Characterizing Practices, Limitations, and Opportunities Related to Text Information Extraction Workflows: A Human-in-the-loop Perspective

Sajjadur Rahman
sajjadur@megagon.ai
Megagon Labs
Mountain View, USA

Eser Kandogan
eser@megagon.ai
Megagon Labs
Mountain View, USA

ABSTRACT

Information extraction (IE) approaches often play a pivotal role in text analysis and require significant human intervention. Therefore, a deeper understanding of existing IE practices and related challenges from a human-in-the-loop perspective is warranted. In this work, we conducted semi-structured interviews in an industrial environment and analyzed the reported IE approaches and limitations. We observed that data science workers often follow an iterative task model consisting of information foraging and sensemaking loops across all the phases of an IE workflow. The task model is generalizable and captures diverse goals across these phases (*e.g.*, data preparation, modeling, evaluation.) We found several limitations in both foraging (e.g., data exploration) and sensemaking (e.g., qualitative debugging) loops stemming from a lack of adherence to existing cognitive engineering principles. Moreover, we identified that due to the iterative nature of an IE workflow, the requirement of provenance is often implied but rarely supported by existing systems. Based on these findings, we discuss design implications for supporting IE workflows and future research directions.

CCS CONCEPTS

• **Human-centered computing** → **User studies**.

KEYWORDS

Information extraction, Data science workflows, Human-AI collaboration

ACM Reference Format:
Sajjadur Rahman and Eser Kandogan. 2022. Characterizing Practices, Limitations, and Opportunities Related to Text Information Extraction Workflows: A Human-in-the-loop Perspective. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3491102.3502068

1 INTRODUCTION

Information extraction (IE) [14] is the process of extracting structured information from unstructured text document. The extracted structured information is semantically well defined and enable downstream task, such as entity matching [38], knowledge-base creation and population [31], text summarization [18], via logical reasoning on the structured representation of the document. Therefore, information extraction is often the first step in text analysis workflows. Similar to any other data science work, information extraction involves human(s) in the loop who intervene at various phases to steer the workflow. Therefore, a more systematic fine-grained characterization of the IE workflow is required to develop tools that enhance practitioners' productivity.

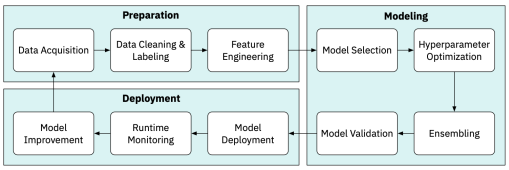

Figure 1: A data science workflow, consisting of three high-level phases: data preparation, model building, and model deployment [70].

There has been a recent shift in focus from the ML and NLP community towards human-in-the-loop data science [17], focusing on productivity tools for improving user experience and data management practices for supporting task-specific workflows. However, these tools are developed from a generic interpretation of data science workflows consisting of abstract processes, pipelines, and workflows [82] (see Figure 1). However, these phases are complex coarse-grained processes that involve a sequence of user actions to accomplish specific tasks. For example, a data cleaning task within the data preparation phase would require the users to 1) view the data (view) 2) form a general understanding (assess), 3) define cleaning objectives (hypothesize), 4) develop a cleaning model (pursue), and 5) evaluate the model (verify). Therefore, the phase-based coarse-grained characterization results in the development of tools that often lack crucial features to support fine-grained actions [57]. HCI and CSCW community have explored the fine-grained details of specific modalities such as data [46], tasks such as data wrangling [26], settings such as collaboration [82]. However, these approaches don't capture the interplay among the coarse-grained phases.

In this work, we strike a balance between the two objectives as we analyze text information extraction workflows from a fine-grained task-centric viewpoint. We propose an iterative task model

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3502068

comprising five tasks — view, assess, hypothesize, pursue, and verify — which is generalizable to all the phases of an IE workflow. The task model captures user actions and goals in each phase in a fine-grained manner. When observed from the lens of the sensemaking process for analyst technology [55], we obtain a coarse-grained view, where the task model comprises both the foraging and sensemaking loops. The iterative foraging and sensemaking loops capture how data science workers may iterate within a single-phase and across phases of an information extraction workflow.

We conducted a semi-structured interview-based study to capture fine-grained details of these workflows, where we interviewed data science workers in an industrial research lab. We focused on understanding essential elements like reasoning, motivation, and experiences as we discussed their IE practices and associated challenges. We analyzed the interviews by employing the grounded theory method [11] and performed an iterative analysis to formulate the aforementioned *task model* for conceptualizing IE workflows. Further investigation of the participant challenges uncovered a lack of adherence to the cognitive engineering principal [23] among the tools for supporting IE workflows. Moreover, we observed that provenance, a feature missing from the existing systems as built-in support, is a crucial requirement within the iterative setting of IE workflows. Based on those observations, we distilled eight design considerations for IE tools.

Contributions. Our primary contributions are as follows:

- We designed and conducted a semi-structured interview study on ten industry projects involving IE and conceptualized IE workflows from fine- and coarse-grained perspectives through qualitative analysis.
- We proposed a task-centric model that captures a fine-grained representation of phases within IE workflows while revealing the iterative nature of a phase.
- We further analyzed the tasks to capture user actions corresponding to each task across all phases and concretized task-specific limitations.
- We analyzed the limitations from the perspective of cognitive engineering principals to identify key design considerations for IE tools to assist in information foraging and sensemaking.
- We discussed how the adoption of the design considerations might impact the development of IE tools in supporting rapid prototyping.

2 RELATED WORK

Text information extraction is a specific instance of data science work. In this section, we ground our discussion on the data science process as we review literature on analyzing work practices, conceptualization of workflows, and tool usage.

2.1 Analyzing data science work practices

HCI researchers have conducted studies to analyze and assess data science work practices along various dimensions. For example, studies have been conducted to understand work practices along dimensions such as users engagement (*e.g.,* collaborative [34, 82], single user [46, 70]), modalities (*e.g.,* data [46, 50]), tools (*e.g.,* notebook [35]), roles (*e.g.,* with domain experts [43]), themes (*e.g.,* interpretability [29], trust [51], explanation [60]), and goals (*e.g.,* wrangling [26]), among others. While exploring the work practices, many of these studies considered more than one dimension. These studies have resulted in identifying the challenges faced by stakeholders, design implications for tool development, and future trends of the data science process. In this work, we focused on a specific text analysis workflow spanning multiple phases, *i.e.,* information extraction, in a collaborative setting within the industry and analyzed the current practices and limitations. Information extraction from text is a diverse and complex process and, within the industry setting, involves many stakeholders beyond those that actively work with data or write code, *i.e.,* data science workers. Our work focuses on this latter group — data science workers – to build an initial understanding of how they accomplish their workflows.

2.2 Conceptualizing data science workflows

Besides analyzing the data science workflows and the associated challenges and limitations, sensemaking studies [75] have been conducted to conceptualize the workflows into high-level concepts beyond just phases. For example, studies have characterized the data science workflow as a multi-phase process [70] by building upon work on the conceptualization of complex activities around data practices [46]. Researchers in HCI and CSCW have developed insights into how data science workers approach their data [46, 50, 53]. Passi and Jackson examined how imposing rules impacted data science workers and described an ongoing tension over the use of algorithmic rules [50]. Pine and Liboiron further explored how data science workers formulate their rules for defining what constitutes data, representation of data in a formal repository, and their combination process [53]. Muller et al. explored human formative work practices in data science and proposed five types of human interventions in relation to data and defined data as a human-influenced entity [46]. Our work is inspired by these sensemaking studies and proposes a task-based model to capture the inherently iterative phases of information extraction workflows. According to this model, tasks represent fine-grained goals within a phase. We decompose the tasks into user actions to capture transitions between the tasks within a specific phase and transition across phases. Such characterization helps us in formulating design principles for tools that can support IE as a single continuum.

2.3 Tools capturing the data science process

Users interact with multiple modalities within a data science workflow such as data, code, models. Such multi-modal interaction necessitates the usage of different types of tools. Data science workers use spreadsheets for exploring and manipulating data [47]. The exploratory nature of data science work often requires visual analytics, for example, TensorBoard module in TensorFlow [39, 48]. These visual analytic tools enable data scientists to understand their data set and develop models quickly. Computational notebooks (*e.g.,* JupyterLab [24], Jupyter Notebook [37]) have become increasingly popular among data scientists for organizing data science work. Novel techniques are also being developed to help them find, clean, recover, and compare code in notebooks [27]. There are also bespoke solutions such as Leam [57] and notebook extensions such as B2 [78], Glinda [15] developed to help data science workers in

various aspects. These advanced tools and features enable users to perform coordinated visualization and interactive data exploration. IE workflows necessitate the use of these tools in various phases. There are various machine learning libraries to support two fundamental tasks within IE workflows: named entity recognition [64] and relation extraction [10, 19]. We refer readers to existing surveys [32] for a more complete discussion of the broader literature. In this work, we discuss how existing tools lack proper support for IE workflows and propose additional design considerations to support rapid prototyping, interactive exploration, seamless documentation, and built-in provenance.

Earlier work such as CRISP-DM [63] and ASUM-DM [3] characterized data science as an iterative multi-phase process. Recent work on managing data science workflows [5, 42, 81] address challenges related to experimentation, reproducibility, and deployment within the same multi-phase framework. However, such characterization is coarse-grained and do not capture the corresponding user actions in each phase. In this work, we propose an iterative task model which generalizes to all the phases of an IE workflow and enables us to perform a fine-grained analysis of user actions within the workflow.

2.4 Data Science with Human-in-the-loop

Research on data science workflows with human-in-the-loop, *i.e.*, *DaSH*, have garnered significant interest in recent years [1, 17, 59]. DaSH is a paradigm wherein human users incorporate their knowledge into and intervene at various stages of a data science workflow — from data preparation and exploration to model building and evaluation [2, 52, 68]. Prior work examines roles of automation in DaSH [79] and informs design rules for related tools [80] through interview-based studies. Besides conducting interviews, we also examine these systems using heuristic evaluation methods [23, 49] to identify usability problems in the user interface design. For example, Gerhardt-Powals's cognitive engineering principles [23] leverage empirical findings from the cognitive sciences to inform the design of an interface. The principal has been widely used to evaluate human-in-the-loop interfaces in various domains such as e-commerce [12], computer-based testing [21], smartphone applications [33], among others. We focus on IE as a representative DaSH workflow due to its widespread usage in text analysis. We investigate how the cognitive engineering principals can guide the design of the corresponding human-in-the-loop IE tools and propose several design considerations.

3 STUDY DESIGN

The purpose of our study is to examine existing information extraction practices and identify associated limitations and potential improvement opportunities. Data science projects (including information extraction) can often take "months" to complete [73] and may involve exploring multiple approaches before a stable version with sufficient quality is developed. Given an information extraction project, we wanted to become aware of the decision-making process as data science workers reasoned over these strategies. Therefore, we opted for retrospective semi-structured interviews as our data-collection method. In particular, the study aimed at answering the following three research questions:

RQ1 : What type of (a) tasks and (b) actions do users perform in various phases of an information extraction project?
RQ2 : What are the challenges in accomplishing the tasks in various phases of information extraction?
RQ3 : How to improve users' experiences with their current information extraction workflows?

3.1 Participants and projects

We conducted ten interviews on ten separate projects involving information extraction at Megagon Labs, an industrial research lab, with natural language processing, data management, and machine learning as the primary research areas. Megagon Labs is a subsidiary of a large holdings and conducts research and development for the other subsidiaries with worldwide businesses in staffing, human resources, travel, marketing, and other online consumer services. We interviewed 10 data science workers at Megagon Labs, one from each project — in total, the projects involved 38 collaborators with at least two collaborators per project. We asked the participants to discuss the project retrospectively, from its inception to completion. The retrospective discussion-based setting enabled the participants to discuss their strategic reconsiderations in various phases of the projects.

We interviewed six researchers, three data scientists, and one graduate student working as a research intern at Megagon Labs. Note that none of the participants were authors of the paper and did not participate in the study design and analysis process. All researchers have Ph.D. degrees in computer science with experience ranging from one to five years in the industry. The data scientists have various degrees (BSc., MSc.) in fields such as computer science and statistics, with similar years of experience in the industry. The research intern holds a bachelor's degree in computer science and is currently a fourth-year Ph.D. student at a university focusing on natural language processing and machine learning. All of the participants had completed more than one large-scale information extraction project and are expert users of NLP/ML techniques, software, and libraries. 30% of participants were women, which compares favorably with recent estimates of 15% women in tenure-track faculty in computing [73] and 20% women in data science positions worldwide [36].

Projects. Each project in the study involved information extraction. Information extraction was the primary task for six projects, whereas information extraction was employed upstream to facilitate the downstream tasks for the rest of the projects. Many of these projects were accepted for publication in academic conferences ($N = 6$), open-sourced ($N = 7$), and deployed in a real-world setting as part of technology transfers ($N = 4$). All of these projects were collaborative. We describe the projects in more detail in Section 3.3.

3.2 Methods of inquiry and analysis

3.2.1 Semi-structured interviews. Each of the interviews lasted about an hour. Interviews were semi-structured, starting with background questions on the participants' roles, jobs, tools, datasets, teams, daily practices, followed by in-situ and follow-up project-specific questions that emerged during the interview process. The first author was the interviewer. The interviews were conducted

remotely in Zoom and recorded (both audio and video) using the built-in recording feature.

Following study introduction and background inquiries, we asked the participants to describe their project. We then asked them to explain their view of the phases and tasks within their IE workflow. Based on their description, we then further discussed each phase in detail in the selected project context. We discussed various challenges related to their workflow as and when the participants mentioned those. We also asked the participants to list desirable feature enhancements for improving their IE experience. Finally, we asked the participants specific questions on adoption, usage, and maintenance status of their projects. Before concluding the interview, we asked the participants to share any additional observations.

3.2.2 Analysis method. The audio recording of the interviews were transcribed using an automated transcription service. The authors further curated the transcriptions to address inconsistencies. As the interview contained discussions on diverse and complex topics, we used grounded theory method for coding and examination of concepts and relationships [11]. The interviews were then codified and analyzed using a qualitative data analysis software, Dedoose [62].

The coding process was inductive where concepts emerged based on the data analyzed and high-level categories — such as phases, tasks, actions, operations, challenges, enhancement requests — were used to help organize overarching themes. We extracted 493 passages from the interviews with each containing one key topic sentence. As in grounded theory analysis, we started with open coding. We identified concepts and properties, followed by axial coding, where we aimed to relate the concepts and identify the context of these relationships. All coding was done by one researcher and refined through periodic overview and discussions with other researchers. We repeatedly examined the extracted passages in the interviews to look for additional evidence and to validate and revise our emergent understanding. These iterative analyses led to a core set of 41 axial codes, which we combined into 4 selective codes.

Presentation of quotes. In presenting participant responses in Sections 4, 5, and 6, we replaced or anonymized Megagon Labs-specific terminology. Since the details of the interview were generated via an automated transcription service, we further corrected spelling and grammatical mistakes in those documents by referring to the actual recording. Therefore, the quotes presented in this paper are essentially paraphrases.

Limitations. We acknowledge that our study is limited in scope, specifically targeting information extraction in the broader landscape of natural language processing tasks, and also in sample, as they were situated in an industrial setting. Furthermore, even though the projects involved several subsidiaries of a large holdings, all participants were employed at a single company. The choice of participants inevitably impacted the observed practices due to organizational norms, policies, and infrastructures. We also acknowledge that each interview was conducted with only one collaborator per project, as such our interpretations and broad conclusions may be limited. Although data science workflows nowadays are collaborative, our study focused on individual work practices while briefly exploring collaboration in the context of project planning. We acknowledge this limitation and propose the analysis of the collaborative aspects of IE workflows as future work.

3.3 Projects

We grouped the projects into four themes: entity extraction (EE), entity matching (EM), knowledge-base construction (KBC), and content generation (CG).

P1. Salient fact extraction: Company Reviews (EE): four researchers and a data scientist worked to develop supervised and unsupervised techniques for identifying salient facts such as fine-grained details, about companies from a large number of reviews. Using language models such as BERT [16], the team extracted salient facts from a dataset of five million company reviews using about ten thousand labeled data.

P2. Aspect-opinion Extraction: Hotel Reviews (EE): two researchers and a data scientist developed techniques for extracting aspect-opinion pairs from a dataset of about a million hotel reviews using only few thousand labeled reviews [44]. They leveraged BERT, fine-tuned augmented data, to extract about three million aspect-opinion pairs.

P3. Structured data extraction: Job Benefits (EE): a data scientist, in collaboration with a researcher, developed and compared several text extraction techniques [32], including rule-based and deep-learning (BERT) based techniques over a dataset with about one hundred thousand company reviews to collect list of benefits provided to employees.

P4. Knowledge-base creation: Product Reviews (KBC): a team of six researchers developed unsupervised methods to capture implications between opinions from user reviews using matrix factorization [9]. The resulting framework was used to extract millions of opinions from reviews across multiple domains such as hotels, restaurants, movies, and travel.

P5. Knowledge-base extension: QA (KBC):. a team of five researchers developed a system to extract millions of tuples from two real-world QA datasets, with more than 300,000 question-answer pairs, to extend the concepts within domain-specific knowledge-bases [8]. Given a sequence-to-sequence learning framework, the system combines distributed representations of a question and an answer to generate facts.

P6. Knowledge-base population: QA and Reviews (KBC): five researchers developed an entity set expansion method [65] for populating facts in knowledge-bases from community QA and reviews. The system used two real-world datasets on community QA ($\approx 1M$ QA pairs) and customer reviews ($> 5M$ reviews) to populate the concepts in the knowledge-base.

P7. Entity matching: Job - Candidate (EM): one researcher and a data scientist developed an entity matching solution to match job seekers to open job positions. The task was performed by combining transformers-based models and content structure-aware pooling methods. The system was validated on a synthetic dataset of five thousand candidate profiles and half a million job postings job postings [41].

P8. Explainable summarization: Reviews (CG): a research team of four developed techniques for producing explainable, easy-to-interpret abstractive summaries from reviews. The team utilized

a sequence-to-sequence deep learning model to generate generate the restaurent summaries from 600,000 reviews of nine thousand restaurants in the Yelp dataset [71].

P9. Controllable Summarization: Reviews (CG): a team of five researchers developed an unsupervised system for extractive opinion summarization. The system utilizes vector-quantized variational autoencoders to extract popular opinions to generate summarized text from reviews [4]. The system was employed on 1.1M Tripadvisor hotel reviews.

P10. Topic modeling: Article Topics (CG): two data scientists developed techniques to generate 100 news article topics from five million user reviews of companies. They applied a deep-learning method to extract aspects and then applied LDA [74] over text spans to generate the article topics.

4 IE TASKS AND PHASES (RQ1A)

Based on participants' explanations of their projects, we divided an IE workflow into four phases: data preparation, model building, model evaluation, and deployment. Each phase consists of multiple tasks and corresponding user action(s). In this section, we first define the tasks and then describe each of the phases of an IE workflow in the context of these tasks.

4.1 The task model of IE workflows

All of the IE phases involve five high-level tasks:

- **View** refers to the act of viewing data. In this work, we refer to data in its most general sense, *i.e.*, data can be a raw dataset or derived information (*e.g.*, extractions and their labels, evaluation metrics).
- **Assess** refers to the act of reconnaissance over the presented information, for example, examining data distribution, finding interesting patterns.
- **Hypothesize** refers to the act of reasoning over the observations to define model semantics.
- **Pursue** refers to the act of actualizing the hypothesized model over the data. We refer to a model as any process derived based on a hypothesis that users can execute. Examples of models include extraction model, data cleaning operation, issuance of crowdsourcing tasks.
- **Verify** refers to the act of qualitatively and quantitatively evaluating the model outcomes. Verify is a special case of the assess task only focused on the evaluation of a user's pursued action.

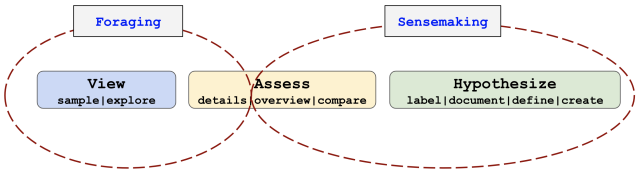

Figure 2: Both Foraging and sensemaking loops emerge across tasks within the task model.

The tasks, phases, and their dynamics within an IE workflow can be explained using the notional model of sensemaking [55]. Figure 2 maps the IE tasks to the loops within the model. The overall process is organized into two major loops — a foraging loop [54] and a sensemaking loop [61]. The foraging loop, which involves the view task and the *details on demand* action corresponding to the assess task (see Table 1), enables users to form an understanding of the data. The sensemaking loop, which involves the hypothesize task and the rest of the actions in the assess tasks, helps define semantics for the downstream tasks to be pursued. Both the loops also emerge during verification tasks (see Figure 4 in Section 5.2). Figure 3 shows how both these loops emerge in all the phases.

4.2 Characterization of IE phases

4.2.1 Data Preparation. The first step in this phase is data understanding (*i.e.*, foraging) where users **view** a sample of the entire dataset, often via eyeballing, to **assess** the data domain (*e.g.*, subjective reviews, factual information), its structure (*e.g.*, tabular, semi-structured), and quality (*e.g.*, cleanliness). When users find inconsistencies within the dataset, they reason over methods, *i.e.*, **hypothesize**, for cleaning the data. Users then **pursue** those methods, often selected based on experience, to transform the data into the desired format. The entire process is captured by the following participant comment: "*A lot of times it'll just be opening the data by hand, seeing what's the format, and getting a sense of what kind of information is in there and based on that either write a cleaning script to process things nicely [P10].*" Users may further **verify** the quality of the data and repeat the steps as mentioned above until the data achieves the desirable quality.

4.2.2 Model Building. In this phase, users again **view** a sample of the entire dataset to identify and examine, *i.e.*, **assess**, diverse and representative patterns and their distributions using methods such as clustering. One participant commented: "*So I would say that the challenge is mainly in the domain, what sort of information you're interested in finding whether it is opinions or not [P5].*" Based on their assessment users then define pattern semantics (**hypothesize**). Defining pattern semantics often involves multiple collaborators who label the patterns manually, then document their observations, and iteratively refine the pattern definitions to create a final rubric for extracting patterns. In the absence of ground truth labels or a benchmark, users **pursue** an additional process of labeled data collection. Data can be collected by assigning annotation tasks to in-house experts or crowdworkers (**pursue**). Once data collection is completed, users may further **verify** the annotation quality using qualitative (*e.g.*, eyeball) or quantitative (*e.g.*, annotator agreement) measures. Depending on the quality of the labeling, users may further **assess** the labels and redesign the annotation task based on discussion **hypothesize**.

After finalizing the extraction pattern definitions and pursuing the optional data collection step, users utilize the refined rules to create models (**pursue**) for extracting information from the dataset. Alternatively, users may reuse already existing models. In this stage of the workflow, users often lack an understanding of the final evaluation metric. So, users first **view** and **assess** the quality of the extractions using simple measures such as error rate and support count of rules. This observation is captured by the following participant comment: "*... we didn't always have a benchmark to begin with. The benchmarks came later, when we were finally evaluating this*

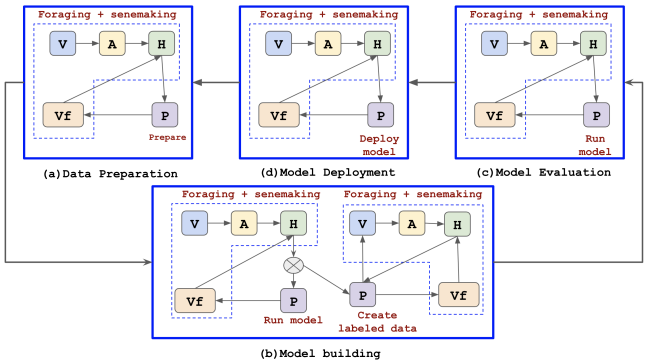

Figure 3: Phases of IE workflow depicted using the task model (V = View, A = Assess, H = Hypothesize, P = Pursue, and Vf = Verify). Counter-clockwise from left: (a) data preparation, (b) model building, (c) model evaluation, (d) model deployment. Each phase involves an initial foraging loop that explores information (*e.g.*, data, metrics, extractions) followed by sensemaking loop to formulate hypothesis for pursuing the next action (*e.g.*, clean, label, extract) via the pursue task. Users then verify the outcomes of the pursue task via foraging and through a sensemaking process reconsider their hypothesis or move on to the next logical phase.

system. Initially, when we are doing the exploration to know whether this idea works [P5]."

Based on the results, users may choose to select the current version of the model for a more rigorous evaluation in the model evaluation phase with the entire dataset. The same participant commented: "*The rules I will base on maybe 10 or 15 examples, but then I would test it on at least 100 or 200 examples to see whether things are making sense before running it on the entire corpus [P5].*" However, if the results are not satisfactory, users may explore the results in further detail to explain the model's behavior by investigating the positive and negative examples (**hypothesize**). Users then update the models accordingly (**pursue**). These tasks are repeated until a stable version of the model is developed.

4.2.3 Model Evaluation. In this phase, users employ the finalized version of the model on the entire dataset (**pursue**). Users then *verify* the model on the labeled dataset using metrics such as precision, recall, coverage, top-k extractions. By this stage, users have a better understanding of the suitable evaluation metric and their acceptable values. In this phase, users tend to evaluate the model on multiple datasets. Similar to the previous phase, based on the results users may select the current version of the model as the final one or perform further assessment of the model (**view** and **assess**). The assessment may result in new observations on the model behavior (**hypothesize**) and users augment the model accordingly (**pursue**). Users then again employ the model (**pursue**) and evaluate (**verify**). For example, one participant commented: "*... at times the rules will capture the 80% of the data, but I'm really interested in modeling the remaining 20% ... we wanted to figure out these long tail concepts ... We will run (model), look at the output, make a judgement of whether it is capturing the long tail, and go over again to change the rules a bit [P6].*"

At first glance, it may seem that both model building and model evaluation phases satisfy the same objective. However, model building is less restrictive and less rigorous than model evaluation. Model building is exploratory, where users operate on a small sample of data to formulate a basic understanding of the model behavior. Model evaluation, on the other hand, is confirmatory where users extrapolate their understandings on a larger scale or test the system rigorously (*e.g.*, addressing edge cases). One participant commented: "*... what is the quality of the things you extract? And because it fails, then you go back. You look at the examples and see, okay, so this is a noun phrase, but I don't want these noun phrases [P9].*"

4.2.4 Deployment. In an industry setting, deployment involves putting the finalized model into production (**pursue**). However, publication of a model in an academic setting is also quite similar where creators open-source their code or models for others to reproduce the solution. However, users need to monitor the performance of the deployed model continuously. The model performance may degrade for a new dataset or due to concept drift in existing data domain. As a result, the users may again repeat tasks such as **verify**, **assess**, and **hypothesize** to update the model. One participant commented: "*I've set up a dashboard with a number of plots ... ratio of positive and negative examples, distribution of extractions ... that monitoring is very important. when the data requirement changes (for the application) that's not going to be explicit. So you need to be able to detect when this change [P3].*"

4.3 Discussion and takeaways

4.3.1 The emergence of an iterative task model. The tasks in an IE phase can be repeated across multiple iterations (see Figure 3). These tasks emerge within a general flow of "view-assess-hypothesize-pursue-verify" in all phases of an IE workflow. Iteration can happen across phases. For example, in the aspect-opinion extraction project (*P2*), collaborators identified that a collection of duplicated reviews generated by bots were impacting the model performance, which they addressed in the next iteration by employing deduplication strategies: "*So if the results are not good we will do our analysis ... we retraced back to the original reviews and found that those are bots that are sending the same reviews multiple times.*" Therefore, the observation triggered a transition from the deployment phase to the data understanding and preparation phase.

4.3.2 The impact of iteration on an IE workflow. As described in Section 4.2.2 and Section 4.2.3, users often repeat the process of data collection and model evaluation to obtain a high quality ground truth and an efficient model, respectively. To construct the ground truth, users iterate over several versions of the labeled data as well metadata such pattern semantics, annotator agreement. Similarly, as users iterate over model versions they keep track of metadata related to model performance such as qualitative observations and quantitative metrics (*e.g.*, precision, recall error rate). For example, one participant commented: "*So, the crucial point being when you train the models, you will specify a lot of hyper-parameters. then we*

need to keep track which set of hyper-parameters resulting in what performance ... F1 score, accuracy [P7]."

Metadata management and provenance help users assess data and models. For example, users may assess labeling performance by comparing the annotator agreement rate between subsequent iterations. Similarly, they may compare the error rate of different model versions. Based on the assessment users may decide to pursue next steps such as issuing new labeling task or updating existing model. We identified several challenges related to metadata management and provenance in their existing practices, which we discuss in Section 6.

5 IE TASKS AND USER ACTIONS (RQ1B)

We now characterize the user actions corresponding to the tasks across the IE phases (see Table 1). In particular, we identified that users' IE operations map to 17 unique user actions.

5.1 Characterization of user actions

5.1.1 View. Sampling and exploring data are two actions that correspond to the view task. The goal, as explained earlier, is to view data in its raw form. For example, viewing the dataset during data preparation (DP, hereafter), exploring labeled ground truth during model building (MB, hereafter), and analyzing the outputs of the models during model evaluation (ME, hereafter) and deployment (MD, hereafter). Note that the sample action is often a crucial first step in an IE phase as it makes the information perceptually scalable to the users and help them in qualitative analysis. One participant shared their sampling experience during data preparation (DP): "*... the dataset has more than a one million rows ... So we had to sample a subset like 10,000 rows [P1].*" View actions can be performed either programmatically (*e.g.*, random sampling) or by direct manipulation (*e.g.*, sampling data in a spreadsheet by scrolling.)

5.1.2 Assess. The assess task is performed in various phases to observe the presented information and ascertain interesting features or patterns. Examples include computing distributions or summaries (DP, ME), examining data corresponding to the summaries (DP, MB, ME), and comparing diversity of patterns (MB, ME, MD) (see Section 4.2). Such reconnaissance helps in hypothesis creation and decision making in subsequent steps of an IE phase. The operations corresponding to the assess task can be grouped into three high level actions: overview, details on demand, and compare.

Overview: Overview action enables users to get a birds eye view of the underlying information space. The high-level information often helps in isolating errors, surfacing diverse and representative patterns, and obtaining data summaries. For example, one participant used k-means clustering to evaluate data quality (DP): "*... we almost always end up with a cluster or two that are devoted purely to grammatical or spelling errors ... [P10].*" Besides clustering, another popular overview operation is computing distributions on various features of the data such as length of text and pattern count. These summaries help in decision making in the subsequent steps. For example, distribution of length of text in the corpus can influence the model building decisions as mentioned by one participant (MB): "*I want to try to understand how long texts tend to be if there is significant variety ... that's going to significantly affect your model ... that is a much harder problem than modeling texts that are all of a similar size [P3].*"

Details on demand: Another important action related to assessment is obtaining further details of the summary information. Seeking such details helps in confirming the observations gathered as users can understand the context of the overview by examining the raw data. For example, one participant commented how clustering helps while qualitatively evaluating models (ME): "*one thing that's very common is I'll try to cluster the text that I'm looking at. Then look at a few examples from each cluster ... And so looking at a few examples from each cluster means that hopefully I'm looking at very diverse examples ... [P3].*" All of the operations belonging to this action class involve searching for the raw data corresponding to the overview or summary. Users perform the action in various ways such as printing samples using the *grep* command in bash script or print command Python. For example, one participant commented how they assessed potential extraction rules (MB): "*... I just searched for the word (using grep) and got text preceding that word and following that word so that I don't read an entire three paragraph review. I would just read the sentence that mentions the word [P4].*"

Compare: Compare action enables users to compare and contrast the diverse information, for example, during model building as mentioned by one participant (MB): "*... at that point you have to go quickly glance at each cluster and get a sense of the main topic or what sets it apart. I then see if there's a few clusters that have the same labels, maybe revisit them and try to see what differences are there [P10].*" However, the compare operation is also used to assess models across iterations using metrics such as error rate, ratio of positive and negative examples, support count of an extraction pattern (ME). For example, one participant commented: "*... in that phase I kind of quickly do a sanity check that if the error rate matches what I recorded before and make sure the code doesn't have a bug [P4].*"

5.1.3 Hypothesize. The user actions corresponding to the hypothesize task often involve multiple collaborators reasoning over subjective observations to reach consensus, *e.g.*, defining pattern semantics, creating extraction rubric. We identified four actions corresponding to the hypothesize task: label data, document observations, define semantics, and create rubric.

Label data: The label action involves annotating information such as raw data (MB): "*... we start from labeling the data sets, with all of us label 1000s of sentences and we pick the top 10 that will feel the most salient from these labels we try to observe [P1].*" Users also annotate outputs of models during model evaluation (ME) to characterize model behavior.

Document observations: Participants documented their observations during hypothesis using handwritten (paper) or digital (spreadsheets) notes. For example, documenting potential extraction candidates with explanations during model building (MB): "*I'll actually try to record it down. For example, let's say if a review talks about directions and hotel, maybe my assumption was that that review is definitely talking about asking for directions to the hotel [P4].*" Similarly, to compare different candidate extractors, *i.e.*, models, participants may record performance, version, and the underlying heuristic of the corresponding model (ME): "*I look at 20 results and*

Table 1: Users actions per IE task and example of corresponding operations across phases (DP = data preparation, MB = model building, ME = model evaluation, MD = model deployment.)

Task	User Action	Phases	Operations
View	Sample	{DP, MB}	{scroll spreadsheet/text editor; sampling}
	Explore	{ME, MD}	{print; cat; view spreadsheet/text editor}
Assess	Overview	{DP, ME}	{feature distribution; clustering}
	Details	{DP, MB, ME}	{grep; view cluster members; view data corresponding to overview}
	Compare	{MB, ME, MD}	{compare cluster patterns; compare error rates}
Hypothesize	Label data	{MB, ME}	{label interesting patterns; label positive and negative examples}
	Document	{MB, ME, MD}	{performance log; handwritten or typed notes}
	Define	{DP, MB}	{define pattern semantics or HITs via collaboration with experts}
	Create rubric	{DP, MB, MD}	{finalize semantics; finalize rules}
Pursue	Prepare	{DP, MB}	{filter; replace}
	Expertsource	{MB}	{issue labeling task from experts}
	Crowdsource	{MB}	{issue labeling task from crowd workers }
	Create model	{DP, MB, ME}	{create rules; train model}
	Reuse Model	{DP, MB, MD}	{reusing existing models ; reuse existing rules }
	Update model	{DP, MB, ME}	{add or augment rules; tune parameters; update training data}
Verify	Validate	{DP, MB, ME}	{measure agreement, annotation quality, error rate, ratio of +ve and -ve examples}
	Evaluate	{ME, MD}	{measure accuracy, precision, recall, coverage, rule support count}

write down somewhere the quality of this heuristic . . . So I have a sense of how much noise I'm introducing (in later versions) [P4]."

Define semantics: Actions related to defining semantics involve multiple collaborators who often reach consensus via ad hoc discussions: "*It's subjective. We will have meetings with others for agreement . . . Do we actually add something or remove something or refine the label [P1].*" For example, one participants explained how they designed human intelligence tasks (HITs) for crowdsourcing labeled data (MB): "*. . . I will annotate and also, team members will annotate to achieve consensus of the labels. During this process, we will create our labeling instructions so that they are clear [P8].*"

Create Rubric: Using this action users map the defined semantics to actionable rules. For example, in the salient fact extraction project (*P1*) collaborators identified two properties of salient facts, which they later used for extraction (MB): "*we observed two characteristics. First one is the uncommon attributes, like some attribute that do not belong to all entities. And the second one is the scope, something measurable numerically.*"

5.1.4 Pursue. The actions corresponding to the pursue task operationalize a hypothesized model. Actions belonging to this class are employed for data cleaning (DP), issuing crowdsourced labeling tasks or creating new models (MB), reusing existing rules or models (MD). One participant commented: "*first we want to see if any of the existing models can accomplish the task. So, maybe we can use that model [P2].*" Revising models is also an example pursue action, which may involve adding new rules or training data as mentioned by one participant (ME): "*For example, you can add rules to in addition to the current model, or you like you can like collect more data. You usually we will need to collect more labeled data [P10].*"

5.1.5 Verify. The verify tasks aim at evaluating the outputs of an action corresponding to a pursue task. Two actions corresponding to this task are validate and evaluate. The actions related to *validation* are often informal measurements of user actions. For example, measuring crowdworker quality and annotator agreement for crowdsourcing action (MB). Other examples include assessing model performance during iterative refinement such as error rate and ratio of positive and negative examples per rule (ME). The evaluate action measures the performance of the actions pursued in the model evaluation and deployment phases (ME, MD). Examples include measuring precision and recall/coverage, rank of top-k, mean avg rank, accuracy and F1.

5.2 Discussion and takeaways

5.2.1 The exploration-confirmation loop. As shown in Figure 4, each phase in the IE workflow has two processes — an exploratory process and a confirmatory process. The exploratory process is bottom-up, *i.e.*, formulates semantics from data. The confirmatory process is top-down, *i.e.*, verifies the defined semantics using

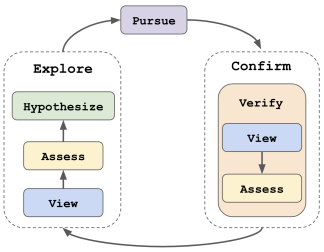

Figure 4: Inherently iterative processes of exploration and confirmation.

evidence. Users continue to iterate over the exploratory and confirmatory processes until stable semantics for the model is defined.

5.2.2 Human-in-the-loop model building. The projects in the study can be divided into two approaches: deep-learning-based and rule-based. From an implementation standpoint, the key difference between the two approaches would be the presence (*i.e.,* rule-based) or absence (*i.e.,* deep-learning-based) of feature-engineering. In deep-learning-based extraction, participants employed models such as BERT [16] trained on crowdsourced data. However, they were required to form an understanding of the data through labeling and documentation to define the semantics of HITs. On the other hand, in rule-based projects, participants performed data reconnaissance to define and handcraft extraction rules. While the goals of the two approaches are different, the underlying process remains the same where humans are involved in the loop for conceptualizing pattern semantics.

Similar sensemaking-oriented task model can be seen for defining semantics related to other phases such as model evaluation and deployment. For example, in the knowledge-base population project (*P6*) participants initially used precision and recall as metrics and later refined the final evaluation metric to be precision*@k* and recall*@k* due to the unbounded nature of the extraction sets. The underlying task model for creating the evaluation metrics were the same as shown in Figure 4: viewing a summary of performance (view), assessing the results by examining raw data (assess), documenting data and model characteristics and defining evaluation criteria (hypothesize), creating metrics accordingly (pursue), and verifying the metrics (verify).

6 IE WORKFLOW CHALLENGES (RQ2)

We now describe challenges related to various IE tasks.

6.1 Challenges of performing IE tasks

6.1.1 Difficulty in foraging information. Foraging information can be challenging as mentioned by one participant: "*The biggest challenge is generally knowing what information we are interested in extracting [P5].*" Following are some of the challenges as users seek information.

Perceptual scalability: One participant pointed out how perceiving even small samples of data can be challenging: "*It was difficult for us to explore a fraction of the data [P2].*" Another participant commented: "*... There's there's no really good way to verify rules (quality) because the corpus is huge [P4].*"

Lack of context: Participants also requested features to automatically highlight interesting information as they sought details on demand: "*... just having indicators that show this is a good example. This is negative and this is not [P3].*" While viewing information in context is helpful in understanding the data, features capturing such context are lacking in existing solutions: "*... show the results and let you explore those results interactively. I think even that is a really useful feature. Currently, for example, I will do that and go to my files and see what's wrong and change [P3].*"

Lack of semantic search capabilities: Spreadsheets and bash commands (*e.g.,* grep) lack advanced search capabilities. For example, the search and filter operations in spreadsheets are limited to exact match and don't consider semantic similarity. One participant commented: "*... maybe some kind of filtering of synonyms ... So for example we want to label everything with salary into benefits. There are many synonyms of salary. Especially I need to enumerate the synonyms myself ... because Google Sheets doesn't provide that functionality [P2].*" While bash commands such as grep are more expressive and support regular expression-based search, they also lack semantic search functionalities. For example, one participant requested searching by parts-of-speech tags with grep: "*The only thing that I wish grep had is that ... I would (search) delicious followed by two (placeholder) words and then a NOUN, as opposed to say, food [P4].*"

Direct manipulation vs. programmable search: While computational notebooks enable users to implement bespoke semantic search, they impede free-form data exploration due to a lack of direct manipulation capabilities. One participant commented:"*main downside of notebooks is sometimes its harder to dig deep into the data. Because the print field is a bit limited. And you can't do things like sorting or editing data [P10].*"

6.1.2 Difficulty in sensemaking. While sensemaking is crucial for users to make informed decisions about their subsequent actions, there are several challenges with the existing set up.

Difficulty in qualitative validation: To qualitatively assess extractions users need to manually explore the raw data corresponding to extractions, which can be cumbersome as mentioned by one participant: "*... the other one (recall) is a little bit more tricky, because you need to read reviews ... once you implemented your extraction then you need to go back and see, was there something you missed? ... I would create another notebook or program and repeatedly print out documents (reviews). That's a very bad experience [P8].* Similar observations can be found for debugging of extractions: *... what we have to do, which is actually very tedious, is to get an extraction, go back and see where it came from. Or get sentences that didn't have extractions or suspiciously too few extractions to see if that made sense or we missed something [P10].*"

Labor intensive documentation: The documentation process is even more tedious and time consuming as users need to reason over the presented information and then document their observations. For example, in the structured data extraction project (*P3*), the participants maintained a log of the errors of rules and their explanation in Jupyter Notebook cells: "*I'll copy all of the code within*

that cell, paste it in a cell down below it, make modifications there to the rule, and then go through that process again."

Lack of an overview interface: Due to the lack of a built-in overview interface that conveys diverse and representative information, participants faced difficulty in comprehending the information space and often resorted to creating custom features to obtain overviews or summaries from the data. One participant commented: "*... what we need right now is a visualization UI. So we hope that there will be like a visualization ready for us to for example, help us draw key features to analyze prediction results [P1].*" Another participant requested: "*... a clustering process that's part of the labeling process that you can label the most important examples instead of just labeling random things [P3].*"

Providing interactive feedback: Participants also requested automated mechanisms for focusing their attention during the sensemaking process. For example, one participant requested an alerting feature to inform data quality issues after annotation by crowdworkers or experts: "*... there is a control board that shows me all these statistics and sends me alerts. For example, annotation alerts that with these we were not able to train a (quality) model [P8].*" The same participant requested automated notification on negative model performance: "*... notify training model doesn't converge. It will definitely make my life easier [P8].*"

Facilitating comparison: Another important action within the sensemaking process is comparison such as comparing model performance or impact of different parameter settings. One participant commented: "*... say we have 10 different models. And then I'm pretty sure I like to like do analysis in an interactive manner, so that I can pick up the best model for the purpose. Otherwise, I think it'd be time consuming see 10 different outputs to from the result file [P9].*" Participants also requested what-if style dashboard (similar to LIT [69]) to perform counterfactual analysis: "*... a table where I have some columns that could be used as parameters, and then I have the true label and the predicted label as two columns. Then that would be all you need to feed into (a system) that outputs a plot where you can select and parameters and their values and see how the metrics change [P10].*"

6.1.3 Difficulty in "pursue" tasks: The challenges related to the pursue task are often action specific such as labeling, augmenting models.

Labeling and cognitive burden: The labeling of data often puts cognitive burden on the users as mentioned by one participant: "*So it was fairly labor intensive labeling. I would end up spending a full 30 seconds to a minute on each entry so it wasn't very efficient [P10].*" While there are various solution now available to make labeling easier, they often do not capture the entire spectrum of labeling requirements [6]. In multiple projects (*P2*, *P8*, *P9*, *P10*), participants had to perform bespoke labeling tasks that were not supported by existing tools.

No-code/low-code model configuration: While reusing a model, participants requested GUI-based features to configure model parameters, inputs, and outputs. One participant commented: "*I can just do some very high level configuration that doesn't require any understanding of the code structure. I say column A is the input, column B is the classifier output and then do all the work without writing the code [P8].*" Participants even created custom wrappers to models for configuring parameters. For example, one participant implemented a parameterized search function over a bash command: "*For some projects I was doing this (grep) so much that I created aliases in the shell script to search with parameters like fetch delicious and food with words in between ... I didn't want to type that long regex command which is about the spaces in between. So if there was only something that would do that (automatically) [P4].*"

6.2 Challenges related to iteration

From the interviews, we identified several challenges related to metadata management and provenance that participants experienced to keep track of both task outcomes and data across iterations. We also observed challenges in context switching due the iterative nature of IE workflows.

6.2.1 Bespoke provenance management. Users often manually created data provenance mechanisms to keep track of data across iterations. Integrating provenance practices in the IE workflow enables users to explore the lineage of data, extractions, and models. For example, one participant commented: "*So basically every step of the extraction we just append new information to the reviews ... aspect and opinions. we keep track of which tokens are these aspects coming from? also the character ID, the token ID, sentence ID. This is something that we developed manually [P4].*" Therefore, the onus is on the user to integrate provenance measures. Absence of such a measure can lead to loss of information, which can hamper the IE task as captured by the following participant comment: "*I never recorded the sample. Later, when there was a mistake, I could not find it (the samples), because I was not sure which slice of the data I got it from. ... because random sampling doesn't produce the same seed, it doesn't give me the same instances [P4].*"

6.2.2 Cumbersome experiment tracking. Metadata can be both quantitative and qualitative. To track quantitative metadata (*e.g.*, metrics, parameters) related to their pursued actions, users employ various strategies such as naming result files with tags related to various metadata. One participant commented: "*I tend to be very verbose in my file names and essentially include almost every parameter that went into training, ends up expressed in the filename ... And that becomes really cumbersome to keep track of [P5].*" Another participant created separate log files for each iteration and added the logs in a spreadsheet: "*after I finish (model building) at that point I'll be in Google Sheets, and I'll have essentially a separate sheet for each iteration, and then I'll generally have a single master sheet at the beginning, that has a summary data statistics and parameters for each of the sheets comprising the file [P10].*"

6.2.3 Labor intensive metadata tracking. Tracking qualitative metadata (*e.g.*, user comments, documentation) can be an even more tedious experience as these are often verbose comments or documents shared among collaborators. For example, in structured data extraction project (*P3*), the users maintained a log of the errors of rules and their explanation in jupyter notebook cells, one cell for each rule version. The participant commented: "*(For each rule version) I will document the examples that were wrong grouped by the reason for why each one was wrong. Once I'm done with the process, I essentially have N versions of the rule. And I can look at why and*

where each version the rule failed. So that I can easily understand the provenance of the rule and how and why I made improvements."

6.2.4 Tedious context switching. Participants employed various tools such as spreadsheets, computational notebooks, and bash commands, for completing their tasks. However, each tool has usage across multiple tasks. For example, spreadsheets were used for viewing data (view), labeling patterns (hypothesize), preparing/cleaning sample data (pursue). Computational notebooks were used across all the tasks. Bash commands were used for viewing sample data (view), regular expression-based search (assess), and running models (pursue). Since both the phases and tasks are iterative, users were often forced to move back and forth between multiple tools as they accomplished their IE workflow, which can be cumbersome.

7 DESIGN CONSIDERATIONS FOR IE TOOLS (RQ3)

In this section, we distill several design consideration for supporting human-in-the-loop IE workflows based on the observed challenges. We discuss the design considerations at both feature- and system-level. We situate our discussion in the context of Gerhardt-Powals's cognitive engineering principles [23], a widely used heuristic evaluation method for evaluating human-in-the-loop interfaces (see Section 2.4).

Cognitive engineering principles. We focus on the following principles: automating unwanted workload (*CP*1), reducing uncertainty of information (*CP*2), fusing data to provide high level abstraction (*CP*3), using known metaphors for ease of interpretation (*CP*4), displaying information in a logical manner (*CP*6), providing visual aids during information seeking (*CP*7), maintaining context of current focus (*CP*8), and presenting information at multiple levels of detail (*CP*9). Of the two remaining principals, one principal promotes context-dependant naming of actions, which is observed by any standard systems nowadays. The other principal on judicious redundancy is related to the design and organization of interface components, which is beyond scope of this discussion.

7.1 Feature-level considerations

D1. Facilitate advanced search capabilities: Search and filter operations help focus a user's attention to information being explored (*CP*8). An IE tool should support typical semantic search functionalities — *e.g.,* search by synonyms, POS tags, regular expressions — as defaults (*CP*4). Besides programmatic specification (*e.g.,* python script), users should be able to specify these operations either via direct manipulation, *e.g.,* a menu bar (*CP*1). For example, TextEdit [13] is a wrapper for Pandas dataframe [67] that enables users to programmantically perform semantic search.

D2. Provide interactive feedback: Interactive feedback can help in improving trustworthiness of a user action while focusing user attention to the desired information (*CP*2, *CP*8). Participants requested interactive feedback for various user actions across tasks for highlighting information (in overview, details on demand actions) and conveying updates or alerts (in validate and evaluate actions). Feedback should be automatically provided as visual aids using known metaphors such as color highlight (*CP*1, *CP*4, *CP*7).

D3. Generate automated summary: Overview interfaces are extremely popular in the information visualization domain — overviews make the information space perceptually scalable thus reducing cognitive burden of users [25]. Viewing the information at multiple levels of detail, *i.e.,* summary and raw data, also provides more context to the users (*CP*3, *CP*9). The grouping within the overviews should be constructed automatically and should convey information in a meaningful (*e.g.,* clustering text by semantic similarity) and a visually consistent manner (*CP*1, *CP*6, *CP*7).

D4. Provide means for comparison: Comparison actions are fairly common while assessing data and models. However, comparison operations often lead to visual discontinuity of the information being explored leading to users loosing context of their task [56, 72]. The comparison feature should automatically convey comparative information meaningfully with visual aids, to enable informed decision making (*CP*1, *CP*6). Moreover, information should be conveyed at multiple levels of detail to add validity and reduce uncertainty (*CP*2, *CP*9). For example, highlighting difference between model performance across iterations via both charts and tables.

D5. Ensure ease of hypothesis creation: Hypothesize actions such as labeling and documentation are crucial for sensemaking and happen in a collaborative setting. As these actions are labor intensive, an IE tool should automate the process, for example, via recommendation of potential labels, explanations, metrics (*CP*1). Both labeling and documentation should be integrated as default features within any IE tool (*CP*9).

D6. Enable configurable actions: While not as common as the previous features, configurable pursue actions (*e.g.,* tuning models parameters) still have their benefits. Configurable user actions should be designed in a way such that unwanted workload of users are reduced (*CP*1). IE systems should introduce the "configurability" feature wherever appropriate.

7.2 System-level considerations

The system-level considerations are based on our observation of usage of tools across IE tasks and iterations. Users employ various tools to accomplish their goals — spreadsheets and bash commands for data exploration and preparation, computational notebooks and scripts for both model development and data exploration. However, none of these tools capture IE phases within a single continuum and users need to move back and forth between multiple tools which can be tedious. As mentioned earlier, while provenance and metadata management across iterations are crucial to IE, existing solutions lack built-in mechanisms to support such iterative process.

D7. Reduce context switching: Context switching is an unwanted workload that puts cognitive burden on the users and leads to loss of context [56, 72]. One approach to reducing context switching between tools is to design a system that groups multiple views related to an IE process, *i.e.,* data view, summary view, script/code view (*CP*1, *CP*9). For example, Leam [57], a general-purpose text analysis tool combines a spreadsheet, a code editor, and an interactive visualization pane to support integrated text analysis. However, a crucial requirement for such a system should be supporting metaphors that are already known to the users. Therefore, instead of developing a new solution, a better approach is to infuse those metaphors into existing solutions (*CP*4). For example, enhancing

Table 2: Design consideration for IE tools and their relationship with cognitive engineering principles [23].

Design considerations	Automate workload (CP1)	Reduce uncertainty (CP2)	Fuse data (CP3)	Known metaphor (CP4)	Group data (CP6)	Visual aids (CP7)	Focus attention (CP8)	Multilevel detail (CP9)
D1. Facilitate advanced search	✓	–	–	✓	–	–	✓	–
D2. Provide interactive feedback	✓	✓	–	✓	–	✓	✓	–
D3. Auto-generate summaries	✓	–	✓	–	✓	✓	–	✓
D4. Provide means for comparisons	✓	–	–	–	✓	✓	–	✓
D5. Ensure ease of hypothesis	✓	✓	–	–	–	–	–	✓
D6. Enable configurable actions	✓	–	–	–	–	–	–	–
D7. Reduce context switching	✓	–	–	✓	–	–	–	✓
D8. Support for provenance	✓	–	–	–	–	–	–	–

computational notebooks with extensions to include a data view and interactive visualizations.

D8. Support for provenance: As discussed in Section 6.2, IE workflows are iterative and require integration of provenance and metadata-management practices for ease of assessment and hypothesis. Therefore, features that enable automated provenance and metadata management should be promoted as first-class citizen of any IE system (*CP*1).

8 DISCUSSION

We now discuss the implications of the task model and our proposed design considerations for developing IE tools.

IE as a continuum: enhancement vs. creation. The notional model of IE tasks is inherently iterative and often requires users to make transitions to different phases or tasks on-demand, thus necessitating context switching between tools. The system-level design consideration (**D1**), outlined in Section 7.2, recommends that an IE tool should reduce such context switching to reduce cognitive burdens of the users. One possible approach is capturing the modalities — data, charts/summaries, code — within a single continuous process. Recent efforts in such combination have resulted in either creation of bespoke tools (*e.g.*, Leam [57]) or enhancements (*e.g.*, B2 [78], Lux [40]). These bespoke solutions often offer new interactions but are not feature-complete. Therefore, they suffer from a lack of adoption. Enhancements, on the other hand, often capture a subset of the modalities and requirements. However, iterative enhancement focusing on capturing the end-to-end process and the complete set of features offers more promise. In fact, Xin et al. [79] identified the self-sufficient end-to-end workflow support as a crucial factor in ease-of-use and efficiency of human-in-the-loop AutoML platforms.

Iterative planning and collaboration: a creative design perspective. A common theme across the projects is iterative hypothesis creation along multiple threads, such as model refinement and concretization of an evaluation metric. The hypothesize tasks are collaborative and are crucial for defining the action items for the subsequent pursue task. To this end, the hypothesize task is similar to the creative design tasks (*e.g.,* web design [7]), which also involves such iterative planning and collaboration. Stakeholders of the design task create and share unstructured documentation of requirements updated both synchronously and asynchronously. The documentations are then distilled into actionable plans either manually [7] or semi-automatically [58]. Throughout the task, stakeholders iterate over the plan. Similarly, the hypothesize task in IE contributes to iterative formation of rubrics such as extraction rules and evaluation metrics. Several approaches for model documentation (*e.g.,* Factsheets [20] and Model Cards [45]) and data documentation (*e.g.,* Datasheets [22] and Nutrition Labels [28]) have been proposed to be used as checklists to ensure the quality of models and data. However, recent work [45, 82] argue for greater standardization around documentation to record discussions and decisions made within data science workflows. Moreover, within a collaborative environment, where team members co-ideate and deliberate, any computer-mediated solution may be vulnerable to conflict and misunderstanding [76, 77]. To ensure ease of hypothesis, as recommended by design consideration **D5**, a better understanding of the pain points of such a human-in-the-loop process is crucial.

Maintenance and reproducibility: an afterthought or a necessity? Throughout various sections of the paper, we discussed how provenance and metadata management are required for reproducibility of IE workflow and effective monitoring and governance of data and models. Zhang et al. [82] identified lack of provenance as a contributing factor in obfuscation and loss of knowledge when data science teams share data. As outlined in our design consideration, **D8**, provenance and metadata management should be promoted as a first-class requirement of any information extraction tool. One approach can be to instill MLOps [42] practices by integrating suitable platforms with the tool. However, MLOps practices are designed to deploy and maintain machine learning models in production reliably and efficiently. Therefore, further research is required to identify ways to integrate such practices into research environments, which can be highly experimental and more iterative than production environments.

Human agency vs. automation. A key feature in all of the design considerations discussed in Section 7 was automating unwanted workload, one of the design principles of cognitive engineering. However, the tension between human agency and automation is long-discussed in HCI research [66] and poses vital challenges for designing and engineering data science platforms. How can we effectively integrate automated reasoning into interactive systems without impeding human agency? Recent work exploring human-AI collaboration [70] argues that a completely automated process may impede a data science worker's deep understanding of the data and models. They envision an *augmented data science* environment where data scientists, in collaboration with subject matter experts, steer automated agents to produce outputs that satisfy business goals. This viewpoint aligns with mixed-initiative systems [30] that aim to offer the best of both worlds with principles on when an automated agent should proactively take action and when a user should. Therefore, a more in-depth investigation of approaches

to reconciling agency and automation is required as the design considerations are incorporated into IE systems.

Broader implication of the task model. While we derived the task model based on grounded theory-based analysis of IE workflow practices, it is not clear whether the task model generalizes to any data science workflow. There are similarities between the phases of a typical data science workflow and our observed IE workflow. However, as shown in Figure 1 there may be additional operations and modeling requirements depending on the task at hand. Moreover, as mentioned in Section 2.1, data science workflows may involve other stakeholders beyond data scientists, *e.g.,* product managers. Besides exploring the generalizability of the task model, a deeper investigation of various tasks is thus required. For example, tasks such as assess and verify are often associated with trust and interpretability in data science. Passi and Jackson [51] argued that the perceived value of the quantitative metrics of assessment and verification may vary across stakeholders. Therefore, understanding how to work with such plastic nature of quantified trust is crucial when designing trustworthy systems involving multiple stakeholders.

9 CONCLUSION

In this paper, we presented a semi-structured interview-based study to understand IE work practices and observed an iterative fine-grained task model that emerged across all the phases. We identified several challenges with the existing IE workflows and proposed a set of design considerations, based on cognitive engineering principals, for developing human-in-the-loop IE tools. The design principals may guide the development of future tools and help identify enhancement opportunities within existing systems. Therefore, an immediate next step is to operationalize the principals within popular data science environments such as computational notebooks and conduct large-scale studies to evaluate their effectiveness. Moreover, the study can be extended to understand the role of collaboration in the task model and characterize the pain points related to aspects such documentation, hypothesis formulation, and conflict resolution. Finally, additional studies can be conducted to understand the broader implication of the task model along dimensions such as generalizability, trust, and interpretability within the human-in-the-loop data science setting.

ACKNOWLEDGMENTS

Conducting the study in the industry setting required a larger cast of characters than the author list on this paper does justice. Our study participants set aside time for interview and reflection. Our anonymous reviewers provided valuable feedback: their suggestions immensely improved the quality of the paper. We're grateful for their contributions to this work.

REFERENCES

[1] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35, 4 (2014), 105–120.

[2] Saleema Amershi, Max Chickering, Steven M Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. 2015. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 337–346.

[3] Santiago Angée, Silvia I Lozano-Argel, Edwin N Montoya-Munera, Juan-David Ospina-Arango, and Marta S Tabares-Betancur. 2018. Towards an improved ASUM-DM process methodology for cross-disciplinary multi-organization big data & analytics projects. In *International Conference on Knowledge Management in Organizations*. Springer, New York, NY, USA, 613–624.

[4] Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata. 2021. Extractive opinion summarization in quantized transformer spaces. *Transactions of the Association for Computational Linguistics* 9 (2021), 277–293.

[5] Rob Ashmore, Radu Calinescu, and Colin Paterson. 2021. Assuring the machine learning lifecycle: Desiderata, methods, and challenges. *ACM Computing Surveys (CSUR)* 54, 5 (2021), 1–39.

[6] Jürgen Bernard, Marco Hutter, Michael Sedlmair, Matthias Zeppelzauer, and Tamara Munzner. 2021. A Taxonomy of Property Measures to Unify Active Learning and Human-Centered Approaches to Data Labeling. *ACM Trans. Interact. Intell. Syst.* 11, 3–4, Article 20 (Aug. 2021), 42 pages.

[7] Aditya Bharadwaj, Pao Siangliulue, Adam Marcus, and Kurt Luther. 2019. Critter: Augmenting Creative Work with Dynamic Checklists, Automated Quality Assurance, and Contextual Reviewer Feedback. In *Proceedings of CHI 2019*. ACM, New York, NY, USA, 1–12.

[8] Nikita Bhutani, Yoshihiko Suhara, Wang-Chiew Tan, Alon Halevy, and HV Jagadish. 2019. Open Information Extraction from Question-Answer Pairs. In *Proceedings of NAACL-HLT*. ACL, Stroudsburg,PA, USA, 2294–2305.

[9] Nikita Bhutani, Aaron Traylor, Chen Chen, Xiaolan Wang, Behzad Golshan, and Wang-Chiew Tan. 2020. SAMPO: Unsupervised Knowledge Base Construction for Opinions and Implications. In *AKBC*. Openreview, Openreview.net, 1–17.

[10] Razvan C Bunescu and Raymond J Mooney. 2007. Extracting relations from text: From word sequences to dependency paths. In *Natural language processing and text mining*. Springer, New York, NY, USA, 29–44.

[11] Akemi T Chatfield, Vivian N Shlemoon, Wilbur Redublado, and Faizur Rahman. 2014. Data scientists as game changers in big data environments. In *Proc. ACIS*. ACIS, Australia, 1–11.

[12] Sherry Y Chen and Robert D Macredie. 2005. The assessment of usability of electronic shopping: A heuristic evaluation. *International journal of information management* 25, 6 (2005), 516–532.

[13] Codait. 2021. *Text Extensions for Pandas*. IBM. Retrieved March 17, 2021 from https://codait.github.io/text-extensions-for-pandas/

[14] Jim Cowie and Wendy Lehnert. 1996. Information extraction. *Commun. ACM* 39, 1 (1996), 80–91.

[15] Robert A DeLine. 2021. Glinda: Supporting Data Science with Live Programming, GUIs and a Domain-specific Language. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–11.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[17] Eduard Dragut, Yunyao Li, Lucian Popa, and Slobodan Vucetic (Eds.). 2021. *Proceedings of the Second Workshop on Data Science with Human in the Loop: Language Advances*. ACL, Online. https://aclanthology.org/2021.dash-1.0

[18] Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications* 165 (2021), 113679.

[19] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Commun. ACM* 51, 12 (2008), 68–74.

[20] Sebastian S Feger, Sünje Dallmeier-Tiessen, Paweł W Woźniak, and Albrecht Schmidt. 2019. The Role of HCI in Reproducible Science: Understanding, Supporting and Motivating Core Practices. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–6.

[21] Glenn Fulcher. 2003. Interface design in computer-based language testing. *Language testing* 20, 4 (2003), 384–408.

[22] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2018. Datasheets for datasets.

[23] Jill Gerhardt-Powals. 1996. Cognitive engineering principles for enhancing human-computer performance. *International Journal of Human-Computer Interaction* 8, 2 (1996), 189–211.

[24] Brian Granger, Chris Colbert, and Ian Rose. 2017. JupyterLab: The next generation jupyter frontend.

[25] Jonathan Grudin. 2001. Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, New York, NY, USA, 458–465.

[26] Philip J Guo, Sean Kandel, Joseph M Hellerstein, and Jeffrey Heer. 2011. Proactive wrangling: Mixed-initiative end-user programming of data transformation scripts. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, New York, NY, USA, 65–74.

[27] Andrew Head, Fred Hohman, Titus Barik, Steven M. Drucker, and Robert DeLine. 2019. *Managing Messes in Computational Notebooks*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300500

[28] Sarah Holland, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. 2018. The dataset nutrition label: A framework to drive higher data quality standards.

[29] Sungsoo Ray Hong, Jessica Hullman, and Enrico Bertini. 2020. Human factors in model interpretability: Industry practices, challenges, and needs. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (2020), 1–26.

[30] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 159–166.

[31] Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. ACL, Stroudsburg,PA, USA, 1148–1158.

[32] Jing Jiang. 2012. Information extraction from text. In *Mining text data*. Springer, New York, NY, USA, 11–41.

[33] Ger Joyce and Mariana Lilley. 2014. Towards the development of usability heuristics for native smartphone mobile applications. In *International Conference of Design, User Experience, and Usability*. Springer, New York, NY, USA, 465–474.

[34] Eser Kandogan, Aruna Balakrishnan, Eben M. Haber, and Jeffrey S. Pierce. 2014. From Data to Insight: Work Practices of Analysts in the Enterprise. *IEEE Computer Graphics and Applications* 34, 5 (2014), 42–50. https://doi.org/10.1109/MCG.2014.62

[35] Mary Beth Kery, Marissa Radensky, Mahima Arya, Bonnie E John, and Brad A Myers. 2018. The story in the notebook: Exploratory data science using a literate programming tool. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–11.

[36] John King and Roger Magoulas. 2015. *2015 data science salary survey*. O'Reilly Media, Incorporated, Sebastopol, CA, USA.

[37] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B Hamrick, Jason Grout, Sylvain Corlay, et al. 2016. *Jupyter Notebooks-a publishing format for reproducible computational workflows*. Vol. 2016. IOS Press, Amsterdam, Netherlands.

[38] Hanna Köpcke and Erhard Rahm. 2010. Frameworks for entity matching: A comparison. *Data & Knowledge Engineering* 69, 2 (2010), 197–210. https://doi.org/10.1016/j.datak.2009.10.003

[39] Georgia Kougka, Anastasios Gounaris, and Alkis Simitsis. 2018. The many faces of data-centric workflow optimization: a survey. *International Journal of Data Science and Analytics* 6, 2 (2018), 81–107.

[40] Doris Jung-Lin Lee, Dixin Tang, Kunal Agarwal, Thyne Boonmark, Caitlyn Chen, Jake Kang, Ujjaini Mukhopadhyay, Jerry Song, Micah Yong, Marti A Hearst, et al. 2021. Lux: Always-on Visualization Recommendations for Exploratory Data Science.

[41] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, Jin Wang, Wataru Hirota, and Wang-Chiew Tan. 2021. Deep Entity Matching: Challenges and Opportunities. *Journal of Data and Information Quality (JDIQ)* 13, 1 (2021), 1–17.

[42] Sasu Mäkinen, Henrik Skogström, Eero Laaksonen, and Tommi Mikkonen. 2021. Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help?

[43] Yaoli Mao, Dakuo Wang, Michael Muller, Kush R Varshney, Ioana Baldini, Casey Dugan, and Aleksandra Mojsilović. 2019. How data scientistswork together with domain experts in scientific collaborations: To find the right answer or to ask the right question? *Proceedings of the ACM on Human-Computer Interaction* 3, GROUP (2019), 1–23.

[44] Zhengjie Miao, Yuliang Li, Xiaolan Wang, and Wang-Chiew Tan. 2020. Snippext: Semi-supervised opinion mining with augmented data. In *Proceedings of The Web Conference 2020*. ACM, New York, NY, USA, 617–628.

[45] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*. ACM, New York, NY, USA, 220–229.

[46] Michael Muller, Ingrid Lange, Dakuo Wang, David Piorkowski, Jason Tsay, Q Vera Liao, Casey Dugan, and Thomas Erickson. 2019. How data science workers work with data: Discovery, capture, curation, design, creation. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. ACM, New York, NY, USA, 1–15.

[47] Bonnie A Nardi and James R Miller. 1991. Twinkling lights and nested loops: distributed problem solving and spreadsheet development. *International Journal of Man-Machine Studies* 34, 2 (1991), 161–184.

[48] Vinh Nguyen, Tommy Dang, and Fang Jin. 2018. Predict saturated thickness using tensorboard visualization.

[49] Jakob Nielsen and Rolf Molich. 1990. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, New York, NY, USA, 249–256.

[50] Samir Passi and Steven Jackson. 2017. Data vision: Learning to see through algorithmic abstraction. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*. ACM, New York, NY, USA, 2436–2447.

[51] Samir Passi and Steven J Jackson. 2018. Trust in data science: Collaboration, translation, and accountability in corporate data science projects. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–28.

[52] Kayur Patel, Naomi Bancroft, Steven M Drucker, James Fogarty, Andrew J Ko, and James Landay. 2010. Gestalt: integrated support for implementation and analysis in machine learning. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, New York, NY, USA, 37–46.

[53] Kathleen H Pine and Max Liboiron. 2015. The politics of measurement and action. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 3147–3156.

[54] Peter Pirolli and Stuart Card. 1999. Information foraging. *Psychological review* 106, 4 (1999), 643.

[55] Peter Pirolli and Stuart Card. 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, Vol. 5. McLean, VA, USA, 2–4.

[56] Sajjadur Rahman, Mangesh Bendre, Yuyang Liu, Shichu Zhu, Zhaoyuan Su, Karrie Karahalios, and Aditya Parameswaran. 2021. NOAH: Interactive Spreadsheet Exploration with Dynamic Hierarchical Overviews. *Proceedings of the VLDB Endowment* 14, 6 (2021), 970–983.

[57] Sajjadur Rahman, Peter Griggs, and Çağatay Demiralp. 2021. Leam: An Interactive System for In-situ Visual Text Analysis. In *CIDR*. CIDRDB, cidrdb.org, 1–7.

[58] Sajjadur Rahman, Pao Siangliulue, and Adam Marcus. 2020. MixTAPE: Mixed-initiative Team Action Plan Creation Through Semi-structured Notes, Automatic Task Generation, and Task Classification. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW2 (2020), 1–26.

[59] Gonzalo Ramos, Christopher Meek, Patrice Simard, Jina Suh, and Soroush Ghorashi. 2020. Interactive machine teaching: a human-centered approach to building machine-learned models. *Human–Computer Interaction* 35, 5-6 (2020), 413–451.

[60] Adam Rule, Aurélien Tabard, and James D Hollan. 2018. Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–12.

[61] Daniel M Russell, Mark J Stefik, Peter Pirolli, and Stuart K Card. 1993. The cost structure of sensemaking. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. ACM, New York, NY, USA, 269–276.

[62] Michelle Salmona, Eli Lieber, and Dan Kaczynski. 2019. *Qualitative and mixed methods data analysis using Dedoose: A practical approach for research across the social sciences*. Sage Publications, Thousand Oaks, CA, USA.

[63] Colin Shearer. 2000. The CRISP-DM model: the new blueprint for data mining. *Journal of data warehousing* 5, 4 (2000), 13–22.

[64] Hemlata Shelar, Gagandeep Kaur, Neha Heda, and Poorva Agrawal. 2020. Named entity recognition approaches and their comparison for custom ner model. *Science & Technology Libraries* 39, 3 (2020), 324–337.

[65] Jiaming Shen, Zeqiu Wu, Dongming Lei, Jingbo Shang, Xiang Ren, and Jiawei Han. 2017. Setexpan: Corpus-based set expansion via context feature selection and rank ensemble. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, New York, NY, USA, 288–304.

[66] Ben Shneiderman and Pattie Maes. 1997. Direct manipulation vs. interface agents. *interactions* 4, 6 (1997), 42–61.

[67] LA Snider and SE Swedo. 2004. PANDAS: current status and directions for research. *Molecular psychiatry* 9, 10 (2004), 900–907.

[68] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S Tan. 2009. EnsembleMatrix: interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, New York, NY, USA, 1283–1292.

[69] Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. 2020. The Language Interpretability Tool: Extensible, Interactive Visualizations and Analysis for NLP Models. In *EMNLP*. ACL, Stroudsburg,PA, USA, 107–118.

[70] Dakuo Wang, Justin D Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. 2019. Human-ai collaboration in data science: Exploring data scientists' perceptions of automated ai. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–24.

[71] Xiaolan Wang, Yoshihiko Suhara, Natalie Nuno, Yuliang Li, Jinfeng Li, Nofar Carmeli, Stefanos Angelidis, Eser Kandogann, and Wang-Chiew Tan. 2020. ExtremeReader: An interactive explorer for customizable and explainable review summarization. In *Companion Proceedings of the Web Conference 2020*. ACM, New York, NY, USA, 176–180.

[72] Jennifer Watts-Perotti and David D Woods. 1999. How experienced users avoid getting lost in large display networks. *International Journal of Human-Computer Interaction* 11, 4 (1999), 269–299.

[73] Samuel F Way, Daniel B Larremore, and Aaron Clauset. 2016. Gender, productivity, and prestige in computer science faculty hiring networks. In *Proceedings of the 25th International Conference on World Wide Web*. ACM, New York, NY, USA, 1169–1179.

[74] Xing Wei and W Bruce Croft. 2006. LDA-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 178–185.

[75] Karl E Weick. 1995. *Sensemaking in organizations*. Vol. 3. Sage Publications, Thousand Oaks, CA, USA.

[76] Mark E Whiting, Allie Blaising, Chloe Barreau, Laura Fiuza, Nik Marda, Melissa Valentine, and Michael S Bernstein. 2019. Did it have to end this way? Understanding the consistency of team fracture. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–23.

[77] Mark E Whiting, Irena Gao, Michelle Xing, N'godjigui Junior Diarrassouba, Tonya Nguyen, and Michael S Bernstein. 2020. Parallel worlds: Repeated initializations of the same team to improve team viability. *proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (2020), 1–22.

[78] Yifan Wu, Joseph M. Hellerstein, and Arvind Satyanarayan. 2020. B2: Bridging Code and Interactive Visualization in Computational Notebooks. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 152–165.

[79] Doris Xin, Eva Yiwei Wu, Doris Jung-Lin Lee, Niloufar Salehi, and Aditya Parameswaran. 2021. Whither AutoML? Understanding the Role of Automation in Machine Learning Workflows. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–16.

[80] Qian Yang, Jina Suh, Nan-Chen Chen, and Gonzalo Ramos. 2018. Grounding interactive machine learning tool design in how non-experts actually build models. In *Proceedings of the 2018 Designing Interactive Systems Conference*. ACM, New York, NY, USA, 573–584.

[81] Matei Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, et al. 2018. Accelerating the machine learning lifecycle with MLflow. *IEEE Data Eng. Bull.* 41, 4 (2018), 39–45.

[82] Amy X Zhang, Michael Muller, and Dakuo Wang. 2020. How do data science workers collaborate? roles, workflows, and tools. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (2020), 1–23.