence to determine how well it fits their personal context, as well as using multiple information sources in order to verify them [176]. Another factor could be the exploratory, dynamic and opportunistic nature of online exploratory search [150] – as users develop a personalized framework for comparing options, they might discover new criteria and iteratively refine their goals and preferences, potentially invalidates prior decisions and opens new research directions [171, 172]. Fundamentally, users have a need to deeply explore and interpret each piece of evidence to discover options and criteria that align with their own personal goals and needs, but the overwhelming amount of available evidence, options and information sources can be prohibitive for them to evaluate such local insights under the global context.

In the second half of this dissertation, I explore three novel systems and interaction techniques

¹<http://reddit.com>

that can better support global context in this bottom-up qualitative process and scaffold users' online exploration and decision-making process:

- **SearchLens**: an interactive restaurant review search interface that enabled users to explore reviews to discover and build up a set of structured queries (i.e., sets of weighted keywords) that reflected their different nuanced interests to search for restaurants. This enabled users to maintain their evolving interests throughout exploration. Using the queries, the system generated personalized visual explanations for each restaurant in the search results, allowing users to interpret and explore new options based on their current interests. (Chapter 5)
- **Weaver**: a browser extension that enabled the browser to identify common entities (i.e., restaurants and destinations) mentioned across open tabs to support travel planning. Weaver's interface allowed users to both evaluate a new option they encountered on one webpage using evidence about it extracted across information sources, as well as allowed them to efficiently forage, accumulate and re-access evidence about different options across their browser tabs throughout the process. (Chapter 6)
- **Mesh**: an interface that scaffolds users in exploring online evidence (reviews and webpages) about products and progressively builds up a comparison table that reflects their personalized criteria and evaluation of online evidence. Allowing them to both evaluate how useful a newly discovered criteria was for differentiating the options to prioritize their effort, as well as keeping track of their own interpretation and summarization of evidence as they explore. (Chapter 7)

The three systems were evaluated using controlled lab studies and field deployment studies. I found evidence that by providing better global context across multiple information sources to individuals can lead to higher incentives for externalizing user interests (Chapter 5), gather and accumulate evidence across information sources with lowered effort (Chapter 6) and discovering deeper insights from data (Chapter 7).

Two high level models for providing global context during individual online sensemaking emerged from the benefits provided in the three systems for providing global context tasks. My first insight is that users can not confidently make decisions based on a single piece of evidence, whether it is a product recommendation or an important criteria mentioned in a review. In this case, providing them with a better global landscape of using other information sources as confirmation can lower both the interaction costs of cross-referencing and also the mental costs of evaluating many pieces of evidence. My second insights is that individuals often have evolving goals throughout the process, encountering both new criteria or interesting soft preferences as they explore more evidence. Allowing them to keep track of their changing interests and using them to interpret new and existing options can provide a scaffold that leads to better decision-making.

1.3 Thesis Statement and Overview

In many sensemaking scenarios, users often face large quantities of unstructured data and take a bottom-up approach to explore them in order to gain deep insights from data. However, individuals with limited capacity often can not process all the data to see a complete landscape of information. With only a local view of the whole picture, users can risk generating incoherent structures when organizing data or be overwhelmed by the amount of choices and evidence

leading to high interaction and mental effort. To investigate methods that can better support global context in such scenarios, I consider the following as the thesis of this dissertation:

Using interaction and visualization techniques, we can dynamically provide global context that matches users' evolving intentions throughout their exploration of large unstructured datasets. Supporting this will allow users to gain deeper insights from data and make better decisions with lowered efforts. Specifically, this dissertation first investigates this thesis in the domain of crowdsourced sensemaking where both the requesters and crowd-workers each only saw a small portion of data but needed to create globally coherent and consistent structures. This dissertation also investigates the domain of individual online sensemaking where the amount of available choices and evidence are often well beyond an individual's capacity to process them.

Concretely, this thesis makes the following contributions:

1. An interaction pattern of allowing users to explore large quantities of information that supports qualitative knowledge discovery from individual pieces of information and evaluating them under the global context.
2. Five novel system designs and interaction techniques that can better support this new model in two domains – crowdsourced and individual online sensemaking.
3. The implementation of the systems and extensive lab and field evaluation that investigated their costs and benefits to the users.

Chapter 2: Background

2.1 Evolving User Intention when Exploring Large Data

Users often need to explore individual pieces of information in large and unfamiliar datasets. As they gradually examine more data, they build up a better mental landscape of the space of information, potentially adjusting their goals in the exploration process. One instance of this process are online exploratory search tasks where users start out with a high level, sometimes vague, idea about their goals and criteria. Consider a user starting out with a search query for finding “the best laptop for college.” Being unfamiliar with the topic, the user must first explore online reviews and articles in order to figure out what were the common options recommended in these sources, what were the important criteria these recommendations were based on, and which of these criteria fits the user’s personal preferences and context. This process is often exploratory, dynamic and opportunistic in nature, requiring users to learn from the individual webpages and reviews to iteratively refine their goals and preferences based on a better understanding of global context [150]. However, prior studies have shown that this process can incur high mental and interaction efforts as new discoveries can potentially invalidate prior decisions, lead to better query terms [15, 188], and open new research directions to pursue [171, 172] requiring users to use a combination of external tools to keep track of all their decisions and progress made throughout these mental changes [38].

There have been several decades of research that have explored ways of getting users to more deeply externalize their intents and goals beyond short search queries in order to provide better support for this process [111]. For example, using prompt and text field designs that promote longer query terms [15, 83], asking for relevance feedback on the results provided [170, 185, 188], explicitly asking users to build up sets of query terms of different topics [95, 97], or providing in-situ interfaces for note-taking [212]. However, research also found that it is very difficult to get users to put in the work to externalize and maintain their evolving interests tasks due to its volatile nature during exploratory search. In addition, interactions such as eliciting longer query terms or explicit relevance feedback, can have the perception that the work will not be sufficiently paid off in the future or not understanding how their work will affect their results. In this thesis, I introduced two mechanisms for providing immediate and sufficient benefits to exploratory searchers: 1) generating personalized and interactive visualizations for explaining items in a search results based on users’ current interest profile (Chapter 5); and 2) allowing users to keep track of their interpretation of online evidence about their different criteria and options to build a product comparison table (Chapter 7). I tested these mechanisms in two systems in controlled lab studies and field deployment studies, and found that users expressed significantly more to the system, and valued the benefits they provided.

2.2 Structuring Information with Crowdsourcing

Human computation approaches present new opportunities to harness deep semantic knowledge for exploring and organizing complex and unstructured data. For example, Cascade [33, 58] generated hierarchical categories from online forum discussions, but suffers from categories at the same level having varying specificity due to the limited context of each crowdworkers. Crowd Synthesis [8] showed that simply showing more items to each crowdworker can lead to significant improvements, suggesting global context is a key element for crowd structuring. Fundamentally, most prior systems provide context by showing a small sample of items, hoping that they capture the distribution of information in the larger dataset. A complementary set of approaches has focused on the scaling through computation, applying approaches such as partial clustering [227], learning similarity metrics [207], or matrix completion [228]. While these have shown to be powerful on simple information such as visual patterns or colors using large numbers of split-second judgments, structuring complex exploratory search information can be difficult without providing novice crowdworkers with richer context or opportunities to learn from data. In chapter 3, I propose an alternative approach that builds up workers' mental models by allowing them to actively request for more context, identify discriminative keywords, and search the dataset for similar items, taking advantage of people's capacity of information foraging [171]. The resulting structures were found to be more coherent than a state-of-the-art crowd and machine learning-based systems and at a lowered monetary cost compared to other crowd-based approaches.

2.3 Exploratory Search Interfaces

Due to the ubiquity and high costs of exploratory search tasks to the users [150], a major thread of work includes novel personalized search interfaces such as semantic web interfaces [221], or computational approaches such as automatic or interactive result clustering [63]. Several exploratory search interfaces have been developed in order to help searchers orient themselves in the information space, review and explore the different subtopics, and keep track of their overall progress [93, 151, 164, 168, 211]. Two closely related studies include Topic-Relevance Map and Exploration Wall, which explored ways to provide overviews of search results of academic papers using document keywords and entities and easily choose keywords to build up subsequent queries [126, 169].

Past studies have shown users rely on aggregating from multiple sources in order to verify online information as credible and make decisions [62, 81, 177], but the process can be “tedious and cumbersome” leading to “opening several tabs ... and then manually switch[ing] between them while trying to remember information on different pages” [22, 85]. Another domain of research focused on aggregation of information scattered across sources. For example, summarizing search results for complex exploratory search tasks has been an area of high research interests. Early threads of research include search results clustering [230, 231], review summarization [146, 229], and identifying criteria about products from reviews [99, 140]. While these top-down approaches have shown great benefits in helping people get an initial overview of the space of information with many making their way into commercial ecommerce websites, prior studies on consumer behavior have also shown that consumers often also rely on bottom-up approaches of deeply examining each pieces of information to gain insights and gradually build up a personal understanding of the information space [84, 165]. In the second half of this dis-

sertation, I presented three systems for supporting global context by allowing users to express personal interests and use them to interpret multiple options (Chapter 5), cross-reference information about entity options scattered across webpages (Chapter 6), and keeping track of how users interpret individual pieces of evidence to gradually build up a global understanding of their options and criteria during online shopping research (Chapter 7). Through controlled lab studies and field deployment studies, I examine the costs and benefits of dynamically providing global context based on users' current interests.

Chapter 3: Alloy

Coherent Categorization with Crowds and Computation

This work was previously published in ACM SIGCHI 2016 ([44] and [88]) and has been adapted for this document.

This chapter describes the first of the two crowd systems in this dissertation that explored ways to provide global context in crowdsourcing. This first system focused on the task of data clustering, a common approach to data analysis. In the domain of crowdsourcing, this typically involves assigning sets of items to different crowdworkers and using human judgements to both creating categories and assigning items under them. Crowdsourced clustering approaches present a promising way to harness deep semantic knowledge of human computation for identifying coherent categories and clustering complex information. However, existing approaches have difficulties supporting the global context needed for workers to generate meaningful categories, and are costly because all items require human judgments. We introduce Alloy, a hybrid approach that combines the richness of human judgments with the power of machine algorithms. Alloy supports greater global context through a new “*sample and search*” crowd pattern which changes the crowd’s task from classifying a fixed subset of items to actively sampling and querying the entire dataset. It also improves efficiency through a two phase process in which crowds provide examples to help a machine cluster the head of the distribution, then classify low-confidence examples in the tail. To accomplish this, Alloy introduces a modular “*cast and gather*” approach which leverages a machine learning backbone to stitch together different types of judgment tasks. In an application-oriented evaluation, Alloy clustered were further synthesized into comprehensive overview articles using a workflow described in [88]. Results show that Alloy structures can lead to coherent and comprehensive overviews that out performed top Google search results published by experts in scenarios where there are a lack of authoritative sources.

3.1 Introduction

Clustering, or pulling out the patterns or themes among documents, is a fundamental way of organizing information and is widely applicable to contexts ranging from web search (clustering pages) to academic research (clustering articles) to consumer decision making (clustering product reviews) [109]. For example, a researcher may try to pull out the key research topics in a field for a literature review, or a Wikipedia editor may try to understand the common topics of discussion about a page in order to avoid or address previous conflicts. Doing so involves complex cognitive processing requiring an understanding of how concepts are related to each other and learning the meaningful differences among them [16, 129, 157].

Computational tools such as machine learning have made great strides in automating the clustering process [29, 41, 60]. However, a lack of semantic understanding to recognize the important differences between clusters leaves the difficult task of identifying meaningful concepts to the human analyst [61]. This reflects an inherent advantage for humans over machines for the complex problem of understanding unstructured data beyond merely measuring surface sim-

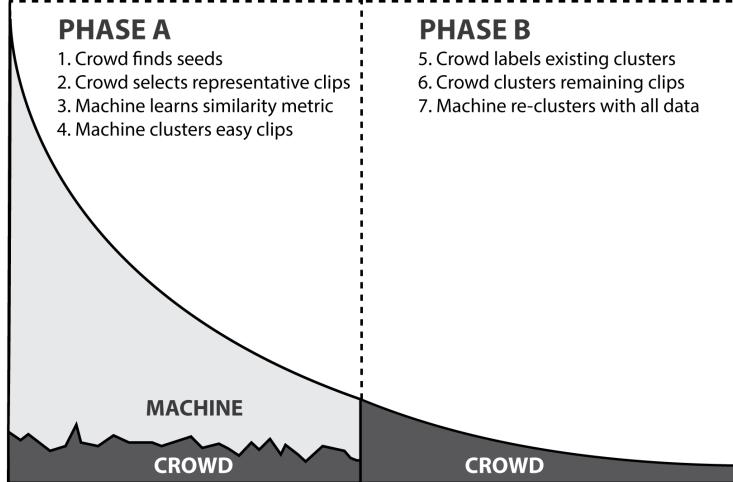


Figure 3.1: A conceptual overview of the system. In the first phase, crowd workers identify seed clips to train a machine learning model, which is used to classify the “head” of the distribution. In the second phase, crowd workers classify the more difficult items in the “tail”. A machine learning backbone provides a consistent way to connect worker judgments in different phases.

ilarity, and a corresponding opportunity for research in combining human and computational judgments to process complex information [79, 101, 132].

One such promising avenue of research harnesses the power of crowds to identify categories and cluster rich textual data. Crowdsourcing approaches such as Cascade, Deluge, and Crowd Synthesis [8, 33, 58] have demonstrated the power of splitting up rich, complex datasets into small chunks which can be distributed across many human coders. However, all of these approaches must grapple with a fundamental problem: since each human coder is seeing only a small part of the whole dataset, a lack of global context can lead to incoherent results. For example, if the items sampled are too similar, the worker might create overly fine-grained clusters. On the other hand, if the items sampled are too dissimilar, the worker might create overly broad clusters. Clusters found in many worker segmentation sets may give rise to redundant clusters, while clusters whose items are sparsely split among segmentation sets may never be realized at all. As an example, [8] cite redundancies in Cascade’s top level clusters having both “green” and “seafoam green”, “blue” and “aqua”, as well as the encompassing category of “pastels”. While Crowd Synthesis used an iterative approach to address these redundancy problems, it trades this off with lowered robustness as issues with early workers’ categories can cascade throughout subsequent workers’ judgments. This suggests the design space of approaches for crowd clustering may be being critically limited by the assumption of splitting up the dataset into small, fixed pieces that prevent workers from gaining a more global context.

Another challenge with current crowd clustering approaches is that using human judgments to label each piece of data is costly and inefficient. Deluge addresses some issues with efficiency, improving on Cascade by reducing the number of human judgments elicited as the rate of new category generation slows [58]. However, these crowd clustering algorithms still require human judgments for every item, which is costly. In the real world data often follows a long-tailed distribution in which much of the data is captured by a small number of categories in the head of the distribution [219]. For such data in which many items in the head of the distribution are likely

to be highly similar, once humans have identified the meaningful categories and representative examples it would be more efficient if a machine could classify the remaining items in those categories. A danger with such an approach is that the sparse categories in the tail of the distribution with few examples may be difficult to train a machine to recognize, and so human judgments may have another important role in “cleaning up” low frequency categories.

This chapter describes Alloy, a hybrid approach to text clustering that combines the richness of human semantic judgments with the power of machine algorithms. Alloy improves on previous crowd clustering approaches in two ways. First, it supports better global context through a new “*sample and search*” crowd pattern which changes the crowd’s task from classifying a fixed subset of items to actively sampling and querying the entire dataset. Second, it improves efficiency using initial crowd judgments to help a machine learning algorithm cluster high-confidence unlabeled items in the head of the distribution (prominent categories), and then uses later crowd judgments to improve the quality of machine clustering by covering the tail of the distribution (edge cases and smaller categories). To achieve these benefits, Alloy introduces a novel modular approach we call “*cast and gather*” which employs a machine learning backbone to stitch together different types of crowd judgment tasks. While we provide a particular instantiation of the cast and gather approach here (with a hierarchical clustering backbone which gathers three types of crowd tasks, or “casts”), the general framework for modularizing multiple types of human judgments with a common machine-based backbone may inspire application to other contexts as well.

3.1.1 Related Work

Document and short text classification are well researched topics in natural language processing and machine learning. With enough labeled training data, state-of-the-art algorithms can often produce good results that are useful in real world applications. Yet building such systems often requires expert analysis of specific datasets both to manually design an organization scheme and to manually label a large set of documents as training data. Unsupervised approaches, or clustering, aim to discover structures on-demand and without expert preparation [91, 110, 202]. While these data mining approaches may discover dimensions (features) that provide a good separation of the dataset, the inferred categories can be difficult for a human to interpret, and many of them may not capture the most meaningful or useful structure in a domain due to high dimensionality or sparseness in the word vector space [16, 129]. To deal with these issues, researchers have explored ways to automatically discover topical keywords that can help identify useful categories in unstructured data such as TF-IDF, latent semantic analysis, and latent Dirichlet allocation [29, 66, 113, 147]. However, even with these improvements, automatic methods often still perform poorly, especially when the number of document is small, the lengths of the documents are short, or when the information is sparse.

More recently, researchers have begun to use crowds to organize datasets without predefined categories. Cascade [58] attempts to address abstraction and sampling problems by first having multiple workers generate categories for each item and then later having workers choose between them. By providing limited context to each worker (8 items or 1 item with 5 categories), it suffers from categories that can have varying levels of specificity. As a follow up study, Deluge [33] produces comparable results, but with significantly lower cost by optimizing its workflow using machine algorithms. In another line of research, Crowd Synthesis [8] showed that providing more context by simply showing more items can lead to significant better categories, suggesting

that global context is one of the key elements for crowd clustering algorithms. In general, most current systems provide context by showing a small sample of items, hoping that they captures the distribution of information in the larger dataset. We propose an alternative approach that builds up workers' mental models by asking them to repeatedly sample for new items, identify discriminative keywords, and search the dataset for similar items, taking advantage of people's capacity of information foraging [171].

A complementary set of approaches to crowd clustering research has focused on addressing the scaling problem through computation, applying approaches such as partial clustering [227], learning similarity metrics through triad-wise comparisons [207], or using matrix completion to reduce the number of labels needed from workers [228]. While these approaches have shown to be powerful on simple information such as images or travel tips, synthesizing more complex information can be difficult without providing novice crowdworkers with richer context or opportunities to deeply process the data.

3.2 System Design

The Alloy system clusters a collection of clips, or short text descriptions (Figure 3.3), using a machine learning backbone that gathers various judgments from human workers. In our terminology, each human task is a “Cast” for human judgements which are then “Gathered” together with the machine learning backbone. Alloy enables Casts (here, crowdworker tasks) of different types and in different orders to be fused together by calling a Gather after each one. In each Cast stages, arbitrary number of workers can be hired for better robustness or lower cost. In this chapter, we present three types of Casts with different purposes as well as one type of Gather. At a high level, the “Head Cast” is aimed at finding common categories in the head of the distribution, while the “Tail Cast” is aimed at classifying categories in the tail of the distribution for which machine clustering has low confidence. The “Merge Cast” aims to clean up existing categories by combining highly similar categories. We also describe a Gather Backbone that fuses the judgements from multiple crowdworkers, and connects multiple casts to form complete workflows. For ease of exposition we introduce each component in the context of a typical workflow: the Head Cast, the Gather, the Merge Cast, and the Tail Cast.

3.2.1 The Head Cast

The Head Cast aims to identify salient keywords to uncover the most common categories in the head of the distribution. Doing so involves challenges in providing workers sufficient context to know what a good category is, and also in how to structure their work process in order to train a machine learning algorithm to take over the classification of categories based on human-identified seeds and keywords. Previous studies show that presenting multiple items from a collection can help provide context to human workers [72], increasing the likelihood of obtaining better clusters. However, it can be difficult to determine how much context is sufficient and how to produce a good sample that captures the distribution of information of the whole dataset. Therefore, we introduce a new crowd-pattern we call “*sample and search*” for providing global context through active sampling and searching with keywords. We ask crowdworkers to identify coherent categories by presenting with four random items, but allowing them to replace each item by random sampling from the entire dataset until they are confident that the items will be in different categories in the final output. This requirement gives them the motivation to build up

Head Cast

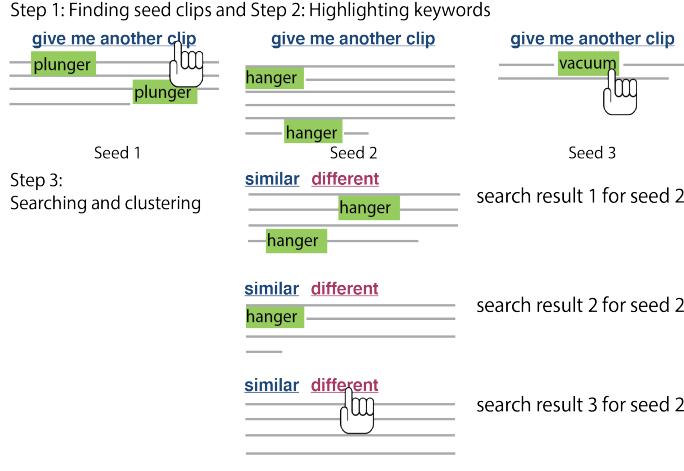


Figure 3.2: The interface and steps of the Head Cast HIT.

better global understanding of the dataset through repeated sampling. After obtaining the four seed items, we ask crowdworkers to identify keywords in each clips to search for related items in the dataset. This process takes advantage of people’s capacity of finding new information [171]. To create a familiar experience, we allow the workers to freely change their query terms and update the search results in real time. This way they can refine their searches based on the results, the same way as when conducting online information foraging tasks [112]. As shown in Figure 3.2, the Head Cast HIT interface consists of three steps:

- Finding seeds:** Four random seed clips are presented to each crowdworker. Over each clip, there is a button that allows them to replace the clip with another random clip from the dataset. They are then asked to replace any clips that are too similar to the other seed clips. The workers repeatedly replace the seed clips until the four clips at hand belong to four different answer categories.
- Highlighting keywords:** The crowdworker is then instructed to highlight one to three unique keywords from each of the four seed clips that best identify their topics.
- Search and label:** For each seed clip, we automatically search for similar clips from the entire corpus based on the highlighted keywords and TF-IDF cosine similarity. The crowd-worker is asked to label the top nine search results as *similar* to or *different* from their seed clips.

In Step 1, the crowdworkers need some understanding of the global context before they can confidently judge that the seeds belong to different categories in the final output. Previous work usually address this problem by presenting multiple items to each crowdworker, in hopes of sampling both similar and dissimilar items to give some sense of the global context. In reality it could be difficult to judge how many items is sufficient for different datasets, and overly small size could lead to bad samples that are unrepresentative of the global distribution. We took a different approach by presenting fewer items at first, but allowing workers to replace the seeds with random clips from the dataset. This provide them both the mechanism and motivation to explore the dataset until they have enough context to find good seed clips.

Tomato seedlings will need either strong, direct sunlight or 14-18 hours under grow lights. Place the young plants only a couple of inches from florescent grow lights. Plant your tomatoes outside in the sunniest part of your vegetable plot.

In its astrobiology roadmap, NASA has defined the principal habitability criteria as "extended regions of liquid water, conditions favourable for the assembly of complex organic molecules, and energy sources to sustain metabolism

Figure 3.3: Example clips from two datasets with crowd keywords.

The intuition behind Step 2 is that people are already familiar with picking out good keywords for searching documents related to a concept via their online information seeking experiences. In addition, requiring them to highlight unique keywords in the seeds first, further ensures that they are familiar with the concepts in the seed clips, before they search for similar items. In Step 3, the crowdworkers can still change and refine their highlights from Step 2, and the system will refresh the search results in realtime. This gives the crowdworkers both the motivation and mechanism to extract better keywords that lead to better search results to label. In Figure 3.3, we show two example clips from the datasets collected using the two questions: *How do I get my tomato plants to produce more tomatoes?* and *What does a planet need to support life?* The highlighted words in each clips are the keywords selected by one of the crowdworkers, showing that workers are finding useful words for classification.

To learn a similarity function between clips, we use the crowd labels and keywords to train a classifier that predict how likely two clips to be labeled as similar. Although the judgments from workers via the HIT interface about which clips go together provide valuable training information, we need to leverage these judgments to bootstrap similarity judgments for the clips that they did not label and to resolve potentially conflicting or partial category judgments. To do so we trained an SVM classifier in real-time to identify the set of keywords that are most indicative of categories and predict whether two clips in the dataset belonged to the same cluster. The training events are all possible pairwise combinations of clips in the clusters obtained with the HIT interface, which may include both positive (similar) and negative (different). The feature dimensions are all the keywords highlighted by the crowdworkers, and the value of each dimension is the product of the number of times that keyword occurred in the two clips. In general, the keywords labeled by the crowdworkers contain little irrelevant information compared to all words in the clips, but there could still be some highlighted words that are not indicative of a category. For example, one crowdworker worked on the dataset for "*How do I unclog my bathtub drain?*" labeled "use", "a", and "plunger" as three keywords. Even though *plunger* is a very indicative feature for clustering this dataset, the first two highlighted words seem too general to be useful. Using a linear kernel to estimate the weights for the different dimensions (i.e., keywords) seems well suited for our purpose [42, 223]. Further, if the same keyword is used by different crowdworkers but lead to very different labels, the linear SVM model will give lower weight to the corresponding dimension and thus lower the effects of keywords that are less indicative of the categories. We use LIBSVM which implements a variant of Platt scaling to estimate probability [141, 174]. The overall intuition is that the SVM classifier is doing a form of feature selection, weighting those words in clips that could maximally distinguish clips amongst clusters.

In a preliminary experiment, we tested using all words in the clips as features to train the SVM model. The intuition is machine algorithms might do a better job at identifying keywords that can outperform keywords identified by crowdworkers. However, the results show that using all words

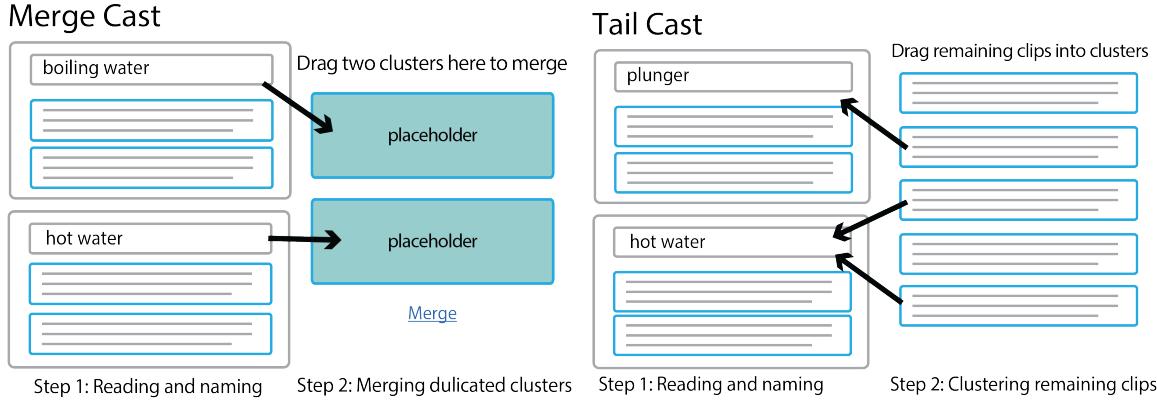


Figure 3.4: The HITs for Merge Cast: Naming and merging existing clusters and Tail Cast: Clustering remaining clips.

as features did not yield better results, and having much higher feature dimensions increases the training time significantly.

Finally, with the probability output of the SVM model as a similarity function between clips and a stopping threshold of 0.5 probability, we use a hierarchical clustering algorithm that serves as the Gather Backbone to capture head clusters.

3.2.2 Gather Backbone: Hierarchical Clustering

Using a multiple-stage approach with different types of microtasks can make it difficult to fuse together the different crowd judgements to form a coherent result. A key element to our approach in *casting* for category judgments in different ways is that we have a unifying mechanism to *gather* them back together. For example, throughout our process we cast for human category judgments in very different ways, including having people identify seed clusters (the Head Cast), merge duplicated categories (the Merge Cast), and classify the tail of the distribution (the Tail Cast). Instead of creating ad-hoc links between these judgments we propose using a unifying gathering mechanism composed of a machine learning backbone which translates the different *casted* judgments into similarity strengths used as the basis of clustering. We believe this *Cast and Gather* pattern may be useful as a way to conceptualize the relationship between machine algorithms and crowd judgments for a variety of tasks.

To build a complete clustering workflow with multiple casts, we use a hierarchical clustering algorithm as the backbone that connects different casts. More specifically, the backbone algorithm fuses the judgements from different crowdworkers working on the same cast into clusters, which, in turn, become the shared context transferred to the next cast of the workflow.

With a clip similarity function from the prior cast and a stopping threshold, the hierarchical clustering method initially treats each clip as a cluster by itself, and iteratively merges the two most similar clusters until a threshold is reached. The result is a partially clustered dataset with clusters and singletons. When the backbone is used after the last cast in the workflow, each singleton is then merged into the most similar cluster. The similarity between two clusters is defined as:

$$ClusterSim(\omega_1, \omega_2) = \frac{1}{|\omega_1||\omega_2|} \sum_{t_j \in \omega_1} \sum_{t_k \in \omega_2} ClipSim(t_j, t_k) \quad (3.1)$$

where ω_1 and ω_2 are the two clusters, t_j and t_k are each of the clips in ω_1 and ω_2 , respectively, and the $ClipSim()$ function is the given similarity function between clips.

3.2.3 The Merge Cast

While the Head Cast is designed to find the large clusters in the head of the distribution, since each crowdworker works independently, some of those clusters may actually be different subsets of the same larger category or the same categories based on different keywords (e.g., *sunlight* vs *natural lighting*). The Merge Cast is designed to consolidate existing clusters by merging duplicated categories. The input to this cast is a set of clusters that may or may not cover the entire dataset, and the output is fewer or equal number of clusters each with a list of ranked short descriptions. The challenge with detecting duplicate categories is that people need to understand what is in each category first. We start by presenting a set of existing clusters, and asking crowdworkers to name each of them. This acts as a defensive design[122] that ensures the crowdworkers understand the current context (scope and abstraction level), and also to obtain short descriptions for each of the clusters. Crowdworkers are then asked to merge identical categories by dragging them into the placeholders on the right (Figure 3.4).

If there are too many head clusters to fit into a microtask, the Merge Cast can be run recursively by first running on disjoint sets of existing clusters to consolidate them independently. Then, run another sets of Merge Cast on the output of each initial Merge Casts, and recurse until the output reduces to a set of clusters that could be presented in a global Merge Cast to ensure consistency. The assumption here is that the set of clusters in the final output of Alloy should be manageable by a single person to be useful. We also wanted to point out that the number of clusters is likely to scale much slower than the size of the dataset for many real-world data.

With the labels from the crowdworkers, we will again use the Gather Backbone to combine the judgements. The goal is to merge existing clusters if more than half of the crowdworkers also merged them in their solutions. Since in the Merge Cast workers can not break up existing clusters or reassign clips, we can formulate the clip similarity function as:

$$ClipSim(t_1, t_2) = \frac{1}{N} |\{\omega : t_1, t_2 \in \omega \text{ and } \omega \in \Omega\}| \quad (3.2)$$

where t_1, t_2 are the two clips, N is the total number of crowdworkers, Ω is the set of all clusters created by all crowdworkers, and ω is any cluster that contains both clips. This function is robust against a few workers doing a poor job. For example, if one crowdworker assigned every clip in the dataset to a single, general cluster (e.g., *answers*), the effect to the similarity function would be equivalent to having one less crowdworker and applying Laplacian smoothing. It is a common concern for crowd-based clustering methods that novice workers may create overly abstract categories (e.g., *solutions* or *tips*), that covers all items in the datasets. With our approach, it would require more than half of the workers to merge all items into a single cluster to generate a single cluster in the output.

From the output of the Gather Backbone, we rank the short descriptions associated with each

Dataset	sources	workers	clips	bad	clusters
Q1: How do I unclog my bathtub drain?	7	16	75	25%	8
Q2: How do I get my tomato plants to produce more tomatoes?	18	13	100	10%	8
Q3: What does a planet need to support life?	19	19	88	31%	7
Q4: What are the best day trips possible from Barcelona, Spain?	12	12	90	18%	16
Q5: How to reduce your carbon footprint?	20	11	160	14%	11
Q6: How do I unclog my bathtub drain?	17	23	159	14%	11
Wiki: Talk page sections for the Wikipedia <i>Hummus</i> article	N/A	N/A	126	0%	13
CSCW: Abstract sections of CSCW 2015 accepted papers	N/A	N/A	135	0%	45

Table 3.1: Datasets used for evaluation

cluster. Since clips are labeled by multiple crowdworkers, each cluster is associated with multiple descriptions via its clips. We use the F1 metric to rank these names to find the most representative description for each cluster, where the precision of a name label is defined as the number of clips in the cluster that it associates with divided by the size of the cluster, and recall as divided by the total number of clips associated with it.

3.2.4 The Tail Cast

The Tail Cast is designed to clean up the remaining singleton clips by classifying them into existing clusters or creating new clusters. The intuition is that even though machine learning techniques can produce high performance output, sometimes it is achieved at the expense of sacrificing the border cases. Human-guided “clean up” is often necessary for data produced by a machine learning model. The input of this cast is a set of existing clusters (with or without short descriptions) and a set of remaining clips. The output is a set of clusters with short descriptions.

We use an interface similar to the Merge Cast (Figure 3.4), and asked crowdworkers to review or name each of the existing clusters first, so that they build up better global understanding of the dataset before they organize the remaining clips. If Merge Cast was performed previously, their names are presented to lower cognitive load. The crowdworkers are then instructed to cluster the unorganized clips shown on the right by assigning them into existing clusters, creating new clusters, or removing uninformative clips. If there are too many remaining clips to fit into a single microtask, they are partitioned into groups of 20 items. Even though we may be dividing the remaining clips into partitions, all workers in the Tail Cast starts with learning the same global context that is the set of existing clusters from the Head Cast.

Finally, we use the Backbone Gather again to combine the multiple solutions from the crowdworkers. The goal is analogous to the goal of the Merge Cast: if two clips are assigned to the same category by more than half of the crowdworkers, they should be in the same cluster in the combined solution. For the similarity function, we simply replace the variable N in Equation 2 by the degree of redundancy.

3.3 Evaluation

3.3.1 Evaluation Metric and Datasets

Unlike evaluating a classification task, which would typically be based on the precision and recall of pre-defined classes, evaluating clusters is not as straightforward due to the potentially different

number of classes in the gold-standard and the system output. For example, high precision can be achieved by simply having more clusters in the output and the mapping between them. To address this, we use the normalized mutual information metric (NMI), which is a symmetric measurement sensitive to both the number of clusters, and the precision of each cluster. Specifically, it compares all possible cluster mappings to calculate the mutual information, and normalizes by the mean entropy so that the numbers are comparable between different datasets:

$$NMI(\Omega, C) = \frac{I(\Omega, C)}{0.5 * [H(\Omega) + H(C)]} \quad (3.3)$$

where Ω is the output clusters and C is the gold-standard clusters. The mutual information I is defined as:

$$I(\Omega, C) = \sum_k \sum_j P(\omega_k \cap c_j) \log \frac{P(\omega_k \cap c_j)}{P(\omega_k)P(c_j)} \quad (3.4)$$

where ω_k and c_j denotes each of the clusters in Ω and C , respectively. The probability $P(\omega)$ of item set ω is defined as $|\omega|/N$, where N is the number of total items. Finally, the mutual information is normalized by the mean entropy of Ω and C , so that the scores are comparable across datasets. To give some intuition, given w that maps to a gold-standard cluster c , we can calculate the precision by $P(w \cap c)/P(w)$ and recall by $P(w \cap c)/P(c)$, and the metric considers both with $P(w \cap c)/P(w)P(c)$. However, in reality it may be difficult to obtain such mappings, and the metric simply sums up scores of all possible mappings weighted by probability $P(w \cap c)$.

We use NMI for it is widely found in the literature for clustering evaluation. A more recent study found that it might favor datasets with more clusters, and proposed a variant that adjusts for randomness (AMI, [215]). We acknowledge this is a potential limitation, but found that the number of clusters Alloy produced were quite close to the gold-standard (average 10.3 vs 10.2), suggesting the concerns may be minimized. To be on the safe side, we also measured Alloy’s performance using AMI on two datasets and found similar results.

In order to evaluate Alloy, we compared it to other machine learning and crowdsourcing clustering approaches in three different contexts: information seeking, Wikipedia discussions, and research papers. These contexts all involve rich, complex data that pose challenges for automated or existing crowd approaches. Below we describe each dataset and how we either generated or collected gold-standards.

3.3.2 Information Seeking Datasets

We picked five questions asked on popular Q&A forums (e.g., Quora, reddit, and Yahoo! Answers) that covered a diverse range of information needs. We then posted these questions to Amazon Mechanical Turk (AMT), and asked each crowdworker to find 5 webpages that best answered the questions in Table 3.1. The top sources were sent to workers to highlight clips that would help answer the question via an interface similar to that described in [125]. The first four datasets (Q1 to Q4) collected consist of 75 to 100 clips, extracted from 7 to 19 webpages using 12 to 19 crowdworkers. In addition, we also collected two datasets with more than 150 clips (Q5 and Q6) by gathering more clips from the sources.

To generate gold standards, two graduate students clustered each dataset independently. Raters were blind to Alloy’s clusters, and no discussion on clustering strategies nor predefined cate-

gories were made prior to the process. Raters initially read every item in the dataset to build global understanding before they started organizing. Conflicts between raters were resolved through discussion. The first author participated in labeling two (out of the seven) datasets, but was always paired with another annotator outside of the research group. To measure inter-annotator agreement, we used the symmetric NMI metric as described in the previous section.

The agreements between raters are shown in Table 3.2. The datasets for “*How do I unclog my bathtub drain?*”, “*How do I get my tomato plants to produce more tomatoes?*” and “*What are the best day trips possible from Barcelona?*” had high agreement between the two annotators of 0.7 to 0.75 NMI. For the “*What does a planet need to support life?*” dataset, the agreement was significantly lower (0.48). We kept this dataset to show the limitations of the proposed method, and we will discuss further in later sections. For the two larger datasets Q5 and Q6, the agreement scores were around 0.6.

3.3.3 Research Papers

Since some of the questions in the above dataset were about common daily life problems, an open question is whether crowd judgements were based on workers’ prior knowledge or the context we provided them. To evaluate the system using more complex data where workers would likely have little prior knowledge we turned to research papers from the 2015 CSCW conference. For this dataset we used the official conference sessions as the gold standard for evaluation. The intuition is that conference organizers would place similar papers together in the same session. We acknowledge that the objectives of organizing conference sessions are not entirely the same as Alloy; most notably, conference session planning requires schedule conflict resolution and fixed size sessions. However, session co-occurrence represents valuable judgments from experts in the community about which papers belong to a common topic, and even though each cluster is smaller in size (e.g., 3-4 papers per session) we can look at whether papers put together by experts are also put together by Alloy and the other baselines [59].

3.3.4 Wikipedia Editor Discussion Threads

Wikipedia relies on its editors to coordinate effectively, but making sense of the archives of editor discussions can be challenging as the archives for a single article can consist of hundreds or thousands of pages of text. We use as a dataset the discussion archives of the *Hummus* article, a popular yet controversial article, and use the discussion threads as the set of documents. The talk page consists of 126 discussion threads about various issues of the main articles that spans over the past 10 years (Table 3.1). Two annotators read the main article and the full talk threads before they started the labeling process. The NMI score between the two annotators was .604, which is comparable to the two other large datasets Q5 and Q6.

Wikipedia data can be more difficult to organize than previously mentioned datasets, because it can be organized in very different ways, such as topics, relations to the main article sections, and mention of Wikipedia guidelines [8]. The annotators also had a hard time coming up with a gold standard through discussion, and found both their categorization solutions to be valid. Therefore, instead of creating a single gold standard, we report the NMI scores between Alloy’s output and each of the annotators.

DS	InterAnnot.	Workflow1	Workflow2	TFIDF	Keywords	LSA	LDA	#clusters	
								Alloy	exp
Q1	.734	.759* $\sigma=.033$.550* $\sigma=.093$.510	.647	.512	.478	7	8
Q2	.693	.687* $\sigma=.016$.467* $\sigma=.046$.534	.562	.537	.506	8	8
Q3	.477	.468	.425	.390	.440	.467	.442	7	7
Q4	.750	.727	.633	.673	.676	.704	.603	14	16
Q5	.630	.576	-	.568	.508	.582	.551	16	11
Q6	.588	.588	-	.462	.492	.497	.456	10	11
AVG	.645	.634	-	.523	.554	.550	.503	10.3	10.2
CSCW	-	.748	-	.584	.652	.691	.725	23	45

Table 3.2: Evaluation Results. * indicates mean of 11 runs using different workers.¹

3.3.5 External Validation, Robustness, and Generalizability

In the following sections, we will describe three experiments and their results followed by an application-oriented evaluation. For the three experiments, two workflows that uses the Gather to connect the different Casts are tested. The first experiment is an external evaluation that compares Alloy with other approaches. We use the full workflow that consists of the Head Cast, the Merge Cast, and the Tail Cast to cluster the six information seeking datasets (Q1-Q6), and compare with previous crowd-based methods and four machine algorithm baselines. The second experiment is an internal evaluation that tests the robustness of Alloy by using different number of workers in the Head Cast and the Tail Cast. Finally, in our last experiment, we test Alloy’s performance on two different types of datasets: Wikipedia editor discussions and research papers. Finally, to investigate the usefulness of the structures produced by Alloy, we used a prototype system called Knowledge Accelerator [88] to synthesize Alloy clusters for the information seeking datasets into report-styled articles and compare the articles against top Google search results.

3.3.6 Experiment 1: External Validation

We first look at how Alloy compares with machine algorithms, other crowd algorithms, and inter-expert agreements. In the Head Cast, crowdworkers highlight keyword and cluster similar clips via searching, and in the Tail Cast another set of crowdworkers organizes all remaining clips.

We compare this Workflow 1 to three baselines that are commonly used in the clustering literature: latent Dirichlet Allocation (LDA) [29], latent semantic analysis (LSA) [66], and TF-IDF [113, 147]. We also compare against a hybrid baseline that uses human-identified keyword vectors from the Head Cast. This aims to test the value of the approach beyond the human identification of keywords by trying to cluster using only the keywords. In addition to comparing against automatic methods, we also compare Alloy to a popular crowd based method. The evaluation conditions are summarized below:

- *Workflow1*. The workflow with ten crowdworkers each for the Head Cast and the Tail Cast for Q1-Q4. An additional five workers for the Merge Cast for Q5-Q6. Each HIT costs 1 USD.
- *TF-IDF*. Weighted cosine similarity as the similarity function for the Gather. No human-computation was employed.
- *Crowd keywords*. Cosine similarity based on worker-highlighted keywords from the Head

Cast as the similarity function for the Gather.

- *LSA*. The LSA model is used as the similarity function for the Gather. No human-computation was employed.
- *LDA*. The LDA topic model is used as the similarity function for the Gather. No human-computation was employed.
- *Cascade*. A version of Cascade with only one recursion using the default parameters as described in the paper.

Results

Alloy introduces a novel approach for providing context in the microtask setting with the sampling mechanism in the Head Cast. We captured crowdworkers' behavior during the tasks and found that nearly all (97.5%) workers used the sampling mechanism to gain context beyond the initial four items. On average, each worker sampled 15.1 items, and more specifically, 11.3% sampled more than 25 items, 23.8% sampled 15~24 items and 62.5% sampled 5~14 items.

Comparing with Machine Algorithms

On average, the proposed method performed significantly better and more consistent than all machine baselines (Table 3.2). In the worst case, Alloy clusters measured 0.058 NMI lower than the inter-annotator agreement, while the baseline systems measured more than 0.1 NMI lower in most cases. In a few cases some baselines also performed well (e.g., LSA performed slightly better on Q5), but none of them produced good results consistently across all datasets. Compared to the gold-standard clusters, Alloy produced clusters about as close to the gold-standard clusters as the two human annotators were to each other, despite the judges' advantages of having a global view of the datasets and multiple rounds of reading, labeling, and discussion. In addition, worker-identified keywords consistently outperformed TF-IDF, showing that the crowd-workers are extracting keywords in the Head Cast that are salient for identifying clusters each dataset. On the two larger datasets (Q5 and Q6), Alloy achieved similar performance as the four smaller datasets; better and more consistent comparing to the baseline systems, and near experts agreement comparing to the gold-standard.

Note that for every machine algorithm baseline we explored multiple parameters for each of the four questions, (hyper-parameters, number of topics, stopping threshold), and report the highest scores. The results of the baseline algorithms are likely over-fitting to the data, but we wanted to compare Alloy to these algorithms under their best possible settings [60].

Comparing with Previous Crowd Methods

We compare Alloy with Cascade using datasets Q1-Q4, a popular crowd-based method for discovering taxonomies in unstructured data based on overlapping crowd clusters [58]. We implemented a simplified version of Cascade using the parameters described in the paper, but with only one recursion. We acknowledge that fine tuning and multiple recursion might improve Cascade's performance, but the numbers from our evaluation are consistent with the results reported in the Cascade paper based on the same metric and similar datasets.

On average, 84% of categories generated with Alloy were shared with clusters in the gold standard, versus 50% for Cascade. Cascade produced soft clusters where child clusters did not necessarily have all the items included in their parents, which breaks the assumptions of us-

Expert	Alloy	Cascade Single Pass
Hot Water	Hot Water / Clearing a drain with hot water	problem/symptom (21) drain clean (55) slow drain (36) plumbing (52)
Plunger	Plunge / Vigorous Plunging Plunger / Dealing with Overflow	comments on why solutions are not working (8) how do i unclog a drain (46) bathroom (39) clean drain (42) chemical solution (16) drano (11) mechanisms that unclog drain (28) steps in unclogging the drain (45) manual solution (32) internet forum suggestion (33)
Plumbing Snake	Snake the Drain / Plumbing Snake	how do i unclog my bath tub (49) drain water (31) help unclog the drain (47) helpful tip (45)
Remove Cover	Remove the Drain Cover / Check drain cover	what should i do with a plunger cup to open a drain (9) plunge (15) help (37)
Chemicals	Drain Cleaner / Use drain cleaner for hair clogs	what to do with a drain clog (43) helpful response (35) technical advice (26)
Bent Hanger Wire	Remove Hair Clusters / Using a bent wire to clear a drain	drain (50) reasons why drains become clogged (15) reasons why a drain becomes clogged (13) uncategorized (2)
Call a Plumber		
Shop Vacuum		

Figure 3.5: Categories comparison for Q1

ing NMI. To produce a direct comparison, we use the gold standard to greedily extract best matching, overlapping clusters that cover all items, and evaluated them using the average F1. In essence, this simulates an omniscient “oracle” that gives Cascade the best possible set of cluster matches, and so is perhaps overly generous but we wanted to err on the conservative side. The average F1 scores for each questions using Alloy are .72, .54, .48, .52, and using Cascade are .50, .48, .42, .39, showing a consistent advantage across questions. Furthermore, Alloy achieved this better performance at a lower cost (average \$20 for Alloy vs \$71 for Cascade), suggesting that machine learning can provide valuable scaling properties. We show categories created by experts and elicited from the two systems in Figure 3.5 to give a better sense of the datasets and the output.

3.3.7 Experiment 2: Robustness

In this section, we examine the robustness of Alloy by varying the number of crowdworkers employed in the Head and the Tail Cast on datasets Q1-Q4. We start with having only 1 worker in the Head Cast, and evaluate performance as we hire more workers until we have 20. To test the two phase assumption, in a second condition, we switch to the Tail Cast after hiring 10 workers in the Head Cast, and continue to hire 1 to 10 more workers. This way, we can characterize the cost/benefit trade-offs in hiring different amount of human judgments. Further, by omitting the

¹We also evaluated Q1 and Q2 using the AMI metric that accounts for randomness. The inter-annotator agreements are .674 and .643, respectively, and Alloy performed .674 and .609, respectively. See the Evaluation Metric Section for detail.

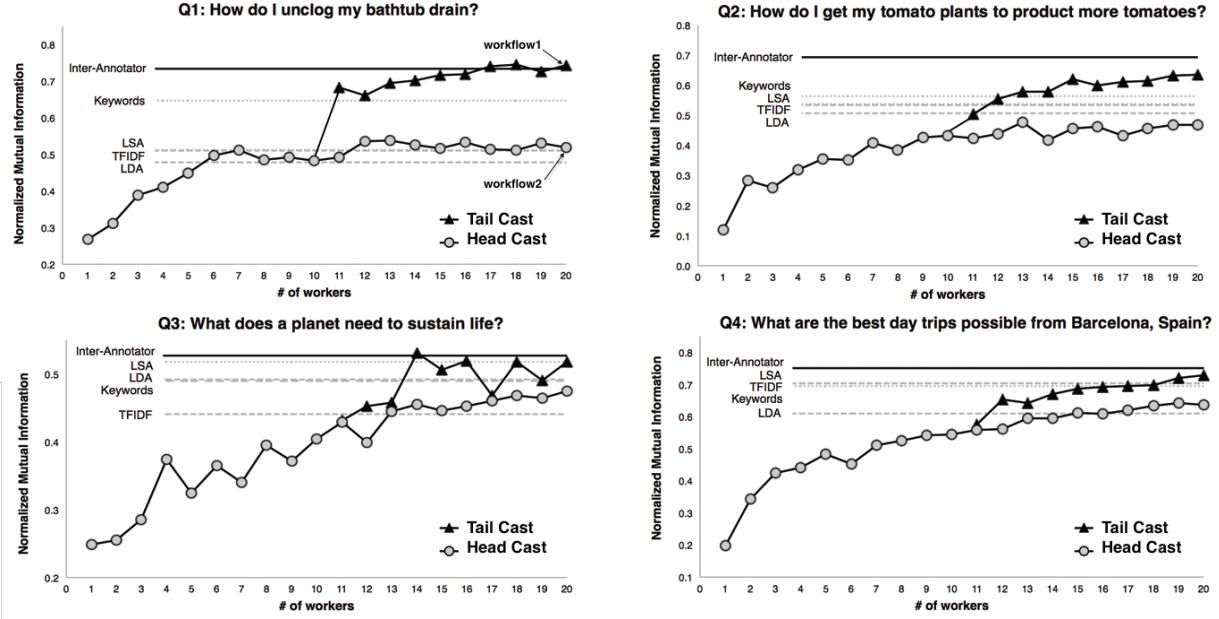


Figure 3.6: Performance comparison of using different number of crowdworkers in the Head Cast and the Tail Cast.

Tail Cast completely in the first condition, we can verify the two phase assumption by comparing the performance of a two-phase process (Head Cast and Tail Cast) with a one-phase control (Head Cast only) while equaling the number of workers:

- *Workflow1*. The workflow with ten crowdworkers each for the Head Cast and the Tail Cast. Each HIT costs 1 USD.
- *Workflow2*. The workflow with twenty crowdworkers and the Head Cast only. Each HIT costs 1 USD.

In addition, to test how robust Alloy is to the variance of crowdworkers on Amazon Mechanical Turk, we also hired eleven sets of ten different crowdworkers (a total of 440) for each Head and Tail Casts for Q1 and Q2.

Results

In Figure 3.6, we show the performance of employing different number of workers in the Head and the Tail Cast. Initially, increasing the number of workers in the Head Cast shows significant performance improvements. However, after gathering training data from around 10 workers, the performance gain from hiring additional crowdworkers decreases notably. Instead, performance improved significantly even with only a few additional crowdworkers in the Tail Cast to refine the clusters. Overall, having 10 crowdworkers in each of the Head and Tail Cast consistently outperformed having all 20 crowdworkers in the Head Cast across all four questions (Table 3.2), suggesting there is significant value in the Tail Cast.

For Q1 and Q2, we also ran Alloy eleven times using different crowdworkers, and compared the results against the gold-standard labels and also with each other. Comparing to the gold-

standards, which have inter-annotator agreements of .734 and .693 for Q1 and Q2 respectively, Alloy produced an average NMI of .759 (SD=.016) and .687 (SD=.016), respectively. Further, the average pair-wise NMI score of the 11 runs are .819 (SD=.040), and .783 (SD=.056), respectively, suggesting Alloy produces similar results using different crowdworkers on the same datasets.

3.3.8 Experiment 3: Other Datasets

In this experiment, we use the same distributed workflow to test Alloy using the Wiki and CSCW datasets as described in the Dataset Section, in order to test how Alloy generalizes to other types of data. These datasets contain long academic documents or editorial discourses that are infeasible to present multiple items to the crowdworker in one HIT. Instead, we show a small portion of each item in the datasets to the crowdworkers. For each item in the Wiki dataset, we display the thread-starter post and the first two replies. For the CSCW dataset, we present the abstract section of each paper, and compare results with the official conference sessions. Machine baselines were however given access to all of the text of the paper and the full discussion threads in order to provide a strong test of Alloy’s approach.

Results

For the CSCW dataset, Alloy outperformed all machine baseline systems with .748 NMI score using conference sessions as the gold standard Table 3.2. The Keyword baseline outperformed the TF-IDF baseline (.652 vs .584), showing that the crowdworkers are extracting valuable keywords in the Head Cast, despite that research papers may be difficult or impossible for crowdworkers to understand. On the other hand, Alloy produced 24 categories out of 135 abstracts, more than all other datasets. One possible assumption is that it may be more difficult for novice workers to induce abstract categories when organizing expert dataset, leading to higher number of more lower level categories in the outcome.

For the Wiki dataset, the NMI score between annotators was .604, which is comparable to the two other large datasets Q5 and Q6. Comparing to the two sets of expert labels independently, Alloy’s output measured .528 and .507. Compared to all previous results, Alloy seemed to perform less favorably on this dataset. As mentioned in the Dataset Section, the raters found this dataset the most difficult to organize, as there are many different valid structures that the two annotators were unable to reach an agreement also hints that the space of valid solutions may be larger on this dataset. In addition, we only showed the first three comments of each discussion to the crowdworkers, whereas the annotators and the machine baselines have access to the full discussion. We acknowledge length of items is a limitation, and will discuss in detail in the Discussion Section.

3.4 Application: Knowledge Accelerator

This work was previously published in ACM SIGCHI 2016 [88] and has been adapted for this document.

To evaluate the usefulness of structures generated by Alloy in a more realistic scenarios, we first used Alloy to clusters a larger set of information seeking datasets (Table 3.3) collected using the same procedure as described in section 3.3.2. We then developed a prototype system called the

How Do I Get My Tomato Plants To Produce More Tomatoes?

Contents <ol style="list-style-type: none"> 1. Tomatos - Feeding 2. Pruning Is Love 3. Maintenance And Harvesting 4. Tomatos - Proper Potting Procedure 5. Weather And Sunlight Conditions 6. Growing Tomatoes 7. Tomatos - Stakes And Support 	<p>Tomatos - Feeding</p> <p>Producing better tomato plants is as simple as picking the perfect soil. There are many market soils or one can add a few things to their own soil. Extra nutrients go a long way in producing more tomatoes per plant.</p> <p>Tomatoes are heavy feeders since they are smaller plants that depend on the bushy growth to support fruit production. They can benefit from some added nutrition even if you use the best soil. Cutting back on nitrogen will ensure a big, gorgeous pile of fruit coming your way in no time!</p> <p>Tomatoes take up nutrients the best when the soil pH ranges from 6.2 to 6.8. They need a constant supply of major and minor plant nutrients. Following the rates on the fertilizer label, mix a balanced timed-release or organic fertilizer to the soil as you prepare planting holes.</p> <p>Feeding tomatoes regularly is critical for a good yield. At the very least, you need a good liquid food that is high in potassium.</p> <p>Any tomato feed from a garden center should do the job. If you want take it a step further, check out Sea Nymph's natural seaweed-based feed or BioBizz's BioGrow, which include molasses to feed the microbes in the soil. About half way through the season, I add a 1 inch (2.5 cm) layer of worm compost or local farm manure to the top of my containers. This adds extra nutrients and soil life.</p> <p>Amend your plant beds with your own or purchased compost; dry, timed-release fertilizer; and most importantly, worm castings. Add 5 cubic feet of Gardner & Bloome compost; 5 quarts of Gardner & Bloome 4-6-3 Tomato, Herb & Vegetable fertilizer; and a quart of 100% pure worm castings for every 50 square feet of garden space.</p> <p>References:</p> <ul style="list-style-type: none"> • Vertical veg man: how to grow tomatoes successfully (www.theguardian.com) • Tomatoes..How To Get The Most From Your Plants In The Garden! (oldworldgardenfarms.com) • Love Apple Farms (www.growbetterveggies.com) • 10 Tips for Growing Great Tomatoes (gardening.about.com) 	 <p>Tomatos - Feeding</p> <p>Producing better tomato plants is as simple as picking the perfect soil.</p> 
--	---	--

Figure 3.7: Example report synthesized by the Knowledge Accelerator system. The table of content on the left listed cluster names generated from Alloy, each corresponded to a different section in the report.

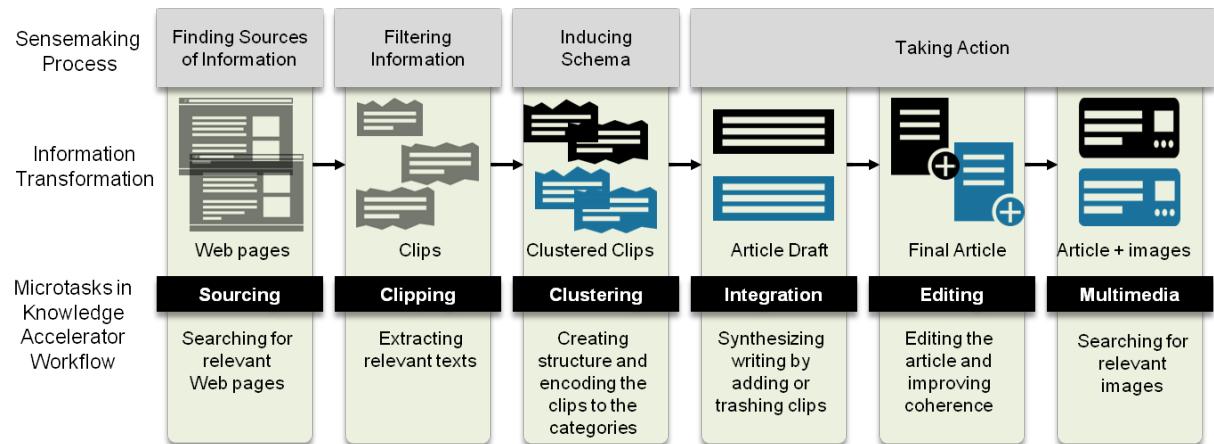


Figure 3.8: The process of the Knowledge Accelerator (KA). Alloy is used for the Clustering Stage of the pipeline.

Question	N	Score
Q1: How do I unclog my bathtub drain?	116	0.292 *
Q2: How do I get my tomato plants to produce more tomatoes?	177	0.420 *
Q3: What are the best attractions in LA if I have two little kids?	158	-0.044
Q4: What are the best day trips possible from Barcelona, Spain?	98	-0.109
Q5: My Worcester CDi Boiler pressure is low. How can I fix it?	139	0.878 *
Q6: 2003 Dodge Durango has an OBD-II error code of P440. How do I fix it?	138	0.662 *
Q7: 2005 Chevy Silverado has an OBD-II error code of C0327. How do I fix it?	135	0.412 *
Q8: How do I deal with the arthritis in my knee as a 28 year old?	139	0.391 *
Q9: My Playstation 3 has a solid yellow light, how do I fix it?	119	0.380 *
Q10: What are the key arguments for and against Global Warming?	138	0.386 *
Q11: How do I use the VIM text editor?	138	0.180

* = significant at $p < 0.01$ after Bonferroni correction

Table 3.3: Average difference between the KA output and top websites for the eleven questions (positive indicates higher ratings for KA, negative indicates higher ratings for the competing website). Each rating was an aggregate of 6 questions on a 7-point Likert scale.

“Knowledge Accelerator” (KA) to synthesize the output of Alloy into articles. Each of the cluster produced by Alloy corresponds to a different section in an article. An example of the output of the system for the target question “How do I get my tomato plants to produce more tomatoes?” can be found in Figure 3.7.

In addition, the KA system probes how to accomplish a complex information synthesis task entirely through relatively small contributions. We limited our maximum task payment to \$1 US, aimed at incentivizing a Target task time of approximately 5-10 minutes. Critically, the KA system accomplishes this process without a core overseer or moderator. Figure 3.8 shows the overview of the KA System with Alloy being the Clustering Stage. For more details on the KA system refer to [88].

We evaluated the usefulness and coherence of the articles by comparing them against webpages an individual might use if they were to complete the same tasks without KA and Alloy — Top Google search results that consists of expert-written articles published by trusted sources such as CDC.gov or the New York Times, as well as popular online forums such as TripAdvisor and Yahoo Answers.

3.4.1 Experimental Settings

Eleven topics were selected for evaluation by browsing question and answer forums, Reddit.com, and referencing online browsing habits [39]. For questions Q3 and Q8 we added additional constraints (i.e., having kids and age) to test the performance of the system for more personalized questions. To compare the two conditions, participants were recruited through the Amazon Mechanical Turk US-only pool and paid \$1.50 for rating two webpages. Each participant was randomly assigned an output article from KA and a top search result webpage for the same topic (Figure 3.9), and rate both webpages based on six criteria using 7-point Likert scale questions and provided free-form explanations: *comprehensiveness, confidence, helpfulness, trustworthiness, understandability, and writing*. We averaged ratings on these dimensions into a single score representing the overall perceived quality of the page.

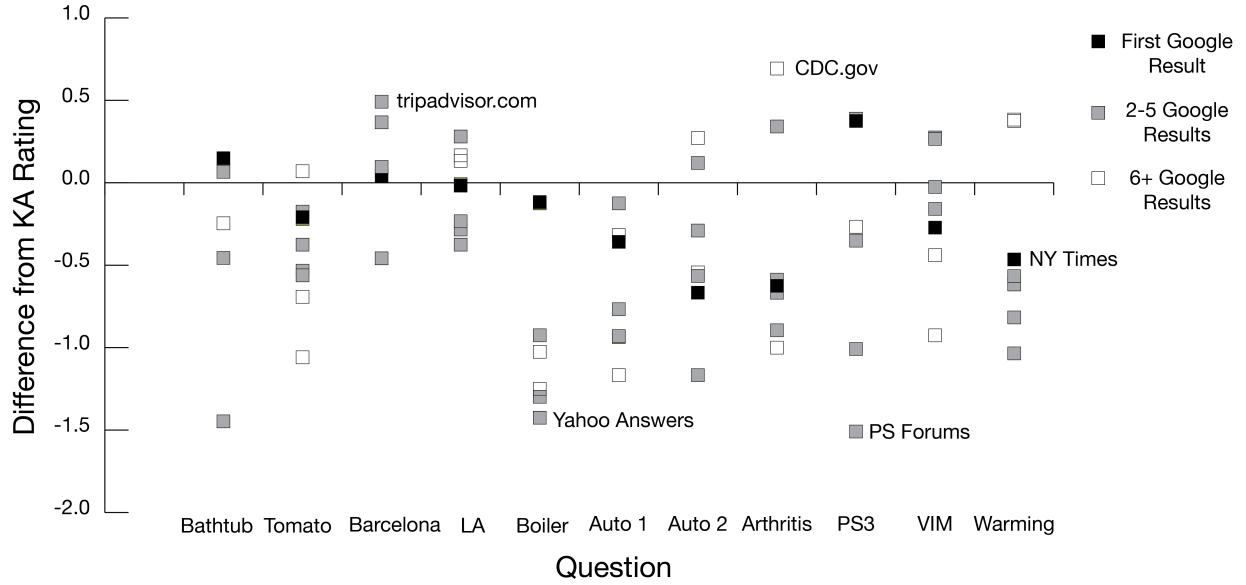


Figure 3.9: Results across questions and websites. Points represent the average aggregate score difference between the KA answer and an existing site

Phase	Task Pay	Avg. # of Tasks	Avg. Cost
Sourcing	\$0.25	15	\$3.75
Clipping	\$0.50	21.6	\$10.80
Alloy Head Cast	\$1.00	10	\$10.00
Alloy Merge + Tail Cast	\$1.00	10	\$10.00
Integrate	\$0.50	37.2	\$18.60
Edit 1	\$0.75	28.8	\$21.60
Edit 2	\$1.00	28.8	\$28.80
Images	\$0.50	9	\$4.50
Total		160.4	\$108.05

Table 3.4: Average number of worker tasks and average cost per phase, and overall, to run a question.

categories induced during clipping (without Alloy):

Boil Water, use hot water, Plunger, try a snake, How to Remove drain stopper, bleach, Use Drano Max Gel, baking soda, drain, tips to unclog, problem, tools, research, internet research, ..., etc.

categories induced by Alloy:

Hot Water, Plunge, Plunger, Snake the Drain, Remove the Drain Cover, Drain Cleaner, Remove Hair Clusters.

gold-standard categories:

Hot Water, Plunger, Plumbing Snake, Remove Cover, Chemicals, Bent Wire Hanger, Call a Plumber, Shop Vacuum.

Figure 3.10: Categories induced from different stages for Q1: *How do I unclog my bathtub drain?*

3.4.2 Results

The costs of running a question through the KA system is shown in Table 3.4. Across the 11 topics we tested, a full run with around 100 short text clips costed an average of \$108.50, of which around \$15 is spent on searching and extracting the text clips from webpages, \$20.00 is spent by the Alloy system, and the rest on synthesizing each of the Alloy clusters into a section in the final article and making sure the different sections are coherent.

Aggregating across all questions, KA output was rated significantly higher than the top 5 Google results (KA: $\bar{x} = 2.904$ vs Alt. Sites: $\bar{x} = 2.545$, $t(1493) = 13.062$, $p < 0.001$). An analysis of individual questions corrected for multiple comparisons is shown in Table 3.3.

The strongly positive results found were surprising because some of the websites in the comparison set were written by experts and had well-established reputations. Only on the two travel questions, Barcelona ($\bar{x} = -0.109$) and LA ($\bar{x} = -0.044$), and the VIM question ($\bar{x} = 0.180$) did the KA output not significantly outperform the comparison pages. A closer examination of these pages suggests that for the two travel questions, because of the strong internet commodity market surrounding travel, a considerable amount of effort has been spent on curating good travel resources. Even with the slightly more specific LA query, there were still two specialized sites dedicated to attraction for kids in LA (MommyPoppins.com and ScaryMommy.com). The VIM question represented a mismatch between our output and the question style. A number of the sources for the question were tutorials, however in the clipping phase, these ordered tutorials were broken up into unordered clips, creating an information model breakdown. This points out an interesting limitation in the KA approach, and suggests that adding support for more structured answers (e.g., including sequential steps) could be valuable future work.

The strong performance of the system is perhaps surprising given that its output was generated by many non-expert crowd workers, none of whom saw the big picture of the whole, and Alloy is a core component that provided useful and coherent structures for producing the final report. Initially we had workers provide labels to categorize each clip, which we planned to use to develop a structure for the article. However, the lack of context of the bigger picture made these labels poorly suited for inducing a good structure. For example, in Figure 3.10 the top box shows the category structure induced by crowdworkers during clipping and without using Alloy during clipping, categories induced using Alloy, and gold standard categories developed by two independent annotators with access to all clips and sources, respectively. Categories induced

without using Alloy matched poorly with the gold standard categories, and include categories with very different abstraction levels (e.g., *Use Drano Max Gel vs tips*). On the other hand, Alloy produced categories that were more coherent and matched more with gold-standard categories.

While we do not believe that this should be interpreted as a replacement for expert creation and curation of content, instead, the power of the system may actually be attributable to the value created by those experts by generating content which the crowd workers could synthesize and structure into a coherent digest. This explanation suggests that the approach would be most valuable where experts generate a lot of valuable information that is unstructured and redundant, such as the automotive questions in which advice from car enthusiasts was spread across many unstructured discussion forums. In contrast, KA's output did not outperform top web sources for topics such as travel, where there are heavy incentives for experts to generate well structured content. We believe its performance is likely due to its aggregation of multiple expert viewpoints rather than particularly excellent writing or structure per se, never the less, the KA system showcased that the structures produced by Alloy can be synthesized into coherent articles that were useful for exploratory searchers.

3.5 Discussion

In this chapter, we took a step towards tackling the problem of clustering high-dimensional, short text collections by combining techniques from natural language processing and crowdsourcing. By using a two-phase process connected by a machine learning backbone, our proposed method compensates for the shortcomings of crowdsourcing (e.g., lack of context, noise) and machine learning (e.g., sparse data, lack of semantic understanding). As part of the system we introduced an approach aimed at providing greater context to workers by transforming their task from clustering fixed subsets of data to actively sampling and querying the entire dataset.

We presented three evaluations that suggest Alloy performed better and more consistently than automatic algorithms and a previous crowd method in accuracy with 28% of the cost (Exp.1), is robust to poor work with only 20 workers (Exp.2), and is general enough to support different types of input (Exp.3). Qualitatively, we noticed Alloy often produced better names for categories than machine algorithms would be capable of, including names not in the text (e.g., a cluster including items about *smart thermostats* and *solar panels* was named “*Home Improvements*” which was not in the actual text).

One potential concern might be whether Alloy’s tasks take too long to be considered microtasks. While Alloy deploys HITs that take more than a few seconds to finish, we think they are still comparable to other complex microtask systems such as Soylent [19] and CrowdForge [123]. Specifically, based on a total of 281 HITs, the median run-time for the Head Cast HITs is 7.5 minutes ($M=8.3$, $SD=4.1$), for Merge Cast 8.3 minutes ($M=16.2$, $SD=15.6$), and for Tail Cast 11.4 minutes ($M=13.2$, $SD=6.1$). Despite having less workers doing longer tasks, Alloy performed consistently across different sets of workers on the same datasets.

During development, some assumptions, both explicitly and implicitly, were made about the input of the system: 1) there are more clips than categories. 2) the categories follow a long-tailed distribution. 3) clips belong to primarily one cluster. 4) there is a small set of gold-standard clusters. 5) workers can understand the content enough to cluster it. Note that we do not assume the crowdworkers can understand the semantics of the content, but just enough to identify ideas

that are salient and common in the dataset. Thus they may be able to cluster complex topics such as machine learning without understanding those topics if enough relational context is embedded in the clips. For example, an abstract of a research paper may say “this paper uses POMDP machine learning approaches to cluster text”, they might put it in a “clustering” cluster without knowing what a POMDP is.

One obvious limitation to our approach is clustering long documents. This is a common limitation for crowd-based systems that rely on workers reviewing multiple items for context (either from random selection or active sampling). It becomes infeasible to fit multiple items in a single HIT if the length of each item is long. Another related limitation is organizing documents that describe multiple topics. Lab studies in a past work [125] showed that individuals are able to decompose long documents into short clips of single topics during information seeking tasks. One way to expand the proposed method to overcome the length limitation could be splitting documents into short snippets, either with the crowds or machine algorithms, and create topical clusters using Alloy.

Another limitation is organizing datasets that are inherently difficult to structure categorically. For example, concepts in Q3 (*planetary habitability*) have causal relationships without clear categorical boundaries (e.g., *distance to sun*, *temperature* and *liquid water*). As a result, all approaches had significant trouble, including low agreement between human annotators. On the other hand, some dataset can be organized categorically in multiple ways. In Q4 (*Barcelona*) we found that some categories fit a *place* schema (e.g., *Sitges*, *Girona*) while other categories fit a *type* schema (e.g., *museums*, *beaches*). One approach for addressing this could be trying to cluster workers to separate the different kinds of schemas; however, upon inspection we found that individual workers often gave mixtures of schemas. This interesting finding prompts further research to investigate what cognitive and design features may be causing this, and how to learn multiple schemas.

Looking forward, we identified a set of patterns that may be useful to system designers aiming to merge human and machine computation to solve problems that involve rich and complex sensemaking. The hierarchical clustering backbone we use to integrate judgments from a variety of crowdworker tasks allows us to *cast* for different types of crowd judgments and *gather* them into a coherent structure that iteratively gets better with more judgments. We also introduce useful new patterns for improving global context through self-selected *sampling* and keyword *searching*. One important consideration these patterns bring up is that while previous ML-based approaches to crowd clustering have focused on minimizing the number of judgments, we have found it is at least as important to support the rich context necessary for doing the task well and setting up conditions that are conducive for crowdworkers to induce meaningful structure from the data.

We hope the patterns described in this chapter can help researchers develop systems that make better use of human computation in different domains and for different purposes. For example, the *sample and search* pattern could potentially be adapted to support other tasks such as image clustering, where crowdworkers could use the sampling mechanism to get a sense of the variety of images in the dataset, highlight discriminative objects, and label images queried based on features extracted from the highlighted regions. Furthermore, the *cast and gather* pattern may provide a useful framework for combining crowds and computation that is both descriptive and generative. For example, Zensors [135], a crowd-based real-time video event detector, could be considered a form of the cast and gather pattern which uses a classification algorithm

instead of a clustering algorithm as a backbone, and casts for human judgements whenever its accuracy falls below a threshold (e.g., if an environmental change lowers precision), with the classifier backbone retrained with the new human labels. While we used a clustering backbone in this work, future system designers might consider other machine learning backbones (e.g., classification or regression algorithms) for different tasks. Overall, we believe this approach takes a step towards solving complex cognitive tasks by enabling better global context for crowd workers and providing a flexible but structured framework for combining crowds and computation.

Chapter 4: Revolt

Exploiting Disagreements For Concept Evolution in Crowd Labeling

This work was previously published in ACM SIGCHI 2017 [45] and has been adapted for this document.

This chapter describe the second of the two crowd systems in this dissertation that explored ways to provide global context in sourced data exploration. Unlike Alloy that focused on providing global context to the crowdworkers who were constraint by their capacity to process large amounts of data in microtasks, this second system focused on providing global context to the requesters who typically turned to crowdsourcing for its ability to scale to large datasets that they themselves do not have the capacity to process fully. Here I focused on another common approach of data analysis of labeling items in a dataset with predefined categories, a crucial process for generating labels for training machine learning models. Crowdsourcing provides a scalable and efficient way to construct labeled datasets for training machine learning systems. However, creating comprehensive label guidelines for crowdworkers is often prohibitive even for seemingly simple concepts. Incomplete or ambiguous label guidelines can then result in differing interpretations of concepts and inconsistent labels. Existing approaches for improving label quality, such as worker screening or detection of poor work, are ineffective for this problem and can lead to rejection of honest work and a missed opportunity to capture rich interpretations about data. We introduce *Revolt*, a collaborative approach that brings ideas from expert annotation workflows to crowd-based labeling. Revolt eliminates the burden of creating detailed label guidelines by harnessing crowd disagreements to identify ambiguous concepts and create rich structures (groups of semantically related items) for post-hoc label decisions. Experiments comparing Revolt to traditional crowdsourced labeling show that Revolt produces high quality labels without requiring label guidelines in turn for an increase in monetary cost. This up front cost, however, is mitigated by Revolt's ability to produce reusable structures that can accommodate a variety of label boundaries without requiring new data to be collected. Further comparisons of Revolt's collaborative and non-collaborative variants show that collaboration reaches higher label accuracy with lower monetary cost.[45]

4.1 Introduction

From conversational assistants on mobile devices, to facial recognition on digital cameras, to document classifiers in email clients, machine learning-based systems have became ubiquitous in our daily lives. Driving these systems are machine learned models that must be trained on representative datasets labeled according to target concepts (e.g., speech labeled by their intended commands, faces labeled in images, emails labeled as spam or not spam).

Techniques for collecting labeled data include recruiting experts for manual annotation [209], extracting relations from readily available sources (e.g., identifying bodies of text in parallel online translations [48, 180]), and automatically generating labels based on user behaviors (e.g., using dwell time to implicitly mark search result relevance [1]). Recently, many practitioners have

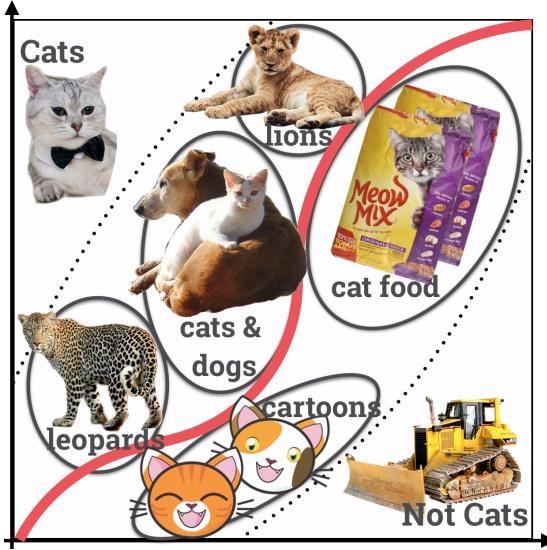


Figure 4.1: Revolt creates labels for unanimously labeled “certain” items (e.g., *cats* and *not cats*), and surfaces categories of “uncertain” items enriched with crowd feedback (e.g., *cats and dogs* and *cartoon cats* in the dotted middle region are annotated with crowd explanations). Rich structures allow label requesters to better understand concepts in the data and make post-hoc decisions on label boundaries (e.g., assigning *cats and dogs* to the *cats* label and *cartoon cats* to the *not cats* label) rather than providing crowd-workers with a priori label guidelines.

also turned to crowdsourcing for creating labeled datasets at low cost [198]. Successful crowd-sourced data collection typically requires practitioners to communicate their desired definition of target concepts to crowdworkers through guidelines explaining how instances should be labeled without leaving room for interpretation. The guideline generation process is similar but often less rigorous than the process used by expert annotators in behavioral sciences [145, 217] whereby experts independently examine a sample of data, generate guidelines especially around possibly ambiguous concepts discovered in the data, and then discuss and iterate over the guidelines based on feedback from others [133]. The guidelines are used as instructions in crowdsourced labeling tasks given to multiple crowdworkers for redundancy. Label disagreements are commonly seen as noise or failure to carefully follow the guidelines, and later corrected through simple majority voting.

While traditional crowd-based labeling has produced many novel datasets used for training machine learning systems [68, 131, 175], a common assumption in labeled data collection is that every task has one correct label which can be recovered by the consensus of the crowd [12]. This assumption, however, rarely holds for every item even for simple concepts (e.g., *cat* vs. *not cat* as illustrated in Figure 4.1) and even experts have been shown to vary their labels significantly on the exact same data due to their evolving interpretations of the target concept [133]. Specifying comprehensive guidelines that cover all the nuances and subtleties in a dataset would require close examination of much of the data which is typically infeasible in the crowdsourcing settings. Crowdworkers are then often presented with incomplete guidelines and left to make their own decisions on items open to interpretation. Not only can this lead to poor quality labels and machine learning models with low accuracy, efforts to detect poor quality work (e.g., [36, 90, 108]) in these cases can actually be harmful due to rejection of honest work. More

fundamentally, limiting crowdworkers to providing feedback only in terms of predefined labels, failing to capture their confusions and reasoning, presents a lost opportunity to discover and capture rich structures in the data that the crowdworkers had encountered.

In this paper, we present *Revolt*, a collaborative crowdsourcing system that applies ideas from expert annotation workflows to crowdsourcing (e.g., supporting flagging of ambiguous items and discussion) for creating high quality training labels for machine learning. Revolt enables groups of workers to collaboratively label data through three stages: *Vote* (where crowdworkers label as in traditional labeling), *Explain* (where crowdworkers provide justifications for their labels on conflicting items), and *Categorize* (where crowdworkers review explanations from others and then tag conflicting items with terms describing the newly discovered concepts). The rich information gathered from the process can then be presented to requesters at various levels of granularity for post-hoc judgments to define the final label decision boundaries.

Revolt requires no pre-defined label guidelines aside from the top-level concept of interest (e.g., faces, spam). As a result, this approach reverses the traditional crowdsourced labeling approach by shifting label requester efforts from guideline creation to post-hoc analysis. The rich structures provided by our approach has the additional benefit of enabling label requesters to experiment with different label decision boundaries without having to re-run label generation with the wide variety of possible label guidelines. For example, for collecting labels of images of *Cats* (Figure 4.1), Revolt produces structures that group together ambiguous sub-concepts such as *cartoon cats*, *cat food*, *leopards* and *lions* along with descriptive explanations about the structures. Requesters can then review these structures and experiment with machine learning models that are trained to identify *leopards* and *lions* as *Cats* or not.

This paper makes the following contributions:

- A new approach to crowdsourcing label collection that employs crowds to identify uncertain aspects of the data and generate rich structures for post-hoc requester judgments, instead of trying to clearly define target concepts beforehand with comprehensive guidelines.
- *Revolt*, an implementation of our collaborative crowdsourcing approach that builds structures containing rich enough information for generating training labels for machine learning. We present both real-time and asynchronous versions of our approach.
- An experiment comparing Revolt to traditional crowd-based labeling on a variety of labeling tasks showing Revolt can produce high quality labels without the need for guidelines.
- An experiment comparing Revolt to its non-collaborative variants showing the benefits of collaboration for reducing cost and increasing quality.

4.2 Related Work

4.2.1 Data Labeling Techniques

Data labeling or annotation is a common practice for many research areas. In social and behavioral sciences, researchers annotate (or code) data to build up theories about collected data, and then analyze the annotated results to discover interesting phenomena [203]. This approach often involves multiple experts working in iterative and collaborative workflows. For example, annotators typically first examine and manually label a dataset (or subset of the dataset) independently and then compare and discuss their labels to iteratively refine a set of combined label

guidelines [145, 217]. Multiple iterations of data examination, label discussion, and guideline refinement may also occur to ensure the quality and coverage of the final guidelines. Once the guidelines stabilize, annotators can then independently label additional data accordingly to produce consistent final labels with high agreement.

Similar collaborative and iterative workflows have been reported for creating high-quality labeled datasets used in natural language processing and machine learning (e.g., [133, 152, 220]). For example, Kulesza et al. [133] found that annotators often evolved their conceptual definition of a target concept and their corresponding labels throughout the course of observing more items in a dataset. Here, allowing annotators to create explicit structures designating ambiguous items discovered during labeling enabled them to gradually build up a better global understanding of the data and generate more consistent final labels. Wiebe et al. [220] also proposed an iterative and collaborative workflow that relies on comparing and discussing conflicting labels amongst expert annotators to construct and refine shared labeling guidelines for producing training labels for complex datasets. These iterative and collaborative processes provide expert annotators systematic ways to learn about and discuss different interpretations of data during labeling.

While expert annotation has been used in creating labeled datasets for machine learning, this process is often too costly and time consuming to scale to the large datasets required for modern machine learning algorithms. As an example, the Penn Treebank dataset that is commonly used in natural language processing for training part-of-speech sequence labelers and syntactic parsers, was built by teams of linguists over the course of eight years [209]. Another example from a previous work showed labeling 1,000 English sentences took four experts nine hours each to iteratively refine their guidelines by labeling items independently then discussing together [220]. Many researchers and practitioners have therefore recently turned to crowdsourcing to label data for its scalability and relatively low cost [68, 131, 175]. However, despite its efficiency, researchers have also reported difficulty obtaining high quality labels using crowdsourcing [5, 67]. Multiple factors can contribute to poor quality labels, such as poor work from inattentive labelers, uncertainty in the task itself (resulting from poorly written guidelines or confusing interfaces), varying worker backgrounds and prior knowledge, or items that are difficult to understand by novice workers [127].

4.2.2 Improving the Quality of Crowdsourced Labels

While disagreements between expert annotators are typically resolved through discussing and refining guidelines [152], disagreements in crowdsourcing are commonly seen as labeling errors to be corrected through majority voting over independent redundant judgments of crowdworkers [115]. Methods for further improving the quality of crowdsourced labels can be mainly broken down into two camps [124]: techniques for preventing poor quality work and techniques for post-hoc detection of poor quality work. Prevention techniques include screening for crowdworkers capable of different tasks [71, 116], pre-training crowdworkers [75], maintaining quality while controlling for cost via dynamic task allocation [33, 210], or designing interfaces or payment structures to motivate good work [90, 163, 182]. Post-hoc identification techniques include probabilistic modeling based on crowdworker agreements for weighted voting [108], analyzing crowdworker behaviors during tasks [187], and using additional crowdworkers to review the work of others [36, 90].

A common assumption in previous work is that every item has one correct label, and conflicts

among crowdworkers are the result of poor quality work from novice or inattentive workers. However, constructing comprehensive and clear instructions about how to correctly label a dataset is often not possible due to the large variety of nuances and subtleties that may exist in the data, even for seemingly simple topics. For example, requesters wanting to identify *cat* photos in a dataset might not be aware that the dataset also contains photos of *leopards*, and/or that leopards are sometimes referred to as *big cats*. As a result, crowdworkers often have to label with incomplete information. Concepts not specified in guidelines are then open to interpretation and confusion amongst crowdworkers (e.g., “should *leopards*, *lion cubs*, or *cartoon cats* be labeled as *cats*?”) Figure 4.1), potentially leading to inconsistent labels (e.g., only some *leopard* items being labeled as *cats*). Methods for identifying poor work are ineffective in these cases and can be harmful to both crowdworker and requester reputations due to rejection of honest work. More fundamentally, this suggests a lost opportunity for requesters to discover interesting new concepts already identified by human computation during the labeling process since the crowdworkers are typically constrained to provide feedback in the form of predefined labels (e.g., *cats* or *not cats*, but not *leopards*).

Even if requesters attempt to create comprehensive guidelines, they often have to review large portions of a dataset to do so which can be prohibitively expensive. Moreover, as guidelines become more complete, they can also become longer and more complicated (e.g., [105] and [155]), requiring more crowdworker training or resulting in more errors. If the resulting label quality is inadequate, requesters will typically have to go through the tedious process of reviewing inconsistencies, identifying sources of confusion, updating the guidelines, and collecting entirely new labels given the updated guidelines [133], potentially doubling the monetary cost of the process.

4.2.3 Harnessing the Diversity of Crowdsourcing

Instead of treating crowdworker disagreement as noise introduced by poor work or lack of expertise, researchers have recently begun exploring methods to harness these disagreements as valuable signals. For example, researchers have found that disagreements amongst novice workers in syntactic labeling tasks often mirror disagreements amongst linguists [173] and are useful signals for identifying poor task designs [107]. In another example, Kairam and Heer [115] used label agreements to cluster crowdworkers into worker types (e.g., *liberal* and *conservative* labelers that identified different amount of targets in an entity extraction task). Manual analysis of these clusters were then used to improve future task designs. In contrast to this previous work, we use crowd disagreements to identify and explain ambiguous concepts in data for the purpose generating labeled data for machine learning.

Most closely related to our work is the MicroTalk system [77] which used crowd diversity to collect and present counterarguments to crowdworkers during labeling to improve label quality. However, this approach still assumes that a single correct label exists for every item and requires crowdworkers to pass a training stage to learn label guidelines before participating. In our work, we tackle the problem of labeling with untrained crowdworkers under the assumption that a single correct label might not exist for every item. Diversity in interpretation is then used to create rich structures of uncertain items for post-hoc label decisions, avoiding the burden of creating comprehensive guidelines beforehand and pre-training crowdworkers. We compare our approach to a condition inspired by MicroTalk called *Revote*, showing that Revolt can produce training labels with higher accuracy under the scenario where comprehensive guidelines are not

available.

4.2.4 Structuring Unlabeled Data with the Crowd

Crowd structuring refers to tasks that make use of crowdsourcing for organizing information without predefined target concepts. For example, categorizing [9, 44] or create taxonomies [57] for a set of documents. In contrast, our work focuses on the task of *crowd labeling* which is the task of assigning predefined target concepts to each item in a dataset. In our approach, crowdworkers perform structuring within a labeling task, and only to resolve different interpretations of the same items for post-hoc requester review. Past work in crowd structuring also typically involves multiple stages completed by different crowdworkers working independently, while we took a different approach by utilizing real-time crowdsourcing to maintain a shared global structure synchronized across groups of crowdworkers working collaboratively.

4.2.5 Real-time Crowdsourcing

Real-time crowdsourcing has been employed for a variety of purposes including minimizing re-action time in crowd-powered user interfaces [20, 25, 35, 135, 136], increasing the speed of data collection [130, 137], and improving data quality by incorporating expert interventions to guide novice workers [40, 76, 121]. In our work, we use real-time crowd-based collaboration to improve the quality of labeled data without expert intervention. Workers engage in real-time collaboration to build rich structures of data that incorporate different crowdworker perspectives and can be analyzed post-hoc. Moreover, while most previous real-time systems employ multiple crowdworkers in a single shared workspace, our approach dynamically coordinates crowdworkers to move synchronously between stages of different subtasks. Within each stage, crowdworkers make independent judgments to be revealed to others in subsequent stages. In this way, our approach still benefits from common crowdsourcing mechanisms of verification through redundant independent judgments, but can also capture and utilize diverse crowd perspectives through collaboration.

4.3 System Design

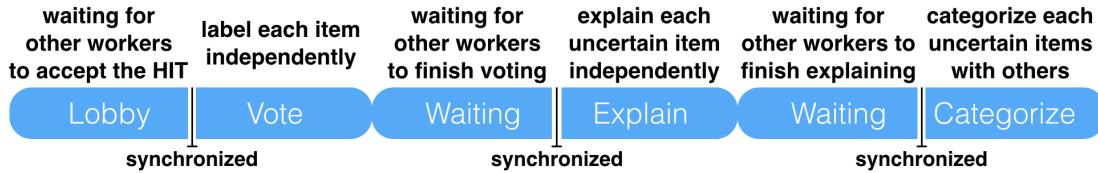


Figure 4.2: Overview of Revolt Stages: Synchronized stages requires all crowdworkers in the group to complete in order to move on.

In this section, we describe Revolt, a collaborative crowdsourcing system for generating labeled datasets for machine learning. Throughout this section, we use the task of labeling images as being about “*Cats*” or “*Not Cats*” as a running example (Figure 4.1).

At a high level, Revolt divides a dataset into multiple batches and then coordinates crowdworkers to create labels for *certain* items (items receiving unanimous labels from multiple crowdworkers) in each batch and identify *uncertain* items (items receiving conflicting labels) for further explana-

tion and processing. In the synchronized version (Revolt), the system coordinates small teams of three crowdworkers through three synchronized stages: Vote, Explain, and then Categorize (see Figure 4.2). In the asynchronous version (RevoltAsync), the system elicits different crowdworkers to work independently in the Vote and Explain stages, maintaining the same redundant judgement of three crowdworkers per item while eliminating the cost of coordinating crowdworkers in real-time. After collecting crowd judgments and explanations across all batches, both systems algorithmically produce structures (groups of semantically related items) at various levels of granularity for review by label requesters to determine final label decision boundaries (e.g., assigning the “*Cartoon Cats*” category as “*Not Cats*”) before training a machine learning model. To minimize redundant information, the rest of this section describes Revolt in the context of the synchronized version. We then describe the differences of the RevoltAsync condition.

4.3.1 The Vote Stage

We want to know if the main theme of the items below are “Cats”. Label “Cat” if you think the main theme of the item is Cats, otherwise label “Not Cat”. Label “Maybe/Not Sure” for items that you are uncertain about or if you think other workers might pick different labels.

	<input type="radio"/> Cat <input checked="" type="radio"/> Not Cat <input type="radio"/> Maybe/NotSure
	<input checked="" type="radio"/> Cat <input type="radio"/> Not Cat <input type="radio"/> Maybe/NotSure
	<input type="radio"/> Cat <input type="radio"/> Not Cat <input checked="" type="radio"/> Maybe/NotSure

Figure 4.3: Human Intelligence Task (HIT) interface for the Vote Stage. In addition to the predefined labels, crowdworkers can also select *Maybe/NotSure* when they were uncertain about the item.

Revolt initially keeps crowdworkers in a lobby until enough crowdworkers have accepted the task and can begin as a group (Figure 4.2). The Vote stage then begins by collecting independent label judgments from multiple crowdworkers using an interface similar to that used in traditional crowdsourced labeling (see Figure 4.3). In addition to showing predefined labels as options at this stage (e.g., “*Cat*” or “*Not Cat*”), we also include a “*Maybe/NotSure*” option to ensure crowdworkers are not forced to make arbitrary decisions for uncertain items that should instead be explained further in subsequent stages. Through task instructions, crowdworkers at this stage are informed that others in the same group are also labeling the same items at the same time, and that they will be asked to compare their labels in subsequent stages. By allowing workers to express their uncertainty in the data and provide feedback in subsequent stages, Revolt avoids

unfairly rejecting honest work [156].

Before Revolt can proceed to the next stage, all crowdworkers in a group must finish labeling all items in their batch. Crowdworkers who finish early are put into a waiting area where they can see in real-time how many crowdworkers in their group are still labeling items. Once the group is ready to continue, desktop and audio notifications are sent to all crowdworkers in case any stepped away while waiting. Once all labels are received, *certain* items are assigned their final labels as usual, and *uncertain* items (including items that received “*Maybe/NotSure*” labels) proceed to the Explain stage.

4.3.2 The Explain Stage

The other workers have also finished labeling the same items you just labeled. The following items received different labels. Please provide an explanation for each of your labels below.

 You labeled "Not Cat". Please focus on describing things about the item that could have made it difficult or ambiguous for others.

 You labeled "Maybe/NotSure". Please focus on describing things about the item that could have made it difficult or ambiguous for others.

Figure 4.4: Human Intelligence Task (HIT) interface for the Explain Stage. Crowdworkers enter a short description for each item that was labeled differently in the Vote Stage. They were informed that disagreement occurred, but not the distribution of different labels used.

In the Explain stage, crowdworkers are asked to provide short explanations about their labels for items flagged as *uncertain* in the previous stage. Instructions informed each crowdworker that others in the group disagreed on the labels for these items and therefore their task was to describe their rationale for each label to the rest of the group (see Figure 4.4).

Note that early prototypes of our system also revealed the individual votes from other crowdworkers on each item at this stage. However, pilot experiments showed that this resulted in less descriptive explanations that were more focused on reacting to other crowdworkers. For example, people who picked the majority vote labels often simply reaffirmed or expressed confidence in their original label (e.g., “*nothing says to me that this is a cat*”), whereas people who were in the minority often just yielded to the majority (e.g., “*this could be a cat, i might have messed this one up*”). Instead, hiding the labels and only stating that a disagreement had occurred resulted in more conceptual explanations helpful for the following stage (e.g., “*This is not a cat, but rather one of the big felines. Leopard or Cheetah I think.*” and “*Although leopards are not domesticated, they are still cats.*”). As in the Vote Stage, crowdworkers who finished early were placed in a waiting area before they could move on together.

4.3.3 The Categorize Stage

You labeled differently on the following items. Please review all the explanations provided by other workers and pick or come up with good category names so the requesters can make an informed decision afterwards.

	<input type="text"/> Create	worker1: This is a tiger.
	<input type="checkbox"/> big cats	worker2: This is a big cat.
	<input type="checkbox"/> cartoon cats	worker3: Do lions and other big cats
	<input type="checkbox"/> cats with dogs	
	<input type="text"/> Create	worker1: This is a cartoon drawing of a cat.
	<input type="checkbox"/> big cats	worker2: Cat drawing.
	<input type="checkbox"/> cartoon cats	worker3: Do cartoon cats count?
	<input type="checkbox"/> cats with dogs	

Figure 4.5: Human Intelligence Task (HIT) interface for the Categorize Stage. Crowdworkers select or create categories for items that were labeled differently in the Vote Stage, based on explanations from all three crowdworkers in the same group.

In the Categorize stage, crowdworkers were tasked with grouping uncertain items into categories based on their explanations. In this stage, we present the same uncertain items to each crowdworker again, but this time also reveal the explanations from others in the group (Figure 4.5). Crowdworkers were then instructed to categorize each item based on its explanations. Categories could either be selected from a list of existing categories presented next to each item or added manually via a text input field. Whenever a new category was added by a crowdworker, each list of categories was synchronized and dynamically updated across all items within the current group, and also across groups working on different parts of the dataset. To encourage category reuse and reduce redundancy we also implemented two mechanisms: First, the text field for creating categories also acts as a quick filter of the existing categories so that crowdworkers may more easily see and select an existing category rather than create a new one when appropriate. Second, the list of existing categories is sorted by the number of crowdworkers (across all batches of the same dataset) that have used each category, a similar strategy to that used to filter out low quality crowd categories in [57, 59]. After assigning categories, crowdworkers could submit their HITs independently without waiting for others to complete.

4.3.4 Post Processing of Crowdworker Responses

After crowdworkers in all groups have gone through all three stages, Revolt collects the crowd feedback for all batches. Revolt assigns labels to *certain* items directly, and then creates structures of *uncertain* items by applying simple majority voting on the category names suggested by crowdworkers for each item. In cases where all crowdworkers suggested a different category, a random crowdworker’s category is used as the final category. At this point, structures can be presented to label requesters for review and to make final label decisions. For example, after reviewing structures, a label requester may decide that *leopards* and *lions* should be considered *Cats* while *cartoon cats* and *cat food* should be considered *Not Cats*. In this way, label assignments can be applied to the data in each category prior to training a machine learning system.

Revolt can also expand the crowd-generated categories to different numbers of clusters, sup-

porting inspection at different levels of granularity. To do this, Revolt performs a hierarchical clustering method that uses the crowd categories as connectivity constraints. This post-processing approach works as follows: First, a term frequency-inverse document frequency (TF-IDF) vector is used to represent each uncertain item where each dimension is the count of a term in its explanations divided by the number of uncertain items with the same term mentioned in their explanations. Then, hierarchical clustering with cosine similarity is applied. That is, initially, each item is treated as a cluster by itself. Clusters are then iteratively merged with the most similar clusters, prioritizing clusters with items in the same crowd category, until all items are in the same cluster.

Generating clusters at various levels of granularity allows label requesters to adjust the amount of effort they are willing to spend in making labeling decisions, allowing them to manage the trade-off between effort and accuracy. For example, labeling low level clusters allows for more expressive label decision boundaries, but at the cost of reviewing more clusters.

4.3.5 RevoltAsync

RevoltAsync removes the real-time nature of Revolt as follows: One set of crowdworkers label items independently in the Vote stage. RevoltAsync then still uses the results of three crowdworkers per item to identify uncertain items. Uncertain items are then posted to the crowdsourcing market again for a different set of crowdworkers to explain the labels. That is, in RevoltAsync’s Explain stage, crowdworkers are presented with an item and a label given by another crowdworker and then asked to justify that label given the knowledge that there were discrepancies between how people voted on this item.

RevoltAsync does not include a Categorize stage, which would require synchronization. Instead it uses the explanations collected at the Explain stage directly for clustering during post-processing. Clustering of explanations is still performed using hierarchical clustering, to produce structures at different levels of granularity, but without connectivity constraints based on the crowd categories provided by the Categorize Stage.

4.4 Evaluation

In this section, we describe experiments we conducted to investigate the cost-benefit trade-off of Revolt compared to the traditional crowdsourcing approach for collecting labeled training data. We also examined several variants of Revolt to better understand the benefits of different components of the Revolt system and workflow.

To compare these workflows, we ran each condition on a variety of datasets and measured the accuracy of the resulting labels with respect to requester effort and crowdsourcing cost. To prevent learning effects, we do not reuse crowdworkers across conditions for the same dataset, and randomize posting order of condition and dataset combinations so that crowdworkers subscribed to postings from our requester account using third party services¹ were distributed across conditions.

¹<http://www.turkalert.com/>

4.4.1 Baselines and Conditions

Our conditions include Revolt, RevoltAsync, three variants, and two baselines representing traditional labeling approaches:

- *NoGuidelines*. A baseline condition where crowdworkers label items without guidelines. This condition should be considered a lower bound baseline, since in most real world scenarios requesters are likely to have some knowledge of the data or desired labels to create some initial guidelines.
- *WithGuidelines*. A baseline condition where crowdworkers label items according to provided guidelines. For this condition we endeavored to create comprehensive guidelines that left no room for subjective assessment as explained in the next Datasets and Guidelines section. Since creating comprehensive guidelines is often infeasible in realistic machine learning tasks, the results from this baseline should be considered an upper bound for what can be achieved with traditional crowdsourced labeling.
- *Revolt*. Our proposed Vote-Explain-Categorize workflow with synchronized real-time collaboration.
- *RevoltAsync*. Our Revolt variant with asynchronous collaboration mechanisms.
- *Revote*. A Revolt variant with similar strategies used in [77] wherein crowdworkers re-label uncertain items after considering each others' explanations instead of categorizing them for post-hoc requester review. This variant replaces Revolt's Categorize stage with a Revote stage (without the *maybe* option) and uses simple majority voting to assign final labels to all items.
- *Solo*. A Revolt variant with no collaboration. In this condition, each crowdworker labels and explains their labels for all items independently. The system still computes uncertain items from three redundant labels and clusters uncertain items using their explanations.
- *SoloClusterAll*. A variant of *Solo* where the system clusters all items based on their explanations. Note that clustering all items (certain and uncertain) is only possible in the *Solo* variants where explanations were collected on all items. This approach creates categories for certain items as well as uncertain, requiring requester review of even items that reached consensus through crowd labeling.

Note that no post-hoc requester effort is required in the *NoGuidelines*, *WithGuidelines* and *Revote* conditions and only the *WithGuidelines* baseline requires requesters to create guidelines prior to crowd labeling. We implemented the Revolt, Revote, Solo, and SoloClusterAll conditions using the TurkServer library [148], which provided the infrastructure for recruiting and coordinating crowdworkers in real-time. Labels for the RevoltAsync, *NoGuidelines*, and *WithGuidelines* conditions were collected through the Mechanical Turk form builder feature on the requester interface.

4.4.2 Datasets and Guidelines

We evaluated each of our conditions with eight tasks made up of different data types (images and webpages) and sizes (around 100 and 600 items, respectively). All of our datasets were obtained from the publicly available ImageNet [68] or Open Directory Project² databases, both commonly used for machine learning research.

²<https://www.dmoz.org/>

Each labeling task asked crowdworkers to label each item in a dataset as belonging or not belonging to a given concept. We used target concepts of *Cars* and *Cats* for our image datasets and *Travel* and *Gardening* for our webpage datasets to show that interpretations can vary for even seemingly simple and generally familiar concepts (Table 4.1). For our *Cars* and *Cats* image datasets, we collected images from ImageNet [68] by first collecting all images that corresponded to WordNet [162] concepts containing the keyword “car” or “cat”, and then sub-sampling the set down to approximately 600 images per dataset while ensuring no sub-concept (such as *sports car* or *cable car*) was overrepresented (>10%) in the final set. We obtained our *Travel* and *Gardening* webpage datasets from [133] which has approximately 100 webpages for each concept obtained from the Open Directory Project by selecting half from each category, “travel” or “gardening”, and then selecting the remainder randomly from the database.

For each dataset, we generated two sets of gold-standard labels and corresponding guidelines (making either datasets total) representing two different interpretations of the same concept in the following way: Independently, each author first manually labeled the datasets using a structured labeling process [133] where they would categorize items as they examined them and then assign final labels at the end. This resulted in gold-standard labels and guidelines describing those labels (defined by rules each author would write down describing their categorizations and final label assignments) for that dataset. These guidelines can be considered comprehensive given that each item was examined during labeling. In realistic tasks with potentially large or complex datasets, it is often infeasible for label requesters to manually examine each item in order to create a set of guidelines (instead they typically examine a subset). Table 4.1 summarizes our datasets. To give some insights into the level of ambiguity that existed in each datasets, we report the proportions of items that received conflicting labels under the NoGuidelines conditions as μ . The average proportion of items being assigned the positive labels in each dataset is 0.41 ($\sigma = 0.12$).

Dataset	Type	N	μ	NoGdlns.	W/Gdlns.	Revote	Revolt	Solo	SoloAll	RVASync	#Cats
Cars1	img	612	.27	.843	.887	.820	.904	.863	.884	.882	32
Cars2	img	612	.27	.756	.804	.775	.827	.794	.807	.820	32
Cats1	img	572	.12	.844	.939	.845	.916	.720	.900	.902	14
Cats2	img	572	.12	.920	.962	.904	.935	.787	.916	.918	14
Travel1	web	108	.24	.759	.870	.787	.880	.815	.806	.870	22
Travel2	web	108	.24	.769	.870	.759	.889	.796	.796	.870	22
Garden1	web	108	.12	.806	.843	.787	.889	.861	.759	.852	8
Garden2	web	108	.12	.778	.833	.787	.843	.815	.787	.787	8

Table 4.1: Accuracy of different labeling conditions. The number of clusters of the Solo, Solo-ClusterAll, and RevoltAsync conditions were fixed to the number of categories observed under the Revolt condition. Bold numbers indicate the best performing condition for each dataset.

4.4.3 Results

We present our experimental results in terms of accuracy and cost of labels produced by each of our conditions. Final labels for the NoGuidelines, WithGuidelines, and Revote condition are assigned using simple majority voting. The Revolt, RevoltAsync, Solo, and SoloClusterAll conditions produce labels for unanimously voted items and categories (or clusters) for uncertain items. To simulate post-hoc requester judgments and measure accuracy of these conditions, we assign each uncertain category the label corresponding to the majority label of its items as defined by

Category	Size	Car1GdStdLabel	Cars2GdStdLabel
train car	19	95%	not car
train	19	100%	not car
military vehicle	16	100%	not car
car	15	73%	car
vehicle mirror	14	100%	car
bumper car	12	100%	not car
tow truck	11	91%	car
wheel	8	100%	car
truck	8	100%	car
trolley	7	86%	car
vehicle interior	6	100%	car
			not car

Table 4.2: Revolt categories for the Car datasets and the corresponding gold-standard label determined with majority voting for each category.

the gold-standards. As an example, in Table 4.2 we show the top eleven categories generated by Revolt for uncertain items in the *Cars* datasets and the proportion of the corresponding majority labels in two sets of gold-standard labels (e.g., 95% of the items in the *train car* category were labeled as *not car* in the gold-standard for both Car1 and Car2). This simulation allows us to produce final labels for all items which we can then compare directly to the gold-standard to compute accuracy. It is important to note that the gold-standard labels are only being used to simulate the post-hoc requester input and none of our approaches use gold-standard labels in their workflows.

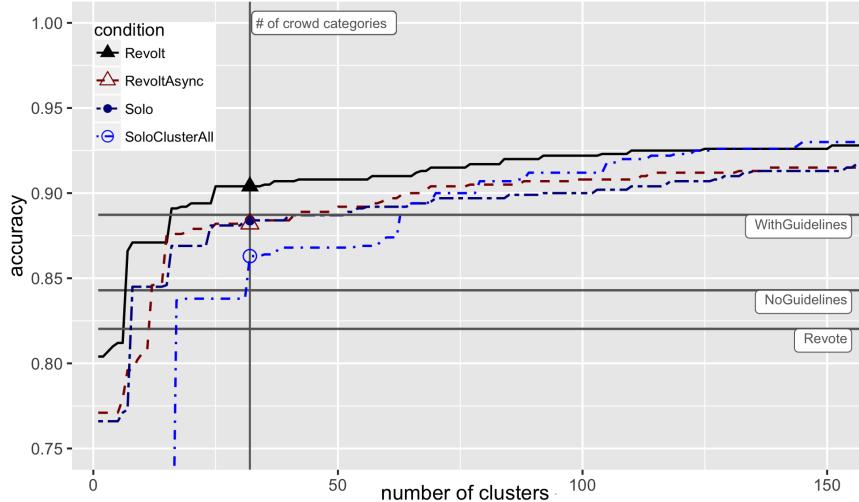


Figure 4.6: Accuracy of different approaches as a function of post-hoc requester effort (i.e., number of clusters) for the Car1 dataset.

In addition to presenting crowd-generated categories, Revolt (and its variant conditions) can also algorithmically produce clusters of uncertain items at various levels of granularity for requesters to review (see the Revolt Section). As a result, requesters can vary the amount of effort they are willing to provide to produce final label decision boundaries in these conditions. Therefore, for these conditions, we also report on accuracy achieved at various levels of post-hoc requester effort. As an example, Figure 4.6 shows how the accuracy of Revolt changes for different amounts

of requester effort required to assign labels (estimated by number of clusters needing labels) on the *Car1* dataset. For this example, receiving requester input for 32 categories produced by Revolt (see vertical line in Figure 4.6) achieved an accuracy higher than the upper bound WithGuidelines baseline, while other conditions did not.

We compare the accuracies of different conditions in two ways. In Table 4.1, we compare conditions at a fixed amount of post-hoc requester effort (i.e., the number of clusters needing examination by the requester). We fix effort to be the number of categories generated by the crowd under the Revolt condition for each dataset. For example, for the *Cars1* dataset, we compute accuracy at the point where 32 clusters would need to be examined. The accuracy results presented in the *Cars1* row in Table 4.1 therefore corresponds to a vertical cut of Figure 4.6 at the 32 number of clusters mark. To compare different conditions and baselines, we fit one generalized linear model per baseline, predicting correctness as a function of condition, with dataset as an additional factor to account for item variation. Both models significantly improve fit over a simpler model with dataset as the only variable ($X2(5)=160.1$, $p < 0.01$, and $X2(5)=180.9$, $p < 0.01$). Using the models, we ran general linear hypothesis tests for pairwise comparisons between conditions, and used Tukey's honestly significant difference as the test statistic. The models showed both Revolt and RevoltAsync to be significantly more accurate than the lower bound NoGuidelines condition ($B=0.56$ and 0.38 , both $p < 0.01$) while no significant differences were found when comparing to the upper bound WithGuidelines condition ($B=0.05$ and -0.13 , $p=0.99$ and 0.63).

In addition to using a fixed numbers of clusters, Figure 4.7 shows the of accuracy improvement rate of each condition under different levels of requester effort relative to the NoGuidelines baseline. Since the smaller datasets only had less than 30 uncertain items, for conditions that generate rich structures (Revolt, RevoltAsync, Solo, and SoloClusterAll) we show the accuracy improvement rate for 10, 15, 20, and 25 post-hoc judgments for the smaller webpage datasets, and 10, 20, 30 for the larger image datasets. We also report the accuracy improvement for the WithGuidelines and Revote conditions that do not require post-hoc judgments.

In our experiments, \$3 were paid to each worker for participating in a batch of Revolt, Revote, Solo, SoloClusterAll conditions, where \$1 was paid as base payment for completing the first stage, and \$2 bonuses were added for completing the rest of the stages. For the RevoltAsync condition, \$1 was paid for each Vote and Explain task. We adjusted the number of items in each batch so that crowdworkers could finish batches under 20 minutes including time spent waiting for other crowdworkers. Each batch in the image datasets contained around 60 items while each batch in the webpage datasets contained around 27 items. For the baseline conditions, we paid \$0.05 for labeling one image, and \$0.15 for labeling one webpage.

We also compared cost of each condition in terms of crowdworker work duration (Figure 4.8). For our Revolt, Revote, Solo and SoloClusterAll, we measure work duration directly by tracking crowdworker behaviors using our external HIT interface, tracking mouse movements to identify the exact time crowdworkers started working after accepting the HIT. Our NoGuidelines, WithGuidelines, and RevoltAsync conditions were implemented via the Mechanical Turk form builder feature. While Mechanical Turk does report work duration, crowdworkers often do not start work immediately after accepting a HIT. To correct for this, we approximate the work duration for these interfaces in the following way. We approximate the work time of the NoGuidelines and WithGuidelines conditions (our baseline conditions) using the timing statistics collected from the Vote Stage of the Revolt workflow, as the crowdwork involved in these baselines are of the same na-

ture as the Vote stage. We similarly approximate the total work duration for the RevoltAsync condition by using the timestamps from the Solo condition (where crowdworkers provided explanations for each item), and multiplying the average duration with the number of uncertain items identified for each dataset in this condition.

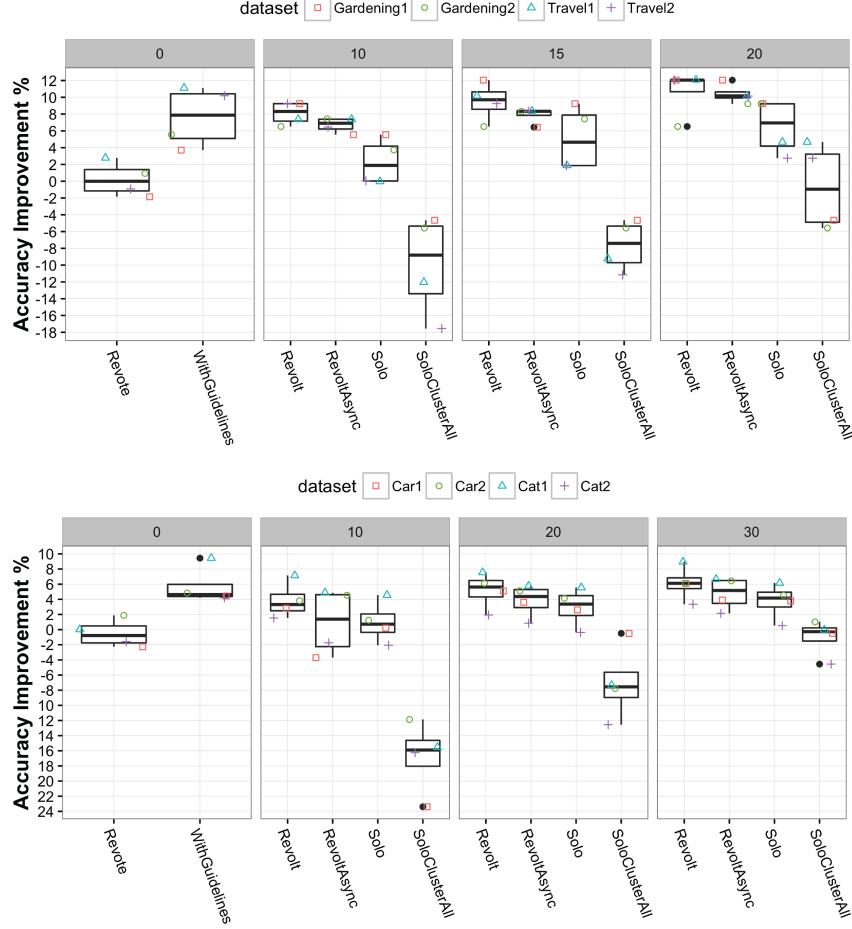


Figure 4.7: Accuracy improvement of different conditions over the NoGuidelines baseline as a function of requester effort.

Revolt vs Traditional Crowdsourced Labeling

In both traditional crowd-based labeling and Revolt, requesters examine uncertain items to refine the label boundaries. However, in Revolt, this is done at the category level in a post-processing step as opposed to reviewing items, refining instructions, and launching more tasks in a loop. The latter may lead to wasted work, particularly when refinements require workers to relabel the same items. In Revolt, structures captured from crowdworkers during labeling allow requestors to refine label boundaries post-hoc without having to launch more tasks.

When compared against the NoGuidelines condition (the lower bound of traditional crowdsourced labeling), Revolt was able to produce higher accuracy labels across all eight datasets (Table 4.1). The generalized linear models also showed both Revolt and RevoltAsync to be significantly more

accurate than the NoGuidelines condition. The comparison of the NoGuidelines and WithGuidelines conditions shows that comprehensive guidelines indeed increase labeling accuracy across all eight datasets (Figure 4.7), but at the cost of the effort needed to create comprehensive guidelines in advance. In contrast, Revolt was able to produce comparable accuracy without any guidelines. In fact, in 6 out of the 8 datasets we tested, Revolt was able to produce labeling accuracies slightly higher than the upper bound baseline (Table 4.1). The generalized linear models also showed that neither Revolt nor RevoltAsync were significantly different than the upper bound condition ($B=0.05$ and -0.13 , $p=0.99$ and 0.63). This suggests that Revolt can outperform current best practices for crowdsourcing training labels where guidelines provided by the requesters are likely to be less comprehensive than the ones provided in the WithGuidelines condition. That is, Revolt shows promise to improve the quality of labeled data collection while removing the burden of comprehensive guideline generation by making use of collaborative crowdsourcing approaches.

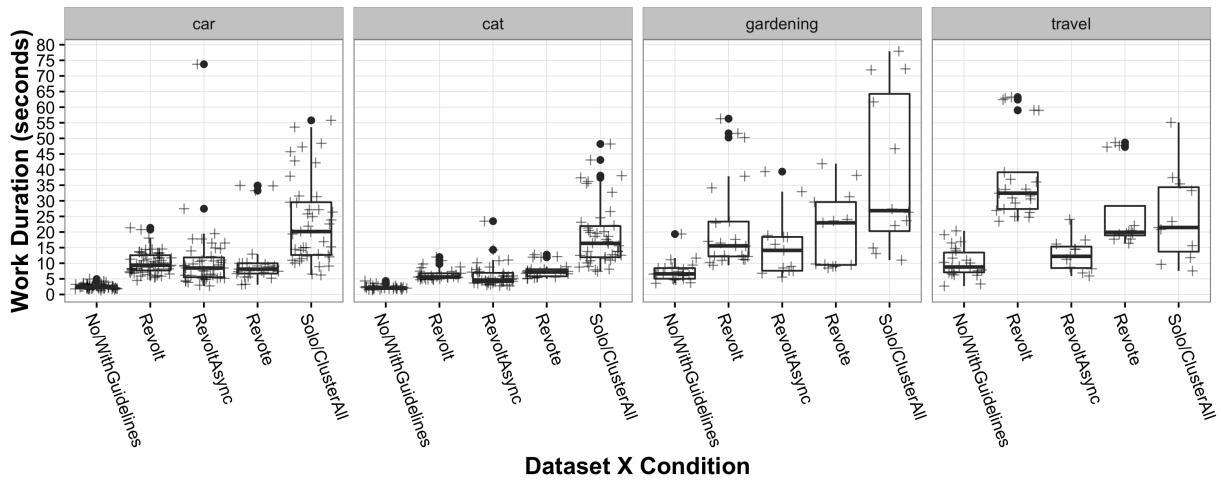


Figure 4.8: Work duration of each crowdworker under different conditions, normalized by the number of item in each batch.

Forcing Crowdworkers to Revote

An alternative way of explaining why we see uncertain items with conflicting labels in Revolt’s Vote Stage is that crowdworkers could converge on true concepts for all items but they are simply making mistakes while labeling. To test this, the Revote condition allowed crowdworkers to reconsider their labels after seeing explanations from others, providing the opportunity to correct mistakes. While previous work has shown accuracy improvement using this strategy under a scenario where clear guidelines were given to pre-trained crowdworkers [77], results from our study showed that the Revote condition did not improve labeling accuracy compared to the NoGuidelines lower bound baseline ($B=0.03$, $p>0.99$), with near zero median accuracy improvement (Figure 4.7). This suggest that in scenarios where it is infeasible to generate comprehensive guidelines to guide workers towards a single correct answer, accuracy cannot simply be improved by quality control on individual workers; instead allowing crowdworkers to inform the requesters about their confusions and discoveries may be a better strategy than forcing them to make arbitrary decisions and then performing post-hoc quality control.

Benefits of Collaborative Crowdsourcing

Traditionally, crowdsourcing techniques require independent crowdworker judgments and do not permit knowledge sharing. In this work, we investigated the benefits of collaborative crowdsourcing by allowing limited and structured communications between crowdworkers. Our collaborative conditions (Revolt, RevoltAsync, Revote) presented crowdworkers with different combinations of conflicting judgements, justifications for the conflicting judgements, and proposed structures (i.e., category names) from other crowdworkers, either synchronously or asynchronously. On the other hand, in NoGuidelines, WithGuidelines, Solo, and SoloClusterAll conditions, workers were not presented with any judgments from others.

Comparing Revolt to RevoltAsync, Revolt with synchronous stages performed slightly better than RevoltAsync at the cost of slightly higher worktime (Figure 4.8), but the difference was not significant ($B=0.18$, $p=0.28$). Comparing collaborative and non-collaborative conditions, results show that both Revolt and RevoltAsync outperformed the non-collaborative Solo condition for each dataset we tested (Figure 4.7). Based on the generalized linear models, the real-time collaborative Revolt condition achieved significantly higher accuracies than the non-collaborative Solo condition, while the RevoltAsync variant did not ($B=0.24$ and 0.06 , $p=0.04$ and 0.97).

Interestingly, we initially expected the RevoltAsync condition would yield poorer results compared to the non-collaborative Solo condition due to cognitive dissonance (i.e., asking one crowdworker to explain the label of another). However, the results showed no significant difference between the two conditions. On the other hand, the non-collaborative SoloClusterAll condition, where explanations were collected for all items to cluster both certain and uncertain items, performed worse than the lower bound baseline. These results suggest that identifying and presenting disagreements is an important factor for eliciting meaningful explanations, even when the disagreements were presented to different crowdworkers in an asynchronous setting (Figure 4.7).

Cost Analysis

In general, items in the webpage datasets took longer to label compared to items in the image datasets. This is to be expected since webpages typically contain both text and images and therefore often require more effort to comprehend (Figure 4.8). Comparing different conditions, traditional labeling (WithGuidelines, NoGuidelines) that only required crowdworkers to label each item had the lowest work times, and the real-time collaborative conditions, Revote and Revolt, had similar and higher work times. This suggests categorizing or re-labeling items has similar costs, but creating rich structures for post-hoc requester judgments can lead to better accuracies. The RevoltAsync condition showed lower work time compared to the Revolt condition. This is also to be expected since crowdworkers did not need to wait for others during the task for progress synchronization. The non-collaborative workflow conditions Solo and SoloClusterAll has the highest work time since it required explanation of all certain and uncertain items. Therefore, using the voting stage to identify uncertain items and guide efforts on structure generation can improve accuracy while also lowering cost.

One concern for synchronous collaborative crowdsourcing is crowdworkers idling or returning the HIT before the task is competed. This was especially important since we did not include a method for using labels from groups with missing labels. In sessions with drop-outs, we paid the crowdworkers and discarded their labels. In fact, the first prototype of Revolt had a high dropout rate of around 50% (i.e., half of the crowdworkers did not complete the three stages),

making it infeasible for practical use. Through an iterative task design process, we observed the following mechanisms being effective for reducing drop-outs: Explaining the collaborative nature of the task, providing time estimates in the task instructions, adding example items in the preview screen so crowdworkers knew what to expect if they accepted the HIT, sending desktop and audio notifications to help coordinate workers, and giving clear progress indicators throughout the tasks (e.g., current bonus amount, number of remaining stages, and the amount of bonus for completing each stage). In the final version of the task with these mechanisms, the dropout rate was lowered to an average of around 5% for the eight datasets presented in this paper.

4.5 Discussion

In this work we focused on designing Revolt’s collaborative crowdsourcing workflow and evaluating whether the generated structures contain information with enough richness for label requesters to define accurate label decision boundaries. While we believe the proposed mechanisms of identifying uncertainty with disagreements and creating structures with explanations can generalize to multi-class scenarios, the added complexity to both the crowdworkers and the interfaces should be studied further. Research is also needed to design a requester-facing interface depicting these structures and to compare requester effort in post-hoc analysis with guideline creation.

To gain insights into these future directions, we conducted a small follow up experiment where we ran Revolt on data needed by a group of practitioners from a large technology company for a real machine learning research problem. This group required 300 items from the publicly available 20 Newsgroup Dataset [134] to be labeled as being about “Autos” or “Not Autos”. Prior to our study, the group of three practitioners already spent approximately 2-3 hours each to browse through some of the data and then about 1-2 hours to generate and iterate over the guidelines. This is a typical process for guidelines creation analogous to previous work [220], and should represent a realistic scenario somewhere between our lower bound NoGuidelines and upper bound WithGuidelines conditions. Because the practitioners already had some idea of how they wanted the dataset labeled, we ran Revolt with their guidelines.

We presented Revolt’s results to one member of the research group and asked them to examine the resulting structures. Interestingly, 93 out of the 300 items (31%) were inconsistent even though we gave crowdworkers guidelines about how to label, underscoring the difficulty of creating comprehensive guidelines covering the subtleties in a dataset. These items surfaced 23 unique categories and, to the practitioner’s surprise, 70% were not covered in the original guidelines (e.g., auto accessories, insurance, intoxication). 7 of the categories were mentioned in the guidelines with explicit instructions about how to label (e.g., driving, buying/selling, auto repair), indicating either failure of some workers to learn the guidelines or failure of the guidelines to capture the complexity of these categories. For example, one of the items categorized as driving was about which side of the road people should drive on. While this could be considered about auto driving, it could also be about driving other types of vehicles. Reading crowdworker explanations helped the practitioner to better understand this ambiguity, and led them to second guess their original guideline about labeling driving related items as autos. The practitioner we interviewed also suggested additional ways they wanted to use Revolt’s results, such as removing ambiguous items or certain categories before training a model, or creating features around

categories that surfaced. Further research is necessary to examine the potential for Revolt’s rich structures to support these tasks.

The practitioner also made several suggestions with respect to how one might design the presentation of Revolt’s structures. First, an indicator of category quality or confidence based on the distribution of labels assigned by individual workers would have helped the practitioner prioritize which categories to look at first and how much to examine each category before making a decision (e.g., by reading individual explanations or viewing a few of the individual items within a category). Other suggestions included blacklisting certain items from the category list (e.g., “autos” surfaced as a category), presenting structures within hierarchical categories, and searching explanations for to find related items under different categories. Future research should consider these insights in determining how to efficiently present Revolt’s structures to label requesters.

In this paper, we presented Revolt, a new approach for generating labeled datasets for machine learning via collaborative crowdsourcing. Our experimental results comparing Revolt to traditional crowd labeling techniques demonstrates Revolt can shift the efforts of label requesters from a priori label guideline creation to post-hoc analysis of crowd-generated conceptual structures. This has several benefits including potentially surfacing new or ambiguous concepts unanticipated by label requesters, reducing the amount of crowdworker training and effort required to learn label guidelines, and allowing label requesters to change their minds about label decision boundaries without having to re-collect new data.

Chapter 5: SearchLens

Capturing and Composing Complex User Interests

This work was previously published in ACM IUI 2019 [47] and has been adapted for this document.

While previous chapters focused on providing global context in the domain of crowdsourced sensemaking, starting with this chapter I shift focus to the domain of building interactive systems that can better support global context for individual's conducting online sensemaking tasks, such as trip planning or product comparison research. This is motivated by the application evaluation of Alloy described in Section 3.4, where we found individuals valued articles synthesized from clusters generated by Alloy, suggesting that global context (i.e., information gathered and synthesized across information sources) is also highly valued by individuals conducting online research.

This chapter explore a novel approach to better support the personalization aspects of data exploration. Using a restaurant review corpus, I focused on supporting users to learn from data and iteratively refine and evolve their nuanced interests. Consumer generated reviews are one of the most important influence in online decision making. To make sense of these rich repositories of diverse opinions, searchers need to sift through a large number of reviews to characterize each item based on aspects that they care about. We introduce a novel system, SearchLens, where searchers build up a collection of "Lenses" that reflect their different latent interests, and compose the Lenses to find relevant items across different contexts. Based on the Lenses, SearchLens generates personalized interfaces with visual explanations that promotes transparency and enables deeper exploration. While prior work found searchers may not wish to put in effort specifying their goals without immediate and sufficient benefits, results from a controlled lab study suggest that our approach incentivized participants to express their interests more richly than in a baseline condition, and a field study showed that participants found benefits in SearchLens while conducting their own tasks.

5.1 Introduction

People often rely on reading online reviews and forum posts to make predictions about how well different options might match their personal interests and needs. With the proliferation of online reviews, people now have instant access to millions of online reviews from people with varying perspectives and interests. It was estimated that in 2013 Amazon provided shoppers access to more than one million reviews for just their electronics section [154], and in 2016 Yelp provided around 250,000 reviews for over 6,000 restaurants for the city of Toronto alone [106]. Having access to this rich repository of diverse perspectives based on the past experiences of others has the potential to empower consumers to understand their choices thoroughly and make better decisions for themselves without being overly influenced by marketing and branding [64].

Unfortunately, it is often difficult for users to be able to quickly and efficiently match their personal interests to the large amount of information available for each potential option. One problem is

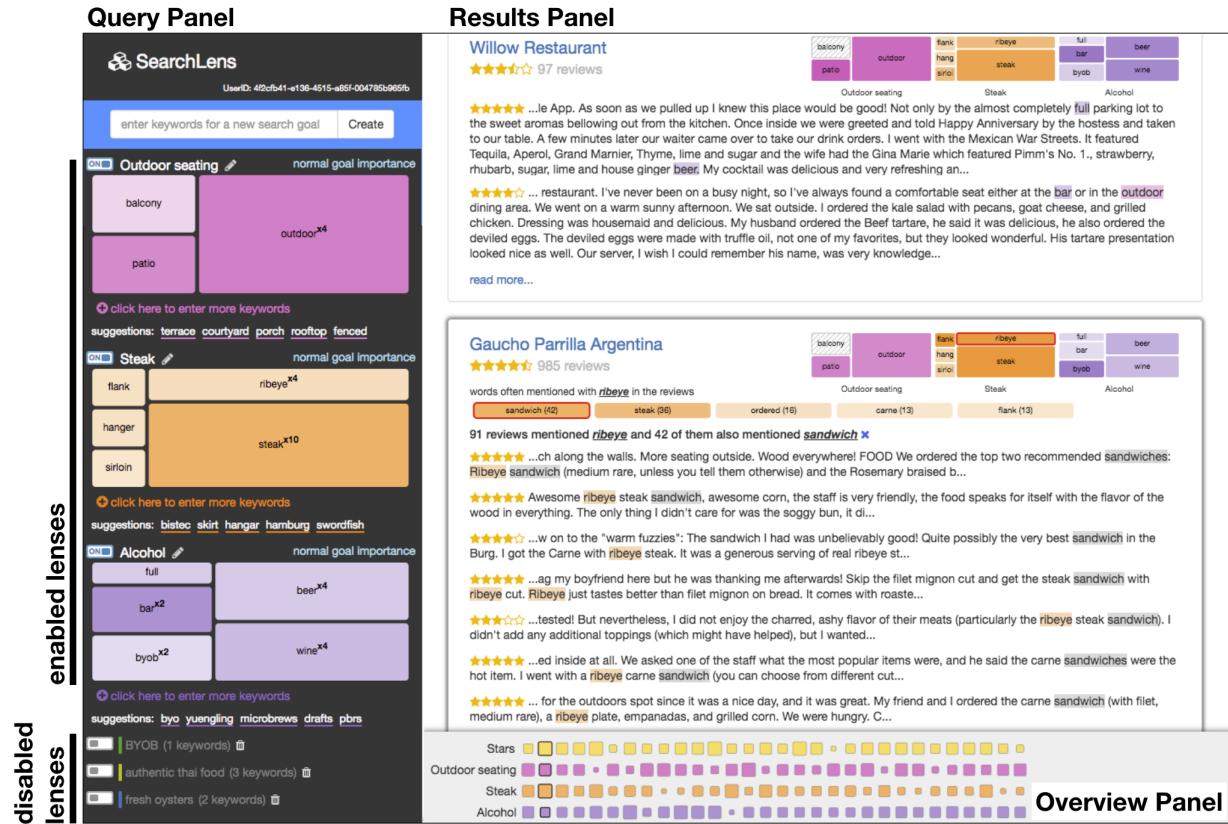


Figure 5.1: An overview of the SearchLens system. The Query Panel on the left allows users to specify search topics, or Lenses, by specifying multiple keywords. The keywords for a given Lens are shown in colored cells sized by importance (weight). Lenses can be freely disabled or enabled for different scenarios. The Results Panel on the right shows a ranked list of search results that best match the enabled Lenses from the searcher. The same visualization for specifying queries are then used for explaining how each result matches with user's interests and mental model, and also serve as an interactive navigation for filtering mentions of specific keywords. The Overview Panel at the bottom shows a collapsed version of the cells that allows for quick comparison between results.

that simple star ratings are often not sufficient, and recent research has shown reviews often play an important role in users online purchase decisions [84, 165]. For example, restaurants might receive negative reviews for its simple decor and lack of good ambiance, but some searchers might value more the authenticity of the food or whether vegan options were available on the menu. Subsequently, finding, reading, and evaluating relevant reviews is time-consuming and challenging. Users have to manually parse through the reviews for each restaurant and match them to their personal interests (e.g., kid friendly, authentic Indian cuisine). They then have to track which restaurant meets which criteria, and if they discover and add any additional criteria, they must back-fill that information and re-evaluate previously seen restaurants. Furthermore, once a user has finished searching, the work performed discovering and evaluating factors is lost, resulting in having to start from scratch even if a similar need arises in the future. For example, a traveler who has spent a lot of time choosing between ramen restaurants in Los Angeles must start from scratch evaluating ramen restaurants in Toronto, despite having discovered several important factors (e.g., thickness and chewiness of noodle, whether the broth is simmered for a long time with pork bones) that will be similarly utilized in their decision making.

Getting users to specify these nuanced interests and preferences has been a long standing challenge. Several decades of research have explored ways of getting users to externalize their interests [14, 111], for example by: using prompt and text field designs that promote longer query terms [15, 83], asking for relevance feedback on the results provided [170, 185, 188], or explicitly asking users to build up sets of query terms of different topics [95, 97]. There are two primary challenges brought up by this work. First, users have trouble specifying their interests, which includes challenges with identifying query terms that were neither too general nor too specific; providing more than a few terms (even when longer queries were more likely to lead to useful results); and learning terms from the content, rather than knowing them all beforehand [15, 188]. The other main issue found is that it is very difficult to get users to put in the work to externalize their interests, either as query terms or as explicit feedback, due to perceptions that the work will not be sufficiently paid off in the future or not understanding how their work will affect their results.

To tackle this issue of capturing, leveraging and exposing user interests, we introduce Search-Lense, where users construct externalized representations of their interests as “Lenses”. Lenses are leveraged as an explanatory tool, providing users with a way to quickly parse, understand and make judgments based on the vast amount of review data instantaneously. Additionally, Lenses can be reused in different contexts and combined in different configurations. In the example above, imagine a system which could capture the factors that the traveler found important for ramen in Los Angeles and reuse them to quickly make a confident, personalized decision about ramen in Toronto. If traveling to Toronto with kids, a “kids” Lens might also be added with factors such as whether the restaurant typically has long lines and how many seats it has. These persistent Lenses could be useful in a variety of situations beyond reviews, ranging from academics keeping track of interesting research topics; travelers deciding which places to visit in an unfamiliar city; consumers deciding between products; lawyers doing case discovery; or voters tracking important issues. We explore this problem in the context of restaurant reviews, conducting a controlled lab study with 29 participants to examine if our visual interface for explanation and exploration is effective in providing immediate benefits to elicit rich interest expressions from the users. Additionally, we performed a three day field deployment study with 5 participants to explore the benefits of Lenses when users were conducting their own tasks. Results suggest

that our prototype system SearchLens was able to learn richer representations of its users' interests when compared to a baseline system by allowing users to fluidly capture, build, and refine Lenses to reflect their interests and needs, and that the user-generated interfaces can be reused over time and transfer across contexts.

5.2 Related Work

Past research has proposed a variety of approaches to collecting, modeling and leveraging users' interests and intents through both interface design and computation. Our work builds on this diverse of literature by allowing the system to learn the personal interests of the users through interaction to retrieve relevant data, and present data based on its understanding of the different users. This allows us to elicit structures that can be reused across different contexts and tasks and are more nuanced and personalized to each users when compared to traditional search structures such as search results clustering or pre-compiled facets.

5.2.1 Eliciting and Modeling Interests and Intents

A significant topic of research has been interfaces that can collect, explicitly or implicitly, the personal goals and interests of users as they search for information and modify their viewing of content correspondingly. While there is extensive literature on doing so in the context of personalized search and re-ranking of search results (e.g., [34, 37, 193, 200]), we focus here on work that enables more interactivity and transparency of users' interests to support more complex searching. One such thread lies in the collection of users' interests through keywords or interest vectors into an agent or user interest or intent model. This includes seminal work such as WebMate [53], which built up an agent composed of sets of TF-IDF [222] vectors to represent the user's different interests. Similar to WebMate, we aim to build collections of terms that represent the user's interests, but focus on explicit user selection of those sets, and making them explainable and composable. Interestingly, WebMate's "Trigger Pair Model" which looked at co-occurrence of words within a sliding window across a set of documents can be seen as a precursor to the word vector model that we use for keyword suggestions. More recent work in this vein includes user modeling of concepts, such as AdaptiveVIBE [2] and Intent Radar [170], which include two dimensional visualizations of documents and their relation to the user's inferred interests. Our work builds upon these but aims at increasing the richness of the structure, nuance, and specificity of the user's expression of interests. Specifically, our Lenses, composed of multiple keywords that can capture multiple levels of specificity, can be themselves composed into more complex expressions and reused across different contexts and tasks. We also focus on supporting users in the discovery process of building good terms that are discriminatory and explanatory.

5.2.2 Concept Discovery and Evolution

Research in interactive machine learning has also explored techniques to support data annotators or searchers in discovering and externalizing useful concepts when working in unfamiliar domains. For example, Alloy used a *sample-and-search* technique to categorize textual datasets with novice crowdworkers where they first explore the space of information through sampling items in the dataset to discover useful categories, then externalize each category using a set of query terms and search for other relevant items [44]. Past work has further suggested that the

working concepts of an annotator may change over time as new items were examined [133]. Different techniques that can better support this concept evolution process were proposed, such as structured labeling [133], crowd collaboration [45], and interactive visualization [54]. These point to the importance of providing mechanisms that allow users to not only discover and define concepts based on data, but also to easily evolve their concept representations during the process of exploring an unfamiliar domain. In a study more closely related to our work, CueFlik allowed image searchers to define conceptual filters (e.g., listing only *action shots* when searching for *baseball* images) by labeling items in a search result list as positive or negative training examples [80]. Previously defined filters are persisted and can be applied to future searches (e.g., applying the same *action shots* filter when searching for *football* images), but evolving existing conceptual filters would require recreating filters from scratch or re-labeling items in existing filters. Our work builds this past work to allow exploratory searchers in unfamiliar domains to discover concepts of interests from data and externalize these concepts in the form of “Lenses” that can be continually refined. Finally, the Lenses are persisted across different search sessions similar to [80], and can be modified and composed for different scenarios and goals.

5.3 System Design

The key motivating concept behind SearchLens was providing users with a way to externalize their complex interest profiles in a way that could be useful for ranking, explanation, and transference to different contexts. We aimed to make the interface simple and transparent but also powerful enough to express higher level, abstract concepts and differing levels of specificity. To do this, we introduce the idea of “Lenses”: reusable collections of weighted keywords that contain “honest signals” of a user’s interests that can be composed in different configurations to match a user’s current needs. The Lenses that are enabled in a particular configuration drive various visualization and explanation elements to help the user understand how the information space meets their needs, and also whether they need to fix or reformulate their Lenses.

A key challenge here is incentivizing users to create rich Lenses by providing sufficient and immediate benefits. For this, SearchLens provides visual explanation of items in the search results based on users’ Lenses, which also serves as an interface for deeper exploration. When a new Lens is created or enabled, its visual representation appears on the interface for each item, allowing users to understand how well each item matches with the Lens, and how frequently each keyword is mentioned in its reviews. To further explore each item, users can click on keywords in each Lens to see relevant reviews.

A typical use case is as follows. A user just moved to Pittsburgh and wants to go out to eat ramen. She starts by pulling up a restaurant she knows she likes from Toronto and goes through some of the reviews, noticing that the reviews of her favorite tonkatsu ramen mention interesting signals such as “bone” and “umami” and adds them to her ramen Lens along with other useful words such as “tonkatsu”, “ramen”, “bowl”, etc. After checking to see that her Lens is bringing up other restaurants that serve ramen she likes in Toronto and adding a few of their terms to her Lens, she switches to Pittsburgh and looks for how her Lens is being used. She also activates her drinks Lens, which she’s built up over time to incorporate her particular interests in unfiltered sakes as well as hoppy beers. Using the Lenses, she quickly see which ramen restaurants in the results list serve unfiltered sakes and/or hoppy beers. To further explore her different options, she can click on each keyword in her Lenses to filter relevant reviews. For example, “tonkatsu”

might be often mentioned with “spicy” in one restaurant, and “creamy” in another, allowing her to further differentiate her options based on aspects that she cares about.

The following subsections describe the designs of the SearchLens system. We will first present our concept of “Lenses,” and how users can use SearchLens to fluidly express and refine their different nuanced interests, and freely compose their Lenses for different contexts. We will also describe how search Lenses can provide immediate benefits once specified, providing users visual explanation of each item in the search results, and also an interface for deeper exploration.

To test our prototype system in a realistic and manageable setting, we focused on the domain of restaurant reviews where personalization and searching with multiple goals is especially important. We used a subset of the dataset from the Yelp challenge [106] that included local business in 11 metropolitan areas.¹ Restaurants and reviews were selected by string matching on the *city* field of each restaurant available as metadata in the Yelp challenge dataset, resulting a subset of 48,485 restaurants and 2,577,298 reviews. This allows us to explore how user-specified Lenses can be composed and reused for different scenarios, as well as for the same scenario across different cities. In addition, we also use the same data to train a Word2Vec model [160] for generating Lens-specific query term suggestions.

5.3.1 Capturing User Interests with Lenses

Our goal was to develop a way to elicit users’ interests which is both highly expressive and immediately beneficial. To explore the natural discovery and collection of users’ interests we conducted a preliminary study in which we asked people to read reviews of their favorite restaurants on Yelp and see if they could identify terms that were good indications of their interests. We discovered that people found it intuitive to identify many different terms that matched their interests. Many of these terms were not simply general descriptors (e.g., “good”, “tasty”) but instead terms they considered indicative of matching their personal interests (e.g., an authentic ramen restaurant would include terms talking about the thickness of the noodles; a popular restaurant might be less favored if it also had very long lines). Terms also fell into different classes of factors users were interested in (e.g., service vs. food quality vs. parking). Users seemed to focus on finding reviews that mentioned these terms and use them in their decision making.

Based on these initial findings we developed a system for users to easily collect terms from reviews into “Lenses” and to use those terms to identify and summarize reviews that mentioned those terms. Similar to [95], we enable users to search with multiple Lenses at the same time. However, our Lenses differ from traditional search queries or faceted metadata in several important ways.

First, our system encourages the iterative development of Lenses as the user explores. A common activity in online exploratory search involves discovering new and interesting aspects from data. SearchLens aims to make it easy for users to add new Lenses and improve existing ones throughout their searching process. Users can create a new Lens by specifying a set of keywords using the text field in the Query Panel on the left (Figure 5.1). As users browse the results on the right, they might find some keywords in their Lenses were too general to be useful (e.g., “tasty broth”), and find discover more indicative keywords either from prior knowledge or from the reviews (e.g., “rich and thick broth”). In this case, users can refine their Lenses by adding

¹Pittsburgh, Charlotte, Phoenix, Las Vegas, Toronto, Montréal, Mesa, Mississauga, Cleveland, Scottsdale, and Edinburgh.

new keywords using three different interactions, each for a different scenario. First, users can click on the plus icon under each Lenses to enter new keywords in a Lens specific text field. Second, as users discover more indicative keywords or new topics of interests from the reviews, they can highlight the keywords and use a context menu to add them to an existing Lens. In addition, a list of keyword suggestions are also listed under each Lens based on current keywords (Figure 5.2). Users can hover over each suggestion to see example mentions, and click on the keyword include it. This allows users to assess the usefulness of the suggestions, such as to avoid ambiguous terms. The Lens-specific suggestions were computed based a word semantic model described in the below subsection. To remove a keyword, users can click on its cell and select remove keyword in the context menu.

Once constructed, Lenses can then be used to visually inspect and adjust their “projections” onto the data. Lenses are represented visually as boxes subdivided into cells, one for each term the user added. Initially, all keywords in the same Lens have equal importance (as reflected by being the same size), but users can click on each cell to select different importance in a context menu (x1, x2, x4, x10, exclude) to better reflect their personal preferences. The size of the cells will adjust accordingly to reflect the importance of each keyword (excluded keywords are represented using fixed size cells with a unique pattern fill). The shade of each cell shows the overall frequency of each keyword in the top 30 search results (Figure 5.1, Query Panel). This allows the user to get a sense of how items in the corpus reflect their mental representation of each topic. For example, a large cell with very light shade represents a concept that the users deemed as an important feature of the topic, but was rarely found in the results. Surfacing this information ensures user are aware of how useful each of their keywords are, and can refine their Lenses to include more indicative keywords.

As Lenses and terms are collected a user can over time build up a repository that reflects her personal interests. Each Lens can be disabled and re-enabled and are persisted across different visits to the SearchLens interface, with disabled Lenses are listed at the bottom of the Query Panel (Figure 5.1). Various combinations of Lenses can be activated depending on the goal and context. For example, for a date night a user might enable their personalized Lenses for “cozy and intimate”, and “vegan”, or for a weekday lunch activate their Lenses for “fast casual”, “vegan”, and “easy parking”. Although our main thrust in this chapter is exploring the viability of this approach, further work will likely be needed to understand as Lenses accumulate how to scale them. For example, in the current prototype all disabled Lenses are shown, but future systems could further contextualize Lenses by inferring the task context (e.g., what type of item someone is searching for).

Keyword Suggestions

While creating a new Lens, listing all keywords from prior knowledge can be mentally taxing and have poor recall. To further reduce the required effort for building expressive Lenses, SearchLens generates Lens-specific keyword suggestions. As an example, when a user created an “Outdoor Seating” Lens with only three keywords (“outdoor”, “patio”, and “garden”), SearchLens automatically suggested relevant keywords including “balcony”, “courtyard”, and “terrace” (Figure 5.2). To do so, we trained a Word2Vec model [160] with 300 dimensions using the entire Yelp dataset of 2,577,298 reviews. The trained word model can project words onto a semantically meaningful vector space, which in turn allows for measuring semantic similarity between words. Alternatively, it can also be used to find a set of words that are semantically similar to a



Figure 5.2: SearchLens provides keywords suggestions based on currently Lenses. Hovering shows a preview panel with mentions of the suggested keyword, allowing users to better understand the effect of adding the suggested keyword. In this case, SearchLens suggested balcony, terrace, fenced, and other keywords for the “Outdoor Seating” Lens. However, further inspection showed that fenced may not be a indicative keyword for the purpose of this Lens.

given term by searching in the vector space of nearby words. To generate Lens-specific keyword suggestions, we first project all its keywords in a Lens onto the vector space and calculate the average vector to obtain a list of similar terms around the average vector. To further increase the chance of presenting useful and discriminatory search terms, we only used terms that appeared more than 50 times in the corpus, were mentioned in reviews of more than three restaurants, and were mentioned in less than 40% of all restaurants.

5.3.2 Interest-driven Explanation

Persistent, decoupled user interest models would be beneficial to users the long run by providing separate reusable and recomposable interests across multiple search sessions. However, without immediate and perceivable benefits, users typically are not willing to spent extra effort expressing their separate interests for future tasks. For this, SearchLens uses each user’s Lenses to provide visual explanation of each item in the search results. This is based on our approach of allowing users to express their multiple topics of interest separately, which enables SearchLens to distinguish between keywords of different topics and opens the possibility of visualizing each result according to users’ interests in easy-to-interpret ways. Explanation is especially important for supporting searching with multiple interests, as it can be difficult for the users to understand which interests and keywords were associated with each result. Consider traditional search interfaces that only offer a short snippet for each result as explanation. These short summaries provide little support for personalized interpretation beyond a few highlighted query terms and their context. Even if users listed keywords of many different topics at once, the linear result list also provides little information about each result beyond their overall relevance ranking.

One obvious approach to explaining items in the search results is to surface mentions and statistical information, such as mention frequencies, at the topic level. For example, [97] visualized the overall frequency of different search terms in different topics for each search result, and [95] visualized the mention locations of different topics within each document. Visualizing at the topic level allowed these systems to provide mechanisms for specifying many topics and keywords, while at the same time visualized deeper information about each result in a way that matches the mental model of the searchers. However, visualizing at the topic level can be prohibitive for keyword-level operations, such as query reformulation and assigning importance levels to

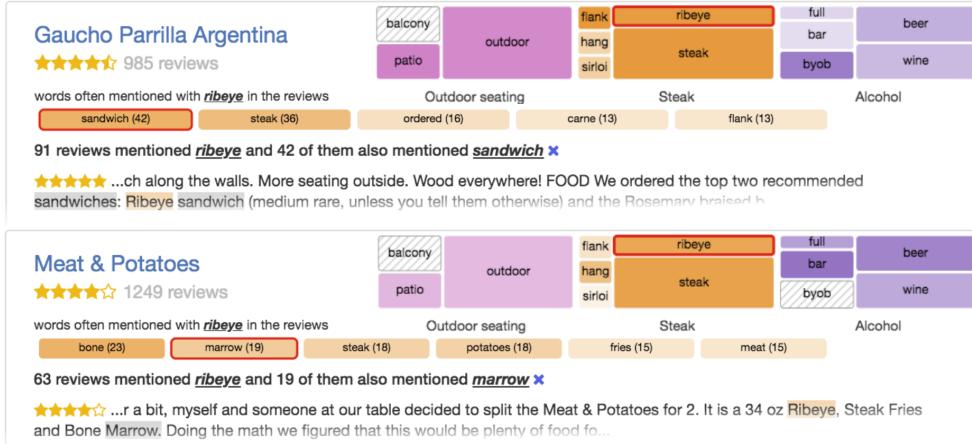


Figure 5.3: The visual explanation and exploration feature allows comparison of results at different levels of granularity using a familiar interface used for specifying queries - at the levels of Lenses, keywords, co-occurring terms, and mentions, allowing users to query with multiple Lenses at the same time, while still being able to comprehend how each result matches their different Lenses.

different keywords based on their frequencies.

SearchLens supports rich explanation at the topic and keyword level through its user-specified Lenses. Explanation occurs by showing the each Lens visualization from the Query Panel (Figure 5.1) on each result and adjusting the term shading to correspond to the frequency of the term within that search result (Figure 5.3). By using identical colors and layouts of each Lenses, and showing result-specific keyword frequencies, users can quickly interpret how each result matches with their different interests at both the topic and at the keyword level using a familiar visualization. As an example, Figure 5.3 shows how a user might examine two restaurants in a search result list using her Lenses for “Steak”, “Alcahol”, and “Outdoor Seating”. At the topic level, both restaurants matched well with her Steak Lens rendered in dark shades that incorporated her stronger preference for “ribeye” steak, and also also her other interests such as “flank” steaks. She can also see that the first restaurant matched her Outdoor Seating Lens better than the second one. Looking at the same Alcahol Lens at the keyword level, she can easily see that the two restaurants matched differently with her “Alcahol” Lens where the first one has many mentions of “byob” in the reviews and the second one with many mentions of “beer” and “bar” instead.

Finally, to provide a more compact, higher-level, topic-centric overview of all restaurants in the search results, SearchLens collapses the colored cells for each Lens into a single cell similar to [97]. The size of each cell to shows the overall frequencies of keywords in different Lenses for each result (Figure 5.1). This allows users to get a quick overview of restaurants in the search results, and compare different options at the topic level using the Overview Panel at the bottom.

5.3.3 Supporting Deeper Exploration of Items

In addition to acting as a visual explanation for each result, the cells in the visualization also act as a navigation tool for deep exploration at the keyword level. Users can explore mentions

of different keywords by clicking on its corresponding cell and the summary will update in real-time to show a list of its mentions. In addition, the Lens also shows the top co-occurring words that were frequently mentioned near the selected keyword as overview and deeper navigation, a strategy found useful in exploratory scenarios by prior work [69, 70, 169]. As an example, Figure 5.3 shows the how the Lenses allow users to explore and compare options at different levels of granularity. At the highest level, users can use the shading of different cells to see that the *Outdoor Seating* Lens has more mentions in the first restaurant (Figure 5.3). Searchers can use the shading of individual cells to compare options at the keyword level. For example, the term “BYOB” was frequently mentioned in reviews for the first restaurant, but did not show up in reviews for the second restaurant. Finally, clicking on the individual cells allows users to explore mentions of its corresponding keywords and words that were frequently mentioned together. For example, when exploring mention of the word “ribeye” for both restaurants, SearchLens shows that there were many mentions of “sandwich” near the word “ribeye” for the first restaurant, and many mentions of “bone marrow” near “ribeye” for the second restaurant (Figure 5.3).

5.3.4 Indexing and Ranking

Traditionally, faceted search systems typically combine factors from multiple facets for ranking using disjunctions (factors within facets, such as brands selected by the user on a shopping website) and conjunctions (factors between facets, such as brands and price ranges). In an early iteration of SearchLens, we tested using the Boolean OR operator between keywords within the same Lens, treating keywords within the same Lens as synonyms while ranking. However, users reported this approach lead them to restaurants that poorly reflected their Lenses, as some restaurants may have many mentions of few keywords in a Lens, but very few mentions of other keywords. Fundamentally, unlike faceted search systems, different keywords in the Lenses typically describe a criteria as a whole. For example, an authentic ramen Lens might contain keywords describing creamy bone broth and freshly made noodles. In this case, the different keywords combined represented what the user considered good ramen restaurants, instead of as alternate options in a facet (such as a set of preferred brands). In a later iteration, we switched to Okapi BM25 for ranking that used inverse document frequencies to weight keywords instead of eliciting importance rating from the users. However, users reported unable to construct Lenses that reflect their priorities and unable to construct expressive Lenses that lead to useful results. This lead to the current iteration where we used a modified version of the standard Okapi BM25 ranking function to combine keywords across Lenses [181], which by default considers both term frequency and document frequency to rank documents similar to TF-IDF ranking function, but also adjust for the length of each documents.

We modify the Okapi BM25 ranking function to account for the importance levels specified by users in the following ways. By default, Okapi BM25 uses the inverse document frequencies to weight each keyword, with the motivation that words appearing in many documents tend to be less important. Since in SearchLens users can specify keyword importance using the interactive visual explanation, we instead weight each keyword according to their user-specified importance level. By default, SearchLens assume each Lens is equally important, and normalizes the weights of keyword q in a Lenses ℓ in proportion to the user-specified importance level of all keywords q in search Lens ℓ :

$$weight(q) = \frac{importance(q)}{\sum_{\hat{q} \in \ell} importance(\hat{q})}$$

SearchLens then uses the normalized keyword weights in place of the inverse document frequency term in the Okapi BM25 ranking function, and the score of each document d in the corpus for a set of Lenses L is therefore:

$$score(d, L) = \sum_{\substack{\ell \in L \\ q \in \ell}} \frac{weight(q) * tf(d, q) * (k + 1)}{tf(d, q) + k * (1 - b + b * |d| / avgDL)}$$

where ℓ is the different user-specified Lenses, q is the different keywords in each Lens ℓ , $tf(d, q)$ is the term frequency of keyword q in document d , $|d|$ is length of the document d , and the constant $avgDL$ is the average document length in the corpus. We used the default parameters $k = 1.2, b = 0.75$ for Okapi BM25. Finally, we sum up the score of each Lens weighted by a coordination factor, which is the proportion of keywords in a Lens that has a non-zero document frequency. This modified version of the Okapi BM25 function can be easily translated to SQL queries for standard relational databases, or as a custom ranking function for the popular open sourced document retrieval engine Apache Lucene. This allows the SearchLens interface to be easily implemented using readily available tools that were already optimized for scaling and computational efficiency. Admittedly, more sophisticated ranking approaches may further improve the quality of results, but this simple method allowed us to explore the costs and benefits of providing reusable, re-composable, explanation-centric Lenses to users.

5.3.5 Implementation Notes

The backend of SearchLens was implemented in Python, using NLTK [27] and gensim [179] for indexing and word semantic model, respectively. In the indexing phase, text in each review is lowercased, tokenized, and stemmed using the Word Punkt Tokenizer [114] and Porter Stemmer [214]. Stop words are filtered out. An inverted index that records the document and the offsets of the mentions of each word stems is computed and stored in a PostgreSQL relational database. The Flask Python framework was used for our HTTP server. We implemented front-end of the SearchLens prototype as a web-based system using Javascript (ES6) and the ReactJS GUI framework, and the interactive visualizations are implemented using the D3.js library. User-specified Lenses were stored on client-side using browser cookies, so that they are persistent for the searchers between multiple visits.

5.4 Evaluation

We evaluated SearchLens in two studies. First, we conducted a usability study in a controlled lab environment. Using predefined tasks, we tested the usefulness and usability of the system, as well as whether the visual explanation and exploration features provide enough benefit to encourage participants to express their rich and multifarious interests. Second, we conducted a field deployment study where participants use SearchLens for their own tasks. This allowed us to explore the benefits and limitations of our reusable and re-composable Lenses in real-life scenarios.

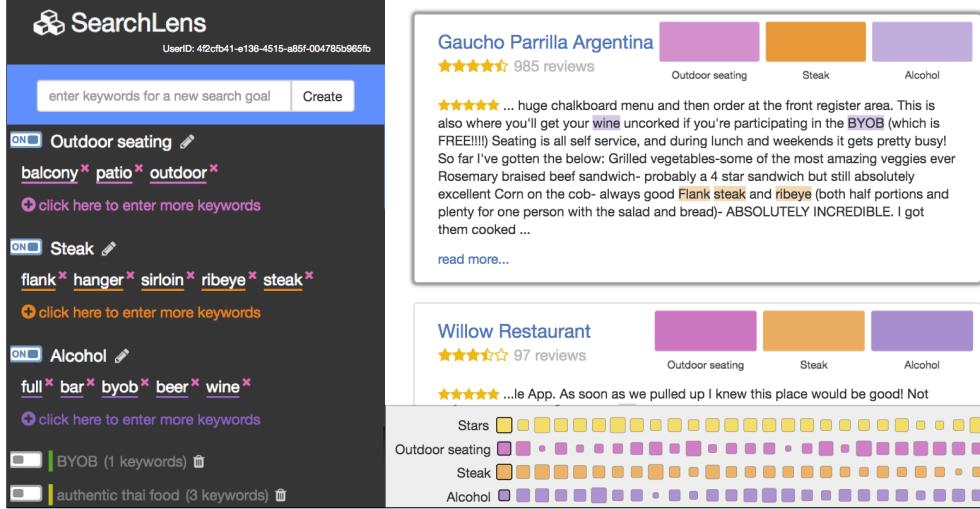


Figure 5.4: A Baseline system with topic-level visual explanation by collapsing the colored cells in each Lens and visualizing results only at the topic level.

5.4.1 Usability Study

The main goal of the usability study was to verify in a controlled lab environment the usability of the interface and whether the visual explanation and exploration features can provide benefits to encourage users to express their nuanced and multifarious interests. We considered these the preconditions for conducting a field deployment study to test the real-life benefits of reusable and re-composable Lenses. Therefore, we focused on the following:

- whether the interface encouraged participants to externalize multiple interests and structure them using Lenses
- whether participants found the visual explanation and exploration feature to be useful
- whether the added benefits of visual explanation and exploration encouraged participants to spend more effort to express, iterate, and refine their Lenses

To test the above, we compared SearchLens to a baseline interface as a between subject condition, where the detailed visual explanation and exploration features were removed by collapsing the colored cells in each Lens and visualizing results only at the topic level (Figure 5.4), resulting an interface similar to the TileBars and the HotMap systems [95, 97]. Unlike in the SearchLens condition, users can only explore each restaurants at the topic-level, but not at the individual keyword level. Since searchers can not assign importance levels for each keyword in the baseline interface, we used the standard Okapi BM25 ranking function that weights keywords based on inverted document frequencies [181]. We chose this baseline as a more conservative test of the interactive explanation features than, for example, a comparison to Yelp or other search query-driven site (which are the implicit comparisons for the field study below).

The three scenarios for the usability study are listed below. The first scenario was designed to have both clear criteria (nice decor and good atmosphere and serves beer or wine), and an exploratory aspect (find a specific type of Japanese restaurant based on your own preferences). Scenarios 2 and 3 were designed to explore whether users would be able to reuse their Lenses

Action	Lab Baseline	Lab SearchLens	Field SearchLens
add terms by typing	3.67 $\sigma=2.82$	5.50 $\sigma=4.86$	7.00 $\sigma=5.39$
add from suggestions	n/a	1.57 $\sigma=1.99$	3.20 $\sigma=1.79$
add from reviews	n/a	0.29 $\sigma=0.73$	0.40 $\sigma=0.55$
total add actions	3.67 $\sigma=2.82$	7.36 $\sigma=6.10$	10.60 $\sigma=3.71$
remove a keyword	4.67 $\sigma=4.27$	3.50 $\sigma=2.79$	4.20 $\sigma=2.68$
adjust weights	n/a	8.93 $\sigma=7.54$	12.80 $\sigma=7.89$
	N=15	N=14	N=5

Table 5.1: Mean statistics for number of Lens editing actions performed by participants. Participants used SearchLens in the lab study more frequently add keywords to refine Lenses compared to baseline ($t(27)=2.12$, $p<0.05$). Participants in the field study conducted their own tasks.

for different contexts and find value in doing so. Scenario 2 had overlapping criteria to Scenario 1 (serves beer, cocktails, or wine), and Scenario 3 involved performing an identical search to Scenario 1 but in a different city.

- **Scenario 1:** Stanley is in Pittsburgh, USA visiting some friends and he is in charge of finding a few good restaurants for the group. They are interested in Japanese restaurants. They're not familiar with Japanese food or the different types of Japanese restaurants, so it is up to you to find Japanese restaurants based on reading the reviews and your personal preferences. The restaurants should have a nice decor and good atmosphere. Some of his friends like to have a few drinks with their meal, so if the place has a bar that serves beer or wine it would also be great. Since its pretty nice out, it would also be nice if the restaurants has outdoor seating or a patio, too.
- **Scenario 2:** John is looking for good seafood restaurants in Pittsburgh, USA, particularly places that serves fresh oysters and has a bar that serves beer, cocktails or wine. Decor or atmosphere are not important, but big plus if they offer outdoor seating, for example, a patio. Some of his friends are allergic to seafood, so the place must also have non-seafood options, preferably steak.
- **Scenario 3:** (Same as Scenario 1 but for finding restaurants in Montreal, Canada instead of in Pittsburgh, USA.)

A total 29 participants were recruited from a local participant pool, where 14 participants were randomly assigned the SearchLens interface with three predefined search tasks (N=14, Age=18-61, M=28.1, SD=12.7, 7 male, 6 female, and 1 other/not listed), and 15 participants assigned the baseline interface with the same search tasks (N=15, Age=18-54, M=28.1, SD=10.7, 7 male, 7 female, and 1 other/not listed). Each participant was given 60 minutes to complete the study and was compensated 10 USD. Before conducting the three tasks, participants watched a five minute introduction video that described the features in their given interfaces, which is followed a step-by-step training where participants created two pre-defined Lenses, report the name of the third restaurant in their search results, and report which keyword is missing from its reviews. Participants finished the training steps using an average of 5.9 minutes (N=29, SD=3.8). For the main task, participants were told to spend 10 to 15 minutes on each of the three tasks listed above in order. Finally, participants answered a short post-survey where we collected their subjective opinions about the systems using 7-point Likert scales and free-form responses.

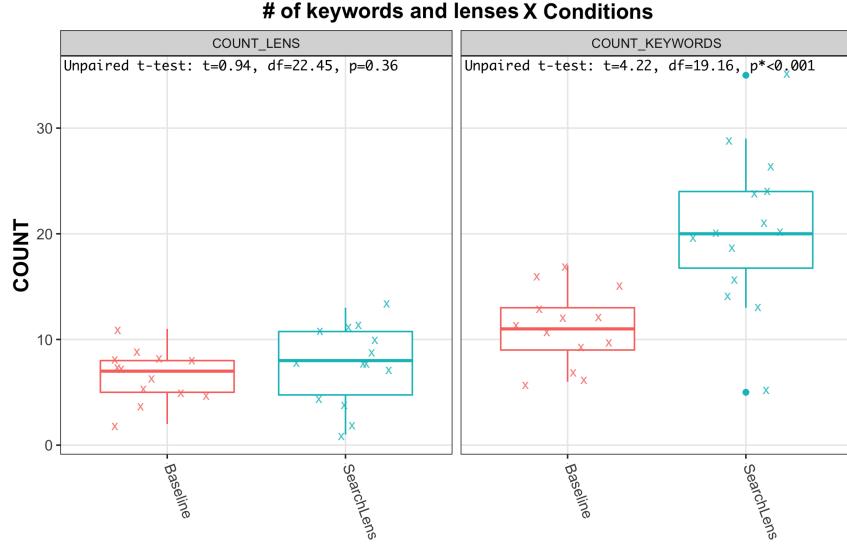


Figure 5.5: Number of Lenses and keywords saved by each participants at the end of the study. Participants in both conditions created comparable number of search Lenses, but participants in the SearchLens condition collected significantly more keywords in their Lenses.

Results for the Usability Study

One of our key hypotheses was that the immediate visual explanation provided by Lenses would encourage participants to express their interests and continually collect and refine those interests throughout the search process. This hypothesis appears to have been validated by the data. On average, participants in the SearchLens condition saved 20.43 keywords across their Lenses ($N=14$, $SD=7.33$), significantly more than participants in the baseline condition who saved 11.15 keywords ($N=15$, $SD=3.58$; $t(27)=4.12$, $p<0.001$). Importantly, this difference is likely not attributable to different perceptions of the task across conditions, as in both the SearchLens and baseline conditions participants generally created one Lens for each task criteria and combined multiple Lenses for each task (e.g., decor, drinks) and there was no difference between the total number of Lenses created between conditions (SearchLens: 7.6, baseline: 6.5; $t(27)=0.92$, $p=0.36$). In other words, the term-based interactive visual affordances supported by SearchLens seemed to encourage people to collect more terms indicative of their interests.

This pattern appeared to hold true throughout the search process for the iterative refinement of Lenses as well (Table 5.1). On average, participants using SearchLens added keywords to existing Lenses 7.4 times ($N=14$, $SD=6.1$) while those in the baseline condition did so 3.7 times ($N=15$, $SD=2.8$), which was found to be a significant difference ($t(27)=2.12$, $p<0.05$). This suggests that the added benefits from the visual explanation and exploration feature encouraged participants to iteratively refine their Lenses and allowed them to discover useful keywords more often.

We also examined whether participants found the added visual exploration features to be useful, and how the added benefits affected their behavior. By examining the behavior logs, we found participants using SearchLens frequently use the visual exploration feature. On average, each participant clicked on 25.86 ($SD=29.19$) keywords to filter reviews that mention a specific

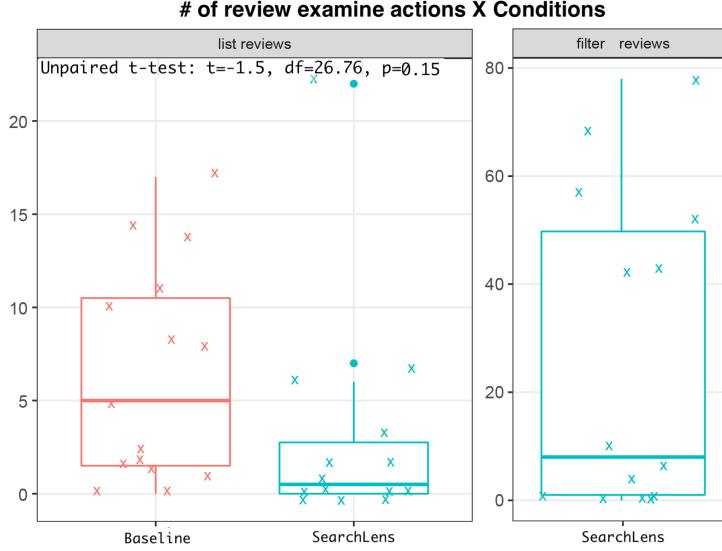


Figure 5.6: Participants in the SearchLens condition were less likely to read through unfiltered lists of reviews than the baseline condition, which was accompanied by increased use of the SearchLens-specific ability to filter reviews relevant to different keywords.

keyword instead of sifting through reviews to find ones that mentioned it (Figure 5.6). In both conditions, participants can also click on the name of a restaurant to see a list of reviews ranked by all active Lenses. While there is suggestive evidence that the filtering of reviews led to less use of the generic review lists, the result was not significant based on the number of participants in the study ($M=6.33, 3.07$; $SD=5.78, 5.92$; $t(27)=1.50$; $p=0.15$).

These results suggest SearchLens allowed participants to maintain a broader search goal with multiple interests, while at the same time explore and compare different options at a finer-grain level interactively instead of sifting through the reviews of each restaurant.

5.4.2 Field Study

Our field deployment study aimed to test our idea of reusable and re-composable Lenses in real-world settings. Five participants were recruited from the first study based on their high self-reported interest in researching restaurants online and in participating in a follow up study ($N=5$, Age=18, 20, 22, 23, and 25, 4 male, and 1 others/not listed). The participants were given access to the SearchLens system via the internet, and were asked to use the system for at least 60 minutes in total over a three day period. Although they were free to choose from any of the 11 cities in the dataset for this study, all five participants conducted tasks for their current city. Afterwards, they return to the lab and were given 45 minutes to finish a survey with primarily free-form questions, and were interviewed for another 15 minutes. Each participant was compensated with 40 USD for finishing the study.

Participants created more Lens keywords when conducting their own tasks comparing to participants in the lab study (Figure 5.7). On average, participants in the field study created 13.40 ($SD=3.65$) Lenses, significantly more than participants in the lab study that created 7.64 Lenses ($SD=6.54$; $t(17)=2.46$, $p<0.05$). They also saved significantly more keywords than participants in

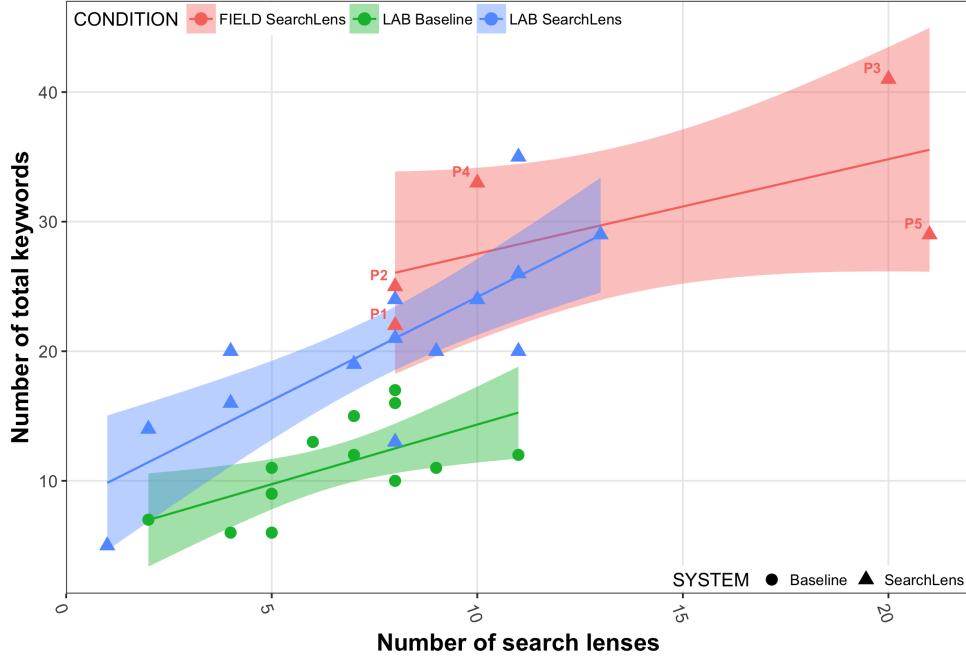


Figure 5.7: Number of Lenses and keywords specified by participants under different conditions. In the lab study with predefined search tasks, participants using SearchLens (blue) created a similar number of Lenses but used more keywords than the baseline condition (green). Participants in the field study (red) conducted their own tasks.

the lab study (lab: 20.4, field: 30.0, $t(17)=2.50$, $p<0.05$). Admittedly, it can be difficult to measure how much time participants actually spent using SearchLens in the field, nevertheless, results suggest that participants were able to accumulate more interests Lenses over a three day period than participants who spent 60 minutes in the lab study.

All five participants conducted multiple tasks during the study. Many explored different types of restaurants that they liked in the city using multiple Lenses, using SearchLens to build “*an overview interface for restaurants in the city that I might like*” (P1, P3, P4, P5). Participants also had more specific goals, including to check if there are vegan restaurants she has not discovered yet (P5), restaurants that serve bubble tea (P2), pizza places that offer Chicago deep dish-styled pizza (P3), and Mexican restaurants that has vegan options on the menu (P2).

Refining Lenses

While participants reported creating Lenses based primarily on prior knowledge, all five participants also reported refining their Lenses throughout the process. Several cited that the shaded cells of the visual explanation helped them quickly noticed some keywords were too uncommon, and that an important concept of interest was missing from the search results (P1, P2, P5). One also mentioned noticing and removing ambiguous keywords when using the mention filtering features (P4). Participants also learned about new keywords which they added to their Lenses, sometimes replacing existing keywords, from both the suggestions (P1, P2, P3, P5) and from the reviews (P1, P2). Interestingly, the behavioral logs (Table 5.1) suggest they frequently discovered them from the systems’ suggestions, indicating the value of the word2vec approach

which we initially were concerned about for being noisy. This also points to potential future work in auto-suggesting Lenses which we intentionally avoided here due to concerns about agency and explainability.

Breadth and Depth

Participants created both general, breadth-oriented Lenses and more specific, depth-oriented Lenses. P4 specifically mentioned that it was useful being able to search for different genre (i.e., American, Mexican, or Indian restaurants) and at the same time pay attention to very specific dishes (i.e., cheese steak sandwich made with chicken), while still being able to see how each result match with different things, citing that “*more specific things are hard to search for on Yelp.*” Alternatively, P3 presented an interesting use case for deeper exploration of a specific genre, by first creating a more general Indian Food Lens, and then creating multiple more specific Lenses describing specific dishes from different regions of India, generating an overview of different styles of Indian restaurants in the city. This suggests that some users may want to create higher level groups of Lenses

Reusing Lenses: Combinations and Task Resumption

Participants reported their strategies for how they reused their Lenses, which can be broken down into two non-exclusive categories. The first use case we observed was task resumption between multiple search sessions (P1, P3, P4). Participants described having the ability to switch to a different sets of Lenses yet still keep the original Lenses for the future being useful (P3). One participant (P1) searched with a single Lens most of the time, but still cited that being able to re-enable Lenses from past sessions and to continue work on previous tasks and refined restaurants being useful. For the second use case, participants mentioned reusing Lenses in combination with other Lenses (P2, P3, P5). When asked about which of their Lenses were used in combination with different other Lenses, participants reported Lenses that concerned style and environment (*Cute and Quirky* (P5), *Atmosphere and Vibe* (P2, P5), *Friendly Staff* (P3)), price (*Inexpensive* (P2, P3), *Large Portion* (P3)), and some food-related but not for a general genre (*Fresh* (P2), *Fast Casual* (P2), *Vegan Options* (P2, P5), *Strong Beer* (P3)).

5.4.3 Overall Usefulness and Other Usecases

Through the lab and the field studies, we found evidence that using user-generated Lenses to provide visual explanation for deeper exploration was beneficial and effective in incentivizing users to externalize and iteratively refine their interests using Lenses. This occurred throughout the search process almost twice as frequently when compared to participants in the baseline condition which did not include the visual explanation and exploration features (Figure 5.4). As a result, participants using SearchLens created richer Lenses with nearly double the number of keywords on average compared to participants in the baseline condition. Participants also frequently used the visual explanation feature to explore the individual items in their search results, filtering reviews using different keywords in their Lenses 25.9 times on average. To test SearchLens in real-world settings, participants in the field study conducted their own tasks, and provided insights into their strategies in building and refining Lenses, as well as their strategies of composing and reusing Lenses across context and across search sessions over a three day period.

From the field study interviews, three out of the five participants said that they actually found and saved interesting restaurants during the study, and intend to visit those restaurant in the near future (P1, P3, P4). P1 in particular went to one of the restaurants he discovered using SearchLens and was happy about the visit, and P3 used SearchLens to complete a previous task, saying “*I wanted to try deep dish pizza for some time since I moved to US. Finally found one near the city. Kudos!*” All participant expressed that they would be interested in using SearchLens in the future if available, many also cited other scenario that might benefit from SearchLens. P2 pointed to scenarios where he needed to “*find a place for many people that may want different things*”, and mentioned that SearchLens would be useful when her family visits her soon for his graduation. These results suggest that SearchLens was effective at helping users effectively find items that matched their specific interests.

5.5 Discussion and Limitations

One limitation of the current implementation of SearchLens is its lack of ability to filter restaurants using their metadata, such as geographic location. We intentionally did not expose this information to our participants so we can focus our studies on allowing them to build personalized Lenses. However, practical systems would likely combine both paradigms to maximize efficiency. Utilizing metadata can also augment user-defined Lenses, for example, taking into account whether the a review that matched a specific Lens was positive or negative and whether the review poster’s interests matched with the user’s personal interests. However, the interactions between the two paradigms would require further studies. On the other hand, utilizing existing techniques for query term generalization beyond stemming or lemmatization, such as synonyms, semantic word models, or query expansion, can potentially improve recall, but their effects on the visual explanations would also require further studies.

Another obvious limitation of SearchLens is that it required more user effort upfront in order to receive the benefits provided by the system, such as reuse, explanation, and exploration. On a 7-point Likert scale, most participants from our lab study responded favorably in the post-survey to this trade-off with 64% agreed or strongly agreed that SearchLens is an improvement to the traditional search interfaces, and another 21% somewhat agreed with the statement, however, the long-term effect remained to be seen. One way to extend SearchLens is to combine machine learning and information retrieval approaches to reduce the effort of building Lenses, such as building interest profiles automatically, or using collaborative filtering and query expansion for expanding or inferring Lenses automatically [3, 189, 224], or word-sense disambiguation techniques for resolving ambiguous keywords [226].

Alternatively, we could also explore ways to allow users to share their Lenses with each other through explicit or implicit collaborations. For example, one participant mentioned “*It would be nice if I can see what Lenses a local person would use if I’m traveling, because I always try to ask the locals about where I should eat.*” Allowing access to Lenses created by previous users or expert users could potentially enable expertise transfer and accumulation through continuing refinement of a set of Lenses. For example, locals and past travelers could iteratively curate a set of Lenses that leads to an interactive and exploratory list of local specialties for future travelers.

Another promising direction is to more deeply explore the idea of user-generated interest profiles and how they could dynamically influence the different interfaces accessible to the user or interacting with users in more proactive ways. Since we asked the field study participants to use

SearchLens for their own tasks, most participants searched for restaurants in the city they lived in. Some participants that conducted more targeted search tasks (P2, P3, P5) mentioned that they were already familiar with most of the options in the city that fits their goals, but would still occasionally search online to see if there were new restaurants that match their interests (P2, P5). As users continue to use SearchLens, the system will accumulate more understanding of what the users is interested in, and can potentially detect and notify the users of new information that might be of interests with high accuracy [225]. Alternatively, existing users may use their repository of Lenses to explore or curate the restaurants in an unfamiliar city. Participants in the field study also pointed to the potential of Lenses being useful for other types of information and domain, including shopping (P2, P3), trip planning (P2, P5), buying a house (P2), and job hunting (P4).

In this chapter we introduced SearchLens, a novel approach that allows users to specify and maintain their profile of multifarious and idiosyncratic interests. This enabled them to reuse and re-compose their different interests across scenarios, as well as maintaining context across multiple search sessions. To encourage users to put in the up-front effort of curating Lenses, we explored ways of using Lenses to provide immediate benefits of visual explanation and deeper exploration of search results. Across a lab and field study we observed that participants expressed their interests with significantly more query terms, and found benefits in the SearchLens approach, including being able to transfer and reuse their Lenses across contexts, being able to interpret new information that reflects their own personal interests with transparency, and working at multiple levels of specificity and hierarchy. More fundamentally, being able to visualize and explore new information in ways that promote transparency can potentially empower users to be more aware of their online information diet. For example, as a way to manipulate their own social media feeds, and being more aware of how posts were selected or hidden. We believe SearchLens represents a first step towards a transparent and user-centered approach to addressing subjective and fragmented nature of information today.

Chapter 6: Weaver

Entity-Centric Foraging across Webpages in the Browser

The previous chapter explore ways users can express and maintain their different criteria for selecting restaurants, and use them to visualize a review dataset and compare different search results. This chapter focus on supporting global context so that users can better evaluate their different options as they encountered them on different webpages. Unlike in the previous chapter where the evidence (i.e., reviews) were already recognized by the different restaurant entries on Yelp, this chapter instead support users' general browsing of different webpages using their browsers. This is enabled powering browsers with modern named entity recognition and linking algorithms, allowing us to identify the same entity mentioned across users' browser tabs.

As people research online to plan trips or shop for new products, they encounter many entities (e.g., attractions, products) and collect evidence across webpages to make informed decisions (e.g., reviews, listicles). Current browsers treat entities on each webpage independently of other pages, making it difficult for users to keep track of what they are interested in and why. We introduce Weaver, a novel browser add-on that weaves pages together through common entity mentions to support sensemaking across browser tabs in the context of trip planning. When users open a webpage, Weaver “infuses” it with evidence from other information sources relevant to entities on the current page. When users save notes, their notes are “diffused” across other pages that mentioned the same entity. We compared Weaver to a baseline and found participants utilized Weaver to gather nearly three times more evidence collected across significantly more webpages, and synthesized evidence to support decision making with lowered interaction costs.

6.1 Introduction

Whether planning a trip to a new city, figuring out which camera to purchase, or researching the different treatments for a medical issue, learning and searching for information online has become the most common way that people make sense of the world today [149, 171]. People spend a significant amount of time exploring available options and gathering evidence about them that are scattered across multiple webpages in order to make informed decisions. Estimates suggest that up to 33% of the time spent online [120, 149, 184], or, as of 2009, around 24 billion hours per year in the US alone, are spent doing this type of aggregation and synthesis [7].

We believe this problem of synthesizing information across sources is increasingly relevant as online information and misinformation (such as fake news and shill reviews) continues to expand. Past studies have shown users rely on aggregating from multiple sources in order to verify online information as credible and make decisions [62, 81, 177], but the process can be “tedious and cumbersome” leading to “opening several tabs ... and then manually switch[ing] between them while trying to remember information on different pages” [85]. Anecdotal evidence for this can also be seen in the rise of aggregation-based sites such as Metacritic or Wirecutter, but such ag-

1. Go Across The **Golden Gate Bridge**



[A]

[B]

Between San Francisco Bay and [Marin County](#), is the world famous [Golden Gate Bridge](#). The bridge has been declared one of the modern wonders of the world. It was opened in 1937 and [at that time it was the longest suspension bridge ever created](#) [F1].

Made from steel and with a total length of 1.7 miles it is the most photographed bridge anywhere in the world. There are six lanes of traffic on the bridge carrying millions of passengers every year. Before the bridge was built people used to have to get a ferry between the two places, the ferry company was called [Golden State Ferry Company](#), and at one point it was the largest ferry company on the planet.

2. Head Down To The Waterfront At **Fisherman's Wharf**

Cromer [C] [September 11, 2018](#)

15 Best Things to Do in San Francisco [D] [August 22, 2018](#)

15 Best Day Trip [E] [August 22, 2018](#)

RELATED POSTS

15 Best Things to Do in San Francisco [F]

HIDE MAP

24 SAVED CARDS AND 4 CATEGORIES

Golden Gate Bridge [G] [Landmarks & Historical Buildings, Art Deco ...](#) San Francisco, CA 94129 - (415) 921-5658 [1735 Yelp reviews](#)

My Notes: We should walk across the bridge, parking can be a bit tricky tho.

Mentions: [W](#) [X](#) [Y](#) [T](#) [I](#) [F](#) [B](#) [G](#)
My Clips: [C](#) [M](#) [U](#) [4](#)

[expand and read more...](#)

[F2] + SAVE SELECTED TEXT AS NEW CLIP

Figure 6.1: An overview of the Weaver browser add-on. Weaver identifies and highlights entity mentions on webpages [A,B] to indicate additional information is available. Highlights in red [B] indicates users had previously interacted with the entity. Hovering over a mention [B] brings up its corresponding entity card [E] as an overlay, with relevant information “infused” from other pages and knowledge bases as *mentions*. Users can also save notes or selected sentences [F1] to a card as *clips* [F2]. Saved clips are then automatically “diffused” to other webpages that mentioned the same entity. Users can also create categories [D] and drag the card under them. Finally, the Map view [C] shows its location in context of previously saved entities.

gregators cannot cover all decision making scenarios nor able to take into account the personal context and goals of different users. In our own informal interviews on people's past experiences with trip planning and examining their notes, we also discovered similar needs and challenges – they compared a large number of options based on evidence gathered from multiple sources, but struggled with managing large numbers of options and sources.

In this paper, we focus on the domain of travel planning, because it has a number of characteristics that make it a good task to test new sensemaking and exploratory search approaches. For example, while it does not require a strong domain knowledge, useful information is often scattered across many sources; there is a strong degree of contextualization and personalization needed (e.g., traveling somewhere with kids is very different than without); and evidence such as reviews can be noisy and subjective [50, 232]. Consider for example planning a trip to a new city: there may be hundreds of possible restaurants to dine at, attractions to see, and places to stay, each with corresponding evidence about its suitability for an individual's goals and preferences. Evidence about each of these options is often spread out across multiple search results, such as Yelp or TripAdvisor reviews, top ten lists, travel blogs, or forum posts. It can be a challenging process of intense cross-referencing and note taking to synthesize this evidence and to record how it meets a user's goals, with little scaffolding or intelligence built into the browser [23, 153, 167, 208].

Currently, such intelligence primarily takes the form of entity-centric approaches in search results interfaces [86, 142]. These approaches include showing entity cards with rich attributes for entity-bearing queries [32, 161], listing related entities as suggestions for subsequent queries (such as listing actors when searching about a movie) [28, 31, 126], or extracting factual attributes about or relationships between entities (such as the director of a movie) [13, 55, 87]. While these approaches can efficiently provide factual and structured information about entities in search interfaces (such as when figuring out the location of a restaurant), they provide little support for the complex sensemaking situations described above, when there is no single objective answer that can be surfaced in a search results page (such as figuring out which city to visit for a vacation).

Instead of using entities as an answer or endpoint to a user's information needs at the query and retrieval stages, here we explore the idea that entities can also be useful during the reading and note taking stages by acting as the fabric connecting different information sources and use them to scaffold the user's mental model in complex exploratory search tasks. Leveraging entities has the potential to enable deeper and more fluid interactions with online information by focusing on meaningful concepts as the units of a user's externalized mental model rather than webpages. Furthermore, recent advances in entity linking algorithms have been particularly promising in bringing the ability to better understand web content to the browser where users read and learn from individual webpages [158]. For example, leveraging common entities mentioned across webpages to provide a sensemaking structure for users conducting complex exploratory searches and foraging across multiple webpages.

Specifically, in this paper, we explore a new paradigm for interacting with unstructured and subjective evidence about entities while reading and foraging from many webpages in exploratory search tasks. Since the user's personal evaluation of subjective information and how it meets their goal is critical in this situation, our design goal was to help the user to see scattered evidence about an entity in one place while also attaching personal notes and web clippings. These together serve as a way to build up an external mental model and track search progress.

To investigate our entity-centric approach, we developed a prototype browser add-on called Weaver. Weaver allows users to keep track of information scattered across multiple sources by “infusing” evidence about an entity from other information sources (webpages and knowledge bases) to the webpage the user is currently reading. It also “diffuses” users’ thoughts about different entities across webpages where the same entities are mentioned for future reference and to accumulate more evidence. In our user study, we tested how participants utilized our entity-centric approaches while conducting a complex exploratory search task, focusing on whether Weaver allowed them to explore, gather, reuse, and accumulate evidence about entity options across multiple webpages. To control for task complexity, the primary domain on which we aimed to test Weaver was a pre-defined travel planning task (described in the Study Design section). With 24 participants and a baseline interface as a between subject condition, we found that participants using Weaver collected nearly 3 times more evidence across 60% more pages while simultaneously being more selective in the options they explored. Furthermore, we describe qualitative evidence of the value in infusing of evidence from other webpages and diffusing users’ notes across webpages. Finally, we discuss implications for the design of future intelligent interfaces that can better understand the information being consumed by their users by taking advantage of advances in natural language processing to support online sensemaking in various scenarios.

6.2 Related Work

6.2.1 Sensemaking and the Web

The importance of sensemaking and complex exploratory search on the web has been studied in depth by many researchers. Past work have identified a persistent challenge that valuable information for many topics is scattered across many different sources that are independent of one another and incur a high cost for bringing them together [22, 125, 149, 153]. Theories of sensemaking suggest several cognitive tasks involved that could be supported through novel interactive tools, ranging from finding potentially relevant items, to triaging items based on reliability and relevance, to collecting evidence relevant to each item and organizing items into categories or structures [94, 186, 206]. We draw on these theories to select the set of cognitive tasks we are interested in supporting through entity-centric interaction approaches, which are typically complex and about synthesizing information, rather than simple fact-finding tasks [149, 218].

6.2.2 Recognizing Entities in Text

Significant research has gone into entity-centric approaches for improving web search results pages due to the ubiquity of entities in online sensemaking. Researchers have found that entity-bearing and -category queries accounted for up to 85% of web search traffic [86, 142]. This has led to significant academic and commercial efforts devoted to building large-scale entity databases (such as DBpedia [11], Yelp¹, and Google Places²), and significant research on ways to identify entity mentions in plain text [158] and using them to enrich search interfaces. This involves both recognizing the same entity mentioned in different surface forms (e.g., MoMA and Museum of Modern Art) and resolving ambiguous surface forms to the right entity based on its

¹<http://yelp.com>

²<https://developers.google.com/places/web-service/intro>

surrounding text. Major threads of research that uses entities to improve search interfaces includes showing entity cards for entity-bearing queries [32, 161], answer factual questions [87], and showing related entities as query suggestions [28, 31, 126]. We build on these recent advances in entity recognition and large-scale entity databases, but instead of focusing on the search interface and simple navigation we investigate the less-explored design space of supporting complex sensemaking across webpages opened in the browser through entity-aware interactions.

6.2.3 Weaving Together Scattered Information across Sources

Due to the scattered nature of information on the Web [22], research has explored ways to connect relevant information distributed across different webpages. One set of top-down semantic web approaches involve incentivizing content publishers to provide machine readable annotations, such as using semantic web markups [17, 18, 119]. However, such approaches have often failed to gain momentum due to issues such as a lack of available end-user tools that can consume these annotations [118]. Alternatively, researchers have built bottom-up systems that exploit detecting entity mentions in articles, and used them as anchor points to connect to other information sources to enhance the reading experience. For example, Wikify identifies entity keywords in articles and creates hyperlinks to their Wikipedia entries for navigation [159], and Experience-Infused Browser links entity mentions in articles to past social interactions (such as emails) for making “serendipitous connections” [89].

Our approach is inspired by these bottom-up approaches in the context of recognizing entities mentioned in webpages opened in the browser. However, our work differs in several important ways: 1) instead of surfacing simple facts or links to articles or emails, we focus on providing context for people to understand complex information spaces; 2) we support the transition from viewing context to saving information; and 3) we propagate saved information to all other instances that entity is shown, allowing users to build up an entity-based mental model of the space.

6.2.4 Note Taking and Saving Information Online

Collecting information online during complex sensemaking tasks can be costly for the users, requiring them to cross-reference between pages and their notes in order to gather all the evidence. This frequent context switching between different documents and taking notes can be distracting, and even prohibitive for users to investigate deeper or to take notes in order to avoid disrupting the flow of reading [23, 153, 167, 208].

On one end of the spectrum, research has focused on allowing users to extract and save structured information from webpages more efficiently from a single document using end-user programming [73, 74, 98, 104] and interaction techniques [24, 204]. Our work is motivated by these studies highlighting users’ desire to collect information about entities, but instead of focusing on structured and objective attributes we are interested in additionally gathering descriptive and subjective evidence, such as how a restaurant is described in a top ten list. Furthermore, many of the approaches above define patterns for extracting multiple entities from a single page, whereas we are more interested in finding evidence related to entities across multiple pages where the structure of those pages might differ significantly.

On the other end of the spectrum, researchers have also explored using *in situ* interfaces, such

as sidebars or on page annotations, to reduce the costs of switching between reading and note taking and collecting from multiple information sources [183, 191, 208, 212, 213]. Our work is also situated in this thread of research on reducing the costs of switching and sensemaking. However, instead of persisting notes on individual documents, our approach persists notes on individual entities which may then appear across multiple documents.

6.3 System Design

We introduce Weaver, a novel browser add-on that uses an entity-centric approach to facilitate sensemaking across webpages in exploratory search tasks. Unlike previous approaches that focused on either enriching search results pages [161, 205], saving information from individual documents [98, 208], or providing separate note taking interfaces [74], we focused on supporting sensemaking across multiple information sources by weaving them together through common entity mentions. This allows users to both evaluate potential options with more context and re-access previously saved information about an option with lowered effort. Figure 6.1 shows an overview of how an exploratory searcher planning a trip might use Weaver. Weaver provides users a lightweight overlay interface embedded on and synced across webpages opened in different browser tabs, allowing users to make quick and lightweight cross-referencing without switching between tabs, windows, or applications.

The two core components of Weaver scaffold sensemaking through entities in two primary ways, which we introduce as “*Infusion*” and “*Diffusion*. First, when users open a webpage from their search results, the system “infuses” the webpage with relevant snippets about mentioned entities from other webpages in their search results and external knowledge sources to help users cross-reference and evaluate newly encountered options. Second, when users save notes or extract content from a webpage, the system “diffuses” them to mentions of the same entities in other webpages of the same task, allowing them to easily access previously saved information without having to switch to and search through their notes a separate interface. In the next subsections, we will first describe in detail both Infusion- and Diffusion-based features, a project overview interface, and finally describe how entities were automatically identified and linked to external knowledge bases.

6.3.1 Infusion: Gathering Evidence from other Webpages

One significant challenge in making sense of a given topic on the internet is that relevant information is scattered across multiple places, and it is difficult to find those places and to synthesize what they say. Doing so is valuable in understanding the popularity and prevalence of a given option (e.g., how often a restaurant is mentioned in lists of top restaurants, what are the various lists of top restaurants) as well as the context and potential biases in how it is described (e.g., is it suitable for a date night, are the pages it is mentioned on reliable). Instead of using them at only the beginning on search results pages, entities could provide a scaffold for improved in situ interactions throughout the browsing process to help users more quickly get a sense of the popularity and context of different options without going to all the different pages on which they are mentioned, as well as providing “pivot points” to see what other sources mention an option.

For example, in Weaver, a user planning a trip to a new city can open an article from a travel blog and see all the destination and restaurant mentions highlighted in yellow. Hovering on a attraction surfaces contextual snippets from other webpages that also mention that attraction, so

Japanese Tea Garden (San Francisco)

Tea Rooms, Gift Shops, Golden Gate Park, Japanese-American culture in San Francisco, ...
75 Hagiwara Tea Garden Dr, Golden Gate Park, San Francisco, CA 94118 - (415) 752-1171

★★★★☆ 1251 Yelp reviews

W The Japanese Tea Garden in San Francisco, California, is a popular feature of Golden Gate Park, originally built as part of a sprawling World's Fair, the California Midwinter International Exposition of 1894. Though many of its attractions are still a part of the garden today, there have been changes throughout the history of the garden that have shaped it into what it is today.

[en.wikipedia.org/wiki/Japanese_Tea_Garden_\(San_Francisco\)](http://en.wikipedia.org/wiki/Japanese_Tea_Garden_(San_Francisco))

Reviews on Yelp

★★★★☆ Pyra-Danny S.: LITTLE, PEACEFUL PLACE EVEN WITH CROWDS [Warning: Young adults from Central Florida in for vacation] This place is much larger than it looks. We weren't... - 2018-09-18 20:44:12

★★★★★ Dennis L.: Beautiful place. 9 bucks entry per person. No regrets. Gets you access to the awesome tea house and great place to take pictures. U can get a ton of... - 2018-09-16 15:48:55

★★★★☆ Stanley W.: Beautiful place to take a break after hours of walking situated in the Japanese Botanical Garden. The tea was unspectacular and subpar compared to what... - 2018-09-10 05:43:07

yelp.com/biz/JcVzpOTp7OH6l8AsM_wGdQ

Your notes

click to type notes

Also mentioned on 1 other page in your project

[Actually Cool Things to Do in San Francisco Right Now When ... - Thrillist](#)
thrillist.com/entertainment/san-francisco/things-to-do-in-san-francisco

Visit the Japanese Tea Garden \$Golden Gate Park If you get the sense that your friend or family member needs to unwind (or you need to after having a house guest for a week), take them to the oh-so serene Japanese Tea Garden where you can meander on the winding paths past koi ponds, a Zen garden, native Japanese plants, and pagodas. And be sure to make them climb the famous Drum Bridge so you can take a picture. [SAVE TO MY NOTES](#)

\$Golden Gate Park If you get the sense that your friend or family member needs to unwind (or you need to after having a house guest for a week), take them to the oh-so serene Japanese Tea Garden where you can meander on the winding paths past koi ponds, a Zen garden, native Japanese plants, and pagodas. And be sure to make them climb the famous Drum Bridge so you can take a picture. [SAVE TO MY NOTES](#)

Figure 6.2: Expanded view for an entity card showing information *infused* from external knowledge sources (Yelp and Wikipedia), user's notes, and evidence of the same entity from other webpages in the exploratory search task. See Figure 6.1 for the non-expanded view of an entity card.

The figure shows a user interface for managing saved entities. On the left, there are two main categories: 'Things to do (9 items)' and 'FR French quarter to dos (6 items)'. Each category contains a list of entity cards. For example, under 'Things to do', there is a card for 'Big Easy Bayou Kayak Tour' which includes a photo of a kayak, notes, and a link to expand and read more. Another card for 'The National WWII Museum' shows its logo. The 'FR French quarter to dos' category includes a card for 'Royal Street, New Orleans' with a photo of a street scene. On the right side of the interface, there is a map of New Orleans with various locations marked, including 'Louis Armstrong Park', 'Royal Street, New Orleans', 'French Market', 'The Presbytere', 'Alger Point', and 'Canal Street Ferry'. The map also shows the Mississippi River and surrounding areas.

Figure 6.3: An project overview page created by one participant after searching for 50 minutes, containing entities saved under different categories, text clips from multiple webpages, and typed notes. Custom cards were created, including one for a kayaking tour that was not available on Yelp and DBpedia. The entity cards were scaled for clarity in this figure.

the user can understand its relevance to their own goals without interrupting their flow of reading (Figure 6.1, B and E). Other sources of information can be brought in and surfaced at the same time, such as Yelp review scores and Wikipedia descriptions. By infusing this information using the entity as a pivot, it is possible that users could both reduce the number of items they need to keep track of (by filtering out non-matching items earlier) and better understand the context of the items they do keep. Meanwhile, the number of sources that mention the item provides implicit feedback as to its popularity within the searches the user has made; their URLs may provide context as to the reliability of those sources; and they may provide additional sources for finding information about other restaurants if they mention a restaurant the user knows they like.

Weaver supports this need by “infusing” entities mentioned on a page with context pulled from other webpages mentioning the same entity or entries in external knowledge sources (in our current implementation, DBpedia and Yelp). When users open a webpage, entity mentions that were recognized by Weaver are highlighted with a half-height yellow highlight (Figure 6.1, A) to indicate they have information from other sources. By hovering over an entity, a user can see an “entity card” (Figure 6.1, E) which displays those sources and relevant information (e.g., number of stars on Yelp, paragraphs from other websites in which the entity was mentioned) which a user can use to gain context about the entity beyond the current webpage [32]. To read the mentions, the user can click on the icon of each external source to see an extracted snippet. Alternatively, the user can also expand the Card to see a larger view (Figure 6.2), showing all mentions, multiple images, and a map of the location using metadata from Yelp and/or DBpedia.

6.3.2 Diffusion: Propagating Notes to other Webpages

After using “infused” context to judge the relevance and suitability of options (i.e., entities), users often need to keep track of and organize the options they found valuable. At the same time, users may evaluate newly encountered options against ones they have already saved. Typically,

this happens by copy-pasting or typing entity names and notes into a separate interface, for example a separate document or email or note taking software (e.g., Evernote). Researchers have tried to lower the switching cost involved in this interaction [167, 208], for example, by adding a sidebar to the browser for taking free-form notes [212]. However, in the cases when the user encounters additional evidence about an option they already have information about, they need to re-find it in the external system before being able to continue. This high interaction cost can be prohibitive as it lead to disruptions of user's flow of reading [43, 153, 167, 208].

Weaver addresses this challenge by “diffusing” notes that users associate with an entity to all other webpages in the project that also mentioned the same entity, reducing the need for user-driven re-finding. Continuing with our running example of trip planning from the previous subsection, imagine that after the user reviews the information in a restaurant entity card, he or she decides to take notes and save the restaurant for future reference. To do so, the user can add various levels of annotation to the card, including just “hearting” it to save it in the Saved Cards view as uncategorized (Figure 6.1, top-left corner of E), typing notes about reasons for saving it (Figure 6.1, yellow region in E), or selecting sentences (Figure 6.1, around B) from the webpage to add to the entity card as a clip (Figure 6.1, E). When the user moves on to other webpages in the project, mentions of the same restaurant will be highlighted in half-height light red (Figure 6.1, B), indicating that the user previously interacted with this entity, and upon hovering will see its entity card with annotations and clips they have previously added.

Using this entity-centric approach, users can save notes of information collected across webpages under entity cards without having to switch back and forth between the browser and note-taking software, and easily re-find and reuse previously saved information when encountering the same entities on other webpages. To recover from cases where an entity of interest was not recognized by Weaver automatically, users can manually create entity cards using interactions as described in the next subsections.

6.3.3 Project Overview and Organizing Entities

As users in exploratory search tasks gradually progress from discovering entities and gathering evidence to focus more on synthesizing and making decisions, they may also need to organize and compare the collected entities. For example, in a travel planning task, users may want to group their entities into categories of restaurants, attractions, and hotels for comparison, and also to figure out the location and distances between the different entities to plan their trips.

In Weaver, in addition to simply “hearting” an entity card, users can also create categories with custom names, colors, and icons in the Saved Cards view (Figure 6.1, D). To categorize an entity card, simply drag and drop it between categories. This allows users to start structuring any time during their exploratory search process when the need arises. Saved geographic entities (entities with coordinates metadata from Yelp and/or DBpedia) will also show up in the Map View (Figure 6.1, C) with their icons and color coded pins. In addition, when users hover over an unsaved geographic entity on the current webpage, its location is also shown on the Map view. This allows users to better situate a newly encountered option with previously discovered entities to make informed decisions. For example, a user could quickly figure out that a hotel recommendation on the current webpage is not relevant by noticing in the Map view that it is too far away from most attractions that they have saved previously from other webpages. At later stages of the exploration process, users might shift their focus from reading and gathering infor-

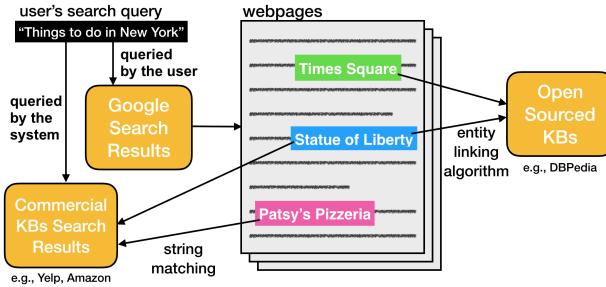


Figure 6.4: Weaver links entity mentions from webpages to both open and commercial knowledge bases.

mation to synthesizing and organizing information. For this, they can open the Project Overview page by clicking on the expand button in the Saved Cards view to see all their entity cards listed in multiple columns of each category along with an integrated map view (Figure 6.3).

6.3.4 Linking to Open and Commercial Knowledge Bases

To drive these operations and connect the different webpages, Weaver uses the DBpedia Spotlight algorithm [158] to automatically identify common entities mentioned in the different webpages. In our implementation, we use Yelp and DBpedia as our entity repositories and focus on travel planning tasks, but other knowledge bases can also be used or added to support other types of projects. For example, using the Microsoft Academic Graph³ and the Gene Ontology [10] as knowledge bases to support literature review projects in biology.

As users search on Google, Weaver parses the HTML of the search results pages to obtain the list of webpages. In the background, Weaver analyzes the content of webpages to identify entities mentioned using the following methods (Figure 6.4). First, it uses the Spotlight library [158] to identify entities mentioned in different surface forms (e.g., “San Francisco Museum of Modern Art” and “SFMoMA”) to DBpedia which contain rich attributes extracted from Wikipedia. Unlike DBpedia, Yelp is a commercial service in which neither the entity database nor a pre-trained entity linking model were publicly available. In order to identify Yelp entities in webpages, we use keywords and a location extracted from the original query term users performed on Google (e.g., “best sushi bars in new york”) to query the Yelp Search API⁴ for a list of 450 related Yelp entities. This allows Weaver to retrieve from closed databases for entities that match with users query intent and information needs. Simple string matching is used to identify mentions of any Yelp entities on each webpage.

To avoid showing duplicate entities from DBpedia and Yelp and to improve the coverage of identifying Yelp entities, Weaver uses a location-based heuristic to merge entities from the two sources: two entities are merged if 1) they are from different knowledge bases, 2) have overlapping surface forms listed in the knowledge bases, and 3) have geographic coordinates that are less than two kilometers from each other. Using simple string matching to identify Yelp entities on webpages can have limited coverage since Yelp only lists one surface form for each entity (e.g., the name of a restaurant). However, if a Yelp entity was merged with a DBpedia entity, it is automatically applied to mentions of different surface forms as identified by the entity linking

³<https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/>

⁴<https://www.yelp.com/Fusion>



Figure 6.5: To create a missing entity, users can select a phrase (here, Japanese Tea Garden) on the page and see a list of candidates to choose from. In this case, the top 3 candidates were 1) a Custom Card not linked to external knowledge bases, 2) an entity card linked to a specific Japanese garden on both Yelp and DBpedia, and 3) an entity card linked to the general entity for Japanese gardens in DBpedia.

algorithm [158]. For example, once the Yelp entity “San Francisco Museum of Modern Art” was merged with its corresponding DBpedia entity, information from Yelp is automatically linked to its mentions in different surface forms listed in DBpedia, such as “SFMoMA”. Finally, Weaver extracts the paragraphs around entity mentions from each webpage as supporting evidence along with the following structured data from the knowledge bases (Figure 6.2): location on a map, phone number, Yelp and Wikipedia categories, images, short descriptions from Wikipedia, average review scores and number of reviews on Yelp, and 3 Yelp reviews.

While the state of entity recognition is continuously improving, a caveat of driving end-user interfaces with machine learning is potentially having the model make occasional mistakes that can degrade the user experience, and it is crucial to provide mechanisms for the users to recover from them [128, 138, 139]. For example, in a trip planning task, a webpage might miss a popular destination not recognized by the Spotlight algorithm, not listed on DBpedia, or not covered in the top 450 results returned from the Yelp API. To recover from cases where an entity of interest was not recognized by Weaver, the user can still create a custom entity card by first selecting the entity name on the webpage, and click on the “Create Card” button in Weaver (Figure 6.5). In the background, Weaver queries the two knowledge sources for candidates, merges the two results lists using the location-based heuristics described previously, and finally presents the list of candidates from which the user can pick. Alternatively, if the entity was not found in the knowledge bases, the user can still create a custom entity card with no external entity information (Figure 6.5). If a user created an entity card that was linked to DBpedia and/or Yelp entities, all the information that was associated with them will also appear on the user-created entity card. This ensures that users can still save and retrieve information to accumulate what they have learned, even when an entity mention was not automatically recognized by Weaver.

During development, we tested this algorithm on the top 10 Google search results for the query “Things to do in new orleans”. The average time to analyze each webpage was 1.8s ($\sigma=1.3$ s, max 4.3s, ran in parallel). The on-page highlighting of entities utilized MarkJS library⁵ which has

⁵<https://markjs.io/>

a speed comparable to the in-page search feature of Chrome.

6.4 Evaluation

6.4.1 Study Design

The main goal was to explore the benefits and challenges of our entity-centric approach, and how it affected the process of reading, cross-referencing, and collecting information during complex exploratory search tasks. For this, we built a baseline system that also had an in situ interface, but with no on-page entity recognition support nor provided entity cards with rich information and as a structure for saving evidence. Similar to a prior system introduced in [212], the baseline system consisted of a sidebar that can be opened on any browser tabs for note taking. However, instead of allowing participants to create itemized lists of short notes, our implementation allowed participants to type notes and/or copy and paste information into a large text input field (Figure 6.6). As participants typed or pasted information into the sidebar, it is automatically saved and synced across browser tabs in real-time where the sidebar was opened similar to online text editors, such as Google Docs, that are commonly used to supporting online research.

We are aware that Evernote web clipper and Google Doc are common tools based on our informal interviews, but we believed our in-situ baseline to be a stronger baseline for what we set out to measure – whether the Infusion and Diffusion mechanisms can promote gathering and sense-making across multiple sources: 1) Infusion and Diffusion benefits users when both collecting and organizing information, the Evernote clipper only supports collecting and would require significant effort for organizing – it creates a new document for every clip, and requires managing and merging multiple documents in a separate interface to organize them. 2) On the other hand, we think our in-situ baseline closely resembles using a separate plain text document, and the in-situ design lowers the cost of collecting by removing the need to switch between application windows. Further, using a separate document as baseline would introduce an additional variable between the two conditions (in-situ vs separate UIs) when the in-situ design is not part of our core contribution

A lab study was conducted with 24 participants recruited from a local participant pool. The participants ranged from the age of 18 to 43 ($\bar{x}=25.38$, $\sigma=5.53$), 63% were female, 33% were college students, and 38% had a bachelor's degree. To control for task complexity, we used a predefined task to compare Weaver with the baseline system as a between subject condition with each system assigned to 12 randomly selected participants. The study began with a pre-survey for demographic information, followed by 50 minutes of exploratory search for the given task (described below). Finally, participants answered a post-survey and were interviewed about their experiences. During the study, we recorded the actions participants performed interacting with the two systems via either event logging (Weaver) or screen recording (baseline) for post-analysis. Each participant was compensated 15 USD.

The main task of travel planning was designed to test Weaver's ability to support collecting evidence from multiple sources to support decision making between many options and had the following instructions:

You and your friends are going on a trip to New Orleans. Help the group figure out

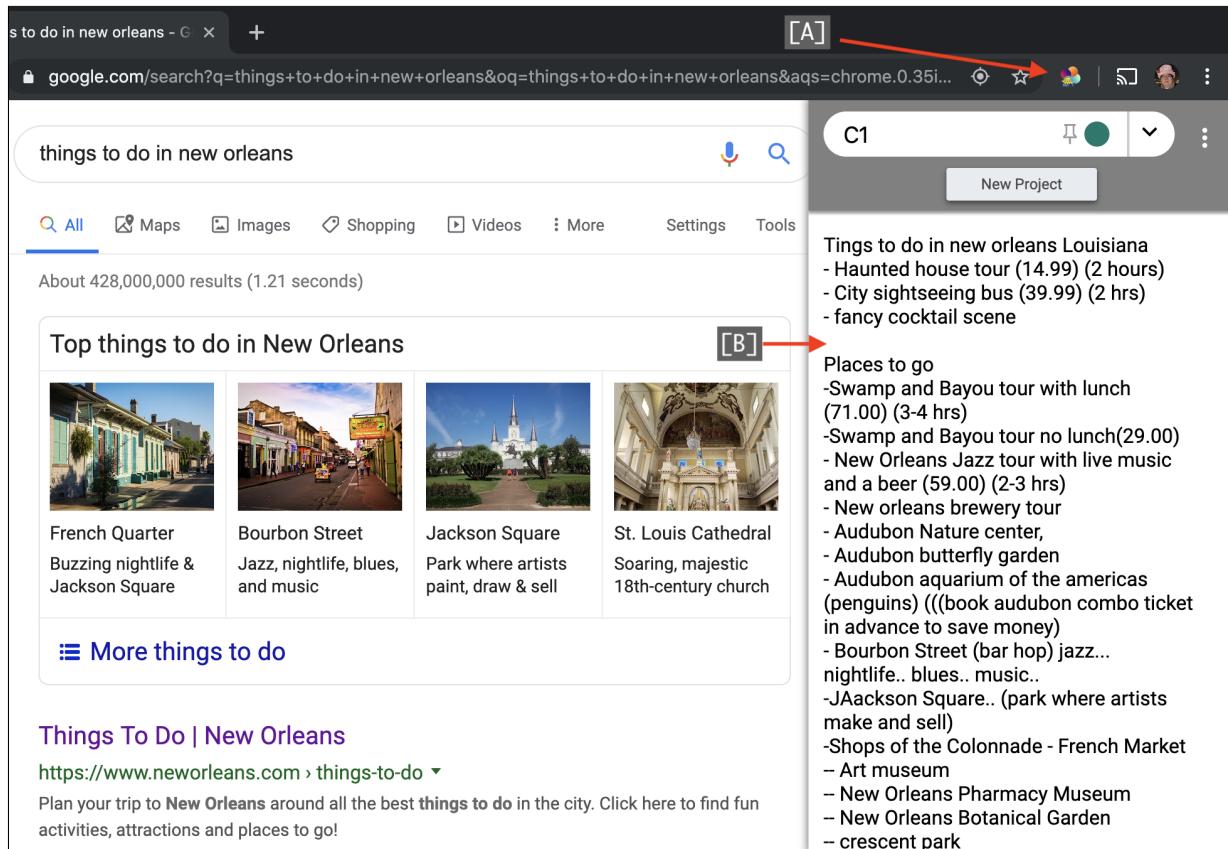


Figure 6.6: The notes of one participant using the baseline system at the end of the study. The sidebar [B] can be opened and closed on any webpages by clicking the extension button [A]. As participants typed into the sidebar, it is automatically saved and synced across browser tabs in real-time where the sidebar was opened.

which places you should go and where to eat during the trip.

We also used the following description as a motivator to encourage the participants to put in more research effort, derived from previous studies of sensemaking:

Imagine after this task you will share you found with your friend(s), along with a short summarization in an email. To convince your friend(s) of your choices, provide enough reasons and information about your choices.

6.4.2 Implementation Detail

Both Weaver and the baseline system were built as add-ons for the Google Chrome browser implemented in JavaScript using the ReactJS⁶ library. The application uses Google's Firestore real-time database⁷ to store mappings between webpages and entities and user-generated notes and clips. To power the entity cards in Weaver, we utilize the Yelp Search API to fetch entities from Yelp, and we use the open-sourced DBpedia Spotlight [158] software in a custom

⁶<https://reactjs.org>

⁷<https://firebase.com>

Behavior \ Systems (between-subject)	Weaver (N=12)	Baseline (N=12)	Independent-samples t-test	
# of webpages the system was accessed from	12.00	$\sigma = 4.81$	5.08 $\sigma = 3.64$ $t(22) = +3.97, p < 0.001^{***}$	
# of webpages evidence was collected from	5.33	$\sigma = 2.58$	3.40 $\sigma = 1.89$ $t(22) = +2.45, p < 0.05^*$	
# of evidence (notes&clips) added to an entity	18.42	$\sigma = 12.65$	6.50 $\sigma = 5.38$ $t(22) = +3.00, p < 0.01^{**}$	
# of unique entities saved in the system	8.92	$\sigma = 6.49$	13.42 $\sigma = 5.94$ $t(22) = -1.77, p = 0.09$	
# of unique entity cards accessed by hovering	29.33	$\sigma = 15.62$	<i>entity cards were not available in baseline</i>	
# of webpages each entity card was accessed from	4.33	$\sigma = 1.40$		
# of times entity cards was accessed by hovering	128.75	$\sigma = 72.11$		

Table 6.1: Evaluation results for our primary domain of Travel Planning – Mean statistics of participant behavior using Weaver and a Baseline system with an in-situ notepad. Results showed participant who used Weaver collected more evidence and from more information sources.

backend that identifies DBpedia entities [11] in webpages and syncs them with the end-user interface through the Firestore service. The studies were conducted using 12.5 inch Chromebooks running Chrome version 69, but both extensions were compatible with other operating systems where the Chrome browser is available.⁸

6.4.3 Results

Our main goal in designing Weaver is to support online sensemaking across different information sources. Weaver supports this using two core mechanisms: *Infusion* that allows users to access evidence about an entity scattered across other information sources, and *Diffusion* that allows users to save supporting evidence about an entity to be resurfaced across other webpages that also mentioned the same entity. Through the two mechanisms, Weaver provides entity cards as a foraging structure where users can collect, reuse, and accumulate evidence from multiple webpages when evaluating a large number of options.

Diffusion: Sensemaking across Webpages

Results from our study in Table 6.1 show participants who used Weaver interacted with the system more frequently than participants who used the baseline system. In addition, they also used Weaver to save nearly 3 times more evidences that were collected from significantly more webpages. On average, participants used Weaver on more than double the number of unique webpages compared to participants who used the baseline system (12.00 vs 5.08; N=12, 12; $t(22)=3.97, p=0.0006 < 0.001^{***}$). In addition to accessing the system across more webpages, participants who used Weaver collected evidence (either by typing notes or copying clips) from more unique webpages. On average, each participant used Weaver to collect evidence from 5.33 different webpages ($\sigma=2.58$), which was significantly more than participants who used the baseline system who collected evidence from 3.40 different webpages based on an independent-samples t-test ($\sigma=1.89$; $t(22)=2.45, p=0.02 < 0.05^*$). These results suggest participants used Weaver to better facilitate sensemaking across multiple webpages in the browser, allowing them to gather new information and access previously saved information across significantly more webpages.

Post-survey and interviews provided insights on how participants used the entity cards provided by Weaver to *diffuse* and *accumulate* evidence across webpages and used them to support decision making:

⁸The Chrome extension APIs are also currently being standardize by the W3C to be used across other browsers:
<https://www.w3.org/community/browserext/>

“The cards were useful because they allowed me to add to things I found on other pages. I could compound things that I had already said and supplement my knowledge to help me arrive at a decision.”

On the other hand, participants assigned the baseline system pointed to the high interaction costs associated with saving evidence to their sidebars. Specifically, they described the tension between manually maintaining a useful organization in the sidebar and capturing all the useful information they had encountered:

“There were times I wanted to like highlight a passage, or save some text from a webpage [but didn’t]... I was sort of using my notes as a TO-DO list, and sometimes I move things around to organize them. But if there are big blocks of text in there it’ll be harder to do that, or even just to look at the list of things I have collected.”

Conversely, participants who used Weaver described lowered interaction costs when saving information to the entity cards, as well as lowered interaction costs to remove all evidence related to an entity that later became irrelevant:

“I liked that I could save entities with notes attached and look back at them all put together. I used to do this with a word doc and links and it wasn’t nearly as easy.”

“They [the entity cards] were very useful... I could save anything, and if I didn’t need it later it was simple to erase them.”

These findings suggest that the entity cards in Weaver were not only used as foraging structures to support sensemaking across multiple webpages, but also allowed participants to collect and synthesize evidence to support decision making with lowered interaction costs. As a result, we also found participants who used Weaver on average collected nearly 3 times more evidence compared to participants who used the baseline system (18.42 vs 6.50; $t(22)=3.00$, $p=0.006 < 0.01^{**}$).

Infusion: Engaging with Entities during Browsing

Traditional entity-centric approaches required users to interact with entities only during the retrieval stage, such as presenting entity cards in search engine results pages. While this approach improves efficiency for factual information needs, such as looking up the weather or the address of a restaurant, for complex exploratory search tasks, users have to consider many more opinions, facets and features about an entity, usually from a number of webpages. Our entity-centric approach explores the benefits of integrating rich entity information to stages beyond retrieval when conducting complex exploratory search tasks.

Our results show that Weaver participants were frequently engaged with the entity cards throughout the study. While participants in both conditions saved multiple entities to keep track of the different options that they were interested in, Weaver participants were actively evaluating potential options with information *infused* from other webpages and external knowledge bases. Participants who used Weaver frequently accessed the entity cards by hovering over highlighted entity mentions on each webpages. On average, each participant who used Weaver hovered over entity mentions 128.75 times (Table 6.1; $N=12$, $\sigma=72.11$). We examined the number of entities participants saved in the two conditions. In Weaver, participants save entities by “hearting” an entity cards which also saves all the entity attributes that was on the card. In the baseline system, participants typed or copy-pasted the names of different entities into their sidebar, and

would sometimes manually collect attributes that were readily available for participants who were using Weaver (such as addresses and Yelp review scores). At the end of the study, each participant on average saved 8.92 entities to Weaver ($\sigma=6.49$), which is slightly lower than participants who used the baseline system (13.42, $\sigma=5.94$), but the difference was only marginally significant based on an independent-samples t-test ($t(22)=-1.77$, $p=0.09$).

Even though participants who used Weaver did not save more entities to their workspace, they hovered over an average of 29.33 unique entities ($\sigma=15.62$) to access their entity cards throughout the study. In addition, participants used Weaver to access the same entity cards while browsing different webpages. On average, each unique entity card was accessed from 4.33 different webpages ($\sigma=4.33$, $N=12$). This further indicates that participants who used Weaver not only collected evidence across multiple information sources for the options that they saved, they also relied on information from multiple information sources *infused* to the entity cards to evaluate many potential options before saving them into their workspaces.

Post-survey and interviews showed that participants in both conditions relied on multiple sources to decide which entities to save in their workspaces:

“There are like clear definite yeses that I would save immediately, but there’s also a lot of maybes. For the maybes if I see something multiple times and they might get added.”

Participants who used Weaver found value in the evidence automatically *infused* from other webpages and external knowledge sources. Participants cited using information on the entity cards to verify uncertain or potentially biased information on the page that they were reading:

“They [the information on entity cards] allowed me to see if the comments on this page was true”

“The cards were useful because I can know the exact condition other than the advertisements provided by their own company [referring to content on an official listing page].”

These results show that entities served as options in the travel planning task, and that entities mentioned in text represented a useful structure for foraging across webpages. By identifying them in the browsers, participants were able to use the entity cards to keep track of interesting options and organize their notes and evidence collected from webpages about them.

Participants who used Weaver also pointed lowered interaction costs for managing additional browser tabs. They cited that the entity cards allowed them to better evaluate options encountered on the current webpage without the need to create additional searches and managing additional browser tabs:

“I liked seeing similar material for an entity [referring to clips from different webpages] and being able to continue research without having to do an actual search.”

“They [the entity cards] were very useful when I needed to pull up information on something I tagged [saved], without the need to use multiple [browser] tabs...”

These responses suggest that information *infused* in the entity cards can help participants more confidently evaluate newly encountered options and used the entity cards to validate information presented on the current webpage. Prior work pointed to the high cost of context switching for note taking can break the linearity of documents, and be disruptive for reading and consuming

information [43, 167, 208]. Specifically, [153] found their participants intend to investigate other articles referenced by the current one, but avoided doing so in order to avoid disrupting their flow of reading of the current article. The above responses in particular, suggested that the lightweight cross-referencing powered by the *infusion* mechanism can potentially address this issue.

Surprisingly, participants also mentioned discovering and navigating to useful information sources from examining information on the entity cards, which was not our original design intention:

“[It was useful when] I was taken to mentions on sites I would not have thought of”
“[entity cards] Give you info about other websites that might be useful later on.”

These results showed that participants were actively engaged with the entity cards in the Weaver system, and that bringing entity support beyond search results pages to support active reading and note taking can also be beneficial for users using multiple information sources to conducting online sensemaking. In addition, participants also found value in the mentions of entities *infused* from multiple webpages, using them as context to evaluate both their entity options as well as potentially biased information on the current webpage.

6.5 Discussion

6.5.1 Limitations and Generalization

In this work, entities were used as a proxy for potential options in complex sensemaking tasks. This allowed us to exploit state-of-the-art entity linking algorithms to automatically identify and disambiguate entity mentions in plain text across webpages and users’ notes. While we believe entity-centric search can cover a wide variety of tasks – past work has shown entity focused queries account for the majority of search traffic [86, 142] – there are still scenarios where potential options can be topical or descriptive. For example, potential options for “How do I get my tomato plants to produce more tomatoes?” could include *fertilization*, *pruning*, and *providing proper support* as shown in [44, 88], which can potentially be supported using micro-interaction driven structuring approaches [44, 101].

With the ability to recognize options for entity-based tasks, Weaver focused on allowing users to gather and synthesize evidence across webpages about each of their options. On a higher level, Weaver provided a simple category structure for organizing multiple saved entity cards, but participants also pointed to the need for further synthesizing their collections of cards in the later stages of sensemaking:

“Weaver is helpful in the first stage of information collection, but when it comes to the final detailed plan of the trip, I still need more place for editing, adding specific time and so on.”

This suggests that more detailed artifacts that leverage these entity-centric cards, such as a calendar for itineraries, could be an promising next step.

6.5.2 Conclusion and Future Work

In this paper we introduced Weaver, a novel approach for weaving together information about entities that were scattered across the Web to support complex sensemaking in the browser. By

presenting *in situ* entity cards, users can both verify potentially biased information on the current webpage with evidence *infused* from other sources, as well as using the entity cards as foraging structures that can *diffuse* their notes across browser tabs and be resurfaced automatically as they become relevant. In a lab study with 24 participants, we compared Weaver to a baseline system with no entity support as a between subject condition. We observed that participants using Weaver gathered nearly 3 times more evidence from 60% more webpages, both significantly more than participants who used the baseline system. Post-interviews revealed how they utilized Weaver to verify information they encountered, to accumulate evidence across webpages, and to synthesize them to support decision making.

We think entity-centric approaches have the potential to support sensemaking in a wide variety of domains that involve collecting evidence and deciding between options. However, the cost of adapting the current framework to support different domains is unclear, especially for domains where high quality knowledge bases are not readily available. In the post-survey, we asked participants if they could think of other search tasks that may benefit from using Weaver, and participants pointed to a variety of different tasks, including making a purchasing decision, essay writing, event planning, literature review, job searching, and deciding on a college major. Here we describe scenarios to illustrate how the infusion/diffusion framework might work for two of these tasks.

- A researcher can create a “paper card” (using Microsoft Academic Graph for metadata) to externalize ideas as she reads a paper and create additional cards for ideas relating to cited papers, providing a foraging and ideation structure during literature review where papers may have overlapping citations.
- A consumer can create “product cards” for different cameras (using online shopping APIs for pricing info and DBPedia for specs) to keep a “short list” as they compare different options by collecting both objective and subjective info in reviews relevant to her personal needs. In this case, users can also configure Weaver to focus on only cameras entities to improve linking and disambiguation performance.

Finally, we also see promise in a number of other research directions. One such direction would be automatically summarizing evidence gathered across different websites for an entity instead of simply listing them. This could help Weaver scale to much larger projects with many sources, while keeping gathered information easy to consume for the users. However, surfacing information sources in the summary so users can better evaluate source trustworthiness is still an open problem. While Weaver supported synthesizing entities and evidence into categories, providing support for creating different structures (e.g., tables or essays) warrants further investigation.

Our results have implications for the design of future intelligent browser interfaces that can better understand the information being consumed by their users, and building novel interactive systems for supporting online sensemaking. While there are already popular commercial browser add-ons that allowed users to collect information from webpages with lowered efforts (such as Evernote Web clipper), almost all existing tools require users to switch to a separate workspace to access saved information. Our findings pointed to benefits in better integration between collecting information from webpages and accessing them. As phenomena such as fake news and shill reviews have demonstrated, there are significant drawbacks to the easy availability and generation of online content. Interactive systems that can provide additional context to users *in situ* may become increasingly necessary to help navigate the information overload. Anecdotal

evidence for this need can also be seen in the rise of aggregation-based sites such as Metacritic or Wirecutter, which act as virtual meta-analyses of evidence and opinions but fail to take into account the personal context of the user and their goals. We believe that this work presents a step forward in illustrating a design space for interactive systems which can take advantage of advances in machine learning and natural language processing to help end users actively gain context and personalize their online sensemaking experience.

Chapter 7: Mesh

Scaffolding Comparison Tables for Online Decision Making

This chapter describe a third and final system in this dissertation that focus on supporting individual online sensemaking tasks. The two previous chapters each focused on providing global context relating to *options* (Weaver in Chapter 6) or *criteria* (SearchLens in Chapter 5), and discovered important insights about users: 1) Users need to keep track of the different options that they were considering, and evaluate them under the global context (Chapter 6); and 2) Users often identify criteria from data and need to evaluate their options based on different criteria. Combining these insights, this chapter focus on a system that can provide holistic support for product comparison, allowing users to both keep track of their options and criteria, as well as keeping track of their personal interpretation of data to scaffold their decision making process.

While there is an enormous amount of information available online for making decisions such as choosing a product, restaurant, or school, it can be challenging for users to synthesize that information into a good decision. Information for users' many different criteria needs to be gathered from many different sources into a structure where they can be compared and contrasted. The usefulness of each criteria for differentiating potential options can be opaque to users, and evidence such as reviews may be subjective and conflicting required users to interpret each under their personal context. We introduce Mesh, which scaffolds users in iteratively building up a better understanding of both their criteria and options by evaluating evidence gathered across sources in the context of consumer decision making. Mesh bridges the gap between decision support systems and exploratory search, changing the cost structure to provide an increasing payoff with greater user investment. Our lab and field deployment studies found evidence that it significantly reduces the costs of gathering and evaluating evidence and scaffolds decision making through personalized criteria allowing our participants to gain deeper insights from data.

7.1 Introduction

Whether figuring out which products to purchase or where to eat in an unfamiliar city, consumers today have instant access online to enormous amounts of information on which to base their decision. However, users could also be overwhelmed by the number of options, the criteria they should use to compare the options, and the number of information sources to collect evidence from [192, 197]. For example, the electronics section of Amazon alone contained more than 1.3 million reviews in 2013 [154], and Yelp has accumulated more than 200 million reviews [201]. Research in consumer behavior has found such reviews to be a major factor in online decision making [84, 165]. Online information has the potential to help consumers make more informed decisions about how well each option satisfies their various criteria [65]. For example, a coffee drinker looking to buy a new espresso machine might read reviews aiming to evaluate how easy it is to use for a novice barista, how well it steams milk, how likely it is to break down, and so on. On the other hand, online reviews can also be messy, biased, subjective and scattered across many sources [50, 96, 177, 232], requiring users to interpret each piece of evidence based on their personal context [194]. The highly bimodal skew of review ratings can lead to compression

☰ Entryway Light Fixture		KINGSO Rustic Pulley Pendant Light One...	Globe Electric 65979 Sansa 3-Light Semi-Flush...	AXILAND Truelite Industrial Metal Spherical...	Progress Lighting P350039-020 Briarwood Close-t...
OPTIONS		 4.17 / 5.00	 3.27 / 5.00	 4.25 / 5.00	 2.58 / 5.00
> WEIGHTED OVERALL	↑ ↓ ⓘ				
ADJUST			 Don't think this is long enough for what we...		
SIZE	x1				
BRIGHTNESS	x1			 Review Tally 4	
DIMENSIONS	x1		 Maybe too small...		
INSTALLATION	x1	 Set criteria importance: x1 x5 x10  Adjustable	 Review Tally 2	 Adjustable, Size seems good Review Tally 1	 Dimmable
STYLE	x1	 Don't know if the style will match the space Review Tally 2	 Review Tally 1	 Adjustable, Size seems good Review Tally 5 2	 Small and doesn't hang- probably won't work
+ NEW CRITERIA		  pulling information from... a KingSo Rustic Pulley Pendant Light One Li...	  pulling information from... a Globe Electric 65979 Sansa 3-Light Semi-...	  pulling information from... a Truelite Industrial Metal Spherical Pendan...	  pulling information from... a Progress Lighting P350039-020 Briarwoo...

Figure 7.1: The Table View. Users can create Option Columns by importing Amazon project pages opened in their browser tabs, and create Criteria rows to see the average review ratings that mentioned each criteria across their options (in yellow). To see explore the reviews more deeply, users can click on the criteria to see the Evidence View (shown in Figure 7.3), where users can overwrite the default Amazon ratings with their own (in purple) based on their own interpretation of data. To prioritize the criteria, users can also adjust the weight to see an weighted average rating across their criteria for each option. This image is an actual project made by P05 in the field deployment study.

of ratings in a narrow band [100], and the increasing number of fake reviews (which now may be in the majority for some categories such as electronics and beauty [78]) mean that solely relying on automatic aggregation such as average ratings or summarizations can be risky or uninformative. Automated approaches to addressing these issues, such as aspect extraction [146, 229], review summarization [99, 140], and direct recommendation [30], can be insufficient due to the long tail of usage contexts [21], the need for nuanced contextualization when reading reviews [46], and the challenge of discovering and learning new criteria along the way.

Consumers doing this task manually must go through the various reviews and information sources, pulling together scattered information, learning about what criteria are useful for picking or ruling out options, evaluating evidence on those criteria, keeping track of their judgments, and weighing them depending on what's most important to them to make a final decision. To assist with the process, consumers utilize techniques such as building comparison tables with spreadsheets or notepads. However, transferring information between information sources and spreadsheets or notepads can be prohibitively time-consuming [6]. Furthermore, as a user encounters and adds new options, they must gather information for each of their criteria in the table in order to evaluate that feature. Similarly, encountering and adding new criteria requires gathering information for all previously added options. This iterative construction is common in new domains [150] and creates an increasing cost the more options and criteria in the table.

Instead of fully automated or manual approaches, we introduce Mesh, a hybrid approach aimed at scaffolding decision making by helping users progressively build up a comparison table that reflects their personalized criteria and evaluations of evidence. Mesh lowers the cost of pulling information into the table, organizing it by the user's criteria, and helping users keep track of their judgments as they evaluate evidence. Importantly, by autofilling rows and columns when new criteria or options are added throughout the process, Mesh makes adding to or changing the table stay at a constant cost as the table grows, changing the cost structure to provide an increasing payoff with greater user investment. Finally, Mesh helps keep users on track by prioritizing where to look, which criteria are most important, and reflecting their current beliefs for each option through an overall weighted average.

We evaluated Mesh through three user studies. We found evidence that Mesh lowered interaction costs, and allowed participants to find answers to objective criteria, such as the size and capacity of coffee machines, significantly faster and more accurately. For subjective criteria, such as the ease of use of coffee machines, participants were able to produce summaries of what they learned in 20 minutes that were rated more insightful when compared to baseline participants who used Google Spreadsheets to conduct the same task. Finally, our field deployment study found participants were able to conduct their own tasks over a prolonged period of time, and described their process as organized and efficient allowing them to make confident purchases based on their own research.

7.2 Related Work

Research in consumer behavior has pointed to the difficulties users face when using online evidence to support making purchase decisions. One major challenge is that online evidence, such as consumer or expert reviews, can be subjective, biased and messy [78, 165]. Furthermore, users may need to go through each piece of evidence in order to interpret them based on their own personal context and unique goals. This process is an important factor in purchase

decision making [84, 165], but can incur high cognitive costs as the user tries to keep track of and summarizing each piece of evidence. Another major challenge is that online evidence is often scattered across many information sources due to the distributed nature of the Web. This includes product listing pages on ecommerce platforms, blog and forum posts, and consumer and expert reviews. On the one hand, having multiple information sources can help users to determine the credibility of online evidence [50, 96, 177, 232]. However, cross-referencing multiple sources can be burdensome and costly [23, 153, 167, 208].

Significant research effort has been focused on using computational approaches to support this process, using the wealth of available data for training machine learning models. One thread of research is focused on making direct product recommendations by either finding patterns from the purchase history of many users [30] and/or the description and metadata of products [144]. Personalized product recommendations can potentially shortcut users' research effort but also face issues with accuracy and explainability [26]. Another thread of research is focused on reducing users' effort when consuming large numbers of reviews by aggregation, such as review summarization and common criteria (aspect) extraction from reviews [78]. However these approaches also face issues with user trust, for example from fake reviews polluting the aggregated results or failing to support the long tail distribution of less common user needs [21]. Furthermore, current approaches fail to support the qualitative process of developing personalized context from subjective data that is vital to human decision making. Not unrelated, recent research has found reading consumer reviews to still play an important role in online purchase decisions [84, 165], even though recommender systems are now widely available on ecommerce platforms.

Another thread of research has focused on building interactive interfaces that aim to support decision making under multi-criteria and multi-options scenarios, such as faceted interfaces [92, 190] and table-based decision support and visualization systems [56, 143, 178, 199]. While these approaches allow consumers to narrow down their options efficiently by navigating to different subsets of a larger collection or investigate trade-offs between them through visualization, the majority of these work rely on pre-compiled metadata or required users to manually clip evidence across different sources. As a result, they do not support accessing criteria that require close examination of a large amount of subjective evidence (such as reviews) that are not in the form of structured metadata. For example, to get a sense of how durable an option is would typically involve consumers evaluating many unstructured reviews describing whether and how an item held up over time. In two studies closely related to our work, [51, 52] allowed users to build comparison tables for camera products by allowing them to pick from a list of precompiled common camera criteria and used sentiment analysis of relevant reviews as summaries across different options. While Mesh also allowed users to build comparison tables with their own options and criteria, in our system users can use arbitrary search terms as their criteria instead of selecting from a common list of criteria, allowing it to support the long tail distribution of user needs [21]. More importantly, Mesh focuses on helping individual interpret reviews under their own personal context, and overwrite the summaries generated by the system to better reflect their own views of data. This approach not provides better support for personal context, but can also allow users to recover from errors made by automated summarization approaches.

Instead of automating away the role of the user, our approach focuses on helping them scaffold their decision-making throughout the process, maximizing their ability to apply their personal context and interpretation to evidence while reducing the costs for doing so. This view unlocks

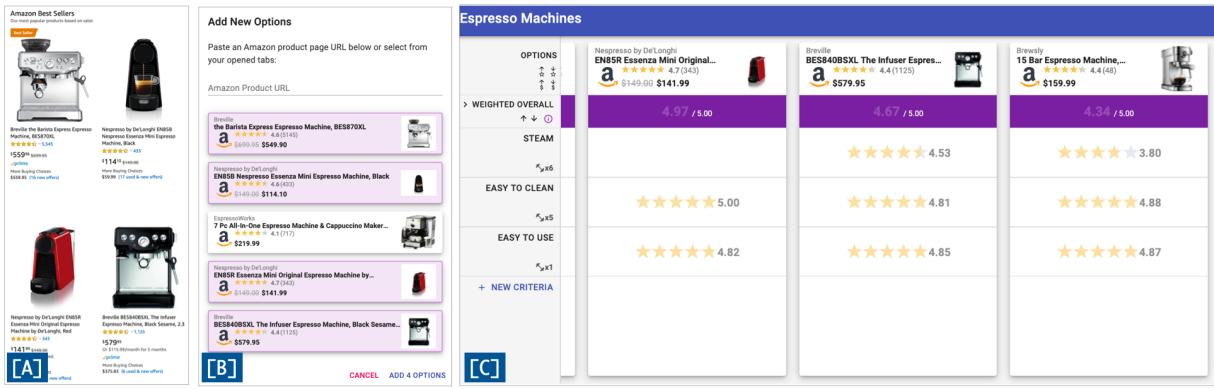


Figure 7.2: Many products on Amazon are highly rated with thousands of views and it can be difficult for users to differentiate them [A]. Users can open them in browser tabs and import them into Mesh to keep track of them [B]. Mesh automatically fetches reviews relevant to different user criteria for each option to help characterize them [C]. Users can uncover meaningful discrepancies between options based on their own criteria. For example, here seeing a larger differences in the “Steam” criteria, with the first option that lack this feature returning no reviews that mentioned “Steam”.

a design space in which the interface supports the human in discovering and sharpening their own understanding of what criteria are important to them in the context of the options and evidence available to them; keeping track of their evaluations of that evidence for them; enabling the human to prioritize their attention to the most discriminative evidence; capturing human perceptions of value; and using those perceptions to drive a final decision that integrates values across their personal criteria. At a high level our work aimed to bridge the gap between decision support research in the literature above (which helps people make decisions by imposing a high degree of structure based on metadata or through computation) and the sensemaking process in which users are learning about the unknown unknowns to develop personalized context from unstructured data [150, 186].

7.3 Exploratory Interviews and Design Goals

To discover the limitations and needs of common online product research practice, we conducted preliminary interviews to inform our design goals. We recruited 30 participants (age: 3% 19-24, 20% 25-34, 33% 35-40, 23% 41-54, 20% 55+; 22 female, 7 male, and 1 not listed) by posting on social media including Facebook, Twitter and Nextdoor, and interviewed each of them for 60 minutes. Prior to the interviews, we generated 10 interface design mock-ups addressing various potential issues discussed in the previous sections, ranging from managing information sources to collecting evidence for purchasing decisions. We discuss these design probes in context of our discoveries below. During the interviews, we walked through each of the design mock-ups and used them as probes to see how strongly participants identified with the issues they tried to address, as well as how they reacted to the designs. We list below three of the most common recurring themes.

Comparing Options with Scattered Evidence

The most common theme mentioned by all participants was the difficulty of managing an overwhelming number of information sources and the amount of evidence scattered across them. Specifically, they pointed to how evidence for options needs to be collected across different web-pages, leading to a *stressful number of opened browser tabs* of ecommerce websites (such as Amazon) and expert review websites (such as CNET reviews). When comparing options, participants were especially frustrated by the high interaction cost of *switching back and forth between tabs to compare options on a metric [criteria]* and that it is *not easy to search for [information that mentioned] specific terms across all products*.

Need for Personal Interpretation of Evidence

Consistent with prior work, we also found reading reviews to be an important factor for supporting online shopping [84, 165]. At the same time, participants felt overwhelmed by the amount of evidence they needed to process in order to confidently make purchase decisions, but were unenthusiastic about designs centered around automating the process. For example, one design had users answer questions about their preferences and provide personalized product recommendations. Participants were reluctant to be convinced by an automated system, but instead saw it as a way to *get some ideas or guidelines about things they should consider*; in other words, they saw it as an additional source for collecting potential options to conduct further comparison. Participants further emphasized the importance of seeing raw evidence and making their own judgments such as *reading through reviews to generate a summary of their own opinion*. Participants were enthusiastic about features that would support this process, such as allowing them to easily rate and tally reviews as positive or negative, or making a summary rating from reading multiple reviews.

Scaffolding Decision Making

Participants pointed to difficulties in keeping track of their overall research, describing their process as “*erratic*” causing them to “*go down many rabbit holes*” and “*get lost in the weeds*.” One main reason is the need to constantly make small and personalized judgments throughout such as interpreting how relevant a review is to their personal context, summarizing how a product fits a criteria, or deciding to keep or rule out a potential option. Participants were frustrated when “*Sometimes I can't remember why a [product] page was kept opened and had to reread the content*.” Currently, participants used spreadsheets, scratchpads and physical notebooks to address this *when things start to get out of hand*, but they also pointed to how this process is cumbersome and only used as a *last resort on important purchases*. When asked about the types of information they would typically save, participants described a mix of factual findings (such as product specifications) and their own interpretation of subjective evidence (such as ease of use as described in the reviews). Participants were enthusiastic about designs that would scaffold them in working in a more organized fashion, such as making a comparison table of options they are considering and with their own criteria and notes and being able to compare options side-by-side and ranking them.

Based on these findings uncovered through the interviews, we formulate the following design goals:

- **D1:** Minimize effort of comparing evidence for the same criteria across different options

- **D2:** Allow users to make their own interpretation of data and summarizing them
- **D3:** Capture users their decisions about options and criteria throughout the process in an organized way

7.4 System Design

Motivated by the design goals uncovered by our exploratory interviews, Mesh focuses on providing a more organized way to conduct research by allowing users to iteratively build up a product comparison table with their own options and criteria.

In a standard spreadsheet, people have to start with a blank table and switch back and forth between information sources to fill out everything manually. In contrast, our system provides an increasing payoff for every criteria and option added by connecting each cell in the table with relevant product information and reviews and summarizing them. One challenge here is that automation and auto-summarizing content goes against the user needs for personal interpretation, so we have carefully constructed interactions that allowed users to both deeply explore the raw evidence and adjust their tables when the auto-summerizations do not fit their own interpretation of data. To support this, Mesh was designed to capture users judgments about data throughout their process with little added effort using different light-weight interactions at different levels of granularity. For example, 1-click to annotate how relevant a review was, summarize how does all the evidence describe a specific criteria for a one product, or click to sort options and see what the trade-offs were. Altogether the system is designed to feel like scaffolding: helping users gain deeper insights from scattered evidence more efficiently, and capturing their own judgements on data in a structured way.

Mesh was implemented in TypeScript with the React library for building UI components and Google FireStore as the database. The full version of the system was implemented as a Chrome extension. Unlike a hosted webpage, this allowed Mesh to make cross-domain requests for fetching evidence from different information sources. Importantly, implementing Mesh as a Chrome extension enabled it to interact with browser tabs, allowing its users to import tabs into Mesh to build a collection of potential options with lowered effort.

Example User Experience

A user wants to purchase an espresso machine for the first time to use in her apartment. She starts by searching on Amazon for popular options to consider, but sees that they all have more than 1000 reviews with average review scores between 4.4 and 4.7, making it difficult for her to tell them apart (Figure 7.2 [A]). To understand which is best for her she needs to dig into the reviews to see which are easy to clean, compact, has great steam for making cappuccinos, and don't require a lot of cleaning – a process that would typically take her hours. Using Mesh she creates a new project and imports the Amazon product pages she had opened in her browser tabs (Figure 7.2 [B]), with the system pulling in their prices, images, and titles. She then adds her criteria to the system as rows, the system automatically fetching reviews for each product that mention it and displaying their average rating in the table (Figure 7.2 [C]). She sees that despite the overall ratings being indistinguishable between her options, there are large discrepancies in review ratings for "steam". She clicks on its row to quickly see reviews mentioning it for all her products in the Evidence View, including one that had no reviews matching it (Figure 7.3). Closer examining the images of that model, she realized it does not support steaming milk,

allowing her to remove it from her project. As she reads a review and processes it in her mind as to whether it is relevant to her, it takes her little extra effort to tally that review as positive or negative and quickly builds up her judgment for each option, replacing the average Amazon rating with her own when it does not reflect her view. She iteratively adds her other criteria, the system autofilling each of them for all her existing options, and as she does so she finds and adds more options, the system autofilling all their criteria as well. At this point she has developed a good understanding of what criteria are important to her goals and discriminative across her options, and changes the weights of her criteria so that the system produces overall scores that are reflect her personal opinions and goals in the Table View (Figure 7.1).

7.4.1 [D1] Comparing Evidence across Options and Sources

Our first design goal was to lower the costs of managing many information sources and examining evidence scattered across them. A fundamental problem we identified was that users often need to compare evidence for a criteria across their different options, but the evidence was typically organized by options and scattered across sources. For example, a user may need to go through multiple Amazon product pages and CNET reviews to get a sense of how different espresso machines were suitable for novices. One way users currently deal with this is by switching back and forth between browser tabs and searching for relevant evidence on each page, and another is to focus on one product at a time and try to remember information from other sources to compare them. Both these strategies can incur high interaction and cognitive costs. As a result, our exploratory interviews found participants had difficulties in keeping track of previous decisions such as which options they were considering, why they considered each in the first place, and the set of criteria they wanted to use for comparing them.

To provide better support for this process, Mesh allows users to progressively build out a product comparison table to keep track of their options, sources and criteria. To keep track of their options and sources, a user can import their browser tabs into Mesh, and group the sources into Option Columns in Mesh (See Figure 7.1 for the Table View). For example, a user could create an option column with an Amazon product page grouped with an expert review article from CNET.com for the same product and its product specification page from the manufacturer website. In the backend, Mesh populates the header of each column with product names, prices, images and review ratings from Amazon. To keep track of their different criteria, a user can create a set of Criteria Rows (Figure 7.1). When a criteria is added, for each option Mesh fetches 60 Amazon reviews by scrapping Amazon's review search API as well as sentences in the product description and imported sources that mention the name of the criteria as evidence. To compare these evidence across options, users can click on each Criteria Row to see all the evidence for their options side-by-side for comparison in the Evidence View, reducing the high cost of switching between information sources (Figure 7.1). Longer reviews are by default collapsed to three sentences surrounding where the criteria name was mentioned so users can stay focused on the current criteria, but can also be expanded when needed for additional context.

In addition, Mesh shows the average rating of the 60 reviews as cell values as a quick overview in the Table View. Our rationale for presenting criteria-specific ratings was to provide users with instant feedback and benefit for externalizing their criteria, which would enable two novel interactions: 1) getting a quick overview of how existing options differ or how a new option compares to existing options and 2) comparing how discriminative their different criteria are for their current options. These have the potential of allowing participants to better prioritize their investigation

efforts. One major challenge here is that while the reviews did mention the criteria, they can often be noisy and include comments on things other than the criteria users were focused on. An alternative design was to use existing sentiment analysis models and analyze sentences around where the criteria was mentioned. During the design phase, we conducted an empirical analysis where we created gold-standard labels for sets of reviews that mentioned specific criteria names. Results suggested that searching reviews based on criteria did retrieve mostly useful results. We also found the average star ratings represented good overall summaries over the reviews, and we did not find significant differences against two modern sentiment analysis techniques we tested [4, 103]. We therefore choose the current design because it is transparent and easy to understand for users. Additionally, we were concerned about whether users would perceive the reviews we retrieved as needing to be exhaustive (instead of the top 60) and perfectly accurate (which was not technically feasible) or otherwise would not trust the system. We instead found that people perceived the reviews as a sampling of the distribution about that criteria, and we did not receive any requests for automated summaries of the rest of the reviews as we initially expected. We believe this further accentuates the importance of personalized evaluation of evidence over an exhaustive aggregation.

7.4.2 [D2] Interpreting Evidence based on Personal Context

Both our exploratory interviews and prior work pointed to an important user need for interpreting evidence based on their own personal context [194], but also pointed to difficulties when trying to keep track of users' judgements throughout the process. For example, how each review mentioned their criteria, their summarization after reading multiple reviews, and how they characterized each option on different criteria. Mesh addressed this using a table structure of options and criteria to scaffold users' research process. Using the Evidence View, where evidence about a criteria were presented side-by-side for their different options, users could externalize their interpretation of data at different levels of granularity using interactions that required little cognitive effort. For example, after examining a review, it only required one click for users to label it as positive or negative using the buttons at the end of each review. As users rate the reviews, Mesh automatically creates a tally of positive and negative reviews for each option, providing immediate payoff to the users for labeling them and reducing working memory load. After examining a couple of reviews about a criteria for one option, users could either leave the average Amazon ratings if they matched their own perceived ratings, or overwrite them with their own ratings (color coded in purple instead of yellow). This can potentially reduces the cost of rating to zero when the default ratings generated by the system matched users' own judgements. In addition, users can also provide additional context through notes, which are shown in the Table View in either case. Based on user feedback, users can also use checks and missing symbols (Figure 7.1) for criteria that had binary value (e.g., does the espresso machine come with a steam wand).

7.4.3 [D3] Scaffolding Decision Making

As a user iteratively builds up a better understanding of their options and criteria, they gradually progress from investigating and interpreting evidence to making a decision between their options. However, participants in the exploratory interviews described spending redundant effort when they lost track of prior judgements about options and had to revisit webpages and reread their content to remind themselves what they liked and disliked about an option. When using Mesh, participants can see all their previous judgements in the Table View presented as cell

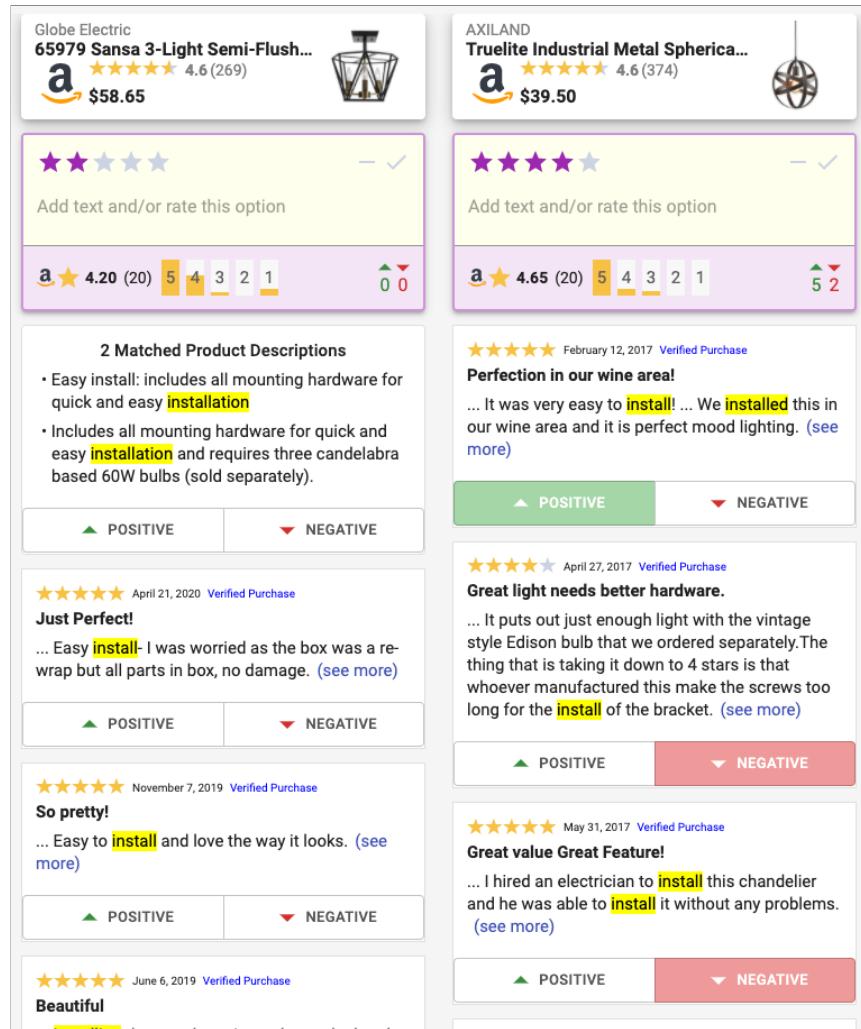


Figure 7.3: The Evidence View. Users can see evidence that mentioned the criteria side-by-side for their options. To capture their interpretation of evidence, users can also label reviews to build a tally or overwrite the average ratings with their own if they do not reflect their views. This is an actual project made by P05 in the field study.

values in each Option Column. This includes the review tallies and their own ratings and notes about each criteria. This allowed users to have an “birds eye view” of their research, seeing which criteria and options contained their own ratings and notes, decide what to focus on next, as well as seeing trade-offs between the options when making purchase decisions.

Participants in the exploratory interviews also described “analysis paralysis” when reaching the decision stage, in which many of their options looked similar on the surface (i.e., highly rated and based on hundreds of reviews) and that it can be difficult for them to see clear trade-offs on multiple criteria for their options. Mesh also provided several affordances for users to interact with their options to explore the trade-offs between them towards making purchase decisions. Firstly, Mesh computes an overall rating for each option by averaging ratings for its criteria. When averaging, Mesh will use users’ own ratings when available, and default to the average Amazon review ratings otherwise. Given that participants in formative studies mentioned the importance of different criteria having differing weights in their decision making, the system also enables them to specify the weight for each criteria which correspondingly alters its impact on the weighted average (e.g., a 5x weight will be counted 5x towards the weighted average more than the default 1x weight). ¹ We also supported “soft” prioritization by enabling users to freely reorder rows and columns via drag-and-drop, for example, to move the most promising options or criteria to the top or the left without altering the overall score. Finally, Mesh could also sort options based on individual criteria ratings or the overall ratings when users click on the sort icon next to the criteria names. This allowed users to quickly explore the best and worst performing options based on their criteria.

7.5 Evaluation

We conducted three studies that focused on exploring the following research questions, respectively:

- The usability of our implementation and the benefits of gathering and presenting evidence across sources. (Study 1)
- Does Mesh enable users to gain more insights from data compared to Google Spreadsheets? (Study 2)
- What are the longer term effects of deploying Mesh to users conducting their own personal tasks? (Study 3)

The first two studies were controlled studies comparing Mesh to a baseline condition using pre-defined tasks to control for task complexity. Participants were recruited from Amazon Mechanical Turk who had more than 100 accepted tasks with above 90% acceptance rate and lived in countries that primarily spoke English. Due to the limitation running Mechanical Turk studies, we could not install Mesh on their computers as a Chrome extension. We therefore deployed the same system as a hosted webpage for participants to interact with. The third study is a field deployment in which participants installed Mesh on their own computers (as a Chrome extension) and conducted their personal tasks over a longer period of time. Participants for the field study were recruited from the local population primarily by posting to discussion boards on NextDoor, a neighborhood based social media. We used video conferencing and screen sharing softwares

¹Details of this calculation is explained to the users via a hovered tooltip. Checks and missing symbols were counted as 1 and 5 stars, respectively.

to assist with the installation process and to conduct two rounds of interviews.

7.5.1 Study 1 - Usability Test and Interaction Costs

The main goals of our first study was to verify in a controlled environment the usability of the Mesh and to test if the mechanism of automatically pulling in evidence from different information sources can allow users to work more efficiently and find more accurate information. For this, we compared Mesh to a baseline variant as a within subject condition where evidence was not automatically pulled in and used objective criteria that had gold-standard answers so we can measure the accuracy of participants' responses. During the baseline condition, participants can use any strategies based on their own product research experiences, such as searching for keywords on Amazon product pages and/or use search engines to find more sources. In order to measure how effective participants were in finding the right answers, we used fixed product options (i.e., 5 popular espresso machines on Amazon) and objective criteria.² One of the authors compiled the gold-standard answers before running the study. Almost all answers were obtained from the manufacturer's website (such as in specification tables and downloading PDF user manuals), with a few resorting to using expert reviews (namely photos or videos that showed a measurement of the portafilters).

The goal of the main task was to find the correct answer for each criteria for the given options. The criteria cells for the first options were filled out to serve as an example. The study was broken down into two segments, and participants worked on two of the four remaining options during each segment with a different condition (counterbalanced for order). During the Mesh condition, evidence was gathered from Amazon reviews and product descriptions, as well as the top two product review webpages returned from Google when searching with the product names appended with the term "reviews". Links to the same sources were also presented during the baseline condition. During the study, we recorded the time each participant spent in the two conditions as well as their responses. After the study, we used the NASA-TLX survey to collect their perceived workload for each of the two conditions. A total of 24 participants were recruited from Amazon Mechanical Turk (age 21-68 M=36.8; SD=10.5; 15 males and 9 females). Each participant was compensated 3 US dollars for an average of 24.9 minutes (median=22.7, SD=8.3).

Study 1 Results

Results suggest that the 24 participants performed the given task more efficiently when in the system condition than when they were in the baseline condition. Comparing Mesh with the baseline, participants completed their tasks faster when using Mesh that gathered evidence automatically across multiple sources (7.2 vs 10.8 minutes; $t(23)=2.6$, $*p=0.017<0.05$ based on a paired t-test). At the same time, they found information that were more accurate based on gold-standard answers (85.0% vs 58.3%; $t(23)=-5.31$, $***p=2.18e-05<0.001$ based on a paired t-test). Combining the two metrics we estimated a x2.30 increase in efficiency, where participants were finding 2.23 correct answers on average each minute when using the full Mesh system, compared only 0.97 correct answers per minute on average when using the baseline variant (based on a paired t-test: $t(23)=4.18$, $***p=0.00036<0.001$).

In addition to speed and accuracy, participants also perceived the process to have lowered

²Dimension, Does it have a built-in grinder, Water tank size, Does it use a solenoid valve and Portafilter size

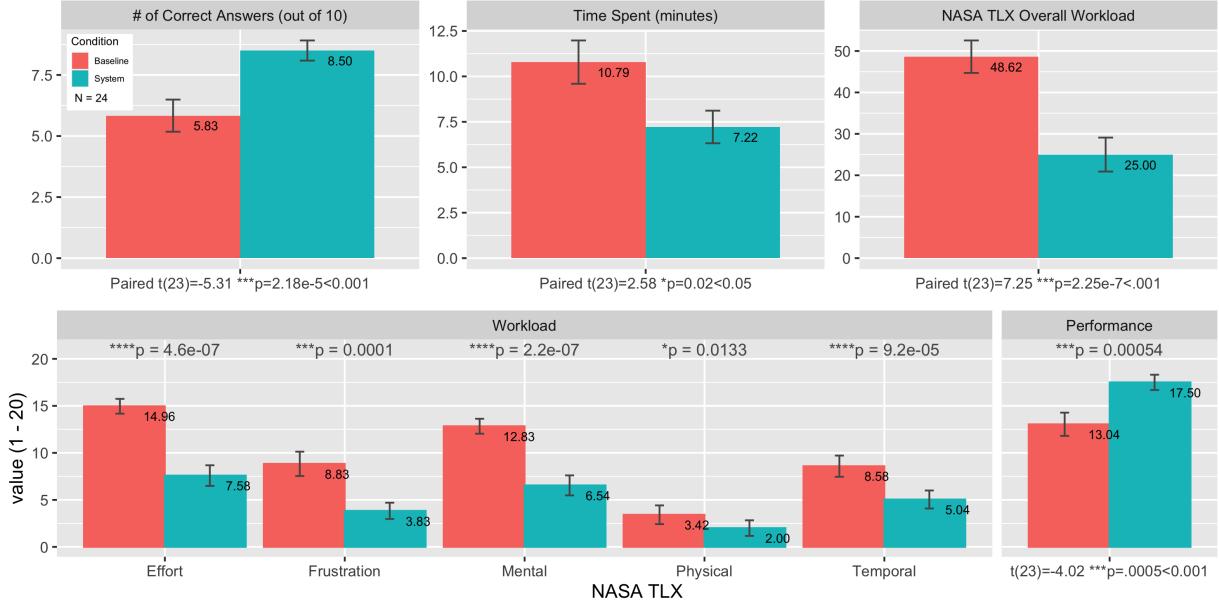


Figure 7.4: Mean statistic of how participants performed under different conditions. Participants who used Mesh were finding more correct answers using a shorter period of time. In addition, they also had lowered perceived workload based on the NASA-TLX survey.

workload when using the full system across effort, frustration, mental, physical and temporal demands based on the NASA-TLX survey (Figure 7.4, combined: 25.0/100.0 vs 48.6/100.0; $t(23)=7.25$ *** $p=2.25e-7<0.001$ based on a paired t-test) as well as increased perceived performance (17.5/20.0 vs 13.4/20.0; $t(23)=-4.02$, *** $p=0.0005<0.001$ based on a paired t-test). This suggests the interface of Mesh can reduce interaction costs when dealing with objective criteria when compared to the baseline where participants relied on their current process, even when they had to learn a new interface.

7.5.2 Study 2 - Interpreting Subjective Evidence

While the first study tested the usability and interaction costs of Mesh when working with objective criteria with gold-standard answers, Study 2 focused on how Mesh can support users when investigating criteria that required subjective and potentially messy and conflicting evidence such as consumer reviews[96]. Unlike looking up the product dimensions in the product description for a coffee machine, investigating its ease of use may require users to read through multiple relevant reviews to get a sense of how previous consumers agreed or disagreed on the criteria while considering how each review fits their personal context. For example, a user buying a robot vacuum who lived in an apartment with wooden floors might down-weight reviews from people who lived in a big house with high pile carpet. For this, we carried out a second study that focused on whether Mesh can provide benefits when researching these types of subjective criteria.

To compare Mesh with people's existing approach, we used Google Spreadsheets as a between subject baseline. We chose this baseline because it is a common tool for consumers building product comparison tables, and that it is an easily accessible hosted service with APIs that allows

Name	Amazon Link	Stuck	Loud	Dog Hair
Roborock S4	https://www.amazon.com/dp/B07TXGQS3H			
iRobot Roomba 614	https://www.amazon.com/dp/B00IPEZKBW			
eufy by Anker	https://www.amazon.com/dp/B07DF9GVK9			
iRobot Roomba 960	https://www.amazon.com/dp/B01ID8H6NQ			
iRobot Roomba 675	https://www.amazon.com/dp/B07DL4QY5V			

Figure 7.5: The initial spreadsheet for the baseline condition.

us to dynamically create a spreadsheet for each crowdworker. To control for task complexity and the personal preferences of participants, we used the following persona and task description for researching 5 robot vacuum cleaners with the 3 subjective criteria in bold:

John is looking to buy a robot vacuum for his house. The most important thing for him is that the robot vacuum **doesn't get stuck too often**. It is also important that it is **not too loud**. He also has a dog, so it would be nice if it's also **effective cleaning up dog hair**.

John already narrowed down to 5 final options. Spend around 20 minutes to build up a comparison table to help John research the best option and explain to him why you think it is the best option.

The five options were all popular models on Amazon that had more than 1,000 reviews and above 4 average review ratings (as of April 15, 2020). In both conditions, we populated their tables with the predefined options and criteria to maximize the time participants spent on exploring and learning from data instead of copying and pasting information from the persona (see Figure 7.5 for the baseline template).

A total of 48 unique participants from Mechanical Turk for the main study. In which 22 (age 31-58, M=38.7, SD=9.6) were randomly assigned to use Mesh and the remaining 26 participants (age 30-70, M=34.7, SD=11.3) used Google Spreadsheet. Each participant was instructed to conduct the above task for 20 minutes using their assigned systems. We assumed participants in the baseline condition to be already familiar with a spreadsheet interface, and gave a brief tutorial (13 sentences and 6 screenshots) to participants assigned to use Mesh. To capture what participants had learned during 20 minutes of research, we asked them to pick one of the options that they recommend and write a short summary for John explaining their choices. This design allowed us to capture the mental models of participants under different conditions through mentions of detailed evidence and how they reasoned and compared the different options, and has shown to be effective for evaluating sensemaking support systems in prior work [117, 166]. Workers who participated in the previous study were excluded from this study to prevent learning effects. Each participant was compensated 3 US dollars.

To compare summaries collected from the two conditions, each summary was rated by 5 additional crowdworkers. Crowdworkers who participated in the Study were excluded to ensure summaries were not rated by the participants who wrote them. In each rating task, crowdworkers first read the same persona used in the study and one of the summaries. Crowdworkers then rated the following statement using 7-point Likert scales for agreement (a score of 7 indicated a strong agreement, a score of 1 indicated a strong disagreement), and we averaged the ratings across 5 workers as the final ratings:

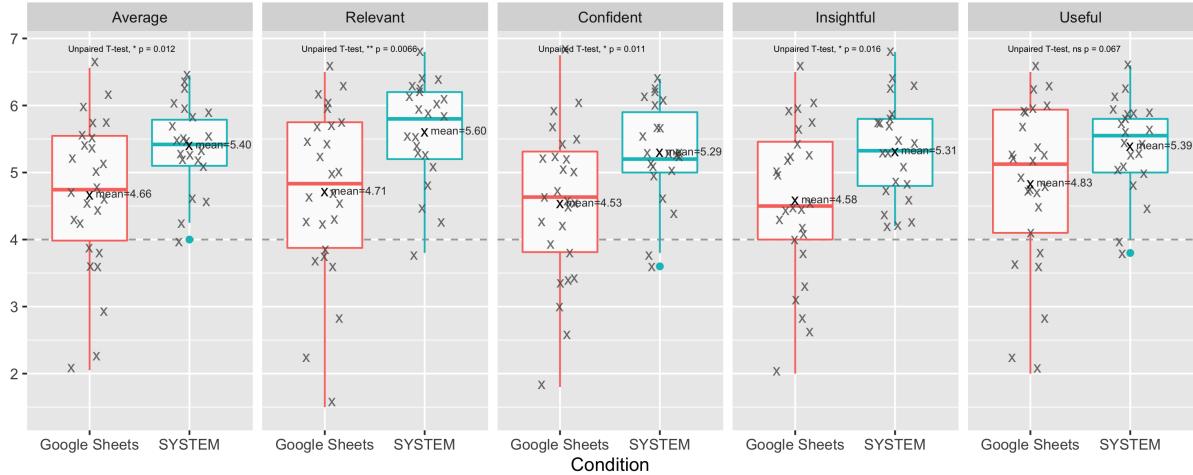


Figure 7.6: Using a 7-point Likert-scale for agreement where 4 is neutral and a MANOVA, we find that participants who used Mesh generated learning summaries that were rated significantly higher compared to participants who used Google Sheets for the same task on the combined dependent variables of relevance, confidence, insightfulness, and usefulness ($F(4, 43)=2.64$, $*p=0.047 < 0.05$).

- I find the summary to be *useful*.
- The summary is *relevant* to the scenario.
- The summary is *insightful*, containing details that may be hard to find.
- I feel *confident* after reading the summary.

The four statements were designed to compare the summaries across conditions on the following aspects: The first statement of usefulness aimed to measure their quality to account for collecting qualitative responses on crowdsourcing platforms [122]. The second statement measured whether participants who used Mesh were able to focus on criteria described in the persona and generate summaries that were more relevant. This is due to the fact that participants in our fact-finding study described their current process as “*a rabbit hole*” and how it can be difficult to “*focus on criteria that really mattered*.” The third statement measured how detailed and insightful the summaries were, an important aspect of consumer review helpfulness identified in a prior work [165]. Finally, the fourth statement aimed to explore whether information in the summaries can support decision making by measuring if they induce confidence.

Workers were paid 0.25 cents for reading the persona and rating summary based on the four statements above.³

Study 2 Results

Figure 7.6 shows the differences between the 22 summaries written by participants using Mesh and the 26 summaries written by participants using Google Spreadsheets for the same task. Averaging across the four aspects, participants who used Mesh generated summaries that were

³Workers read an average of 124.0 words for each task (range: 50.0-220.0, SD=44.4) and the estimated reading speed of English speakers is 200-300 words per minute [195]. Assuming the lower-bound reading speed of 200 words per minute and 15 seconds was required to answer each of the four Likert-scales. Similar to approach in [102], we estimated the average hourly pay rate to be around 9.26 USD.

ID	Tasks	Minutes Spent	# of Sessions
P01	Snow boots. Gourmet cat food.	70	3
P02	Backpacks. Pajamas as a gift for his/her sister.	134	3
P03	Bread machine. Hair cutting kit.	150	4
P04	Running shoes. Printer for learning material for kids.	82	6
P05	Entryway light fixture. Toy play-set for kids.	131	5

Table 7.1: Usage statistics about participants in the field deployment study based on the activity logs. This includes the tasks they conducted using Mesh, as well as the total number minutes they spent using the system and the number of sessions over the deployment period.

rated higher than participants in the baseline condition (Figure 7.6, mean 5.40 vs 4.66). We used a MANOVA to correct for multiple comparisons and found statistically significant difference ($F(4, 43)=2.64$, $*p=0.047<0.05$) between the conditions on the combined dependent variables (relevance, confidence, insightful and usefulness). Below we show two typical summaries from each of the conditions collected after 20 minutes of product research:

Baseline condition: I would pick the Roborock S4 after considering the 3 categories [criteria] that are important to him: how often it gets stuck, noise, and ability to pick up hair. Unfortunately, all of the models he picked do have a tendency to get stuck, which makes it difficult to choose when just using the three factors [criteria]. However, the Roborock was the only model I found, where there weren't many complaints about it being too loud. Additionally, the Roborock is able to pick up dog hair, according to the product description and user reviews.

System condition: It seems that while all options do tend to get stuck from time to time, the reviews that the Roomba 675 does somewhat better in that regard. Additionally, many reviews for the Roomba 675 stated how well it picks up pet hair, which was another important consideration that differentiated the Roomba 675 from other options. The Roomba 960 may be marginally better but it costs \$200 more and so I didn't think it was worth the extra expense. Lastly there were reviews that found the noise of the Roomba 675 to be acceptable.

While many participants who used Google Sheets mentioned the similarity between options and the difficulty of the task, people who used Mesh point out how they differentiated the options on the given criteria based on multiple pieces of evidence.

7.5.3 Study 3 - Field Deployment Study

While the first two studies provided quantitative measures on how Mesh affected learning, efficiency, accuracy and perceived workload when participants were given predefined tasks, we also conducted a field deployment to further investigate the longer term effects of Mesh when participants performed their personal tasks in the wild. Five participants (age: four 25-34 and one 35-40; two females, two males and one non-binary) were recruited by posting to 5 local neighborhood discussion boards on NextDoor (a neighborhood based social media website). The posts contained a link to an online screener survey, and we used the responses to recruit to people who have used a spreadsheet for online research in the past (49.4%, N=89) and prioritized people who had any Chrome extensions installed (49.4 N=89).

We interviewed participants for one hour at each the beginning and the end of the deployment.

Before the initial interview, we asked each participant to email us 1 to 3 upcoming online purchases to ensure they have a real task to work on during the initial interview. At the start of the first interview, we collected their demographic information and assisted with installing Mesh as a Chrome extension on their computers via screen sharing. Each participant then proceeded to perform a think aloud session for around 30 minutes using Mesh to conduct one of the tasks they had proposed. After the first interview, participants continued to use Mesh on their own for the same tasks and/or create new tasks. Based on their availability, we interviewed each participant again after 1-2 weeks. Participants shared their screens and retrospectively walked us through their projects while we probed on their experiences, strategies and issues they had encountered during the deployment. Table 7.1 shows the tasks each participant conducted and the total amount of time they spent using Mesh based on log data. The first tasks in the table were ones created during the initial interview and the rest created during deployment. All 5 participants completed the study and were each compensated a 50 US dollars Amazon gift card. The interviews were video recorded and transcribed for analysis.

Study 3 Results

Following an open coding approach based on grounded theory, the first author went through the 10 hours of recordings and transcriptions in three passes, and iteratively generated potential categories from the dialogue until clear themes emerged [49]. Throughout the iterations, inputs from the rest of the research team were also incorporated, including other researchers who also conducted interviews. Our key findings are presented below.

Efficient and Organized

In general, participants responded favorably to using Mesh in the field for their personal tasks when asked to compare Mesh to their current online product research process (i.e., using spreadsheets and/or notepads). Specifically, all participants pointed to lowered interaction costs when using the Evidence View to access evidence gathered across information sources to compare their options, as well as lowered cognitive costs from being able to rule out options confidently based on evidence.

It is much better than a spreadsheet... I like that I can really quickly add something and it just pulls in all the information, the picture, the price, and [evidence for] all of these different criteria and presents it in a way that is really easy to do comparison across products. I'm able to delete things easily so that I can reduce my cognitive load as I go through my decision making process. - P03

All participants described how Mesh allowed them to take a more organized and structured approach when managing multiple information sources and foraging evidence from them. Specifically, P01 and P02 described how the linear structure of browser tabs can be inefficient when trying to find evidence for a specific criteria across browser tabs for different options. While participants also pointed out that the mechanisms provided by Mesh can technically be performed manually, they also acknowledged that the interaction costs of managing many browser tabs and filtering for relevant information to support their criteria amongst them would be prohibitively high in practice.

In theory I could do all this myself but it would take 10 times [as] long so I would never do it well. I would say is it technically possible? Yes. But would any person ever do this [manually] for themselves? ... It's nice to have a more organized and systematic

approach... Instead of something that right now is very linear. If I pulled up a bunch of boots in different tabs and searched [in] each of them for reviews with the word boar. It's really boring and not a particularly efficient way to look at information. - P01

One participant (P03) pointed to how having a more organized scaffold to their processes, Mesh provided better support task resumption, allowing them to make progress on their overall tasks even with shorter sessions.

I loved being able to come back to this [referring to one project]. It's something we hadn't done in our initial sessions that became so much better when I was using it on my own. I couldn't say, hey, I've got 15 minutes to kill. Let me do some more searching, and then I could say, okay, gotta go to my next meeting. - P03

Post-analysis based on activity logs also found all participants conducted their product research in three to six sessions (Table 7.1).

Prioritizing Effort on Discriminative Criteria

We found two major usage patterns for the criteria-specific average Amazon ratings, both centered around being able to discover discrepancies between participants' options. All participants saw immediate value when the average Amazon ratings populated automatically for their options when they added a new criteria. Allowing them to get an initial overview of how the evidence differs between options for a criteria. Specifically, participants described trying out different criteria as a way to surface meaningful differences (i.e., based on their own criteria) amongst their options. Since participants typically only considered options that were popular and highly rated on Amazon, they described their options as virtually indistinguishable without Mesh:

Having never purchased it before I literally have no idea what to buy. And so this [task] is what I tried to do [with Mesh] and it's actually like super helpful because [otherwise] every single stupid cat food on Amazon just like looks identical... So, it was really helpful especially [with] this picky criteria. - P01

Alternatively, when participants added a new option to a project that had existing criteria, Mesh automatically populated Amazon average review ratings across different criteria for the new option. Participants used this mechanism to quickly characterize new options and see how they fit with existing options based on their own criteria:

But now, you put this kind of data in front of me in this quick hit fashion. Then I go, wow! They're pricey [comparing a newly added option to existing ones], and yet anybody that mentioned cost [a criteria] has given it the full rating. They're more durable [referring to discrepancy between options in the average ratings for a criteria] You know, I could see tangible evidence now. And that makes me want to go – Maybe that's the pair. - P04

Seeing discrepancies between options also influenced participants' process by prompting them prioritize their effort on investigating criteria that had large discrepancies between their different options:

Okay, there wasn't a great difference here in terms of ink [a criteria]. Let me go into what I weighted as more important, and it's this air printing [another criteria] capability... for this middle one [referring to one option], rated pretty poorly... These two have pretty good ratings. So then I went in and started looking [at the evidence]

- P04

By focusing first on criteria that were more discriminative amongst the options, participants could rule out options that compared less favorably earlier to shortcut their process. All participants described prioritizing their options in the system, either by reordering their options via drag-and-drop, or rule out options completely by removing them. One participant in particular, described a sense of relief and progress when removing options in Mesh.

I don't feel like I would delete things [options] in a spreadsheet. Whereas here it actually feels good to delete it [an option], because I'm like, Great! I've decided that I'm not going to deal with it. - P03

Scaffolding Decision Making

Participants also described how Mesh supported deep exploration of individual pieces of evidence in the Evidence View that laid out evidence for specific criteria across their options.

The second thing that I think is really great for me was the ability to dive into the reviews for specific criteria. It's really nice to be able to open this [the Evidence View] up and have it filter out for all of the products, so I can make this comparison across products. - P03

When we introduced the system, we explicitly explained to participants that the average ratings were based on review scores and could be influenced by parts of a review not relevant to their criteria even though the reviews mentioned the criteria. Participants were able to work with this limitation, and replaced the Amazon ratings with their own when they do not reflect how they wish to characterize the evidence. In addition, participants also described creating ratings as a way to keep track of and aggregate how they personally interpreted evidence and saw benefits in how changing criteria ratings was reflected in the overall weight score of each option.

I would say in the event that I was going to differ from what's in front of me, I would rate [the criteria]. - P04

Once I start to make decisions on things, like I put my thing [own ratings and notes] in there and say: Okay, this is what my rating is. And now it starts to change the overall ratings, so it would help me make a better decision based on what I think. . . . like, the tool thinks this is a really good value, but maybe I think this value is not enough for me and it's a two, because I just think it's two - P03

All of the participants also made actual purchases during the deployment based on their research performed with Mesh and expressed confidence in their decisions. This suggests that their tasks represented real world user needs, and our participants were able to use Mesh to conduct research for a prolonged period of time and use it to support making their final purchase decisions.

7.6 Discussion and Limitations

While all participants' initial responses were positive when adding options and criteria to the Table View, some of them find their first impressions of the Evidence View to be overwhelming. While this suggested a higher learning curve for the Evidence View, all participants were able to complete research with it for their own tasks during the deployment.

So initially it was like, Whoa, there's a lot going on here. It's a lot of text but I'm kind of over it once I understood what was going on. Now I'm like, Okay, cool. Let's take a look at this [referring to the criteria] across the things [referring to the options] - P03

More commonly, participants express desire to extract evidence from online sources other than Amazon. While in the current implementation does support extracting evidence from other sources through pasting in their URLs to the appropriate option, participants pointed to two limitations: 1) extracting and tracking price changes across ecommerce platforms other than Amazon and be notified, and 2) extracting from listicles and forum posts that discussed multiple products:

Running shoes are kind of discipline specific. There are other sites solely for this [type of] product that I would go to. [To add a webpage and] track the options to use globally would be cool. But like robot vacuum there's nowhere else [but Amazon] I'm going. Unless I'm tipped off that Target or Bed Bath and Beyond happened to have an incredible sale. - P04

While price tracking could be implemented within Mesh, there are multiple commercial solutions available ⁴ and we considered it outside the scope of this work. On the other hand, extracting information from sources containing evidence about multiple options presents an interesting research challenge of computationally identifying mentions of products and extracting descriptions about them from text. Finally, in our design we relied on the users to generate the list of options and criteria. While recommender systems or aspect extraction techniques could also be incorporated into the system to support this in the future, many were already commercially available on ecommerce platforms which also potentially explain why this was not a salient issue in the interviews.

We introduced Mesh, a novel system where users to build up product comparison tables connected to information sources as a way to scaffold their research. This design is novel because it introduced a new process that scaffolds the building up of context and changes the cost structure from increasing cost to increasing payoffs as the number of criteria and options grow. Through three rounds of lab and field studies, we gained deeper insights into how Mesh can benefit online sensemaking in the context of product comparison research. We found evidence that Mesh not only lowered the interaction costs (i.e., faster and lowered perceived effort) by pulling in information across sources about objective criteria (e.g., the water tank capacities for different espresso machines), but also allowed participants to find more accurate information (Study 1). When dealing with subjective criteria (e.g., ease of use for different espresso machines), we found evidence that participants who used Mesh were more insightful and confident about their choices compared to participants who used Google Spreadsheet in the baseline condition (Study 2). We also tested Mesh in the wild with participants conducting their own tasks over a longer period of time. Post-interview revealed insights on how Mesh allowed them to prioritize their investigation on criteria that were discriminative amongst the options they were considering, enabled them to keep track of their progress by capturing their own interpretation of data, and allowed them to make confident purchases based on their findings (Study 3).

⁴<https://camelcamelcamel.com/> and <https://www.joinhoney.com/>

Chapter 8: Conclusion

At the beginning of this document, I set out to explore the following thesis:

Using interaction and visualization techniques, we can dynamically provide global context that matches users' evolving intentions throughout their exploration of large unstructured datasets. Supporting this will allow users to gain deeper insights from data and make better decisions with lowered efforts.

In the first half of the dissertation, I investigated this thesis in the domain of crowdsourced sensemaking, where both the requesters and crowdworkers each only saw a small portion of data but needed to create globally coherent and consistent structures. In the second half, I extended insights from providing global context to crowdworkers and requesters to support individual online sensemaking where the amount of available choices and evidence are often well beyond an individual's capacity to process them. Five systems were described in this dissertation including crowd-based systems that were able to produce coherent and consistent categories and labels for complex datasets, as well as systems for supporting individuals in online sensemaking tasks allowing them to gain deeper insights from data with lowered efforts. By evaluating the five systems described in this dissertation, I present evidence supporting the above thesis statement listed below as take-aways.

8.1 Take-Aways

8.1.1 Users Expressed Intents to Explore Global Context

Instead of taking a top-down approach such as aggregation or summarization, this dissertation explored ways to support users in the bottom-up exploration of individual items to iteratively refine their goals and evaluate them under the global context. One of the core challenges here is the extra interaction costs for users to express their evolving goals to the systems. For this, one common interaction pattern used in this dissertation was allowing users' to discover and externalize key concepts from data by picking out keywords from data. This allowed the systems a way to provide immediate payoff via searching and summarizing relevant items in the dataset and presenting them to users as context about the concepts.

One instance of such interaction was the *sample and search* pattern in Alloy (Chapter 3) where crowdworkers first identified key categories in the sampling phase, and then highlighted keywords in the text snippets to further externalize their categories to the system. The system in turn presented search results of other items in the dataset that mentioned the keywords, allowing crowdworkers to evaluate both their categories and keywords under the global context. Evaluation results suggest that through *sample and search*, crowdworkers were able to identify coherent categories at the right abstraction levels, as well as picking out discriminative keywords to group items under them. In the second half of this dissertation, I explored how this pattern can be used to benefit individuals conducting online sensemaking tasks, and found direct evidence

that it can incentivise users to express their intents more richly to the system. Specifically, in Chapter 5, SearchLens allowed users to collect sets of weighted keywords from reading reviews to represent their different interests. The system in turn generated visual explanations in the search results to provide context based on users' current interests. I found direct evidence that participants expressed their interests in SearchLens using significantly more keywords when compared to a baseline variant that does not support the visual explanations. Less directly, in Chapter 7, one of the most valued features from the Mesh field deployment study was that it allowed participants to discover criteria from reading reviews and externalize them to see the average ratings of reviews that mentioned the criteria across their different options.

In sum, while traditional information retrieval studies have shown difficulties in getting users to express search goals more richly, these results support the thesis statement of this dissertation by showing that accessing global context is important when trying to make sense of large and unstructured datasets, and can be used to incentivise users to externalize their evolving interests for their own benefits.

8.1.2 Harnessing User's Ability to Identify Good Keywords

One core assumption made by Alloy (Chapter 3) was that crowdworkers are able to identify useful keywords from data during *sample and search* so the system can search for other items to put under the same categories. This interaction was designed in a way to harness crowdworkers' existing proficiency in figuring out good query keywords from data based on their past experiences with online exploratory searches. For example, allowing crowdworkers to freely change their highlighted keywords and update the search results in real time to create a familiar experience of query reformulation based on search results [112]. I found direct evidence for the above assumption by comparing two of the baseline conditions in Chapter 3 – clustering the text snippets using all words as features versus using the same clustering algorithms but using only keywords highlighted by the crowdworkers as features. Results showed that crowdworkers were highlighting discriminative keywords that lead to clusters that better matched with the gold-standard categories. I extended these insights in two of the individual sensemaking support tools introduced later – SearchLens (Chapter 5) and Mesh (Chapter 7), where users identified keywords in the reviews that were representative of their different interests and externalize them to the systems to see an immediate feedback. This allowed users to both evaluate their findings under the global context, and also quickly refine its keywords representation. In sum, since users were already proficient in figuring out good search keywords from data based on their past experiences, we can leverage this ability to both capture their current intents as well as search across the datasets for evidence about their intents to support global context.

8.1.3 In-Situ Global Context Promotes Learning Deep Insights from Data

Based on an observation from the exploratory interviews in Chapter 7, many of the system designs in Chapters 5 to 7 were influenced by how individuals often have two separate structures that they needed to maintain when conducting complex search tasks. The first is their foraging structure consisting of their different searches and webpages opened in their browser tabs. The second structure is their evolving mental structure that they either kept in their minds or externalized to a separate interface, such as a notepad or a spreadsheet consisting of options they were considering, criteria they use to compare them, and their evaluation of these options based on how they interpreted data. However, maintaining the two structures and transferring informa-

tion between them can be cumbersome for users involving cross-referencing information across browser tabs, copying- and pasting information, and re-finding previously saved information.

To bridge this gap, one common approach used in this dissertation is an in-situ design that allows global context to be presented on-demand and embedded into users' exploration process. Specifically, users in Waver can access relevant information across their different browser tabs as new options are discovered; and, users in Mesh can group sets of tabs together into options and compare relevant evidence across different options when a new criteria is discovered. Results presented in Chapter 6 and Chapter 7 showed that these in-situ designs supported the thesis that by providing global context based on users' interests can lead to users learning deeper insights from data. Most directly, in Chapter 6, we found quantitative results that participants assigned to used Weaver collected evidence from 60

In sum, online sensemaking support tools that are available today largely treat the two structures independently – Tab management browser add-ons focused on helping users in managing information sources more efficiently, and personal information management interfaces focused on supporting users in organizing the collected information. However, the process of gathering and cross-referencing evidence across information sources in the foraging structure and transitioning them into the sensemaking structure is poorly supported. This dissertation explored a design space that bridges the foraging structure, where information is divided by their sources, and the sensemaking structure, where scattered information needs to be synthesized. Evidence showed that this allowed the systems to provide context around key concepts (i.e., options and criteria) as they were discovered by the users, allowing them to gain deeper insights from data with lowered interaction costs (Chapter 7).

8.1.4 Providing Global Context in Microtasks for Crowdworkers

Providing global context is a fundamental issue for complex crowdsourcing tasks because each crowdworker is limited by the scope of microtasks and therefore typically only see a small portion of the entire datasets. This lack of global context could lead to crowdworkers creating incoherent categories under the global context based on their local views of data. In Chapter 3 I focused on this core problem in the task of creating globally coherent categories (i.e., clustering) by introducing the *sample and search* interaction. Unlike prior work that tried to provide global context with larger but still fixed sets of items to each crowdworker, Alloy instead instructed crowdworkers to repeatedly sampled from the entire dataset until they were confident that they had found four items that belong to different categories to build up global context. The trade-off made here was that each crowdworkers in Alloy actually started with less items compared to previous systems in order to offset the additional effort of learning global context through sampling (4 items in Alloy vs 10 in [9] and 8-10 in [57]). Evaluation results suggested this to be a favorable trade-off that led to structures that were more coherent when compared to a prior system [57] and with around a third of the monetary costs. While I showed preliminary success in using *sample and search* to provide global context when categorizing collections of text snippets, expanding *sample and search* to create more complex structures in microtasks remained an interesting future research direction. For example, expanding Alloy to create topic models for longer articles that contained sections that would belong to different categories.

8.1.5 Structuring Global Context with Crowdworkers for Requesters

This dissertation also explored a second fundamental issue of complex crowdsourcing tasks that were less discussed in the literature: Requesters also need support for global context since they typically turn to crowdsourcing for its ability to scale to large datasets in the first place and do not have the capacity to examine the data fully. For complex data, this lack of global context could lead to microtasks that were not well defined to cover the long tail of edge cases in data leading to incoherent labels.

The second system, Revolt (Chapter 4), focused on this problem in the context of labeling items using predefined categories but without comprehensive guidelines from the requesters. Revolt introduced the interaction pattern of *vote-explain-categorize* that shifts the effort of requesters trying to generate comprehensive labeling guidelines based on limited understanding of data to the crowdworkers who distributedly explored all items in the datasets. This is achieved by harnessing disagreement between crowdworkers to identify items that were ambiguous under an incomprehensive guideline, and allowing crowdworkers to create new categories to capture the ambiguous cases. This process is analogous to the *representational shift* loops in the sense-making framework for individuals proposed in [186] but do so in a distributed fashion. Here the requesters generated the initial schema (i.e., the predefined class labels) which is handed off to the crowdworkers to explore the individual items to identify *residues* that does not fit the initial schema, and expand the working schema by categorizing the residues.

Fundamentally this distributed sensemaking process can allow crowdsourcing to not only scale and process large numbers of items, but also use crowdworkers to scale to more complex datasets by distributedly exploring the dataset to capture and report back cases unknown to the requesters who had limited global understanding of data. Evaluation results suggested that the crowd can generate consistent structures through *vote-explain-categorize* that can lead to more consistent training labels while eliminating the need for requesters to create comprehensive guidelines. In addition, while traditional crowdsourcing systems typically treated disagreements as noise from crowdworkers doing a poor job, Revolt instead showed that it can be valuable signals for capturing ambiguity in microtasks and structuring them.

8.2 Design Pattern

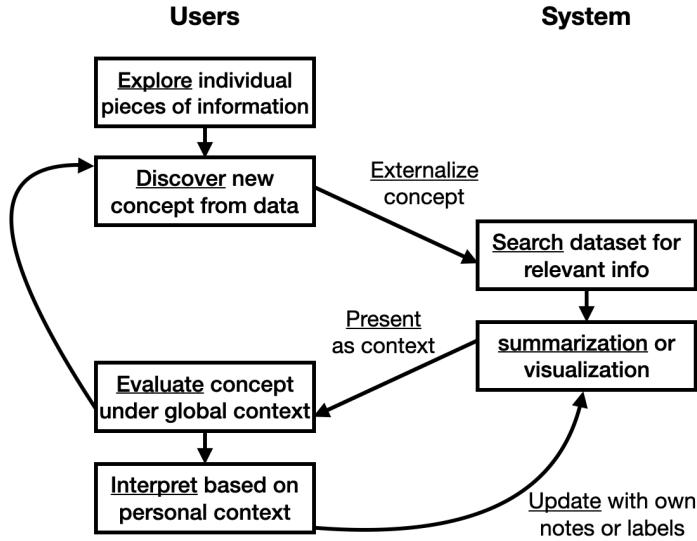


Figure 8.1: A general system design pattern.

In general, the systems described in this dissertation followed the above sensemaking design pattern that aimed to provide better support for global context during bottom-up exploration of large and unstructured data. The framework starts with users **exploring** individual pieces of information in the dataset. For individuals, this is typically the initial queries when conducting online exploratory searches with Yelp, Google or Amazon as in Chapters 5 to 7, respectively. Similarly, the crowd system Alloy (Chapter 3) simulated this process by allowing workers to repeatedly sample from the datasets to iteratively build up a better understanding of the space of information. As users process the individual pieces of evidence, they **discover** concepts that were important to their task. In the case of Alloy, this would be crowdworkers identifying categories to for organizing items in the dataset; in the case of SearchLens and Mesh, this would be individuals identifying criteria from a restaurant or product review that fits users' personal interests; and in the case of Weaver, this would be users identifying a restaurant or travel destination on a webpage.

Since the users often discovered useful concepts from examining a single piece of information, it can be difficult for them to evaluate such concepts under the global context without spending significant effort. For example, a consumer comparing products may find a recommendation on one webpage but wonder whether other websites also recommend it. Reading the review she might discover useful criteria about the type of product she is researching, but figuring out how all the different options performed for that specific criteria can be very high costs, requiring her to go through many reviews for each option. Motivated by this need, users then **externalize** newly identified concepts to the systems to access relevant information across the entire dataset. In the case of Alloy, crowdworkers highlighted keywords in the sampled items to search for other items across the dataset to group them into the same categories; in the cases of SearchLens and Mesh, users used keywords they identified in reviews as criteria to search for reviews across different restaurants on Yelp or products on Amazon; and in the case of Weaver, users simply

hover over an entity mention to express interests to the system.

The system in turn used the keywords or entity mentions to **search** across the entire dataset (i.e., text items in a dataset in Alloy, reviews of different options in SearchLens and Mesh, or text snippets across different browser tabs in Weaver), and **summarizes** the search results to provide concept-specific global context to the users. In the systems presented in this dissertation, this included directly showing a list of search results in Alloy, showing the average ratings of reviews that mentioned a specific criteria in Mesh, and presenting visual explanations for each search result in SearchLens. In addition to the summarizations, all systems also allowed its users to drill-down and explore each item in the search results which often leads to new iterations of discovering and externalizing new concepts for further investigation forming a discovery-exploration loop.

Evaluation results presented in Chapter 3 showed that the same clustering algorithm performed better when only using keywords selected by the crowdworkers when compared to using all words as features. This showed that crowdworkers were capable of selecting discriminatory keywords to represent the categories. Based on this insight, I continued to use this strategy of allowing users to externalize their mental concepts (i.e., criteria) in systems that supported individual online sensemaking later in this document. Admittedly, document search algorithms still make occasional mistakes, and it is important to provide mechanisms to the users to recover from system errors. For this, users can **update** the summaries generated by the systems to better reflect their own interpretation of data. In the case of Alloy, crowdworkers labeled each item in the search results to indicate whether they should be in the same category or not; in the case of Mesh, individuals can change the average rating based on their own judgement after reading the reviews. This pattern is especially important when the summaries themselves are the final artifacts for users such as in Alloy (labeling items with categories) and Mesh (generating a personalized product comparison table).

8.3 Conclusion

Bottom-up exploration of large quantities of unstructured information is ubiquitous, but it can also be challenging to users due to the individual's limited capacity to understand global context throughout the exploration. Crowdsourcing is one example of this where each crowdworker's capacity to process data is further limited by the scope of microscopes, and the requesters often turn to crowdsourcing for scale and do not have the capacity to understand their own datasets fully. Another example is consumer research, where users are faced with making decisions based on their personal interpretation of an enormous amount of online evidence about many different choices.

In this dissertation, I investigated the core thesis of providing global context during bottom-up data exploration by designing and building five novel systems and interaction techniques. The key insight is that users need global context that reflects their evolving interests to support sense-making throughout their exploration. Through extensive lab and field studies, I demonstrated benefits to users over existing approaches, include allowing users to identify patterns in data and propose categories at the right abstraction level under the global context (Chapter 3), identify *residue* and improve current schema Chapter 4, evaluate new concepts based on current interests under global context Chapters 5 and 6, and evaluate newly discovered criteria to prioritize research efforts Chapter 7. Better support for bottom-up data exploration is likely to become

increasingly important these two scenarios explored in this document: For crowdsourcing, modern machine learning models demand increasingly large amounts of high quality training labels; and for individual online research, the rise of online misinformation and shill reviews makes traditional automated aggregation techniques that increasingly vulnerable. This thesis introduced novel interaction paradigms and insights about how users can benefit from them, as well as a general design pattern that can inform the building of future systems that can support global context during bottom-up data exploration.

Bibliography

- [1] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26. ACM, 2006. 4.1
- [2] Jae-wook Ahn and Peter Brusilovsky. Adaptive visualization of search results: Bringing user models to visual analytics. *Information Visualization*, 8(3):167–179, 2009. 5.2.1
- [3] Jae-wook Ahn, Peter Brusilovsky, Jonathan Grady, Daqing He, and Sue Yeon Syn. Open user profiles for adaptive news systems: help or harm? In *Proceedings of the 16th international conference on World Wide Web*, pages 11–20. ACM, 2007. 5.5
- [4] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, 2019. 7.4.1
- [5] Omar Alonso, Catherine C Marshall, and Marc Najork. Are some tweets more interesting than others?# hardquestion. In *Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval*, page 2. ACM, 2013. 4.2.1
- [6] Erik M Altmann and J Gregory Trafton. Task interruption: Resumption lag and the role of cues. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 26, 2004. 7.1
- [7] J Anderson. Forrester Market Report: Consumer Behavior Online: A 2009 Deep Dive. <http://www.forrester.com/go?docid=54327>, 2009. Accessed: 2017-09-10. 6.1
- [8] Paul André, Aniket Kittur, and Steven P Dow. Crowd synthesis: Extracting categories and clusters from complex data. In *Proc. CSCW 2014*, 2014. 2.2, 3.1, 3.1.1, 3.3.4
- [9] Paul André, Aniket Kittur, and Steven P Dow. Crowd synthesis: Extracting categories and clusters from complex data. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 989–998. ACM, 2014. 4.2.4, 8.1.4
- [10] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25, 2000. 6.3.4
- [11] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007. 6.2.2, 6.4.2
- [12] Yoram Bachrach, Thore Graepel, Tom Minka, and John Guiver. How to grade a test without knowing the answers—a bayesian graphical model for adaptive crowdsourcing and aptitude testing. *The proceedings of the International Conference on Machine Learning*,

2012. 4.1

- [13] Krisztian Balog, Pavel Serdyukov, and Arjen P De Vries. Overview of the trec 2010 entity track. Technical report, Norwegian Univ of Science and Technology Trondheim, 2010. 6.1
- [14] Nicholas J Belkin, Colleen Cool, Judy Jeng, Amymarie Keller, Diane Kelly, Ja-Young Kim, Hyuk-Jin Lee, Muh-Chyun (Morris) Tang, and Xiao-Jun Yuan. Rutgers' trec 2001 interactive track experience. In *TREC*, 2001. 5.1
- [15] Nicholas J Belkin, Diane Kelly, G Kim, J-Y Kim, H-J Lee, Gheorghe Muresan, M-C Tang, X-J Yuan, and Colleen Cool. Query length in interactive information retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 205–212. ACM, 2003. 2.1, 5.1
- [16] Richard Ernest Bellman. *Dynamic Programming*. Dover Publications, Incorporated, 2003. ISBN 0486428095. 3.1, 3.1.1
- [17] Tim Berners-Lee and Mark Fischetti. *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*. DIANE Publishing Company, 2001. 6.2.3
- [18] Tim Berners-Lee and James Hendler. Publishing on the semantic web. *Nature*, 410(6832):1023, 2001. 6.2.3
- [19] Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. Soylent: a word processor with a crowd inside. In *Proc. UIST 2010*, pages 313–322. ACM, 2010. 3.5
- [20] Michael S. Bernstein, Joel Brandt, Robert C. Miller, and David R. Karger. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 33–42, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0716-1. doi: 10.1145/2047196.2047201. URL <http://doi.acm.org/10.1145/2047196.2047201>. 4.2.5
- [21] Michael S Bernstein, Jaime Teevan, Susan Dumais, Daniel Liebling, and Eric Horvitz. Direct answers for search queries in the long tail. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 237–246. ACM, 2012. 7.1, 7.2
- [22] Suresh K Bhavnani. Why is it difficult to find comprehensive information? implications of information scatter for search and design. *Journal of the American Society for Information Science and Technology*, 56(9):989–1003, 2005. 2.3, 6.2.1, 6.2.3
- [23] Andrea Bianchi, So-Ryang Ban, and Ian Oakley. Designing a physical aid to support active reading on tablets. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 699–708. ACM, 2015. 6.1, 6.2.4, 7.2
- [24] Eric A Bier, Edward W Ishak, and Ed Chi. Entity quick click: rapid text copying based on automatic entity extraction. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems*, pages 562–567. ACM, 2006. 6.2.4
- [25] Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandy White, Samual White, et al. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342. ACM, 2010. 4.2.5
- [26] Mustafa Bilgic and Raymond J Mooney. Explaining recommendations: Satisfaction vs. promotion. In *Beyond Personalization Workshop, IUI*, volume 5, page 153, 2005. 7.2

- [27] Steven Bird and Edward Loper. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics, 2004. 5.3.5
- [28] Roi Blanco, Berkant Barla Cambazoglu, Peter Mika, and Nicolas Torzec. Entity recommendations in web search. In *International Semantic Web Conference*, pages 33–48. Springer, 2013. 6.1, 6.2.2
- [29] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003. 3.1, 3.1.1, 3.3.6
- [30] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowledge-based systems*, 46:109–132, 2013. 1.2, 7.1, 7.2
- [31] Ilaria Bordino, Yelena Mejova, and Mounia Lalmas. Penguins in sweaters, or serendipitous entity search on user-generated content. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 109–118. ACM, 2013. 6.1, 6.2.2
- [32] Horatiu Bota, Ke Zhou, and Joemon M Jose. Playing your cards right: The effect of entity cards on search behaviour and workload. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*, pages 131–140. ACM, 2016. 6.1, 6.2.2, 6.3.1
- [33] Jonathan Bragg, Daniel S Weld, et al. Crowdsourcing multi-label classification for taxonomy creation. In *First AAAI conference on human computation and crowdsourcing*, 2013. 2.2, 3.1, 3.1.1, 4.2.2
- [34] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM, 2005. 5.2.1
- [35] Michele A. Burton, Erin Brady, Robin Brewer, Callie Neylan, Jeffrey P. Bigham, and Amy Hurst. Crowdsourcing subjective fashion advice using vizwiz: Challenges and opportunities. In *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS ’12, pages 135–142, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1321-6. doi: 10.1145/2384916.2384941. URL <http://doi.acm.org/10.1145/2384916.2384941>. 4.2.5
- [36] Chris Callison-Burch and Mark Dredze. Creating speech and language data with amazon’s mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pages 1–12. Association for Computational Linguistics, 2010. 4.1, 4.2.2
- [37] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM, 2007. 5.2.1
- [38] Robert Capra, Gary Marchionini, Javier Velasco-Martin, and Katrina Muller. Tools-at-hand and learning in multi-session, collaborative search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 951–960. ACM, 2010. 2.1
- [39] Pew Research Center. Generational differences in online activities. Report, July 2015. <http://www.pewinternet.org/2009/01/28/generational-differences-in-online-activities/>. 3.4.1

- [40] Joel Chan, Steven Dang, and Steven P Dow. Improving crowd innovation with expert facilitation. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 1223–1235. ACM, 2016. 4.2.5
- [41] Allison June-Barlow Chaney and David M Blei. Visualizing topic models. In *ICWSM*, 2012. 3.1
- [42] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM 2011 TIST*, 2(3):27, 2011. 3.2.1
- [43] Joseph Chee Chang, Nathan Hahn, and Aniket Kittur. Supporting mobile sensemaking through intentionally uncertain highlighting. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 61–68. ACM, 2016. 6.3.2, 6.4.3
- [44] Joseph Chee Chang, Aniket Kittur, and Nathan Hahn. Alloy: Clustering with crowds and computation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3180–3191. ACM, 2016. 3, 4.2.4, 5.2.2, 6.5.1
- [45] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. Revolt: Collaborative crowdsourcing for labeling machine learning datasets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI ’17, New York, NY, USA, 2017. ACM. doi: 10.1145/3025453.3026044. URL <http://doi.acm.org/10.1145/3025453.3026044>. 4, 5.2.2
- [46] Joseph Chee Chang, Nathan Hahn, Adam Perer, and Aniket Kittur. Searchlens: Composing and capturing complex user interests for exploratory search. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 498–509, 2019. 7.1
- [47] Joseph Chee Chang, Nathan Hahn, Adam Perer, and Aniket Kittur. Searchlens composing and capturing complex user interests for exploratory search. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI ’19, Marina del Rey, CA, USA, 2019. ACM. doi: 10.1145/3301275.3302321. URL <http://doi.acm.org/10.1145/3301275.3302321>. 5
- [48] Joseph Z. Chang, Jason S. Chang, and Jyh-Shing Roger Jang. Learning to find translations and transliterations on the web. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, ACL ’12, pages 130–134. Association for Computational Linguistics, 2012. 4.1
- [49] Kathy Charmaz and Linda Liska Belgrave. Grounded theory. *The Blackwell encyclopedia of sociology*, 2007. (document), 1, 7.5.3
- [50] Chao Chen, Daqing Zhang, Bin Guo, Xiaojuan Ma, Gang Pan, and Zhaohui Wu. Triplanner: Personalized trip planning leveraging heterogeneous crowdsourced digital footprints. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1259–1273, 2015. 1.2, 6.1, 7.1, 7.2
- [51] Li Chen and Feng Wang. Explaining recommendations based on feature sentiments in product reviews. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 17–28, 2017. 7.2
- [52] Li Chen, Feng Wang, Luole Qi, and Fengfeng Liang. Experiment on sentiment embedded comparison interface. *Knowledge-Based Systems*, 64:44–58, 2014. 7.2
- [53] Liren Chen and Katia Sycara. Webmate: A personal agent for browsing and searching. In *Proceedings of the second international conference on Autonomous agents*, pages 132–

139. ACM, 1998. 5.2.1

- [54] Nan-Chen Chen, Jina Suh, Johan Verwey, Gonzalo Ramos, Steven Drucker, and Patrice Simard. Anchorviz: Facilitating classifier error discovery through interactive semantic data exploration. In *23rd International Conference on Intelligent User Interfaces*, pages 269–280. ACM, 2018. 5.2.2
- [55] Tao Cheng, Xifeng Yan, and Kevin Chen-Chuan Chang. Entityrank: searching entities directly and holistically. In *Proceedings of the 33rd international conference on Very large data bases*, pages 387–398. VLDB Endowment, 2007. 6.1
- [56] EH-H Chi, Phillip Barry, John Riedl, and Joseph Konstan. A spreadsheet approach to information visualization. In *Proceedings of VIZ'97: Visualization Conference, Information Visualization Symposium and Parallel Rendering Symposium*, pages 17–24. IEEE, 1997. 7.2
- [57] Lydia B Chilton, Greg Little, Darren Edge, Daniel S Weld, and James A Landay. Cascade: Crowdsourcing taxonomy creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1999–2008. ACM, 2013. 4.2.4, 4.3.3, 8.1.4
- [58] Lydia B Chilton, Greg Little, Darren Edge, Daniel S Weld, and James A Landay. Cascade: Crowdsourcing taxonomy creation. In *Proc. CHI 2013*, pages 1999–2008. ACM, 2013. 2.2, 3.1, 3.1.1, 3.3.6
- [59] Lydia B Chilton, Juho Kim, Paul André, Felicia Cordeiro, James A Landay, Daniel S Weld, Steven P Dow, Robert C Miller, and Haoqi Zhang. Frenzy: Collaborative data organization for creating conference sessions. In *Proc. CHI 2014*, pages 1255–1264. ACM, 2014. 3.3.3, 4.3.3
- [60] Jason Chuang, Christopher D Manning, and Jeffrey Heer. Termite: Visualization techniques for assessing textual topic models. In *Proc. of the International Working Conference on Advanced Visual Interfaces*, pages 74–77. ACM, 2012. 3.1, 3.3.6
- [61] Jason Chuang, Daniel Ramage, Christopher Manning, and Jeffrey Heer. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proc. CHI 2012*, pages 443–452. ACM, 2012. 3.1
- [62] Shelia R Cotten and Sipi S Gupta. Characteristics of online and offline health information seekers and factors that discriminate between them. *Social science & medicine*, 59(9):1795–1806, 2004. 2.3, 6.1
- [63] Douglass R Cutting, David R Karger, Jan O Pedersen, and John W Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *ACM SIGIR Forum*, volume 51, pages 148–159. ACM, 2017. 2.3
- [64] Bart De Langhe, Philip M Fernbach, and Donald R Lichtenstein. Navigating by the stars: Investigating the actual and perceived validity of online user ratings. *Journal of Consumer Research*, 42(6):817–833, 2015. 1.2, 5.1
- [65] Bart De Langhe, Philip M Fernbach, and Donald R Lichtenstein. Navigating by the stars: Investigating the actual and perceived validity of online user ratings. *Journal of Consumer Research*, 42(6):817–833, 2016. 7.1
- [66] Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990. 3.1.1, 3.3.6

- [67] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Zencrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st International Conference on World Wide Web, WWW '12*, pages 469–478, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1229-5. doi: 10.1145/2187836.2187900. URL <http://doi.acm.org/10.1145/2187836.2187900>. 4.2.1
- [68] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 1.1, 4.1, 4.2.1, 4.4.2
- [69] Cecilia di Sciascio, Vedran Sabol, and Eduardo E Veas. Rank as you go: User-driven exploration of search results. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, pages 118–129. ACM, 2016. 5.3.3
- [70] Cecilia di Sciascio, Peter Brusilovsky, and Eduardo Veas. A study on user-controllable social exploratory search. In *23rd International Conference on Intelligent User Interfaces*, pages 353–364. ACM, 2018. 5.3.3
- [71] Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. Pick-a-crowd: Tell me what you like, and i'll tell you what to do. In *Proceedings of the 22Nd International Conference on World Wide Web, WWW '13*, pages 367–374, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2035-1. doi: 10.1145/2488388.2488421. URL <http://doi.acm.org/10.1145/2488388.2488421>. 4.2.2
- [72] M. M. Schaffer D.L. Medin. Context theory of classification learning. *Psychological review*, 85(3):207, 1978. 3.2.1
- [73] Mira Dontcheva, Steven M Drucker, Geraldine Wade, David Salesin, and Michael F Cohen. Collecting and organizing web content. In *Personal Information Management-Special Interest Group for Information Retrieval Workshop*, pages 44–47, 2006. 6.2.4
- [74] Mira Dontcheva, Steven M Drucker, David Salesin, and Michael F Cohen. Relations, cards, and search templates: user-guided web data integration and layout. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 61–70. ACM, 2007. 6.2.4, 6.3
- [75] Shayan Doroudi, Ece Kamar, Emma Brunskill, and Eric Horvitz. Toward a learning science for complex crowdsourcing tasks. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2623–2634. ACM, 2016. 4.2.2
- [76] Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1013–1022. ACM, 2012. 4.2.5
- [77] Ryan Drapeau, Lydia B Chilton, Jonathan Bragg, and Daniel S Weld. Microtalk: Using argumentation to improve crowdsourcing accuracy. *AAAI HCOMP*, 2016. 4.2.3, 4.4.1, 4.4.3
- [78] Elizabeth Dwoskin and Craig Timberg. How merchants use facebook to flood amazon with fake reviews - the washington post. https://www.washingtonpost.com/business/economy/how-merchants-secretly-use-facebook-to-flood-amazon-with-fake-reviews/2018/04/23/5dad1e30-4392-11e8-8569-26fda6b404c7_story.html, 2018. (Accessed on 05/06/2020). 7.1, 7.2
- [79] Jerry Alan Fails and Dan R Olsen Jr. Interactive machine learning. In *Proc. IUI 2003*,

pages 39–45. ACM, 2003. 3.1

- [80] James Fogarty, Desney Tan, Ashish Kapoor, and Simon Winder. Cueflik: interactive concept learning in image search. In *Proceedings of the sigchi conference on human factors in computing systems*, pages 29–38. ACM, 2008. 5.2.2
- [81] Suzannah Fox. The online health care revolution: How the web helps americans take better care of themselves. a pew internet and american life project online report. http://www.pewinternet.org/reports/pdfs/PIP_Health_Report.pdf, 2000. 2.3, 6.1
- [82] Michael J Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin. Crowddb: answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 61–72, 2011. 1.1
- [83] Kristofer Franzen and Jussi Karlgren. Verbosity and interface design. *SICS Research Report*, 2000. 2.1, 5.1
- [84] Qiwei Gan, Qing Cao, and Donald Jones. Helpfulness of online user reviews: More is less. 2012. 1.2, 2.3, 5.1, 7.1, 7.2, 7.3
- [85] Miriam Greis, Emre Avci, Albrecht Schmidt, and Tonja Machulla. Increasing users' confidence in uncertain data by aggregating data from multiple sources. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 828–840, 2017. 2.3, 6.1
- [86] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274. ACM, 2009. 6.1, 6.2.2, 6.5.1
- [87] Kelvin Guu, John Miller, and Percy Liang. Traversing knowledge graphs in vector space. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 318–327, Lisbon, Portugal, September 2015. Association for Computational Linguistics. 6.1, 6.2.2
- [88] Nathan Hahn, Joseph Chang, Ji Eun Kim, and Aniket Kittur. The knowledge accelerator: Big picture thinking in small pieces. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2258–2270. ACM, 2016. 3, 3.3.5, 3.4, 3.4, 6.5.1
- [89] Sudheendra Hangal, Abhinay Nagpal, and Monica Lam. Effective browsing and serendipitous discovery with an experience-infused browser. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, pages 149–158. ACM, 2012. 6.2.3
- [90] Derek L Hansen, Patrick J Schone, Douglas Corey, Matthew Reid, and Jake Gehring. Quality control mechanisms for crowdsourcing: peer review, arbitration, & expertise at familysearch indexing. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 649–660. ACM, 2013. 4.1, 4.2.2
- [91] John A Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc., 1975. 3.1.1
- [92] Marti Hearst. Design recommendations for hierarchical faceted search interfaces. In *ACM SIGIR workshop on faceted search*, pages 1–5. Seattle, WA, 2006. 7.2
- [93] Marti Hearst. *Search user interfaces*. Cambridge University Press, 2009. 2.3
- [94] Marti A Hearst and Duane Degler. Sewing the seams of sensemaking: A practical interface for tagging and organizing saved search results. In *Proceedings of the symposium on human-computer interaction and information retrieval*, page 4. ACM, 2013. 6.2.1

- [95] Marti A Hearst and Jan O Pedersen. Visualizing information retrieval results: a demonstration of the tilebar interface. In *Conference Companion on Human Factors in Computing Systems*, pages 394–395. ACM, 1996. 2.1, 5.1, 5.3.1, 5.3.2, 5.4.1
- [96] Stephen J Hoch and Young-Won Ha. Consumer learning: Advertising and the ambiguity of product experience. *Journal of consumer research*, 13(2):221–233, 1986. 1.2, 7.1, 7.2, 7.5.2
- [97] Orland Hoeber and Xue Dong Yang. A comparative user study of web search interfaces: Hotmap, concept highlighter, and google. In *Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on*, pages 866–874. IEEE, 2006. 2.1, 5.1, 5.3.2, 5.3.2, 5.4.1
- [98] Andrew Hogue and David Karger. Thresher: automating the unwrapping of semantic content from the world wide web. In *Proceedings of the 14th international conference on World Wide Web*, pages 86–95. ACM, 2005. 6.2.4, 6.3
- [99] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, 2004. 1.2, 2.3, 7.1
- [100] Nan Hu, Jie Zhang, and Paul A Pavlou. Overcoming the j-shaped distribution of product reviews. *Communications of the ACM*, 52(10):144–147, 2009. 7.1
- [101] Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. Interactive topic modeling. *Machine learning*, 95(3):423–469, 2014. 3.1, 6.5.1
- [102] Chieh-Yang Huang, Shih-Hong Huang, and Ting-Hao Kenneth Huang. Heteroglossia: In-situ story ideation with the crowd. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI ’20, page 1–12, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376715. URL <https://doi.org/10.1145/3313831.3376715>. 3
- [103] Clayton J Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international AAAI conference on weblogs and social media*, 2014. 7.4.1
- [104] David Huynh, Stefano Mazzocchi, and David Karger. Piggy bank: Experience the semantic web inside your web browser. In *International Semantic Web Conference*, pages 413–430. Springer, 2005. 6.2.4
- [105] Google Inc. Google search quality evaluator guidelines., 2016. URL <http://static.googleusercontent.com/media/google.com/en//insidesearch/howsearchworks/assets/searchqualityevaluatorguidelines.pdf>. 4.2.2
- [106] Yelp Inc. The Yelp Dataset Challenge: Discover what insights lie hidden in our data. <https://www.yelp.com/dataset/challenge>, 2016. Accessed: 2017-09-10. 5.1, 5.3
- [107] Oana Inel, Khalid Khamkham, Tatiana Cristea, Anca Dumitrasche, Arne Rutjes, Jelle van der Ploeg, Lukasz Romaszko, Lora Aroyo, and Robert-Jan Sips. Crowdtruth: Machine-human computation framework for harnessing disagreement in gathering annotated data. In *International Semantic Web Conference*, pages 486–504. Springer, 2014. 4.2.3
- [108] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proc. of the ACM SIGKDD*, pages 64–67. ACM, 2010. 4.1, 4.2.2

- [109] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31:3, 1999. 3.1
- [110] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988. 3.1.1
- [111] Bernard J Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information processing & management*, 36(2):207–227, 2000. 2.1, 5.1
- [112] Bernard J Jansen, Danielle L Booth, and Amanda Spink. Patterns of query reformulation during web searching. *Journal of the American society for information science and technology*, 60(7):1358–1371, 2009. 3.2.1, 8.1.2
- [113] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972. 3.1.1, 3.3.6
- [114] Bryan Jurish and Kay-Michael Würzner. Word and sentence tokenization with hidden markov models. *JLCL*, 28(2):61–83, 2013. 5.3.5
- [115] Sanjay Kairam and Jeffrey Heer. Parting crowds: Characterizing divergent interpretations in crowdsourced annotation tasks. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 1637–1648. ACM, 2016. 4.2.2, 4.2.3
- [116] Ece Kamar, Severin Hacker, and Eric Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 467–474. International Foundation for Autonomous Agents and Multiagent Systems, 2012. 4.2.2
- [117] Yvonne Kammerer, Rowan Nairn, Peter Pirolli, and Ed H Chi. Signpost from the masses: learning effects in an exploratory social tag search browser. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 625–634, 2009. 7.5.2
- [118] David R Karger. The semantic web and end users: What's wrong and how to fix it. *IEEE Internet Computing*, 18(6):64–70, 2014. 6.2.3
- [119] David R Karger and Dennis Quan. Haystack: a user interface for creating, browsing, and organizing arbitrary semistructured information. In *CHI'04 extended abstracts on Human factors in computing systems*, pages 777–778. ACM, 2004. 6.2.3
- [120] Melanie Kellar, Carolyn Watters, and Michael Shepherd. A field study characterizing web-based information-seeking tasks. *Journal of the American Society for Information Science and Technology*, 58(7):999–1018, 2007. 6.1
- [121] Joy Kim, Justin Cheng, and Michael S Bernstein. Ensemble: exploring complementary strengths of leaders and crowds in creative collaboration. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 745–755. ACM, 2014. 4.2.5
- [122] Aniket Kittur, Ed H Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proc. CHI 2008*, pages 453–456. ACM, 2008. 3.2.3, 7.5.2
- [123] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut. Crowdforge: Crowd-sourcing complex work. In *Proc UIST 2011*, pages 43–52. ACM, 2011. 3.5
- [124] Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John

- Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1301–1318. ACM, 2013. 4.2.2
- [125] Aniket Kittur, Andrew M Peters, Abdigani Diriye, Trupti Telang, and Michael R Bove. Costs and benefits of structured information foraging. In *Proc. CHI 2013*, pages 2989–2998. ACM, 2013. 3.3.2, 3.5, 6.2.1
- [126] Khalil Klouche, Tuukka Ruotsalo, Diogo Cabral, Salvatore Andolina, Andrea Bellucci, and Giulio Jacucci. Designing for exploratory search on touch devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 4189–4198. ACM, 2015. 2.3, 6.1, 6.2.2
- [127] James Q Knowlton. On the definition of “picture”. *AV Communication Review*, 14(2):157–183, 1966. 4.2.1
- [128] Rafal Kocielnik, Saleema Amershi, and Paul N Bennett. Will you accept an imperfect ai? exploring designs for adjusting end-user expectations of ai systems. 2019. 6.3.4
- [129] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM TKDD*, 2009. 3.1, 3.1.1
- [130] Ranjay A. Krishna, Kenji Hata, Stephanie Chen, Joshua Kravitz, David A. Shamma, Li Fei-Fei, and Michael S. Bernstein. Embracing error to enable rapid crowdsourcing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI ’16, pages 3167–3179, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858115. URL <http://doi.acm.org/10.1145/2858036.2858115>. 4.2.5
- [131] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 4.1, 4.2.1
- [132] Todd Kulesza, Saleema Amershi, Rich Caruana, Danyel Fisher, and Denis Charles. Structured labeling for facilitating concept evolution in machine learning. In *Proc. CHI 2014*, pages 3075–3084, 2014. 3.1
- [133] Todd Kulesza, Saleema Amershi, Rich Caruana, Danyel Fisher, and Denis Charles. Structured labeling for facilitating concept evolution in machine learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3075–3084. ACM, 2014. 1.1, 4.1, 4.2.1, 4.2.2, 4.4.2, 5.2.2
- [134] Ken Lang. The 20 newsgroups dataset, 1995. URL <http://people.csail.mit.edu/jrennie/20Newsgroups/>. 4.5
- [135] Gierad Laput, Walter S Lasecki, Jason Wiese, Robert Xiao, Jeffrey P Bigham, and Chris Harrison. Zensors: Adaptive, rapidly deployable, human-intelligent sensor feeds. In *Proc. CHI 2015*, pages 1935–1944. ACM, 2015. 3.5, 4.2.5
- [136] Walter S Lasecki, Rachel Wesley, Jeffrey Nichols, Anand Kulkarni, James F Allen, and Jeffrey P Bigham. Chorus: a crowd-powered conversational assistant. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 151–162. ACM, 2013. 4.2.5
- [137] Walter S Lasecki, Mitchell Gordon, Danai Koutra, Malte F Jung, Steven P Dow, and Jeffrey P Bigham. Glance: Rapidly coding behavioral video with the crowd. In *Proceedings*

- of the 27th annual ACM symposium on User interface software and technology*, pages 551–562. ACM, 2014. 4.2.5
- [138] Min Kyung Lee, Sara Kiesler, Jodi Forlizzi, Siddhartha Srinivasa, and Paul Rybski. Gracefully mitigating breakdowns in robotic services. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 203–210. IEEE, 2010. 6.3.4
 - [139] Clayton Lewis and Donald A Norman. Designing for error. In *Readings in Human-Computer Interaction*, pages 686–697. Elsevier, 1995. 6.3.4
 - [140] Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. Structure-aware review mining and summarization. In *Proceedings of the 23rd international conference on computational linguistics*, pages 653–661. Association for Computational Linguistics, 2010. 1.2, 2.3, 7.1
 - [141] Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C Weng. A note on platt’s probabilistic outputs for support vector machines. *Machine learning*, 68(3):267–276, 2007. 3.2.1
 - [142] Thomas Lin, Patrick Pantel, Michael Gamon, Anitha Kannan, and Ariel Fuxman. Active objects: Actions for entity-centric search. In *Proceedings of the 21st international conference on World Wide Web*, pages 589–598. ACM, 2012. 6.1, 6.2.2, 6.5.1
 - [143] Michael Xieyang Liu, Jane Hsieh, Nathan Hahn, Angelina Zhou, Emily Deng, Shaun Burley, Cynthia Taylor, Aniket Kittur, and Brad A Myers. Unakite: Scaffolding developers’ decision-making using the web. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 67–80, 2019. 7.2
 - [144] Pasquale Lops, Marco Degennaris, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, 2011. 7.2
 - [145] Kathleen M MacQueen, Eleanor McLellan, Kelly Kay, and Bobby Milstein. Codebook development for team-based qualitative analysis. *Cultural anthropology methods*, 10(2):31–36, 1998. 4.1, 4.2.1
 - [146] Asha S Manek, P Deepa Shenoy, M Chandra Mohan, and KR Venugopal. Aspect term extraction for sentiment analysis in large movie reviews using gini index feature selection method and svm classifier. *World wide web*, 20(2):135–154, 2017. 2.3, 7.1
 - [147] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008. 3.1.1, 3.3.6
 - [148] A. Mao, Y. Chen, K.Z. Gajos, D.C. Parkes, A.D. Procaccia, and H. Zhang. Turkserver: Enabling synchronous and longitudinal online experiments. In *Proceedings of HCOMP’12*, 2012. 4.4.1
 - [149] Gary Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006. 1, 6.1, 6.2.1
 - [150] Gary Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006. 1.2, 2.1, 2.3, 7.1, 7.2
 - [151] Gary J Marchionini, Gary Geisler, and Ben Brunk. Agileviews. In *Proceedings of the ASIST Annual Meeting*, volume 37, pages 271–280, 2000. 2.3
 - [152] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993. 4.2.1, 4.2.2

- [153] Catherine C Marshall, Morgan N Price, Gene Golovchinsky, and Bill N Schilit. Introducing a digital library reading appliance into a reading group. In *Proceedings of the fourth ACM conference on Digital libraries*, pages 77–84. ACM, 1999. 6.1, 6.2.1, 6.2.4, 6.3.2, 6.4.3, 7.2
- [154] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, 2013. 5.1, 7.1
- [155] Matt McGee. Yes, bing has human search quality raters and here's how they judge web pages, 2012. URL <http://searchengineland.com/bing-search-quality-rating-guidelines-130592>. 4.2.2
- [156] Brian McInnis, Dan Cosley, Chaebong Nam, and Gilly Leshed. Taking a hit: Designing around rejection, mistrust, risk, and workers' experiences in amazon mechanical turk. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2271–2282. ACM, 2016. 4.3.1
- [157] Douglas L Medin and Marguerite M Schaffer. Context theory of classification learning. *Psychological review*, 85(3):207, 1978. 3.1
- [158] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8. ACM, 2011. 6.1, 6.2.2, 6.3.4, 6.4.2
- [159] Rada Mihalcea and Andras Csoma. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007. 6.2.3
- [160] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 5.3, 5.3.1
- [161] Iris Miliaraki, Roi Blanco, and Mounia Lalmas. From selena gomez to marlon brando: Understanding explorative entity search. In *Proceedings of the 24th International Conference on World Wide Web*, pages 765–775. International World Wide Web Conferences Steering Committee, 2015. 6.1, 6.2.2, 6.3
- [162] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 4.4.2
- [163] Tanushree Mitra, Clayton J Hutto, and Eric Gilbert. Comparing person-and process-centric strategies for obtaining quality data on amazon mechanical turk. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1345–1354. ACM, 2015. 4.2.2
- [164] Dan Morris, Meredith Ringel Morris, and Gina Venolia. Searchbar: a search-centric web history for task resumption and information re-finding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1207–1216. ACM, 2008. 2.3
- [165] Susan M Mudambi and David Schuff. Research note: What makes a helpful online review? a study of customer reviews on amazon. com. *MIS quarterly*, pages 185–200, 2010. 1.2, 2.3, 5.1, 7.1, 7.2, 7.3, 7.5.2
- [166] Les Nelson, Christoph Held, Peter Pirolli, Lichan Hong, Diane Schiano, and Ed H Chi. With a little help from my friends: examining the impact of social annotations in sensemaking tasks. In *Proceedings of the SIGCHI conference on human factors in computing systems*,

pages 1795–1798, 2009. 7.5.2

- [167] K O'Hara. Towards a typology of reading goals rxrc affordances of paper project. *Rank Xerox Research Center, Cambridge, UK*, 1996. 6.1, 6.2.4, 6.3.2, 6.4.3, 7.2
- [168] Emily S Patterson, Emilie M Roth, and David D Woods. Predicting vulnerabilities in computer-supported inferential analysis under data overload. *Cognition, Technology & Work*, 3(4):224–237, 2001. 2.3
- [169] Jaakko Peltonen, Kseniia Belorustceva, and Tuukka Ruotsalo. Topic-relevance map: Visualization for improving search result comprehension. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 611–622. ACM, 2017. 2.3, 5.3.3
- [170] Jaakko Peltonen, Jonathan Strahl, and Patrik Floréen. Negative relevance feedback for exploratory search with visual interactive intent modeling. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 149–159. ACM, 2017. 2.1, 5.1, 5.2.1
- [171] Peter Pirolli and Stuart Card. Information foraging. *Psychological review*, 106(4):643, 1999. (document), 1, 1.2, 2.1, 2.2, 3.1.1, 3.2.1, 6.1
- [172] Peter Pirolli and Stuart Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, volume 5, pages 2–4, 2005. 1.2, 2.1
- [173] Barbara Plank, Dirk Hovy, and Anders Søgaard. Linguistically debatable or just plain wrong? In *ACL (2)*, pages 507–511, 2014. 4.2.3
- [174] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. 3.2.1
- [175] Matt Post, Chris Callison-Burch, and Miles Osborne. Constructing parallel corpora for six indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 401–409. Association for Computational Linguistics, 2012. 4.1, 4.2.1
- [176] Pradeep Racherla and Wesley Friske. Perceived 'usefulness' of online consumer reviews: An exploratory investigation across three services categories. *Electron. Commer. Rec. Appl.*, 11(6):548–559, November 2012. ISSN 1567-4223. doi: 10.1016/j.elerap.2012.06.003. URL <http://dx.doi.org/10.1016/j.elerap.2012.06.003>. 1.2
- [177] Pradeep Racherla and Wesley Friske. Perceived 'usefulness' of online consumer reviews: An exploratory investigation across three services categories. *Electronic Commerce Research and Applications*, 11(6):548–559, 2012. 1.2, 2.3, 6.1, 7.1, 7.2
- [178] Ramana Rao and Stuart K Card. The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 318–322, 1994. 7.2
- [179] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. <http://is.muni.cz/publication/884893/en>. 5.3.5

- [180] Philip Resnik and Noah A. Smith. The web as a parallel corpus. *Comput. Linguist.*, 29(3):349–380, September 2003. ISSN 0891-2017. doi: 10.1162/089120103322711578. URL <http://dx.doi.org/10.1162/089120103322711578>. 4.1
- [181] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009. 5.3.4, 5.4.1
- [182] Jakob Rogstadius, Vassilis Kostakos, Aniket Kittur, Boris Smus, Jim Laredo, and Maja Vukovic. An assessment of intrinsic and extrinsic motivation on task performance in crowdsourcing markets. *ICWSM*, 11:17–21, 2011. 4.2.2
- [183] Hugo Romat, Emmanuel Pietriga, Nathalie Henry-Riche, Ken Hinckley, and Caroline Appert. Spaceink: Making space for in-context annotations. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, UIST ’19, 2019. 6.2.4
- [184] Daniel E Rose and Danny Levinson. Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web*, pages 13–19. ACM, 2004. 6.1
- [185] Yong Rui, Thomas S Huang, Michael Ortega, and Sharad Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on circuits and systems for video technology*, 8(5):644–655, 1998. 2.1, 5.1
- [186] Daniel M Russell, Mark J Stefk, Peter Pirolli, and Stuart K Card. The cost structure of sensemaking. In *Proceedings of the INTERACT’93 and CHI’93 conference on Human factors in computing systems*, pages 269–276. ACM, 1993. 6.2.1, 7.2, 8.1.5
- [187] Jeffrey Rzeszotarski and Aniket Kittur. Crowdscape: interactively visualizing user behavior and output. In *Proc. UIST 2012*, pages 55–62. ACM, 2012. 4.2.2
- [188] Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Journal of the American society for information science*, 41(4):288–297, 1990. 2.1, 5.1
- [189] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001. 5.5
- [190] MC Schraefel, Max Wilson, Alistair Russell, and Daniel A Smith. mspace: improving information access to multimedia domains with multimodal exploratory search. *Communications of the ACM*, 49(4):47–49, 2006. 7.2
- [191] Monica C Schraefel, Yuxiang Zhu, David Modjeska, Daniel Wigdor, and Shengdong Zhao. Hunter gatherer: interaction support for the creation and management of within-web-page collections. In *Proceedings of the 11th international conference on World Wide Web*, pages 172–181, 2002. 6.2.4
- [192] Barry Schwartz. The paradox of choice: Why more is less. Ecco New York, 2004. 7.1
- [193] Xuehua Shen, Bin Tan, and ChengXiang Zhai. Implicit user modeling for personalized search. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 824–831. ACM, 2005. 5.2.1
- [194] Steven M Shugan. The cost of thinking. *Journal of consumer Research*, 7(2):99–111, 1980. 7.1, 7.4.2
- [195] Eva Siegenthaler, Yves Bochud, Per Bergamin, and Pascal Wurtz. Reading on lcd vs e-ink

- displays: effects on fatigue and visual strain. *Ophthalmic and Physiological Optics*, 32(5):367–374, 2012. 3
- [196] Herbert A Simon. Designing organizations for an information-rich world. *The Johns Hopkins Press*, 1971. 1.2
- [197] Herbert A Simon. Designing organizations for an information-rich world. *International Library of Critical Writings in Economics*, 70:187–202, 1996. 7.1
- [198] Rion Snow, Brendan O’Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP ’08, pages 254–263. Association for Computational Linguistics, 2008. URL <http://dl.acm.org/citation.cfm?id=1613715.1613751>. 4.1
- [199] Michael Spenke, Christian Beilken, and Thomas Berlage. Focus: the interactive table for product comparison and selection. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 41–50, 1996. 7.2
- [200] Mirco Speretta and Susan Gauch. Personalized search based on user search histories. In *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, pages 622–628. IEEE, 2005. 5.2.1
- [201] Statista. Yelp: cumulative number of reviews 2019 | statista. <https://www.statista.com/statistics/278032/cumulative-number-of-reviews-submitted-to-yelp/>, 2019. (Accessed on 05/06/2020). 7.1
- [202] Michael Steinbach, George Karypis, Vipin Kumar, et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, 2000. 3.1.1
- [203] Anselm Strauss and Juliet Corbin. *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage Publications, Inc, 1998. 4.2.1
- [204] Jeffrey Stylos, Brad A Myers, and Andrew Faulring. Citrine: providing intelligent copy-and-paste. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 185–188. ACM, 2004. 6.2.4
- [205] Rohail Syed and Kevyn Collins-Thompson. Optimizing search results for human learning goals. *Information Retrieval Journal*, 20(5):506–523, 2017. 6.3
- [206] Leila Takayama and Stuart K Card. Tracing the microstructure of sensemaking. In *CHI workshop on Sensemaking, Florence, Italy (April 6, 2008) Fig*, volume 2, 2008. 6.2.1
- [207] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. Adaptively learning the crowd kernel. In *In ICML’11*, 2011. 2.2, 3.1.1
- [208] Craig S Tashman and W Keith Edwards. Liquidtext: a flexible, multitouch environment to support active reading. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3285–3294. ACM, 2011. 6.1, 6.2.4, 6.3, 6.3.2, 6.4.3, 7.2
- [209] Ann Taylor, Mitchell Marcus, and Beatrice Santorini. The penn treebank: an overview. In *Treebanks*, pages 5–22. Springer, 2003. 4.1, 4.2.1
- [210] Long Tran-Thanh, Trung Dong Huynh, Avi Rosenfeld, Sarvapali D. Ramchurn, and Nicholas R. Jennings. Budgetfix: Budget limited crowdsourcing for interdependent task allocation with quality guarantees. In *Proceedings of the 2014 International Conference*

- on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pages 477–484, Richland, SC, 2014. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-2738-1. URL <http://dl.acm.org/citation.cfm?id=2615731.2615809>. 4.2.2
- [211] Simon Tretter, Gene Golovchinsky, and Pernilla Qvarfordt. Searchpanel: A browser extension for managing search activity. In *EuroHCIR*, pages 51–54, 2013. 2.3
 - [212] Max G Van Kleek, Michael Bernstein, Katrina Panovich, Gregory G Vargas, David R Karger, and MC Schraefel. Note to self: examining personal information keeping in a lightweight note-taking tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1477–1480. ACM, 2009. 2.1, 6.2.4, 6.3.2, 6.4.1
 - [213] Max G Van Kleek, Wolfe Styke, David Karger, et al. Finders/keepers: a longitudinal study of people managing information scraps in a micro-note tool. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2907–2916. ACM, 2011. 6.2.4
 - [214] Cornelis J Van Rijsbergen, Stephen Edward Robertson, and Martin F Porter. *New models in probabilistic information retrieval*. British Library Research and Development Department London, 1980. 5.3.5
 - [215] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proc. ICML 2009*, pages 1073–1080. ACM, 2009. 3.3.1
 - [216] Luis Von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. re-captcha: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008. 1.1
 - [217] Cynthia Weston, Terry Gandell, Jacinthe Beauchamp, Lynn McAlpine, Carol Wiseman, and Cathy Beauchamp. Analyzing interview data: The development and evolution of a coding system. *Qualitative sociology*, 24(3):381–400, 2001. 4.1, 4.2.1
 - [218] Ryen W White, Bill Kules, Steven M Drucker, et al. Supporting exploratory search, introduction, special issue, communications of the acm. *Communications of the ACM*, 49(4):36–39, 2006. 6.2.1
 - [219] Ryen W White, Mikhail Bilenko, and Silviu Cucerzan. Studying the use of popular destinations to enhance web search interaction. In *Proc. SIGIR 2007*, pages 159–166. ACM, 2007. 3.1
 - [220] Janyce M Wiebe, Rebecca F Bruce, and Thomas P O'Hara. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 246–253. Association for Computational Linguistics, 1999. 4.2.1, 4.5
 - [221] Max Wilson, Alistair Russell, Daniel A Smith, et al. mspace: improving information access to multimedia domains with multimodal exploratory search. *Communications of the ACM*, 49(4):47–49, 2006. 2.3
 - [222] Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)*, 26(3):13, 2008. 5.2.1
 - [223] Ting-Fan Wu, Chih-Jen Lin, and Ruby C Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5(975-1005):4,

2004. 3.2.1

- [224] Jinxi Xu and W Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 4–11. ACM, 1996. 5.5
- [225] Beverly Yang and Glen Jeh. Retroactive answering of search queries. In *Proceedings of the 15th international conference on World Wide Web*, pages 457–466. ACM, 2006. 5.5
- [226] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics, 1995. 5.5
- [227] Jinfeng Yi, Rong Jin, Anil K Jain, and Shaili Jain. Crowdclustering with sparse pairwise labels: A matrix completion approach. In *AAAI Workshop on Human Computation*, volume 2, 2012. 2.2, 3.1.1
- [228] Jinfeng Yi, Rong Jin, Shaili Jain, Tianbao Yang, and Anil K Jain. Semi-crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning. In *Advances in Neural Information Processing Systems*, pages 1772–1780, 2012. 2.2, 3.1.1
- [229] Jianxing Yu, Zheng-Jun Zha, Meng Wang, and Tat-Seng Chua. Aspect ranking: identifying important product aspects from online consumer reviews. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1496–1505. Association for Computational Linguistics, 2011. 2.3, 7.1
- [230] Oren Zamir and Oren Etzioni. Grouper: a dynamic clustering interface to web search results. *Computer Networks*, 31(11):1361–1374, 1999. 2.3
- [231] Hua-Jun Zeng, Qi-Cai He, Zheng Chen, Wei-Ying Ma, and Jinwen Ma. Learning to cluster web search results. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 210–217. ACM, 2004. 2.3
- [232] Haoqi Zhang, Edith Law, Rob Miller, Krzysztof Gajos, David Parkes, and Eric Horvitz. Human computation tasks with global constraints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 217–226. ACM, 2012. 1.2, 6.1, 7.1, 7.2